Titre: Title:	Direct simulation of flow and rheology of polymeric fluids
Auteur: Author:	Bogdan Zygmunt Dlugogorski
Date:	1993
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Dlugogorski, B. Z. (1993). Direct simulation of flow and rheology of polymeric fluids [Ph.D. thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/57979/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/57979/
Directeurs de recherche: Advisors:	
Programme: Program:	Unspecified

UNIVERSITÉ DE MONTRÉAL

DIRECT SIMULATION OF FLOW AND RHEOLOGY OF POLYMERIC FLUIDS

par

Bogdan Zygmunt DLUGOGORSKI

DÉPARTEMENT DE GÉNIE CHIMIQUE

ÉCOLE POLYTECHNIQUE

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU GRADE DE PHILOSOPHIAE DOCTOR (Ph.D.)
(GÉNIE CHIMIQUE)

Février 1993

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Cette thèse intitulée:

Direct Simulation of Flow and Rheology of Polymeric Fluids

présentée par: Bogdan Z. DLUGOGORSKI

en vue de l'obtention de grade de: PHILOSOPHIAE DOCTOR (Ph.D.)

a été dûment acceptée par le jury d'examen constitué de:

CHAOUKI Jamal, Ph.D., président

CARREAU Pierre J., Ph.D., membre et directeur de recherche

GRMELA Miroslav, Ph.D., membre et co-directeur de recherche

FORTIN André, Ph.D., membre

COHEN Claude, Ph.D., membre

To Eve, David and Annette

"To be surprised, to wonder, is to begin to understand. This is the sport, the luxury, special to the intellectual man. The gesture characteristic of his tribe consists in looking at the world with eyes wide open in wonder. Everything in the world is strange and marvellous to well-open eyes. This faculty of wonder is the delight refused to your football 'fan,' and, on the other hand, is the perpetual ecstasy of the visionary. His special attribute is the wonder of the eyes. Hence it was that the ancients gave Minerva her owl, the bird with ever-dazzled eyes."

José Ortega y Gasset, *The Revolt of the Masses*

ABSTRACT

Placed within the context of non-equilibrium molecular dynamics (NEMD) simulation of model polymer fluids, this work has three underlying threads: rheology of dumbbell fluids, rheology of rigid bodies and direct simulation of flow of atomic liquids.

In the first two papers entitled *Viscometric Functions for FENE and Generalized Lennard-Jones Dumbbell Liquids in Couette Flow: Molecular Dynamics Study* and *Microscopic and Mesoscopic Results from Non-Equilibrium Molecular Dynamics Modeling of FENE Dumbbell Liquids* a very extensive NEMD investigation on the properties of vibrating dumbbells is conducted. Generally accepted NEMD algorithm called SLLOD is used but its application is extended to new areas. The data are collected in terms of macroscopic (*e.g.* viscometric functions), mesoscopic (*e.g.* distribution functions) and microstructural (snapshots) results. In addition, atomic liquids are investigated for new state points. The results at all three levels of physical detail indicate that the underlying microstructure is responsible for trends observed in microscopic and mesoscopic variables. The contracted distribution functions are calculated in both position and velocity spaces, and the fluid structure is probed by two conformation tensors. It is observed that dumbbells form a variety of short and long range structures depending on the imposed shear rate and the size of a single dumbbell. The dumbbell liquids exhibit shear thinning, non-zero first and second normal stress differences, and volumetric dilatancy. These effects are weakly sensitive to detail in shape of the intra-molecular potentials, and to the dominant frequency associated with vibrations of dumbbells. Furthermore, many new phenomena are observed for the first time including the disappearance of the Maxwellian distribution at high shear rates. The Gibbs distribution function in the configuration space collapses to one particle distribution for very stiff dumbbells. In general, dumbbells are more elastic than atomic liquids.

Thermodynamical, microstructural and rheological data on fluids composed of rigid ellipsoids of revolution, of size corresponding to average end-to-end extension of vibrating dumbbells, are assembled in the paper *Rheology of Several Hundred Rigid Bodies*. Here, both the algorithm, based on the equations developed for suspensions, and the NEMD

results are new. The microcanonical equations of motion for the translational and angular momenta as well for the position of mass-centers and orientational unit vectors are derived from a Hamiltonian. These expressions are then augmented by SLLOD-like and Gaussian thermostat terms added consistently to equations for both the rotational and translational degrees of freedom. The thermodynamic data are generated along one isotherm and the rheology is investigated for two state points. At the intermediate shear rates, ellipsoids become aligned at an angle to the direction of flow, the stress tensor begins to be highly nonsymmetric, and the radial distribution functions become significantly anisotropic. At increasing shear rates, this configuration is replaced by the isotropic-like fluid which evolves to a considerably ordered structure exhibiting global orientation of particles in the direction of the vorticity axis. As in the case of vibrating dumbbells, the rearrangements at the microstructural level are closely followed by trends in the viscometric functions. For example, at low densities the first normal stress difference may become slightly negative.

The last paper entitled *Direct Numerical Studies of Viscous Flow of Two Dimensional Liquids* investigates flow of fluids composed of disks between parallel plates as well as in sudden contraction and expansion. The most of the methodology has been already developed, but it is carried over to new situations and original data are generated for flow in new geometries. An excellent agreement is found between the results of simulations and the well known flow structure, such as flow separation and formation of viscous eddies proving that the flow behavior observed experimentally and predicted from the Navier-Stokes equations can be reproduced directly from the microscopic Newtonian equations of motion. The simulations are performed in the Newtonian regime, for medium sized systems comprising up to 8000 disks. In addition to the traditional stochastic and periodic boundary conditions, new diffusive boundary conditions are developed which include a stagnant fluid layer. Furthermore, the viscosity and thermal conductivity coefficients for two dimensional fluids are calculated from velocity and temperature profiles between parallel plates and confirmed by data obtained by a means of SLLOD and the Evans thermal conductivity algorithms.

Direct simulations of the time evolution of a model macromolecular fluid yield a complete set of microscopic, mesoscopic and macroscopic results. A major advantage of NEMD models is the fact that they are formulated in terms of ordinary rather than partial differential equations. They lead directly to the results and avoid complex numerical algorithms and complicated mathematical developments that are essential in obtaining predictions from most rheological models based on the kinetic theory.

RESUME

Ces dernières années, la dynamique moléculaire à l'équilibre (EMD) a été utilisée avec succès pour résoudre les problèmes relatifs à la thermodynamique classique ou à la conception de médicaments, et permet d'obtenir des résultats qui ne pouvaient être atteints qu'au prix d'expériences longues et coûteuses. La dynamique moléculaire hors équilibre (NEMD), est un domaine relativement nouveau qui est utilisé principalement pour des systèmes physiques soumis à des forces externes. Dans ce travail, NEMD est appliquée pour étudier le comportement rhéologique et l'écoulement de plusieurs centaines de particules sphériques, d'haltères en vibration, de particules rigides ou de plusieurs milliers de disques. Des études récentes ont montré que les équations de mouvement SLLOD qui ont été introduites initialement à un niveau de description microscopique dans le contexte de la NEMD, peuvent être réinterprétées dans l'esprit de théories formulées au niveau mésoscopique. C'est le cas notamment des équations gouvernant l'évolution temporelle des variables internes. Les variables du système sont la position et la vitesse des particules (équivalentes, renormalisées) aussi bien que le vecteur d'orientation et quantité de mouvement en rotation dans le cas de particules anisotropiques.

Le travail est divisé en trois grandes parties. Les deux premières présentent la rhéologie d'haltères vibrantes et de particules rigides utilisant respectivement les algorithmes SLLOD et SLLOD généralisés. La dernière partie est consacrée à l'écoulement de disques bidimensionnels.

Dans la première partie, de nouveaux résultats rhéologiques macroscopiques, issus de simulations d'écoulement Couette utilisant la dynamique moléculaire hors équilibre, sont présentés. Des liquides composés uniquement d'atomes, sont étudiés pour des nouveaux points d'états. Des nouveaux résultats sont obtenus pour deux types liquides haltères: (1) "finitely extensible nonlinear elastic" (FENE) et (2) Lennard-Jones généralisé (GLJ), et jusqu'à une vitesse de cisaillement adimentionnelle de 15. Le liquide haltère présente un comportement rhéofluidifiant, une première N_1 et deuxième N_2 différence de contraintes normales différentes de zéro, ainsi qu'une dilatation volumique. Ces effets sont faiblement sensibles au détail de la forme des potentiels intermoléculaires, et à la fréquence

principale associées à la vibration des haltères. Cependant, la viscosité Newtonienne de liquides haltères dépend fortement de la taille des haltères. Le début du comportement rhéofluidifiant d'haltères FENE et GLJ est reporté à des vitesses de cisaillement plus élevées en comparaison avec les liquides atomes.

Le papier intitulé "Microscopic and Mesoscopic Results from non-Equilibrium Molecular Dynamics Modeling of FENE Dumbbell Liquids" poursuit la discussion sur les liquides haltères, et l'étend à de nouveaux domaines. Ici, les résultats microscopiques et mésoscopiques sont présentés pour plusieurs fluides composés d'haltères élastiques et soumis à un écoulement selon la dynamique SLLOD. Les fonctions de distribution contractées sont calculées à la fois dans les espaces position et vitesse thermique. La structure du fluide est examinée à l'aide de deux tenseurs de conformation. On observe que les haltères s'organisent suivant une grande variété de structures qui dépendent de la vitesse de cisaillement imposée et des dimensions de l'haltère. L'hypothèse d'un champ de vitesse Maxwellien, n'est satisfaite qu'à faible vitesse de cisaillement. En cisaillement, la distribution des distances bout à bout de l'haltère est similaire à la fonction de distribution de Gibbs (décrite dans l'espace de position), si les intéractions à l'intérieur des molécules sont plus fortes que les forces entre molécules. En moyenne, les haltères les plus longues sont obtenues pour une inclinaison de 50° à 30° par rapport à la direction de l'écoulement. Bien que les potentiels FENE et GLJ soient bien distincts, le comportement macroscopique des deux types de liquide est similaire lorsque la rupture des haltères est faible et lorsque l'extension bout à bout (moyennée et adimensionnalisée) des haltères FENE est maintenue proche de 0.9. La ressemblance des propriétés macroscopiques des haltères FENE et GLJ découle de la simplicité des deux modèles.

Dans la deuxième partie, une nouvelle dynamique moléculaire hors équilibre qui trouve son origine dans les théories mésoscopiques de suspensions, est utilisée pour apréhender la rhéologie de plusieurs centaines d'éllipsoides de révolution, qui inter-agissent avec le potentiel de Gay-Berne. Le rapport du puit de potentiel pour deux particules prises côte à côte et bout à bout est choisi égal à 2.5. L'équation de mouvement microcanonique pour la quantité de mouvement en translation et en rotation ainsi que pour la position du centre de masse et les vecteurs unitaires dérivent d'un Hamiltonien. Ces expressions sont alors enrichies par les termes similaire à SLLOD et les termes qui maintiennent une température constante que l'on ajoute de façon consistante aux équations de translation et de rotation. Les données thermodynamiques sont générées le long d'un isotherme (la température adimensionelle est maintenue à 1). La rhéologie est étudiée en deux points d'états (la densité du nombre de particule - ρ - est égale à 0.25 et 0.4) qui se trouvent dans la phase isotropique si le système n'est pas soumis à un écoulement. Comme l'indiquent les configurations moléculaires, à des vitesses de cisaillement intermédiaires (vitesses de cisaillement adimensionalisées de l'ordre de 1 à 2), les éllipsoides s'alignent avec une inclinaison par rapport à la direction de l'écoulement, et le tenseur de contrainte commence à ne plus être symétrique. A des vitesses de cisaillement encore plus importantes, cette configuration se casse pour former l'arrangement transitoire d'un fluide isotropique, et ensuite se réorganiser en une structure très ordonnée qui présente une orientation globale des particules perpendiculaire à la direction de l'écoulement. Pour ρ=0.4, la première et la deuxième différence de contrainte normale sont positive et négative respectivement, mais à faible densité (ρ=0.25), N_1 devient légèrement négatif. En plus du tenseur de contrainte, nous avons calculé le tenseur de conformation, le paramètre d'ordre, et les composantes de la fonction de distribution radiale. A vitesse de cisaillement élevée, la fonction de distribution radiale devient anisotropique de façon significative. De plus, nous avons étudié dans une perspective moléculaire le phénomène d'overshoot de la contrainte pour un écoulement en cisaillement simple, ainsi que l'évolution de la distribution des vitesses en translation, en fonction de la vitesse de cisaillement.

Finalement, la troisième partie est dédiée à la simulation directe de l'écoulement d'atomes dans une géométrie bidimensionelle. Ainsi, le comportement du fluide qui est observé

expérimentalement et prédit par les équations de Navier-Stokes, est reproduit directement par les équations de mouvement Newtoniennes microscopiques. Les simulations sont effectuées pour des écoulements entre deux plans, et également pour l'écoulement dans des contractions et expensions soudaines. Les résultats sont présentés en terme de profils de vitesse, de pression, de température, de densité, d'énergie potentielle, de contrainte de cisaillement, et de vorticité. En général, un excellent accord est obtenu entre les profils simulés et les structures du fluide bien connues, telles que la séparation des lignes de courant, et la formation de tourbillons visqueux, prouvant ainsi que la micro-hydrodynamique est un outil pertinent pour liéer les phénomènes macroscopiques avec les méchanismes physiques. Les simulations sont opérées dans le régime Newtonien afin de minimiser les effets viscoélastiques, pour des systèmes de taille moyenne inférieure à 8000 disques. Ce nombre est suffisament important pour contrôler l'effet des frontières et l'effet du nombre de particules. Les conditions aux limites stochastiques et périodiques habituelles, sont employées dans la plupart des simulations, mais on a développé également des conditions aux frontières de diffusion, qui peuvent inclure une couche stagnante. Les problèmes d'échelle, qui sont liés à l'appliquation d'une force externe importante sur un système microscopique (dont la taille est de l'ordre de 100 Å), se traduit en pressions et gradients de température extrêmes. En plus, la viscosité et les coefficients de conductivité thermique obtenus à partir des profils de vitesse et de température, sont présentés pour l'écoulement entre deux plaques. Ces résultats sont confirmés d'une autre manière, en modélisant l'écoulement Couette avec les équations de mouvement SLLOD et l'algorithme d'Evans pour la conductivité thermique (en deux dimensions).

Les simulations directes de l'évolution temporelle d'un fluide macromoléculaire modélisé fournissent une gamme complète de résultats microscopiques, mésoscopiques et macroscopiques. Un avantage majeur des modeles NEMD est le fait qu'ils sont décrits par des équations en dérivées ordinaires, plutôt qu'en différentielles partielles. Ils conduisent directement aux résultats et évitent des calculs mathématiques et numériques complèxes qui sont nécessaires dans la plupart des modèles rhéologiques basés sur la théorie cinétique.

ACKNOWLEDGMENTS

I am indebted to many of my professors, colleagues and fellow graduate students for help and advice that have led to a successful completion of this research project.

Especially, I would like to express my sincere gratitude to my doctorate advisors Drs Miroslav Grmela and Pierre J. Carreau for their inspiration, encouragement, guidance, incisive criticism and insistence on maintaining well defined objectives in the course of this work. I have been motivated by their scientific achievements and their human qualities have made our interaction both congenial and stimulating for me.

Also, I would like to thank members of the USENET community, especially Marc Boucher, for the expert computer knowledge they have not hesitate to share with me. Their quest for freer exchange of computer codes and their devotion to develop high-quality, public-domain software packages deserve the highest praise. Hermel Dubé of the Department of Electrical Engineering provided me with much needed computational resources on which part of this work was completed.

I owe a special debt to my wife Eve, my parents Zygmunt and Aleksandra, my uncle and aunt Chester and Irene Ostrowski for their continuing support and encouragement through the duration of this project. My mother-in-law Josefa Kurek stayed with us and was of a great help to Eve and me when our daughter Annette was born.

In my early years, I was profoundly touched by many summer discussions with my aunt Olga Langenberger whose kind nature has been matched by her intellectual ability. At least now, I wish to acknowledge her influence on the course of my adult life.

It was a pleasure to work along and share the office with Ahmed Chehbouni, Fatiha Elfeninat, Tamal Ghosh, Ali Gonzales, Patrick Hass, Akli Iabbadene, Vincent Langlade and Jacques Thomasset. I could always count on them.

The financial support in form of research scholarships was provided by the *Natural Sciences and Engineering Research Council of Canada* during the first two years, and by *Fonds pour la formation de chercheurs et l'aide à la recherche* of the Province of Québec during the third year of this project. For the last six months, the financial assistance from my advisors' research grants is gratefully acknowledge. Supplementary scholarships

from École Polytechnique and the Department of Chemical Engineering are recognized with gratitude.

LIST OF SYMBOLS

Paper 1&2

Latin Symbols

A	arbitary parameter
b^*	dimensionless group
$\mathbf{c}$	laboratory velocities
$\underline{\underline{c}}$	first conformation tensor
$\underline{\underline{C}}$	second conformation tensor
E^*_{id}	dimensionless total energy per isolated FENE dumbbell
E^*_{vib}	dimensionless vibrational energy
f^*	dimensionless vibration frequency of a single FENE dumbbell
f_v	distribution function based on peculiar velocities
$F_{i\alpha}$	α component of the force exerted on the ith particle
$F_{ij\alpha}$	α component of the force exerted on the ith by the jth particle
$\mathbf{F_i}$	force exerted on the ith particle
g	dumbbell pair radial distribution function
H	spring constant
H^*	dimensionless spring constant
$\mathcal{H}$	Hamiltonian function
k	power law consistency index
k_B	Boltzmann constant
K	kinetic energy
K_{cm}	kinetic energy of center-of-mass
K_{rot}	kinetic energy of rotation
K_{vib}	kinetic energy of vibration
L^*	dimensionless length
M^*	dimensionless mass
n	power law exponent

N	total number of particles
N_1	first normal stress difference
N_1^*	dimensionless first normal stress difference
N_2	second normal stress difference
N_2^*	dimensionless second normal stress difference
P	hydrostatic pressure
$p_{i\alpha}$	α component of the momentum vector of the ith particle
$\mathbf{p_i}$	momentum vector of the ith particle
$r_{i\alpha}$	α component of the position vector of the ith particle
r_{eq}	midway position between r_{min} and r_{max}
r_{min}	minimum extension of a single FENE dumbbell
r_{max}	maximum extention of a single FENE dumbbell
$\mathbf{r_i}$	position vector of the ith particle
R	end-to-end distance
t	time
t^*	dimensionless time
T	temperature
T^*	dimensionless temperature
u_α	α component of the hydrodynamic velocity field
$\mathbf{u}$	hydrodynamic velocity field
$\mathbf{v}$	relative velocity
V	total volume of the primitive cell
$\mathbf{V}$	center-of-mass velocity
$\mathbf{v}_{\|\|}$	vibration velocity
$\mathbf{v}_\perp$	rotational velocity

Greek Symbols

α	Gaussian multiplier
$\dot{\gamma}$	shear rate

$\dot{\gamma}^*$	dimensionless shear rate
Γ	$6N$ dimensional phase space
$\delta_{\alpha\beta}$	Kronecker delta
Δt^*	dimensionless time step
$\Delta\phi$	interval of alignment angle
ϵ	depth of the potential well (Lennard-Jones parameter)
η	viscosity
η^*	dimensionless viscosity
θ	angle
Ξ	contracted velocity distribution function
ρ_B	bulk density of fluid
ρ_d	single dumbbell density
ρ^*	dimensionless density
σ	collision diameter (Lennard Jones parameter)
$\underline{\underline{\sigma}}$	total stress tensor
ϕ	alignment angle
Φ	total potential energy
Φ_{inter}	inter-particle potential
Φ_{intra}	intra-particle potential
Φ^*_{intra}	dimensionless intra-particle potential
φ_{ij}	binary potential
φ_{ij}^{LJ}	Lennard-Jones potential
φ_{ij}^{FENE}	FENE potential
$\underline{\underline{\chi}}$	velocity gradient tensor
ψ_1	primary normal stress coefficient
ψ_1^*	dimensionless primary normal stress coefficient
ψ_2	secondary normal stress coefficient
ψ_2^*	dimensionless secondary normal stress coefficient
Ψ	contracted single dumbbell configuration distribution function

Paper 3

Latin Symbols

$\underline{\underline{c}}$	conformation tensor
$\mathbf{F}_i^r$	translational forces
$\mathbf{F}_i^p$	rotational forces
g	pair radial distribution function
$\underline{\underline{g}}$	resolved pair radial distribution function
$g_{\parallel}$	parallel components of the pair radial distribution function
$g_{\perp}$	perpendicular components of the pair radial distribution function
$\mathcal{H}$	Hamiltonian function
$\underline{\underline{I}}$	inertia tensor
$\mathcal{K}$	Gay-Berne strength parameter
m_i	mass of an ellipsoid
$\mathbf{M}$	momenta of mass-centers
N	number of molecules
N_1	first normal stress difference
N_2	second normal stress difference
L	side length of a primitive cube
$\hat{\mathbf{n}}$	director
$\hat{\mathbf{p}}_\mathbf{i}$	orientational unit vector
$\mathbf{P}$	center-of-mass momenta
$P_{i\alpha}$	α component of the center-of-mass momenta
r	distance between particles
$\mathbf{r}$	position of mass-centers
$\mathbf{r}_{ij}$	center-to-center vector
$r_{ij\parallel}$	component of r_{ij} along $\hat{\mathbf{n}}$
$r_{ij\perp}$	component of r_{ij} normal to $\hat{\mathbf{n}}$

S	order parameter
T	temperature
$\mathbf{u}$	macroscopic velocity

Greek Symbols

α_1, α_2	Gaussian multipliers
$\dot{\gamma}$	shear rate
$\epsilon_{=}$	depth of potential well for side-by-side configuration
ϵ_{--}	depth of potential well for end-to-end configuration
η_1	$\frac{\sigma_{xy}}{\dot{\gamma}}$
η_2	$\frac{\sigma_{yx}}{\dot{\gamma}}$
η^+	stress growth function
κ	anisotropy parameter
λ_i	Lagrange multiplier
ρ	particle number density
σ	collision radius (Lennard-Jones parameter)
$\sigma_{\leftrightarrow}$	major axis of ellipsoid
$\sigma_{\updownarrow}$	minor axis of ellipsoid
$\underline{\underline{\sigma}}$	extra stress tensor
ϕ_{ij}	binary potential
Φ	potential energy
$\underline{\underline{\chi}}$	velocity gradient tensor
$\underline{\omega}_i$	angular velocity vector
$\underline{\Omega}$	vorticity

Superscripts

*	nondimensional quantities

Subscripts

t	time averaged quantities

Paper 4

Latin Symbols

g	scalar gravity
$\mathbf{g}$	gravity
k_B	Boltzmann constant
m	mass of a single disk
N	total number of disks
r_{ij}	interparticle separation
$\mathbf{r_i}$	position coordinates of the i^{th} particle
$\mathbf{v_i}$	velocity coordinates of the i^{th} particle
T	temperature

Greek Symbols

ϵ	depth of potential well
σ	collision diameter
ρ	density
ϕ^{SD}	inter-molecular potential (soft disks)
ϕ^{SLJ}	inter-molecular potential (shifted Lennard Jones)
ϕ^{FENE}	intra-molecular potential (FENE)

1 GENERAL INTRODUCTION

1.1 Introduction

Euler's fluid mechanics is formulated in terms of partial differential equations whose integration is an important and one of the oldest parts of applied mathematics. However, closed analytical solutions have been found only for a small number of particular cases. Nowadays, with an arrival of fast digital computers, these equations can be solved numerically. Hence, one should realize the following steps in obtaining predictions when the spirit of Euler's fluid mechanics is followed: 1. casting physical phenomena into partial differential equations, 2. discretizing the equations in space and time in order to arrive at finite difference equations (truncation of the power series), and finally 3. employing digital computers (round-off errors) to yield solutions [1, 2, 3]. Until recently, the above approach has been accepted in both fluid dynamics and rheology; the computational fluid dynamics (CFD) and computational rheology (CR) have been considered as the final step in the solution of the fluid dynamics and rheological equations on digital computers. Since this scheme is neither direct nor simple, one may contemplate bypassing one or both middle stages in order to model the physics directly. In this work we study rheology and flow of model polymeric fluids without the recurrence to the partial differential formulation of the conservation and constitutive equations.

The techniques that remain at our disposal have been recently developed based on the idea that matter is not really continuous at the atomic level. Especially, molecular dynamics (MD), lattice gases (a class of cellular automata, CA) and an older method of Brownian dynamics (BD) have been receiving much attention. As oppose to Monte Carlo (MC) techniques which can only provide static properties since the modeling is confined exclusively to the configuration (position) space, MD, BD and CA are able to yield dynamic properties and are well suited for simulations of dynamic processes. An ever increasing number of papers published on MD, BD and lattice gases comes out not only from theoretical physics and chemistry departments but also from chemical engineering schools - a sign that all three methods have reached the theoretical maturity, and could be applied to problems of engineering significance. Another indication for the emergence of a new branch of science is a number of books in print. For instance, in the field of molecular dynamics, besides more advanced *Theory of Simple Liquids* by Hansen and McDonald [4] and *Statistical Mechanics of Nonequilibrium Liquids* by Evans and Morriss [5], *Molecular Dynamics* and *Computational Statistical Mechanics* by Hoover [6, 7] there is an introductory text by Haile [8] *Molecular Dynamics Simulations: Elementary Methods*. In addition, several monographs include chapters on MD simulations of liquids [9, 10, 11, 12]. Similar lists might have been prepared for BD and CA.

Both MD and BD yield predictions that are continuous functions of the independent variables and time, whereas lattice gases are discrete in space, time, and variables. MD and BD particles are of course discrete in space but their motion is continuous in time. MD is a deterministic technique since no stochastic forces, as for BD, are included in the equations of motion. Somewhat arbitrarily but motivated by pragmatic reasons we have decided to pursue MD simulations in this investigation, although we have performed some preliminary computations on the lattice gases. Unlike MD and BD particles which may be related to single polymer particles, to single atoms or even to small volumes of fluid, lattice gases are built of abstract particles that collide according to some *ad-hoc* collision rules. Lattice gases have been extensively studied for turbulent but not for low Reynolds number flows, and there is no critical body of literature on theory nor on applications of CA to viscoelastic fluids.

In MD simulations one computes the instantaneous positions, velocities (or momenta) of all particles by solving a very large system of ODE; six equations are necessary for each particle. As the input to a simulation, one provides detailed microscopic properties of polymeric liquids (in the form of model potentials). As the output from calculations, a trajectory of the system in 6 N dimensional space, where N is the number of particles, is obtained. In principle, all properties at any level of description can be calculated along a trajectory. For example, the viscometric functions are computed by statistical averaging of the components of the microscopic stress tensor. In other words, MD allows to correlate microscopic description of model polymeric liquids with their macroscopic properties. This is often regarded as a computer analogue of laboratory work and therefore a "term computer (numerical) experimentation" is used. Actually, John von Neumann who conceived and was responsible for the construction of first electronic serial computers also realized the possible applications of computational machines (especially parallel) for doing numerical experiments.

The general objective of this research project was to investigate the rheology and flow of viscoelastic fluids consisting of model polymeric particles, such as vibrating dumbbells, ellipsoids of revolution, or two-dimensional Lennard-Jones-like disks. The specific objectives were: (1) to extend the existing NEMD rheological algorithms for atomic liquids to model polymeric fluids, (2) to formulate new algorithms better suited for this purpose, (3) to develop fast NEMD computer codes capable of yielding results on work-station type computers (*e.g* IBM's RISC, SUN's SPARC), (4) to generate thermodynamical, microstructural and rheological data for model polymeric liquids, and (5) to investigate the flow of model liquids directly.

In the next two sections, we will review the literature on molecular dynamics simulations relating to systems of interest, in term of rheological and flow simulations. We will concentrate on the literature that has not been reviewed in our publications. Useful links between the papers will be given next. The papers are rather complete, each includes background information, theory, computational details and a set of MD results. The synthesis of the major achievements of this study will be given next, followed by a brief outline of the future work. The computer code for rheological modeling of rigid bodies is assembled in the appendix.

1.2 Review of the Literature

The last 15 years have seen a true explosion of the research work devoted to the study of simple liquids. Scientists of different backgrounds started to realize that substantial advances in molecular simulations may be achieved by imposing external constraints on MD systems in such a way as to induce hydrodynamic fluxes. This new field of research has come to be called nonequilibrium molecular dynamics (NEMD). In equilibrium molecular dynamics (EMD) the transport coefficients (*e.g.* shear viscosity, thermal conductivity) are investigated by fluctuations around the equilibrium states. More specifically by calculation of the time autocorrelation function and by the subsequent evaluation of the Green-Kubo integrals (see for example Table II of ref. [13]).

There are several major problems in the calculation of the autocorrelation functions. Since they represent the response to the fluctuation of the system, which could be minute, the signal to noise ratio is rather small. Moreover, the calculations are very inefficient. Usually, the postprocessing of MD results may take more CPU time than MD simulation itself [14]. For example, calculation of the velocity autocorrelation function, required in the evaluation of the self-diffusion, is proportional to the square of the number of time steps. Incidentally, time necessary to compute radial distribution function is proportional to the square of the number of particles. Also, the stress-stress correlation function depends on number of particles in a system and it displays a tail at long times. Most importantly, the shear viscosity values, calculated from EMD are up to 20% greater than the values obtained from the experiments [15].

On the other hand, in NEMD the signal to noise ratio is maximized by using large externally applied forces and gradients. The difficulties in the calculations of the autocorrelation functions are avoided in favor of direct computation of fluxes and transport coefficients; for example, the viscosity of atomic liquids is calculated from the stress tensor obtained from the Irving-Kirkwood formula [16]. When describing the perspective in NEMD H. Hanley once said [9]: "A goal of NEMD is to connect the macroscopic rheological laws and constitutive equations with the atomistic behavior of a fluid". This is almost exactly what we have set out to achieve in this work - we have simulated the flow and rheology of model particles and the connection between atomistic and macroscopic behavior has come as a by-product of the simulations.

In the introduction we purposely mentioned both, computational fluid dynamics and computational rheology. For us, these terms convey different meaning. In CR the problem is as follows: given well defined boundary conditions and velocity profiles one is asked to determine the internal state of the liquid, for example in terms of the internal stresses. In CFD the situation is more complicated - knowing the boundary conditions (e.g. the nature of interactions of molecules with walls) and a constitutive model one calculates both the flow field and the stresses (it follows that the computational non-Newtonian or viscoelastic fluid dynamics is only a subset of CFD). By introducing the distinction between CFD and CR we aimed at showing drawbacks of the classical approach. In order to compute velocity structures by a means of CFD one needs to know the rheological properties (equation of state) for a given fluid. This is not the case for direct modeling of flow.

There is an extensive body of NEMD literature on modeling of simple liquids, inspired by the work of Lees and Edwards [17]. Lees and Edwards's approach has become known as the molecular modeling of rheology since the flow field is imposed either by the boundary conditions (called LE) alone or in conjunction with an imposed linear velocity profile within the simulation box. There are no fundamental differences, except for the directness and space-discreteness, between the classical CR and its molecular dynamics analogue.

In case of the direct simulation of flow by NEMD the improvement over the traditional CFD is significant. One has to specify the boundary conditions and perhaps body forces, such as gravity, to induce the flow and the nature of microscopic interactions among fluid particles but a constitutive equation is not necessary, as hinted above. CFD is concerned with solving five partial differential equation for density, momenta and energy fields which are coupled to a rheological equations of state or to equation(s) for internal state variables, in case of the mesoscopic theories. These PDE are replaced by a system of ODE equations for MD simulations. For simple liquids, the equations of motion are not complicated, however their number could be high (of the order 10^5), in order to generate data with the sufficient statistical accuracy. Before reviewing the literature on the rheological and flow calculations in the next two section we make two general comments.

It has been realized that any fluid may exhibit the non-Newtonian behavior when the product of its relaxation time and the imposed shear rate ($\dot{\gamma}$) is of the order of unity [18]. As a matter of fact, this has been cited as one of the important reasons for the investigation of simple liquids. The logic is that the physical insight gained for atomic liquids could be extrapolated to complex polymeric liquids. For viscoelastic fluids the relaxation time is close to 10^{-2} s and the onset of shear thinning may be observed at relatively low shear rates. In case of simple liquids which have extremely fast relaxation times, the viscoelastic behavior may be noticed if the imposed shear rate is of the order of 5×10^{11} s^{-1}, significantly higher than can be obtained in the laboratory. In colloidal suspensions, particles are bigger and heavier than for atomic liquids and the fluid relaxation time of colloids could be as high as 10^{-3} s. Consequently, colloidal suspension are specially interesting from the MD point of view [19, 20].

Finally, we mention the subtraction technique (see [21] for the latest reference and [11] for the review) that is useful for studying nonequilibrium time dependent phenomena, such as the inception of the shear flow. In this method, unperturbed and weakly perturbed trajectories are followed and the response of the perturbed system is extracted from the knowledge of both trajectories. Since the neighboring trajectories tend to diverge in time, with the largest Lyapunov exponent of the unperturbed system, after a while a correlation between the trajectories is blurred by the noise. Incidentally, the distance between the unperturbed and perturbed (by a small shear) trajectories is linear in $\dot{\gamma}$ [22].

1.2.1 NEMD Modeling of Rheology

Since the latest literature on the molecular modeling of rheology is reviewed in Papers 1-3, in this section we briefly describe the older publications. The introduction of NEMD rheological simulation by Lees and Edwards [17] as well as by Ashurst and Hoover [23, 24] allowed to study the effect of imposed shear rates on the structural and dynamical properties of simple liquids. A few years later, a shear induced phase transition between amorphous and ordered (string) phases was observed for the first time by Erpenbeck [25]. This discovery was questioned shortly after by Evans and Morriss [26] who maintained that the string phase is an artificial result of an imperfect algorithm.

The classical SLLOD algorithm was introduced by the efforts of many people. Lees and Edwards formulated the boundary conditions [17], Ladd [27] put the SLLOD equations into their final form, and Evans *et al.* [28] invented a method to control the kinetic temperature of a system of molecules by a means of Lagrangian multipliers; they called the method *Gauss's principle of least constraint.* Thus, the algorithm is often denoted as

SLLOD/LE/GAUSS.

But it was only Heyes (for example [29, 30]) who produced a remarkable series of papers on every imaginable aspect of the rheology of liquid argon. On the other hand, Evans's contribution to NEMD has been mostly on the theoretical side. Among his many discoveries are inventions of NEMD algorithms for thermal conductivity [31] or for isochoric and isobaric ensembles [5].

The work of Nosé and Hoover led to the formulation of another method for controlling the temperature, nowadays called the Nosé-Hoover thermostat (see [32] and refs. therein). Hess [33] was the first to suggest the extension of NEMD to short polymeric chains.

1.2.2 NEMD Modeling of Flow

The applicability of NEMD simulations to hydrodynamics was demonstrated by showing that results of computer experiments remain in good agreement with the laboratory evidence and with the steady state hydrodynamic equations for incompressible flows; both simple shear (Couette) and channel (Poiseuille) flows were investigated. In addition, studies on Rayleigh-Bénard convection, on flow at boundaries, on obstructed flows, and on flows in dilute gases were also published. In contrast to the NEMD rheological modeling, NEMD flow simulations do not require an imposed velocity profile.

In one study [15], a dense Lennard-Jones fluid near its triple point was subjected to a moving thermal (stochastic) wall implemented by replacing velocities of all particles crossing the system boundary with velocities drawn from the shifted (by macroscopic velocity) Maxwell-Boltzmann distribution, at a given temperature. Similar technique was used to model the heat transfer in liquid argon [34, 35]. The results indicated that the shear viscosity was Newtonian for shear rates up to $\dot{\gamma} = 10^{11}$ s^{-1}, especially away from the triple point. Close to walls, the fluid showed boundary surface effects, such as slip and temperature drop. These effect were especially significant in a layer immediately adjacent to the thermal wall whose thickness was of the order of a mean free path.

Hannon *et al.* [36] subjected assemblies of argon atoms to a uniform acceleration of 10^{12} m/s^2 in order to induce gravity driven flows; in total 1152 atoms, enclosed in a parallelepiped with the ratio of side lengths equal to 32:6:6, were used in the study. This extremely high acceleration was necessary to increase the signal to noise ratio. The heat generated during the simulation was removed by thermal walls, which are similar to stochastic walls described above. The velocity and temperature profiles were fitted to those predicted by the hydrodynamic equations yielding transport coefficients of shear viscosity and thermal conductivity. For the Lennard-Jones potential with a cut off radius equal to three molecular diameters, viscosities were 20% and thermal conductivities 10% lower that measured experimentally. For simulations with a larger number of particles (over 20,000) the agreement between the experimental data and NEMD dynamics results was even better [37]. If the extreme gravity field is replaced by sink and source of molecules which are supposed to emulated the pressure driven flows, as it has been done in a recent study [38], then the density along a channel becomes nonuniform.

NEMD calculations were applied to investigate flows around a cylinder [39] and past a plate inserted perpendicularly to the direction of flow [37]. In both studies qualitative agreement was found between simulation and the macroscopic theory; the development and the propagation of wake oscillations were observed but the flow had to be induced by very high gradients, as for NEMD flow experiments. Actually, it has been argued that the requirement of extreme gradients is a weak point of NEMD. Similarly, good qualitative agreement was demonstrated for the Rayleigh-Bénard hydrodynamic instability, when a convective current is produced in a cell driven by opposed temperature and gravitational fields [40, 41, 37]. Very recently [42], it was shown by NEMD modeling that the no-slip condition may arise naturally as a result of molecular roughness of walls [42]. In macroscopic theory, the non-slip condition has been presented either as a postulate or as a result from 19^{th} century experiments. Finally, NEMD was used to study the heat and momentum transfer phenomena in dilute gases with Knudsen number (ratio of the mean free path of a particle to the characteristic length of the flow system) less than unity [43, 44].

It is clear that NEMD simulations of flow of simple liquids successfully model different types of flows. Both the Newtonian behavior and the macroscopic flow fields have been extracted from direct NEMD simulations for fluids whose properties were specified at the microscopic level. This is a starting point for our work presented in Chapter 6.

1.3 Organization and Links Among Articles

Modeling of industrial processing operations requires that the complex viscoelastic behavior of polymeric liquids be expressed in terms of the rheological constitutive equations. Together with the equations of the conservation of mass, momentum and energy, the constitutive equations are then discretized and solved on digital computers to yield flow predictions. The discretization of a partial differential equation of state (there are also integral equations but not as widely used) is an obligatory step if a rheological model was conceived at the macroscopic or at truly mesoscopic (*e.g.* conformation tensor) levels of descriptions. If however a model is proposed in terms of real or effective particles which move according to Newton equations of motion then only ordinary differential equations have to be discretized. ODE are easier to solve since they are not plagued by the mathematical and numerical difficulties related to PDE [45].

In this work, we have followed an "effective particle" route, which we believe may eventually lead to industrial applications of NEMD to polymer processing. In computational chemistry, as opposed to (computational) chemical engineering, one is interested in obtaining a set of physico-chemical properties for a well specified material at equilibrium, and therefore real particles investigated by EMD are favored. It stands to reason to suggest that the nature of long nonuniform polymeric chains could be simulated more efficiently by a way of model particles, from the practical point of view. Model polymeric particles studied in this work include vibrating and pseudo-rigid dumbbells (Chap. 4), rigid ellipsoids of revolution (Chap. 5) and Lennard-Jones-like disks (Chap. 6); pseudo-rigid dumbbells are dumbbells whose intramolecular bond is very stiff (Chap. 4). To our knowledge, only one investigation [46] has been devoted to rheology of model polymeric particles, although there are more publications on direct simulations of atomic liquids (see for example [37, 38]).

In Chap. 4, we assemble results from a comprehensive investigation of vibrating dumbbells from the rheological perspective, at all levels of description. Normally as in [46], rigid dumbbells (rigid rotors) are regarded as a particular case of vibrating dumbbell, when the bead-bead extension is maintained constant by an imposed constrain. In Chap. 5, we have chosen an alternative route. We consider the dynamics of rigid rotors to be a particular implementation of the dynamics of rigid bodies. Accordingly, we develop the theory and generate a broad set of results for ellipsoids of revolution, which are similar in size and in eccentricity to dumbbells. Fluids consisting of ellipsoids of revolution are known to exhibit liquid crystal behavior [47]. Our investigation has been conducted for the state points that at equilibrium lie within the isotropic phase, but similar NEMD/rigid body approach could be followed in studying the rheology of the nematic and smectic phases.

Chap. 6 is our first attempt to investigate model polymeric fluids by direct simulations. Here, we study middle-size (around 8000 particles) two-dimensional systems of beads in new geometries and with new boundary conditions. We plan, although not within this doctorate dissertation, to replace disks with two dimensional vibrating dumbbells as well as with rigid ellipsoids to obtain flow predictions for these particles as well.

References

[1] T. Toffoli. Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica D*, 10:117–127, 1984.

[2] G. Y. Vichniac. Simulating physics with cellular automata. *Physica D*, 10:96–116, 1984.

[3] N. Margolus. Physics-like models of computations. *Physica D*, 10:1–35, 1984.

[4] J.-P. Hansen and I. R. McDonald. *Theory of Simple Liquids.* Academic Press, London, 1986.

[5] D. J. Evans and G. P. Morriss. *Statistical Mechanics of Nonequilibrium Liquids.* Academic Press, London, 1990.

[6] W. G. Hoover. *Mlecular Dynamics: Lecture Notes in Physics 258.* Springer, Berlin, 1986.

[7] W. G. Hoover. *Computational Statistical Mechanics.* Elsevier, Amsterdam, 1991.

[8] M. P. Haile. *Molecular Dynamics Simulations: Elementary Methods.* John Wiley & Sons, New York, 1992.

[9] W. G. Ciccotti and W. G. Hoover. *Molecular-Dynamics Simulation of Statistical-Mechanical Systems: Proceedings of the International School of Physics 'Enrico Fermi' - Course XCVII.* North Holland, Amsterdam, 1986.

[10] D. W. Heermann. *Computer Simulation Methods in Theoretical Physics.* Springer Verlag, Berlin, 1986.

[11] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids.* Clarendon Press, Oxford, 1987.

[12] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles.* Adam Hilger, Bristol, 1987.

[13] D. J. Evans. Nonequilibrium molecular dynamics. In Ciccotti G. and W.G. Hoover, editors, *Molecular-Dynamics Simulation of Statistical-Mechanical Systems*, pages 221–240. North-Holland, Oxford, 1986.

[14] Z. A. Rycerz and P. W. M. Jacobs. Molecular dynamics simulation program of order n for condensed matter I. MDPYRS1: Scalar pyramid, short-ranged interactions. *Comp. Phys. Comm.*, 60:53–74, 1990.

[15] C. Trozzi and G. Ciccotti. Stationary nonequilibrium states by molecular dynamics: II Newton law. *Phys. Rev. A*, 29(2):916–925, 1984.

[16] J. H. Irving and J. G. Kirkwood. The statistical mechanical theory of transport processes. IV. The equations of hydrodynamics. *J. Chem. Phys.*, 18(6):817–829, 1950.

[17] A. W. Lees and S. F. Edwards. The computer study of transport processes under extreme conditions. *J. Phys. C: Solid State Phys.*, 5:1921–1929, 1972.

[18] J. C. Rainwater, H. J. M. Hanley, T. Paszkiewicz, and Z. Petru. Non-Newtonian flow of a model liquid between concentric cylinders. *J. Chem. Phys.*, 83(1):339–347, 1985.

[19] W. Loose and S. Hess. Rheology of dense model fluids *via* nonequilibrium molecular dynamics: shear thinning and ordering transition. *Rheol. Acta*, 28:91–101, 1989.

[20] T. Yamada and S. Nosé. Two-phase coexistence of string and liquid phases: Nonequilibrium molecular-dynamics simulation of Couette flow. *Phys. Rev. A*, 42(10):6282–6291, 1990.

[21] G. V. Paolini, G. Ciccotti, and H van Beijeren. Nonequilibrium molecular dynamics *via* a nondiverging subtraction technique. *Phys. Rev. A*, 42(10):5912–5916, 1990.

[22] G. P. Morriss, D. J. Evans, E. D. G. Cohen, and H. van Beijeren. Linear response of phase-space trajectories to shearing. *Phys. Rev. Lett.*, 62(14):1579–1582, 1989.

[23] W. T. Ashurst and W. G. Hoover. Dense fluid shear viscosity *via* nonequilibrium molecular dynamics. *Phys. Rev. A*, 11(2):658–678, 1975.

[24] W. T. Ashurst and W. G. Hoover. Shear viscosity *via* periodic nonequilibrium molecular dynamics. *Phys. Lett. A*, 61(3):175–177, 1977.

[25] J. J. Erpenbeck. Shear viscosity of the hard-sphere fluid *via* nonequilibrium molecular dynamics. *Phys. Rev. Lett.*, 52(15):1333–1335, 1984.

[26] D. J. Evans and G. P. Morriss. Shear thickening and turbulence in simple fluids. *Phys. Rev. Lett.*, 56(20):2172–2175, 1986.

[27] A. J. C. Ladd. Equations of motion for non-equilibrium molecular dynamics simulations of viscous flow in molecular fluids. *Mol. Phys.*, 53(2):459–463, 1984.

[28] D. J. Evans, W. G. Hoover, B. C. Failor, B. Moran, and A. J. C. Ladd. Nonequilibrium molecular dynamics *via* Gauss's principle of least constraint. *Phys. Rev. A*, 28:1016–1021, 1983.

[29] D. M. Heyes. Shear thinning and thickening of the Lennard-Jones liquid. *J. Chem. Soc., Faraday Trans. 2*, 82:1365–1383, 1986.

[30] D. M. Heyes. The nature of extreme shear thinning in simple liquids. *Mol. Phys.*, 57(6):1265–1382, 1986.

[31] D. J. Evans. Homogeneous NEMD algorithm for thermal conductivity - application of non-canonical linear response theory. *Phys. Lett. A*, 91(9):457–460, 1982.

[32] W. G. Hoover. Generalization of nosé isothermal molecular dynamics: Non-Hamiltonian dynamics for the canonical ensemble. *Phys. Rev. A*, 40(5):2418–2419, 1989.

[33] S. Hess. Rheological properties *via* nonequilibrium molecular dynamics: From simple towards polymeric liquids. *J. Non-Newt. Fl. Mech.*, 23:305–319, 1987.

[34] A. Tenenbaum, G. Ciccotti, and R. Gallico. Stationary nonequilibrium states by by molecular dynamics: Fourier's law. *Phys. Rev. A*, 25(5):2778–2787, 1982.

[35] A. Tenenbaum. Local equilibrium in stationary states by molecular dynamics. *Phys. Rev. A*, 28(5):3132–3133, 1983.

[36] L. Hannon, G. C. Lie, and E. Clementi. Molecular dynamics simulation of channel flow. *Phys. Lett. A*, 119(4):174–177, 1986.

[37] L. Hannon, G. C. Lie, and E. Clementi. Micro-hydrodynamics. *J. Stat. Phys.*, 51(5/6):965–979, 1988.

[38] M. Sun and C. Ebner. Molecular-dynamics simulation of compressible fluid flow in two-dimensional channels. *Phys. Rev. A*, 46(8):4813–4818, 1992.

[39] D. C. Rapaport and E. Clementi. Eddy formation in obstructed fluid flow: A molecular-dynamics study. *Phys. Rev. Lett.*, 57(6):695–698, 1986.

[40] M. Marechal, M. Malek Mansour, A. Puhl, and E. Kestmont. Molecular dynamics versus hydrodynamics in two-dimensional Rayleigh-Bénard system. *Phys. Rev. Lett.*, 61(22):2550–2553, 1988.

[41] D. C. Rapaport. Molecular-dynamics study of Rayleigh-Bénard convection. *Phys. Rev. Lett.*, 60(24):2480–2483, 1988.

[42] J. Koplik, J. A. Banavar, and J. F. Willemsen. Molecular dynamics of fluid flow at solid surfaces. *Phys. of Fluids A*, 1(5):781–794, 1989.

[43] D. K. Bhattacharya and G. C. Lie. Molecular dynamics simulations of nonequilibrium heat and momentum transport in very dilute gases. *Phys. Rev. Lett.*, 62(8):897–900, 1989.

[44] D. K. Bhattacharya and G. C. Lie. Nonequilibrium gas flow in the transient regime: A molecular dynamics study. *Phys. Rev. A*, 43:761–767, 1989.

[45] D. D. Joseph. *Fluid Dynamics of Viscoelastic Liquids.* Springer Verlag, New York, 1990.

[46] J. W. Rudisill and P. T. Cummings. The contribution of internal degrees of freedom to the non-Newtonian rheology of model polymer fluids. *Rheol. Acta*, 30:33–43, 1991.

[47] E de Miguel, L. F. Rull, M. K. Chalam, and K. E. Gubbins. Liquid crystal phase diagram of the Gay-Berne fluid. *Mol. Phys.*, 74(2):405–424, 1991.

2 RHEOLOGY OF VIBRATING DUMBBELLS

Viscometric Functions for FENE and Generalized Lennard-Jones Dumbbell Liquids in Couette Flow: Molecular Dynamics Study

Bogdan Z. Dlugogorski, Miroslav Grmela and Pierre J. Carreau

Département de génie chimique
Centre de recherche appliquée sur les polymères (CRASP)
École Polytechnique
Case postale 6079, succursale A
Montréal, Québec H3C 3A7, CANADA.

- *Journal of Non-Newtonian Fluid Mechanics* (in press)

Summary

We report new macro-rheological results extracted from non-equilibrium molecular dynamics (NEMD) simulations of Couette flow. We investigate atomic liquids for new state points, and in addition two types of dumbbell liquids: (1) finitely extensible nonlinear elastic (FENE) and (2) newly defined generalized Lennard-Jones (GLJ), up to a nondimensional shear rate of 15. The dumbbell liquids exhibit shear thining, non-zero first and second normal stress differences, and volumetric dilatancy. These effects are weakly sensitive to details in shape of the intra-molecular potentials, and to the dominant frequency associated with vibrations of dumbbells. However, the Newtonian viscosity of dumbbell liquids strongly depends on the size of dumbbells. The onset of shear thining of FENE and GLJ dumbbells is delayed to higher shear rates in comparison with atomic liquids. In general, for the entire investigated region, we see that dumbbell are slightly more elastic than atomic liquids.

1 Introduction

Modeling of polymer processing operations can be fully appreciated by the polymer processing industry only if the flow predictions are sufficiently accurate. The accuracy can be achieved only if the rheological complexity of the polymeric fluids involved is expressed in the governing equations of the model. If this is attempted, however, the resulting system of partial differential equations becomes so complex that the numerical solutions cannot be found [1]. One way to avoid these difficulties is to simulate the flows directly, *i.e.* molecule by molecule, with no recourse to partial differential equations. The ultimate goal of our research, at the Centre de recherche appliquée sur les polymères, is to perform this type of simulations of flows of polymeric liquids. The time evolution of polymeric fluid is modeled by using ordinary differential equations describing the time evolution of the molecules. These equations are easier to formulate and solve. There are of course many difficulties met in this approach. We believe, however, that in the long term much is to be gained in terms of mathematical and computational simplicity, by performing simulations at the microscopic level. The first step in the molecular simulations of flows of polymeric fluids is to investigate rheological properties of fluids that are defined at the molecular level. We make distinction between rheological simulations, where the velocity profile is given (*e.g.* Couette flow), and the computational fluid dynamics where the velocity profile arises as a result of the simulation. In this paper we present results of rheological simulations. Flow simulations will be given in a subsequent publication [2].

Non-equilibrium molecular dynamics (NEMD) is a technique used to simulate the behavior of a system of interacting particles subjected to an external field. Although the modeling is performed on the microscopic scale, macroscopic transport coefficients can be extracted from the results by using statistical averaging. This has been well demonstrated, especially for simple liquids [3]. Recently, the technique has been applied to predict the rheological properties of dumbbell liquids, whose intra-particle interaction are described by finitely extensible nonlinear elastic (FENE) potential [4] (thereafter called "FENE liquids"). This study extends the investigation to higher shear rates. The technique of NEMD is not constrained to atomic and dumbbell fluids; it may be easily generalized to short polymeric chains, as it was done among others, by Hess [5] and recently by Berker ***et al.*** [6]. In the former publication, the intra-particle (bonded) interactions were modeled by the usual Lennard-Jones potential multiplied by a constant, whereas in the latter article the intra-particle potentials incorporating bond-stretching, angle bending and tortion were used.

The dumbbell fluids that we investigate must not be perceived as truly molecular fluids. They are model fluids composed of small number of model molecules, relative to the Avogadro number. It is highly unlikely that it will be possible to perform direct rheological and especially flow simulations using molecular fluids that preserve the exact one-to-one correspondence between atoms in polymeric chains and in simulated particles. Even replacing methyl and methylene groups by single "computer" particles has been attempted only for medium size alkanes such as eicosane, ($C_{20}H_{42}$) [7] for nonequilibrium simulations, and for short polymeric chains, of less than 400 monomers, in case of equilibrium modeling [8]. In our simulations, the microscopic character of the dumbbell fluids is preserved by the Lennard-Jones type interactions between beads (atoms) belonging to two different dumbbells, whereas the phenomenological component of the liquids is implied by the FENE and GLJ potentials. It is appropriate to emphasize that the fluids investigated in this paper are artificial and some results may be tested exclusively against mesoscopic models.

The FENE and GLJ model fluids are described in Sec. 2 in conjunction with the definition of the dynamic evolution equations called SLLOD [9]. Dimensionless quantities, statistical averaging and a few words about hardware are also given in that section. We have decided to implement the SLLOD algorithm classically in terms of single bead motion rather than using the evolution equations for center-of-mass and end-to-end vectors of each dumbbell. The implications that follow from this choice are examined in Sec. 3. It has been thought that dumbbell liquids should exhibit different macroscopic properties than atomic fluids due to the appearance of new frequencies associated with vibrations. This notion is investigated in detail in Sec. 4, together with a related topic of the average size of a single dumbbell and issues related to physical and numerical breakage of dumbbells. Sec. 5, which contains the figures of the viscometric functions of monatomic and molecular liquids, discusses the main results of this work. Macroscopic predictions from computer simulations of dumbbell fluids and from elastic dumbbell theories are compared in Sec. 6. The conclusions are presented in Sec. 7.

2 Equations of motion and potentials

The model fluid is represented by 256 atoms or 128 dumbbells that are placed in a primitive cell of size adjusted in such a manner that the density of the liquid corresponds to the density of liquid argon. The cell is surrounded by its infinitely many images to simulate a homogeneous macroscopic system. We consider liquid argon as a convenient benchmark example. Since the inception of molecular dynamics, liquid argon has been a favorite subject of computer experimentation due to its ideal physical properties. Therefore, the abundant computer generated data is available for the purpose of comparison. As opposed to quite a few molecular dynamics studies that concentrate on elucidating the exact physical properties of one well defined fluid (be it liquid argon, nitrogen, or light alkanes), we investigate rheological properties within a group of liquids.

Newton's equations governing the time evolution of the atoms are coupled to the hydrodynamic equations according to SLLOD [10]:

$$
\begin{aligned}
\frac{dr_{i\alpha}}{dt} &= \frac{p_{i\alpha}}{m} + u_{\alpha}, && (1)\\
\frac{dp_{i\alpha}}{dt} &= F_{i\alpha} - p_{i\gamma}\chi_{\gamma\alpha}, && (2)\\
\frac{\partial u_{\alpha}}{\partial t} &= -\frac{\partial u_{\alpha}u_{\gamma}}{\partial r_{\gamma}} - \frac{\partial \sigma_{\alpha\gamma}}{\partial r_{\gamma}}, && (3)\\
\sigma_{\alpha\beta}(\dot{\gamma}) &= -\frac{1}{V}(\sum_{i=1}^{N}\frac{p_{i\alpha}p_{i\beta}}{m_i} + \sum_{i=1}^{N-1}\sum_{j>i}^{N} r_{ij\alpha}F_{ij\beta}). && (4)
\end{aligned}
$$

where,

$$i = 1, \ldots, N, \text{ ;in this study } N = 256, \qquad (5)$$

$$\chi_{\alpha\beta} = \frac{\partial u_{\alpha}}{\partial r_{\beta}}. \qquad (6)$$

In the above equations, the Einstein summation convention is implied, and all symbols have their traditional meanings: $\mathbf{r_i}$, $\mathbf{p_i}$, and $\mathbf{F_i}$ are position, momentum and force vectors of the i^{th} particle, m stands for mass of a single atom (bead). Subscripts, such as $_{\alpha}$, denote components of these vectors and of the tensor $\underline{\underline{\chi}}$, in the Cartesian coordinates. $\mathbf{u}$ denotes the hydrodynamic velocity field in case of incompressible and isothermal fluid. In general, $\mathbf{u}$ depends on the location $\mathbf{r}$, V stands for the total volume of the primitive cell, $\mathbf{F_i}$ includes components due to bonded and inter-molecular interactions, as shown below, $\underline{\underline{\sigma}}$ is the total stress tensor. The scalar pressure P will be introduced in Sec. 3.

It has been shown [10] that Eqs. (1)-(4) represent a Hamiltonian dynamic system in which the time evolution is generated by the Hamiltonian function:

$$\mathcal{H} = \int \frac{1}{2}\rho_B \mathbf{u}^2 d\mathbf{r} + K + \Phi, \qquad (7)$$

representing the total energy of the fluid. $\int \frac{1}{2}\rho_B \mathbf{u}^2$ is the kinetic energy of the fluid in the laboratory frame of reference; ρ_B is a bulk density of the fluid. The kinetic energy K is based on the peculiar (thermal) motion of particles and Φ denotes the total potential energy of the system of interacting particles per unit volume.

In order to interpret physically Eqs. (1)-(4), we compare them with the governing equations that arise in the standard rheological modeling. We recall that a rheological model always begins with a selection of state variables (called internal state variables) characterizing the internal structure of the complex fluid under consideration. The equations governing the time evolution of the internal state variables are then coupled to the hydrodynamic equations. The final system of equations possesses a Hamiltonian structure (or a generalized Hamiltonian structure - see [11]). We note that if we regard the position coordinates $\mathbf{r}_1 \dots \mathbf{r}_N$ and the momenta $\mathbf{p}_1 \dots \mathbf{p}_N$ as the internal state variables then Eqs. (1)-(4) become in fact governing equations of traditional rheological modeling (see more in [10]). Furthermore, Eqs. (1) and (2) may be viewed as describing the motion of particles suspended in a carrier fluid whose time evolution is governed by Eq. (3) [12]. Eq. (4) serves as a coupling link between (1-2) and (3). The Langevin dynamics is an alternative (to NEMD) approach to study suspensions at the microscopic level.

Following Lees and Edwards [13] (LE), we supplement the time evolution equations (1-4) with the "sliding bricks" boundary conditions (see [14] for a particularly enlightening explanation of the workings of this type of boundary conditions). In addition, we follow Evans *et al.* [15] and incorporate the Gauss thermostat multiplier (α) in Eqs. (1)-(4). From the physical point of view this modification is equivalent to the condition of constant temperature (see more on this in Sec. 3 and in [16]). In case of simple shear (Couette) flow, Eqs. (1)-(4) become SLLOD/LE/GAUSS,

$$\begin{aligned}
\dot{r}_{i\beta} &= \frac{p_{i\beta}}{m} + r_{i\gamma}\chi_{\gamma\beta}, && (8)\\
\dot{p}_{i\beta} &= F_{i\beta} - p_{i\gamma}\chi_{\gamma\beta} - \alpha p_{i\beta}, && (9)\\
\alpha &= \frac{\sum_{i=1}^{N}(F_{i\beta}p_{i\beta} - p_{i\beta}\chi_{\beta\gamma}p_{i\gamma})}{\sum_{i=1}^{N} p_{i\beta}p_{i\beta}}. && (10)
\end{aligned}$$

Now, the tensor $\underline{\underline{\chi}}$ takes the form,

$$\underline{\underline{\chi}} = \dot{\gamma} \begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} .$$

Three types of potentials are used in the simulation. Atom-atom interactions in atomic liquids and inter-particle interactions in dumbbell liquids are described by the Lennard-Jones (12-6) potential (LJ), whereas the intra-particle forces in dumbbell fluids are derived from the modified finitely extensible nonlinear elastic potential, due to Rudisill and Cummings [4], or from the generalized Lennard-Jones (GLJ) potential, to be defined below. Thus,

$$F_{i\beta} = \sum_{j=1,j\neq i}^{N} F_{ij\beta} = - \sum_{j=1,j\neq i}^{N} \frac{\partial\varphi(r_{ij})}{\partial r_{ij\beta}}, \tag{11}$$

where,

$$r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|.$$

For algorithmic simplicity we say that dumbbell 1 consists of beads 1 and 2, dumbbell 2 is built from beads 3 and 4, ..., dumbbell $\frac{N}{2}$ is constructed from beads $N-1$ and N. Hence,

$$\varphi_{ij}(r_{ij}) = \begin{cases} \varphi_{ij}^{LJ}(r_{ij}) & \text{for interdumbbell interaction} \\ \varphi_{ij}^{FENE}(r_{ij}) & \text{for intradumbbell interaction} \\ \varphi_{ij}^{GLJ}(r_{ij}) & \text{for intradumbbell interaction} \end{cases} \tag{12}$$

where, σ and ϵ are the Lennard-Jones parameters which denote the collision diameter and the depth of the potential well. The potentials are defined as ,

$$\varphi_{ij}^{LJ}(r_{ij}) = \begin{cases} 4\epsilon[(\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^{6}] & \frac{r_{ij}}{\sigma} < 2.5 \\ 0 & \frac{r_{ij}}{\sigma} \geq 2.5, \end{cases} \tag{13}$$

$$\varphi_{ij}^{FENE}(r_{ij}) = \begin{cases} -\frac{H(r_{max}-r_{eq})^2}{2} \, ln \, [1 - (\frac{r_{ij}-r_{eq}}{r_{max}-r_{eq}})^2] & r_{min} < r_{ij} < r_{max} \\ \infty & r_{ij} \leq r_{min} \; or \; r_{ij} \geq r_{max}, \end{cases} \tag{14}$$

and,

$$\varphi_{ij}^{GLJ}(r_{ij}) = \begin{cases} 4\epsilon[(\frac{\sigma}{r_{ij}})^{12} - A(\frac{\sigma}{r_{ij}})^6] & \frac{r_{ij}}{\sigma} < 2.5 \\ 0 & \frac{r_{ij}}{\sigma} \geq 2.5, \end{cases} \tag{15}$$

where H denotes a spring constant and A is an arbitrary parameter that modifies the attraction term in the classical LJ potential. For inter-molecular forces, the cut-off radius was set to 2.5σ. All intra-particle potentials studied are drawn in Fig. 1 and summarized in Tab. 1. r_{eq} denotes a midway position between the minimum (r_{min}) and the maximum extensions (r_{max}) of a single FENE dumbbell. Thus the FENE potential is symmetric around $r = r_{eq}$. In the other study [4], r_{eq} and r_{max} were chosen as 0.9σ and 1.4σ to eliminate the possibility of dumbbell bond crossings. This is also true here with the exception of the longest dumbbells. Arbitrarily, in case of GLJ liquids, a dumbbell is considered broken if its bead-bead extension exceeds 1.7σ. The forces between beads in a broken dumbbell, however minuscule, are taken into account above 1.7σ until the cut-off radius of 2.5σ.

From the microscopic perspective, the Lennard-Jones parameters defined at the molecular level, have no meaning for GLJ potential, and they cannot be meaningfully related to H, r_{eq}, nor to r_{max} of the FENE potential. In this light, SLLOD particles should be thought as renormalized pseudo-particles or effective particles.

We emphasize that bead-bead potentials result in central forces only. Furthermore, the interactions between two dumbbells are approximated by a sum of four Lennard-Jones site-site potentials. An alternative inter-molecular potential could be conceived by considering dumbbells as ellipsoids of revolution. Such a potential was developed by Gay and Berne [17] and was successfully applied to model the behavior of liquid crystals; see for example [18]. Currently, we investigate the Gay-Berne potential in conjunction with the SLLOD algorithm in the context of rheological modeling [19]. In general, the Gay-Berne potential, which could be considered as a generalization of the Lennard-Jones site-site potentials in case of multi-bead molecules, depends on the relative orientation of long axes of the two interacting molecules as well as on the vector that connects their centers-of-masses. It is also important to realize that neither of the above approaches makes an allowance for interactions among three or more dumbbells.

The starting configurations for NEMD runs are prepared by using the following procedure. Firstly, all beads with velocities drawn from the Maxwellian distribution are placed on the faced-centered-cubic (FCC) lattice. During the initial 500-1000 steps all velocities are rescaled in an ad-hoc manner to drive the system to the equilibrium; pragmatically, we consider the equilibrium to be reached if there is no drift in thermodynamic quantities such as hydrostatic pressure, potential energy or kinetic temperature. Over the next 10,000-30,000 steps, the system follows the classical molecular dynamics trajectory. Subsequently, the molecular dynamics equations are replaced by the SLLOD equations. The data are collected only after few thousand time steps, when the transitional effects, such as the stress overshoot, die out. In this study, we present the results that were obtained with Verlet neighborhood [20] switched off. We also use linked lists-cells but only in modeling of two dimensional systems [2].

Dimensionless quantities, denoted by * superscript and defined on the basis of the Lennard-Jones parameters, are used in the simulation. For example, the dimensionless mass, length, time and temperature are defined as,

$$M^{\star} = M\frac{1}{m}, \;\; L^{\star} = L\frac{1}{\sigma}, \;\; t^{\star} = t\sqrt{\frac{\epsilon}{m\sigma^2}}, \;\; T^{\star} = T\frac{k_B}{\epsilon}; \tag{16}$$

m denotes the mass of a single atom of argon, and k_B is the Boltzmann constant. Hence, dimensionless unit quantities of mass, length, time, and temperature correspond to $6.63 \times 10^{-26}\,kg$, $3.405 \times 10^{-10}\,m$, $2.156 \times 10^{-12}\,s$, and $119.8\,K$. It follows that dimensionless unit quantities of shear rate, energy, velocity, pressure, and viscosity are $4.64 \times 10^{11}\,s^{-1}$, $1.65 \times 10^{-21}\,J$, $157.9\,m/s$, $41.9 \times 10^{6}\,Pa$, and $9.03 \times 10^{-5}\,Pa\,s$. In molecular dynamics simulations, density is understood as a number of beads (atoms) in a unit cube of volume equal to σ^3, thus the dimensionless unit quantity of density of liquid argon, is synonymous to $1679.4\,kg/m^3$. Finally, the spring constant $H^\star$ and the dimensionless group $b^\star$ are written as,

$$H^\star = H\frac{\sigma^2}{\epsilon}, \;\; b^\star = \frac{H^\star\,(r^\star_{max} - r^\star_{eq})^2}{T^\star}. \tag{17}$$

Three series of numerical experiments are performed for atomic argon: close to its triple point ($T^\star = 0.722$, $\rho^\star = 0.842$), near the liquid-vapor line ($T^\star = 1$, $\rho^\star = 0.7$), and in the liquid phase ($T^\star = 1$, $\rho^\star = 0.8$). The rheological macroscopic properties of argon at the triple point for the whole range of $\dot\gamma^\star$ [21], and near the liquid-vapor line in the Newtonian regime [5] were investigated, but the other results are new. The dynamic properties of dumbbell fluids are determined at ($T^\star = 1$, $\rho^\star = 0.8$) for nine series of experiments, as indicated in Tab. 1.

The calculations were performed on work-station type computers, such as SUN SPARC II and IBM RISC 320. Rather long runs were necessary to minimize the effects of fluctuations; an average simulation demanded 8 hours of CPU time including the overhead for taking averages and storing parts of trajectories. The number of time steps in production runs was varied between 50,000, for $\dot\gamma^\star$ in the vicinity of 1, and 300,000 for $\dot\gamma^\star$ below 0.1. The time step was normally set to 0.005 but it was decreased to 0.001 at high shear rates.

The errors in viscometric functions were estimated in a pragmatic manner, similarly as it was done by Rudisill and Cummings [4]; physically more correct, but also more involved techniques for estimation of errors my be found in [3, 22]. Here, the variables were calculated at each time step and averaged over bins of at least 5,000 steps long. Since the shortest production runs lasted 50,000 steps, at least 10 bin averages were computed. Subsequently, these averages were considered as independent observations and used to calculate an overall average, and associated with it a standard deviation. Thus the mean values and the confidence intervals, plotted in the graphs, correspond to overall averages and one standard deviation, respectively.

3 Stress tensor and kinetic temperature

The fact that the equations of motion were introduced in the classical (atomic) rather than in molecular (center-of-mass) representation, requires a consistent treatment of the stress tensor, and of the Gaussian thermostating multiplier α; in molecular dynamics terminology, the negative of the stress tensor is often called the pressure tensor. In simple words, the atomic approach demands that six first order differential equations are solved for each bead, the stress tensor is placed at beads and the total kinetic energy is maintained constant by thermostating. On the other hand, if the equations of motion are cast into the molecular form, six equations are solved for translational velocities and coordinates of the center-of-mass. The other six equations are necessary to describe the motion of the bead-bead vector. In this representation, the stress tensor is positioned at the center-of-mass implying that the momentum transfer takes place when the center of mass crosses an arbitrary surface. In addition, the sum of the translational energies is kept constant [23] indicating that the vibrational and rotational energies are not included in the definition of α. In this study the stress tensor was placed at beads, whereas Rudisill and Cummings [4] followed the other approach.

In the atomic representation the total stress tensor is always symmetric due to the interactions under the central forces only. The stress tensor is a sum of the kinetic part, which is represented by the first term on the right hand side in Eq. 4, and the two potential contributions, which are due to the Lennard-Jones, FENE or GLJ interactions. In case of dense atomic liquids the kinetic part of $\sigma_{\alpha\beta}$ is $7-8$ times smaller than the potential term [5].

The viscometric functions, namely the viscosity (η), the first (N_1) and the second (N_2) normal stress differences, as well as the hydrostatic pressure (P) are defined as:

$$\begin{aligned}
\eta(\dot{\gamma}) &= \frac{\sigma_{xy}(\dot{\gamma})}{\dot{\gamma}}, & (18)\\
N_1(\dot{\gamma}) &= \sigma_{xx}(\dot{\gamma}) - \sigma_{yy}(\dot{\gamma}), & (19)\\
N_2(\dot{\gamma}) &= \sigma_{yy}(\dot{\gamma}) - \sigma_{zz}(\dot{\gamma}), & (20)\\
P(\dot{\gamma}) &= -\frac{1}{3}(\sigma_{xx}(\dot{\gamma}) + \sigma_{yy}(\dot{\gamma}) + \sigma_{zz}(\dot{\gamma})). & (21)
\end{aligned}$$

Similarly to the results from real experiments, N_1 and N_2 obtained from the computer simulations are associated with large relative errors, due to the fluctuations at low shear rates. Hence presentation of the results in terms of the first and the second normal stress coefficients $\psi_1(\dot{\gamma}) = \frac{N_1(\dot{\gamma})}{\dot{\gamma}^2}$ and $\psi_2(\dot{\gamma}) = \frac{N_2(\dot{\gamma})}{\dot{\gamma}^2}$ is impractical in the Newtonian regime; division of N_1 and N_2, which at $\dot{\gamma}^* \to 0$ are of the order of their standard deviations, by a very small shear rate leads to a large uncertainty in ψ_1 and ψ_2.

In general, the temperature of any macroscopic system is specified in terms of the thermodynamic temperature which is defined as the partial derivative of the total internal energy of the system with respect to its entropy. In order to ensure the compatibility of the NEMD results with macroscopic phenomena, it is necessary to find the thermodynamic temperature for microscopic systems. This calls for the calculation of the entropy from NEMD simulations. A first attempt of this type was reported by Evans [24] who used histograms to approximate the distribution function at low density for a small number of molecules. At low shear rates the entropy of a fluid consisting of 108 atoms was investigated by Baranyai and Evans [25]. It has been observed that at low densities, the thermodynamic temperature is smaller than the kinetic temperature and the discrepancy increases with shear rate [24]. We are not aware of a technique that would allow to determine the entropy or the thermodynamic temperature for a system similar to ours. We recognize the fact that it is not at all clear how the kinetic temperature relates to the thermodynamic temperature, in case of nonequilibrium steady states; for monatomic fluids modeling an ideal gas at equilibrium both temperatures are the same. The uncertainty in establishing the temperature will in general be also carried over to pressure, since the kinetic temperature is related to the trace of the kinetic contribution to the stress tensor, as follows from Eq. 4. Therefore, the thermodynamic pressure, defined as the negative of the partial differential of the total internal energy of the system with respect to its volume, may not correspond to the hydrostatic pressure which is equal to be the negative of one third of the trace of the stress tensor. Motivated by the necessity, we employ the kinetic temperature and the hydrostatic pressure which are convenient in comparing NEMD results among themselves. However, the temperature and the pressure defined in this manner might not be reliable in correlating experimental and NEMD data.

We conclude this section by making a comment that the molecular representation seems to be more frequently used in conjunction with shorter molecules [23]. This is true especially for the mesoscopic elastic dumbbell theories which tend to be developed in the configurational (position) space only. However, it is known that at least in the Newtonian limit, the use of the atomic and molecular representations leads to the same predictions. Moreover, at higher shear rates, the distribution functions that are produced by the both sets of the equations of motion correspond well with the Zwanzig solution for dilute gases [23]. Thus, it appears that the choice of either of the representations is arbitrary.

4 Oscillating diatoms and rigid rotors

Historically, there have been two major advantages in using rigid rotors in place of vibrating diatomic molecules. It has been deemed that the rigid molecules are able to incorporate most properties of real homonuclear molecules yielding predictions that correspond closely to those obtained for oscillating diatoms, at lower computational costs. Although the former reason is a valid one, the latter is of less importance in modeling of dumbbell fluids.

A FENE dumbbell is a theoretical concept that has proved itself to be extremely useful as a starting point, in the formulation of the rheological models [26]. The microscopic homonuclear molecules are characterized by a high frequency of vibrations that is of an order of magnitude higher than that of the frequencies associated with translations and rotation. This necessitates a very small time step that corresponds to a fraction of the vibrational period. In this context, the rigid rotors approximation give significant savings in computational time by elimination of fast vibrations, however at a cost of more complex computational schemes that must incorporate the constrained bond length. On the other hand, the FENE potential has its origins in the macroscopic rheology where it was first introduced as an improvement over the simple infinitely extensible Hookean spring potential. In the mesoscopic elastic dumbbell theories the strength of the FENE bead-bead bond is considered to be an adjustable parameter which is included in the definition of b^* [26]; see Eq. 17. The microscopic effect of this is such that the vibrational motion of beads has a frequency which is comparable to the other frequencies present in the system. In addition, the FENE spring constant may be increased to result, in the limit, in the strongly bound microscopic-like molecules. As a matter of fact, this was the philosophy followed in this study.

In the paper, a pseudo-rigid rotor fluid is considered as a special case of the FENE dumbbell fluids, namely $H^* \rightarrow \infty$. According to the principles of statistical mechanics, the vibrational energy reservoir for N dumbbells always stores $\frac{N\,T^*}{2}$ amount of energy. This is why vibrations become faster but are of lower magnitude as the bead-bead bond comes to be stiffer. From the perspective of statistical mechanics, rigid rotor limit cannot be reached just by letting $H^* \rightarrow \infty$. In order to investigate rigid rotors one has to replace Eqs. (1)-(3) by the time evolution equations that take into account rigid body motion. Results of this type of simulations will be reported in a separate publication [19].

The inset in Fig. 2a shows vibrations of a single representative FENE dumbbell in unstrained isokinetic fluids as a function of the spring constant; we recall that the equilibrium position corresponds to 0.9 and the maximum and minimum extensions to 1.4 and 0.4, respectively. In order to determine the frequency of vibrations, the bead-bead distances of all dumbbells were Fourier transformed separately and then averaged to produce the amplitude spectra presented in Fig. 2a. A complex function $G(f)$, where f is the frequency, may be expressed in terms of its amplitude $A(f)$ and phase $\phi(f)$ spectra, according to

$$G(f) = A(f)\, e^{i\phi(f)}, \tag{22}$$

where, i denotes $\sqrt{-1}$, $A(f)$ is real and positive (see p. 372 of ref. [27]).

At high values of the spring constant the bond becomes stiff, but the frequency of vibrations increases. $H^{\star} = 37.5, 375, 3750, 37500$ correspond to $b^{\star} = 9.375, 93.75, 937.5, 9375$, for the state point investigated in this study; in mesoscopic elastic dumbbell models $b^{\star}$ is usually between 30 and 1000 for infinitely diluted solutions [28]. At $H^{\star} = 3750$, the frequency of vibrations is around 14 and hence it is necessary, even at small shear rates, to decrease the time step to 2.5×10^{-3} so it corresponds to approximately 30 integrations per one vibration. In Sec. 5, it will become clear that the viscometric functions at $H^{\star} = 375$ and 3750 are nearly the same, therefore no simulations are performed for $H^{\star} = 37500$.

In a similar manner, dominant frequencies of vibrations were identified for unstrained GLJ liquids (Fig. 2b). These frequencies are similar as for FENE liquids with the spring constant being equal to between 375 and 3750. For $A = 3$, broken dumbbell were not taken into consideration in determining of the average amplitude spectrum.

On the assumption that at equilibrium the total energy per isolated FENE dumbbell $E^{\star}_{id}$ may be approximated as the sum of the average potential intra-particle and the vibrational kinetic energies,

$$E^{\star}_{id} = \frac{\Phi^{\star}_{intra} + E^{\star}_{vib}}{N/2} = \frac{64. + 64.}{128} = 1.0, \tag{23}$$

and neglecting the dumbbell-dumbbell interactions, it is possible to calculate the frequency of vibrations of a single FENE dumbbell in one dimension as,

$$f^{\star} = \left\{ \int_0^{x^{\star}_1} \frac{2}{\sqrt{E^{\star}_{id} + \frac{H^{\star}(r^{\star}_{max}-r^{\star}_{eq})^2}{2} ln\,(1 - \frac{x^{\star 2}}{(r^{\star}_{max}-r^{\star}_{eq})^2})}} d\,x^{\star} \right\}^{-1}, \tag{24}$$

where,

$$x^{\star}_1 = (r^{\star}_{max} - r^{\star}_{eq})\sqrt{1 - exp(-\frac{2\,E^{\star}_{id}}{H^{\star}(r^{\star}_{max} - r^{\star}_{eq})^2})}. \tag{25}$$

The numerical values of $E^{\star}_{vib}$ and Φ_{intra} were taken from Figs. 12d and 13a in [16], in the Newtonian limit. As shown in Tab. 2, there is an excellent agreement between the predicted frequency of vibration of a single FENE dumbbell and the dominant frequency of vibration determined from the Fourier analysis for dumbbell liquids (Fig. 2a).

It is worthwhile to add that a typical period of oscillation related to translational motion was estimated by Hess [5] from the Einstein frequency to be 0.4, in case of atomic fluids. This is very close to the predicted period of vibration for $H^{\star} = 37.5$ and shows that, at equilibrium for small values of the spring constant, there is only one time scale present in the system.

As illustrated in Fig. 2c for $H^{\star} = 375$, the dominant frequency of vibration of the unstrained fluid is replaced by an amplitude spectrum that contains high and low frequencies, under shear. The former may be associated with the long range slow structural rearrangements that take place in the fluid, especially at high shear rates. The latter, characterized by smaller magnitudes, might be explained by fast vibrations of single dumbbells which are locked inside linear structures (strings), and are unable to rotate nor vibrate freely due to the caging effect exerted by neighboring strings [16].

It is important to observe that frequencies of vibration of dumbbells have almost no impact on the viscometric functions. One notable exception is a small reduction in the Newtonian viscosities of dumbbell fluids at higher frequencies of vibrations. Rather, the macroscopic results seem to be strongly correlated with an anisotropic structure whose existence is evident even at the smallest shear rates from the results at the mesoscopic level of description [16]. This spatial structure is related to the packing of dumbbells in the primitive cube and hence it depends on the average end-to-end extension of a single dumbbell. For this reason, Fig. 3 summarizes averages of the square of end-to-end distances for FENE and GLJ dumbbells, as functions of the shear rate.

GLJ dumbbells tend to break, especially for $A = 3$ and at high shear rates, as plotted in Fig. 4. The breakage of GLJ dumbbells may be considered as a microscopic realization of the shear-induced degradation (extension of polymeric chains) of polymeric liquids, as was done by Hess [5] for short polymeric chains. An excessive degradation of a dumbbell liquid leads in the limit to atomic fluid-like behavior. Also, the number of broken dumbbells depends on the size of the primitive cube, since two beads are allowed to recombine only if they originally belonged to the same dumbbell. The likelihood of a bead encountering the other matching bead is related to the size of the primitive cube. If trajectories of beads are calculated in the so-called physical coordinates, as oppose to the virtual coordinates which are confined to the primitive cube [16], then all dumbbells may be broken relatively fast, for example, within 150 000 time steps ($\Delta t^{\star} = 0.0025$), at $\dot{\gamma}^{\star} = 5.5$ for $A = 3$.

Even for FENE liquids, for $\dot{\gamma}^{\star} > 5$ and for $H^{\star} = 37.5$ dumbbells break; in numerical sense breaking occurs when the bead-bead distance is either smaller or larger than allowed by the FENE potential. The breaking is eliminated by an infrequent "widening" of the FENE potential curve to force the offending dumbbells to fall within the permitted limits. Since this procedure is invoked only a few times during simulation it has no effect on the final result.

5 Viscometric functions

In order to investigate the finite size effects in the macroscopic results for monatomic fluids, several simulations were performed for 108 and 512 particles. With the exception of runs consisting of 108 atoms at high shear rates, all macroscopic results were reproducible for different populations within the statistical uncertainty. Since at the very high shear rates, not studied here, differences were observed between systems consisting of 256 and 2048 molecules [21] it stands to reason to suggest that the particle number effects are in general a function of the shear rate. That is, larger systems must be investigated at elevated shear rates.

All viscosity curves are alike but not identical (Figs. 5 a-d). In case of atomic fluids, the Newtonian viscosity appears to be independent of pressure, but varies with temperature. This comes as no surprise since such a behavior is consistent with the classical kinetic theory of viscosity. The transition between the Newtonian and the non-Newtonian regimes is sudden, and it corresponds well with the structural rearrangements (see also [16]) taking place at the microscopic level. The slopes of the viscosity curves are exactly the same for argon in the liquid phase and close to the liquid-vapor line, but they are different than the slope of the viscosity line close to the argon triple point.

As a result of the same inter-particle potential used, the shapes of the viscosity curves for the FENE fluids for $r^\star_{eq} = 0.9$ and $r^\star_{max} = 1.4$ are reminiscent of those of atomic liquids. However, as a consequence of the introduced intra-molecular interactions, the transition from the Newtonian regime is shifted to higher shear rates. The Newtonian viscosity of the FENE fluids increases with the spring constant to attain the highest value for the pseudo-rigid rotors fluid, that is, for $H^\star = 3750$ (Fig. 5b). This indicates that for dumbbell fluids the Newtonian viscosity depends on the type of the intra-particle potential. At the lowest shear rates the rotational and the vibrational kinetic energies are not well equilibrated for $H^\star = 3750$ [16], suggesting that for this spring constant the viscosity in the limit of small shear rates has not yet reached its Newtonian value.

Although the onset of shear thinning and the slopes of the viscosity curves in the non-Newtonian regime are independent of the average end-to-end distance of FENE dumbbells, the Newtonian viscosity is not. As a matter of fact, the viscosity in the limit of small shear rates increases rapidly for longer dumbbells, as shown in Fig. 5c. For very small dumbbells ($r^\star_{eq} < 0.67$) and for $0.2 < \dot{\gamma}^\star < 2.0$, the computational fluid separates into two horizontal layers: a void space and a layer that contains all the particles. At the macroscopic level of description this leads to an abrupt reduction in shear viscosity (Fig. 5c).

Interestingly, most of the characteristics of the viscosity curves of FENE liquids seem to be preserved for the generalized Lennard-Jones liquids. For example, the onset of shear thinning is delayed to higher shear rates than for atomic fluids, and the viscosity decreases as rapidly as for FENE dumbbells liquids, in the non-Newtonian region. In addition, the viscosity of GLJ dumbbell fluids for $A = 3$ approaches the viscosity of atomic fluids due to breaking of GLJ dumbbells, at intermediate shear rates (Fig. 5d). Lastly, the Newtonian viscosity for all GLJ fluids is similar as a result of opposing effects related to dumbbell size and rigidity of the bead-bead bond; the GLJ dumbbells at $A = 5$ are the longest but also most rigid.

The first and the second normal stress differences, predicted from the FENE and GLJ models (Figs. 6&7) and also observed experimentally for viscoelastic liquids, are positive and negative, respectively. However, our micro-rheological models foretell that $-N^\star_2$ is approximately half of $N^\star_1$, whereas for polymeric fluids it is known that $-N^\star_2$ remains below 20% of $N^\star_1$ [29]. In the Newtonian limit, $N^\star_1$ tends to zero, for both atomic and dumbbell fluids. As opposed to liquid argon, where $N_1^\star$ is insignificant for amorphous phase, and sharply increases only in the ordered region, for dumbbell fluids $N_1^\star$ grows gradually until it attains a local maximum at $\dot{\gamma}^\star \simeq 5$ that corresponds to the phase change. In contrast, the second normal stress difference, for both types of fluids decreases monotonically until the random phase is replaced by strings, as observed at the microscopic level [16]. Not surprisingly, $N_1^\star$ and $N_2^\star$ peak at higher absolute values for FENE and GLJ than atomic fluids indicating that dumbbell fluids are more viscoelastic.

At high shear rates, $N_1^\star$ and $N_2^\star$ reverse their trends and either descend towards zero or remain approximately constant; note that scatter in the dumbbell data is noticeably larger than in the atomic results, at high shear rates. At the microscopic level, this behavior may be explained by the observed formation of strings that move with ease in the direction of flow as opposed to the disordered phase that has no dominant direction. This leads to lower σ_{xx}, but also to comparable σ_{yy} and σ_{zz}. It should be noted that some curves of $N_1^\star$ and $N_2^\star$ have two local extrema, *e.g.* Fig. 8a; the second extremum, at higher shear rate may signify secondary structural rearrangements. Finally, the functional dependence of $\eta^\star$, $N_1^\star$, and $N_2^\star$ with respect to $\dot{\gamma}^\star$, as shown in Figs. 5b, 6b and 7b, remains similar for $H^\star = 375$ and $H^\star = 3750$ confirming that the pseudo-rigid rotor limit has been reached.

Very small FENE dumbbells that show phase separation for $\dot{\gamma}^\star$ of between 0.2 and 2.0 manifest slightly different behavior. Specifically, N_1 becomes negative around $\dot{\gamma}^\star = 1$ (Fig. 6c) and N_2 exhibit a slight drop for $\dot{\gamma}^\star = 0.35 =$ (Fig. 7c). This is reminiscent of experimental and computational observation made for liquid crystal polymers (see Figs. 9&10 in [30] and Fig. 12 in [31]).

Both atomic and dumbbell fluids exhibit volume dilatancy which manifests itself by the increase in pressure with shear rate (Figs. 8 a-d). For this reason, the viscosity tends to be higher for isochoric than for isobaric systems, as it was shown by Hood *et al.* [32]. The trend in the pressure is closely followed by the inter-particle potential energy [16], which is at the minimum in the Newtonian limit, but then increases due to the structural anisotropy which impedes dumbbells from assuming random positions.

6 Comparison to the elastic dumbbell theory

In the microscopic simulations, the imposed velocity profile is expressed in terms of forces that act on each particle. This approach leads to several consequences. For example, one may rightfully argue that the molecular simulations are no longer truly microscopic since they incorporate the imposed velocity in a way that would be physically appropriate for magnetic, electric or gravity field. In addition, NEMD simulations do not ensure the incompressibility of molecular fluids. This is similar to a problem that we have encountered in modeling of flow of two dimensional liquids [2]. In brief, it is not possible as yet to model the pressure driven flows directly, rather a uniform gravity field has to be applied to all particles in the system.

Another important difference between microscopic and mesoscopic rheological models is the use of boundary conditions. In mesoscopic modeling, the flow field is given, usually either as the steady shear or the steady elongational flow. It is immaterial how the flows are induced. Equations of state are then solved to obtain the material functions. On the other hand, the microscopic simulations do require the explicit implementation of flow. This is accomplished through Lees-Edwards boundary conditions alone, or in conjunction with the SLLOD equations of motion.

Fig. 5 reveals that both Lennard-Jones atomic and FENE dumbbell fluids exhibit shear thinning with power law slopes of less than -1, $\eta(\dot{\gamma}) = k\,\dot{\gamma}^{n-1}$, thus

$n < 0$. This implies that the criterion for stability,

$$\frac{d\sigma_{xy}}{d\dot{\gamma}} > 0, \tag{26}$$

which requires that the shear stress increases with the shear rate, is not satisfied [29], since

$$\frac{d\sigma_{xy}}{d\dot{\gamma}} = kn\dot{\gamma}^{n-1} < 0. \tag{27}$$

For atomic fluids, it has been noted by Heyes [33], that there is a limiting value of the shear stress that can be maintained by the amorphous liquid before it is compelled to reorganize. This is also true for dumbbell liquids. The above-mentioned criterion for stability does not incorporate elasticity and should be considered as an approximation. In fact, Demarquette and Dealy have recently observed that for concentrated polystyrene n may be marginally less that zero [34]. For completeness, it should be added that NEMD rheological simulations of long n-alkanes, such as hexadecane [6], lead to power law slopes that fall within the range predicted by the stability analysis. It is believed that this behavior is due to complex intramolecular interaction within a single chain of alkanes. In other words, dumbbells are too simple to account for complex viscoelastic behavior of polymeric liquids.

Similarly to microscopic results, mesoscopic dumbbell theories predict power law slopes that are independent of the value of the spring constant. However, in contrast to our dumbbells the slopes do not violate the criterion for stability. The mesoscopic elastic dumbbell models are unable to predict monotonically decreasing ψ_1 or N_2 [28, 26].

7 Conclusions

The rheological models that have been developed in the context of NEMD still have many drawbacks. The major disadvantage is the lack of an efficient technique to calculate the entropy as well as the thermodynamic temperature and the thermodynamic pressure. This implies that the averaged microscopic results may not be related, with certainty, to macroscopic phenomena. By following a similar reasoning, one may question the thermostat which was designed to maintain constant the kinetic rather than the thermodynamic temperature. It is a common practice to consider molten polymers as incompressible fluids and yet no NEMD technique exists that could explicitly use incompressibility in establishing a flow pattern. With respect to the Erpenbeck string phase [35], much attention must be given to investigate its very existence, and thus to verify or disprove the structural stability of the model, at high shear rates; see more on this in [16]. This may only be accomplished for very large systems with a sufficient number of molecules that would allow to develop a profile unbiased thermostat in three dimensions with an adequate statistics. There are indications, such as the viscosity curve that implies instability at the macroscopic level, that the string phase is indeed stabilized by the Gaussian thermostat.

Scaling is yet another problem that is implicit in micro-rheological models that incorporate macromolecular structure by means of empirical potentials. Here, two questions are left unanswered: (1) how are macroscopic parameters of polymeric fluids translated to microscopic level, and (2) how is shear rate scaled back to experimentally attainable values ?

At the macroscopic level, the viscometric functions are obtained for atomic and molecular fluids whose bonded interactions are modeled by FENE or GLJ potentials. The FENE and GLJ fluids are studied with respect to the strength of the bonded interactions and to the average size of a single dumbbell. The phenomena associated with viscoelastic fluids, such as shear thinning, Newtonian viscosity, non-zero normal stress differences, and volumetric dilatancy are observed for atomic but also for dumbbell fluids. Atomic liquids tend to thin at lower shear rates and are less elastic than dumbbell liquids.

Although FENE and GLJ potentials are quite distinct, the macroscopic behavior of both types of liquids is similar, if the breakage of GLJ dumbbells is small ($A \geq 4$) and the average dimensionless end-to-end extension of FENE dumbbells is maintained close to 0.9. The resemblance of the macroscopic properties of FENE and generalized Lennard-Jones dumbbells is a manifestation of simplicity of both models. More complex intra-molecular potentials would have to be investigated to account for observed variation in the experimental results of polymeric fluids. In addition, the role played by inter-molecular potentials in determining the Newtonian viscosity of dumbbell fluids is yet to be investigated. Apparently, inter- rather than intra-molecular potentials are important in case of the Newtonian viscosity of n-alkanes [6]. For dumbbell fluids, the choice of an intra-molecular potential seems to have no effect on the slopes of viscosity curves in the non-Newtonian regime, nor on the location of the onset of shear thinning.

Very small FENE dumbbells exhibit phase separation and associated with it reduction in viscosity and change in shapes of N_1 and N_2, at intermediate shear rates. As a matter of fact, both the first and the second normal stress differences show trends that are characteristic of liquid crystal polymers, but more investigations are necessary to confirm these trends for systems consisting of large number of particles.

It is noted that the properties of dumbbell liquids are related to the conformation of single dumbbells, especially their average end-to-end extensions, and that the frequency of vibrations of dumbbells is less important. In addition, the viscometric functions of oscillating and pseudo-rigid rotor dumbbells ($H^* \to \infty$) are similar, but in the Newtonian limit. These observation should be considered as a molecular dynamics justification for the "rigid-in-average" assumption introduced in some mesoscopic elastic dumbbell models [36, 37].

8 Acknowledgments

We wish to thank Professor C. Cohen of Cornell University for helpful discussions. Financial support from the *Natural Sciences and Engineering Research Council of Canada* and *Fonds pour la formation de chercheurs et l'aide à la recherche* of the Province of Québec is gratefully acknowledged.

References

[1] D. D. Joseph. *Fluid Dynamics of Viscoelastic Liquids.* Springer Verlag, New York, 1990.

[2] B. Z. Dlugogorski, M. Grmela, and P. J. Carreau. Direct numerical studies of viscous flow of two dimensional liquids. *In preparation, École Polytechnique*, 1993.

[3] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids.* Clarendon Press, Oxford, 1987.

[4] J. W. Rudisill and P. T. Cummings. The contribution of internal degrees of freedom to the non-Newtonian rheology of model polymer fluids. *Rheol. Acta*, 30:33–43, 1991.

[5] S. Hess. Rheological properties *via* nonequilibrium molecular dynamics: From simple towards polymeric liquids. *J. Non-Newt. Fl. Mech.*, 23:305–319, 1987.

[6] A. Berker, S. Chynoweth, U. C. Klomp, and Y. Michopoulos. Non-equilibrium molecular dynamics (NEMD) simulations and the rheological properties of liquid n-hexadecane. *J. Chem. Soc., Faraday Trans.*, 88(13):1719–1725, 1992.

[7] G. P. Morriss, P. J. Daivis, and D. J. Evans. The rheology of *n* alkanes: Decane and eicosane. *J. Chem. Phys.*, 94(11):7420–7433, 1991.

[8] K. Kremer and G. S. Grest. Dynamics of entangled linear polymer melts: A molecular-dynamics simulation. *J. Chem. Phys.*, 92(8):5057–5086, 1990.

[9] D. J. Evans. Nonequilibrium molecular dynamics. In Ciccotti G. and W.G. Hoover, editors, *Molecular-Dynamics Simulation of Statistical-Mechanical Systems*, pages 221–240. North-Holland, Oxford, 1986.

[10] M. Grmela. Coupling between microscopic and macroscopic dynamics in NEMD. *Phys. Lett. A*, 174:59–65, 1993.

[11] M. Grmela. Thermodynamic and rheological modeling: Polymeric liquid crystals. In A.A. Collyer and L.A. Utracki, editors, *Polymer Rheology and Processing*. Elsevier, London, 1990.

[12] M. Grmela. Mesoscopic hydrodynamics of suspensions. *Proc. Royal Soc. Lond. A (submitted)*, 1993.

[13] A. W. Lees and S. F. Edwards. The computer study of transport processes under extreme conditions. *J. Phys. C: Solid State Phys.*, 5:1921–1929, 1972.

[14] D. M. Heyes. Shear flow by molecular dynamics. *Physica B*, 131:217–226, 1985.

[15] D. J. Evans, W. G. Hoover, B. C. Failor, B. Moran, and A. J. C. Ladd. Nonequilibrium molecular dynamics *via* Gauss's principle of least constraint. *Phys. Rev. A*, 28:1016–1021, 1983.

[16] B. Z. Dlugogorski, M. Grmela, and P. J. Carreau. Microscopic and mesoscopic results from non-equilibrium molecular dynamics modeling of FENE dumbbell liquids. *J. Non-Newt. Fl. Mech. (in press)*, 1993.

[17] J. G. Gay and B. J. Berne. Modification of the overlap potential to mimic a linear site-site potential. *J. Chem. Phys.*, 74(6):3316–3319, 1981.

[18] E de Miguel, L. F. Rull, M. K. Chalam, and K. E. Gubbins. Dynamics of the Gay-Berne fluid. *Phys. Rev. A*, 45:3813–3822, 1992.

[19] B. Z. Dlugogorski, M. Grmela, P. J. Carreau, and G. Lebon. Rheology of several hundred rigid bodies. *J. Chem. Phys. (submitted)*, 1993.

[20] L. Verlet. Computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.*, 159(1):98–103, 1967.

[21] D. M. Heyes. Shear thinning and thickening of the Lennard-Jones liquid. *J. Chem. Soc., Faraday Trans. 2*, 82:1365–1383, 1986.

[22] D. M. Heyes. Transport coefficients of Lennard-Jones fluids: A molecular-dynamics and effective hard-sphere treatment. *Phys. Rev. B*, 37(10):5677–5696, 1988.

[23] A. J. C. Ladd. Equations of motion for non-equilibrium molecular dynamics simulations of viscous flow in molecular fluids. *Mol. Phys.*, 53(2):459–463, 1984.

[24] D. J. Evans. On the entropy of nonequilibrium states. *J. Stat. Phys.*, 57(3/4):745–758, 1989.

[25] A. Baranyai and D. Evans. Comments on thermodynamic integration methods for the determination of nonequilibrium entropy. *Mol. Phys.*, 74(2):353–365, 1991.

[26] R. B. Bird, C. F. Curtiss, R. C. Armstrong, and O. Hassager. *Dynamics of Polymeric Liquids*, volume 2. John Wiley & Sons, New York, 2 edition, 1987.

[27] W. M. Telford, L. P. Geldart, R. E. Sheriff, and D. A. Keys. *Applied Geophysics*. Cambridge University Press, London, 1976.

[28] G. A. Bird, P. J. Dotson, and N. L. Johnson. Polymer solution rheology based on a finitely extensible bead-spring chain model. *J. Non-Newt. Fl. Mech.*, 7:213–235, 1980.

[29] R. I. Tanner. *Engineering Rheology*. Clarendon Press, Oxford, 2 edition, 1988.

[30] J. J. Magda, Seong-Gi Baek, and K. L. De Vries. Shear flows of liquid crystal polymers: Measurements of the second normal stress difference and the Doi molecular theory. *Macromolecules*, 24:4460–4468, 1991.

[31] R. G. Larson. Arrested tumbling in shearing flows of liquid crystal polymers. *Macromolecules*, 23:3983–3992, 1990.

[32] L. M. Hood, D. J. Evans, and H. J. M. Hanley. Properties of soft-sphere liquid from non-Newtonian molecular dynamics. *J. Stat. Phys.*, 57(3/4):729–743, 1989.

[33] D. M. Heyes. The nature of extreme shear thinning in simple liquids. *Mol. Phys.*, 57(6):1265–1382, 1986.

[34] N. R. Demarquette and J. M. Dealy. Nonlinear viscoelasticity of concentrated polystyrene solutions: Sliding plate rheometer studies. *J. Rheol.*, 36(6):1007–1032, 1992.

[35] J. J. Erpenbeck. Shear viscosity of the hard-sphere fluid *via* nonequilibrium molecular dynamics. *Phys. Rev. Lett.*, 52(15):1333–1335, 1984.

[36] M. Grmela and P. J. Carreau. Conformation tensor rheological models. *J. Non-Newt. Fl. Mech.*, 23:271–294, 1987.

[37] M. Grmela. Dependence of the stress tensor on the intramolecular viscosity. *J. Rheol.*, 33(2):207–231, 1989.

Figure Captions

- Figure 1. Inter- (GLJ, $A = 1$) and intra-molecular potentials used to construct dumbbell liquids; $T^\star = 1$, $\rho^\star = 0.8$. GLJ with A=1 implies classical Lennard-Jones (12-6) potential. The reader may note that the effect of changing $H^\star$ in the FENE potential is shown in Figures 5b, 6b, 7b and 8b. Similarly, the consequence of varying $r^\star_{eq}$ and $r^\star_{max}$ in the FENE potential may be examined in Figures 5c, 6c, 7c and 8c. The result of increasing the attraction term in the GLJ potential may be observed in Figures 5d, 6d, 7d and 8d.

- Figure 2. Averaged amplitude spectrum of the bead-bead extension, with the zero frequency amplitude removed: a) unstrained FENE fluids ($r^\star_{eq} = 0.9$, and $r^\star_{max} = 1.4$). in the inset, vibrations of a single FENE dumbbell, represented by the change in the end-to-end distance, as a function of time, b) unstrained GLJ fluids, c) FENE fluid under shear $T^\star = 1$, $\rho^\star = 0.8$, $H^\star = 375$, $r^\star_{eq} = 0.9$, and $r^\star_{max} = 1.4$. Vibrations of all unbroken dumbbells during 8192 time steps were transformed separately into the frequency domain and then the average amplitude spectra were computed. The zero frequency amplitude approximates the end-to-end distance that corresponds to the minimum in the intra-molecular potential. For example, for GLJ fluids at $\dot{\gamma}^\star = 0$ this distance is 0.867, 0.898, and 0.954 for $A = 3, 4, 5$, respectively.

- Figure 3. Average of the square of the end-to-end distance for a few selected FENE and GLJ dumbbell liquids; $T^\star = 1$, $\rho^\star = 0.8$. All GLJ and FENE dumbbells with $r^\star_{eq} = 0.9$ and $r^\star_{max} = 1.4$ are similar in size "on average" although their vibrational frequencies may differ. The macroscopic results (Figures 5, 6, 7 and 8) depend on average size of the dumbbells, but not on their vibrational frequencies, neither on the detailed shape of their intramolecular potentials.

- Figure 4. Average number of GLJ dumbbells that are not broken; $T^\star = 1$, $\rho^\star = 0.8$. A dumbbell is considered broken if the end-to-end distance exceeds 1.7σ. GLJ dumbbell liquid at $A = 3$ was prepared from GLJ at $A = 4$ by a linear decrease in A over 10000 steps.

- Figure 5. Viscosity as a function of shear rate for: (a) atomic, (b,c) FENE, and (d) GLJ dumbbell fluids; dumbbell fluids at $T^\star = 1$, $\rho^\star = 0.8$. Multiple symbols at a given shear rate denote replicates. Note large errors at very small and very large shear rates. The former are the manifestation of large fluctuation in relation to the imposed external force. The latter stem from the interplay between the shear induced long range anisotropic structure in the fluid that may be stabilized by the linear thermostat and a small size of the primitive cube. In Figure c, the arrows indicate a dramatic decrease in viscosity due to separation of FENE liquids which are composed of very small dumbbells ($r^\star_{eq} < 0.67$). For example, for a FENE fluid defined by $H^\star = 375$, $r^\star_{eq} = 0.5$ and $r^\star_{max} = 0.9$, for $\dot{\gamma}^\star = 0.35\,, 0.6\,, 0.8\,, 1.5$, $\eta^\star = 0.050_{17}\,, 0.013_{17}\,, 0.000_{15}\,, 0.0001_{15}$, respectively; the numbers written in subscript signify one standard deviation.

- Figure 6. Shear rate dependence of first normal stress difference for: (a) atomic, (b,c) FENE, and (d) GLJ dumbbell fluids; FENE fluids at $T^\star = 1$, $\rho^\star = 0.8$. Note limited reproducibility of replicates in the shear thinning regime.

- Figure 7. The variation in second normal stress difference against shear rate for: (a) atomic, (b,c) FENE, and (d) GLJ dumbbell fluids; dumbbell fluids at $T^\star = 1$, $\rho^\star = 0.8$.

- Figure 8. Pressure as a function of shear rate for: (a) atomic, (b,c) FENE, and (d) GLJ dumbbell fluids; dumbbell fluids at $T^\star = 1$, $\rho^\star = 0.8$. The results were not corrected for the effect of the cut-off radius. Note large increase in pressure for long FENE dumbbells, in Figure c.

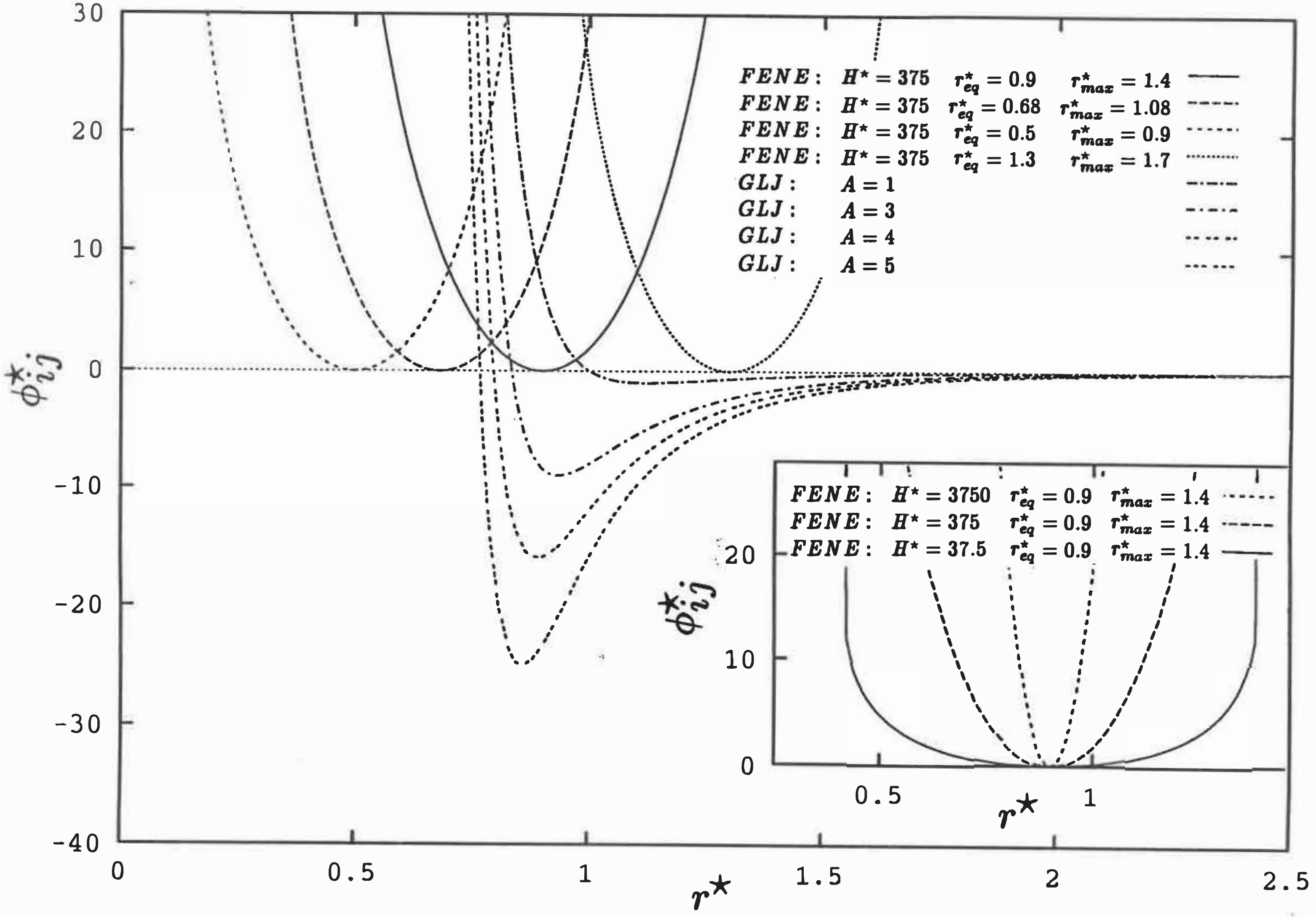
Table 1: Dimensionless parameters used to define FENE and GLJ dumbbells.

Potential	H^*	r^*_{eq}	r^*_{max}	A
FENE	37.5	0.9	1.4	-
FENE	375	0.5	0.9	-
FENE	375	0.68	1.08	-
FENE	375	0.9	1.4	-
FENE	375	1.3	1.7	-
FENE	3750	0.9	1.4	-
GLJ	-	-	-	3
GLJ	-	-	-	4
GLJ	-	-	-	5

Table 2: Predicted (Equations 23-25) and observed dominant (Figure 2a) frequencies of vibrations of a single dumbbell in an unstrained FENE fluid.

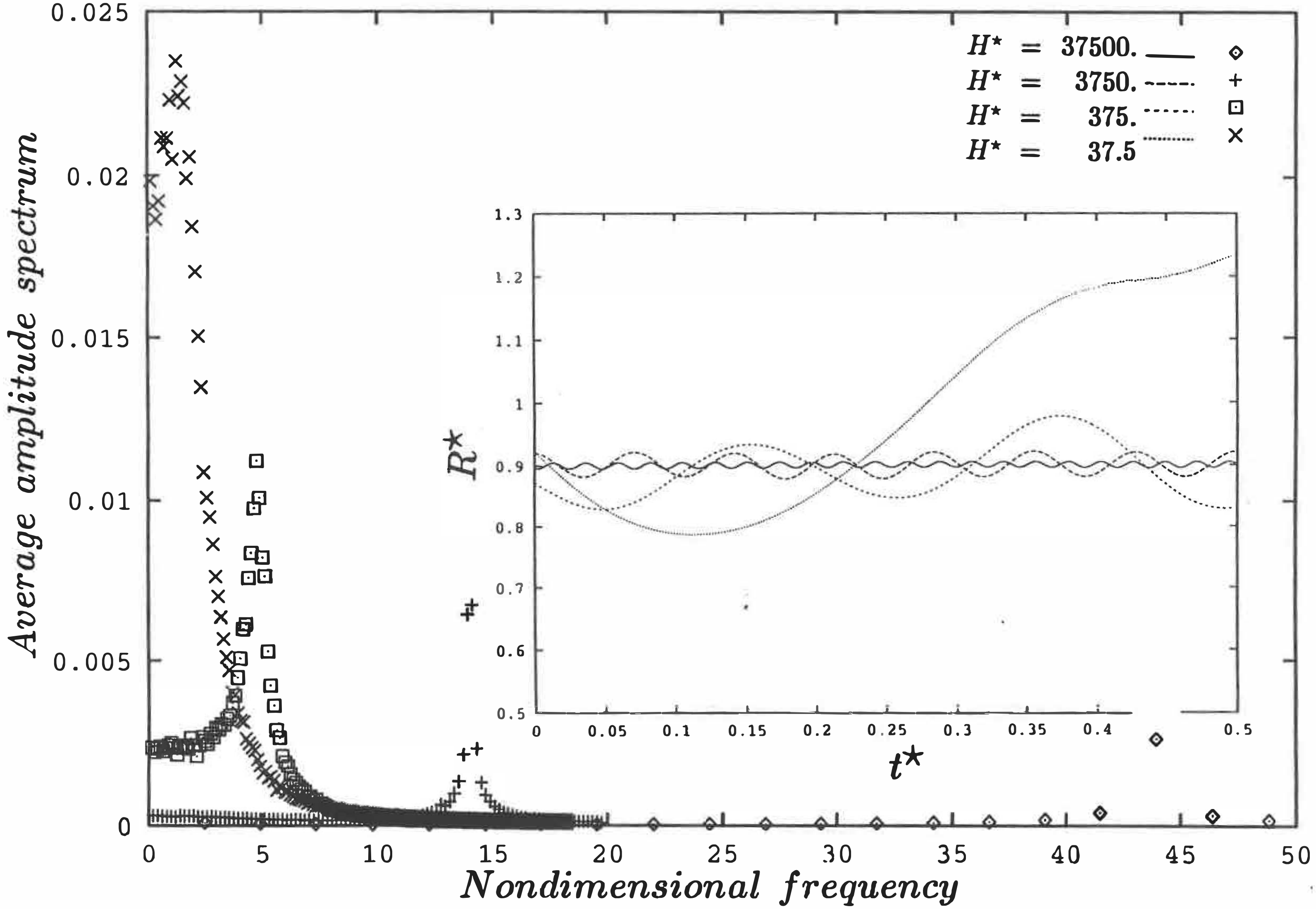
H^*	predicted f^*	observed f^*
37.5	1.5	1.4
375	4.4	4.8
3750	13.8	14.0
37500	43.6	44.0

Figure 1



$FENE$: $H^\star = 375$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$
$FENE$: $H^\star = 375$ $r^\star_{eq} = 0.68$ $r^\star_{max} = 1.08$
$FENE$: $H^\star = 375$ $r^\star_{eq} = 0.5$ $r^\star_{max} = 0.9$
$FENE$: $H^\star = 375$ $r^\star_{eq} = 1.3$ $r^\star_{max} = 1.7$
GLJ: $A = 1$
GLJ: $A = 3$
GLJ: $A = 4$
GLJ: $A = 5$
$\phi^\star_{ij}$
30
20
10
0
-10
-20
-30
-40
0
0.5
1
1.5
2
2.5
$r^\star$
$FENE$: $H^\star = 3750$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$
$FENE$: $H^\star = 375$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$
$FENE$: $H^\star = 37.5$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$
$\phi^\star_{ij}$
20
10
0
0.5
1
$r^\star$

Figure 2a



0.025
0.02
0.015
0.01
0.005
0
0
5
10
15
20
25
30
35
40
45
50
Average amplitude spectrum
Nondimensional frequency
$H^{\star}$ = 37500.
$H^{\star}$ = 3750.
$H^{\star}$ = 375.
$H^{\star}$ = 37.5
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.5
$R^{\star}$
$t^{\star}$

Figure 2b

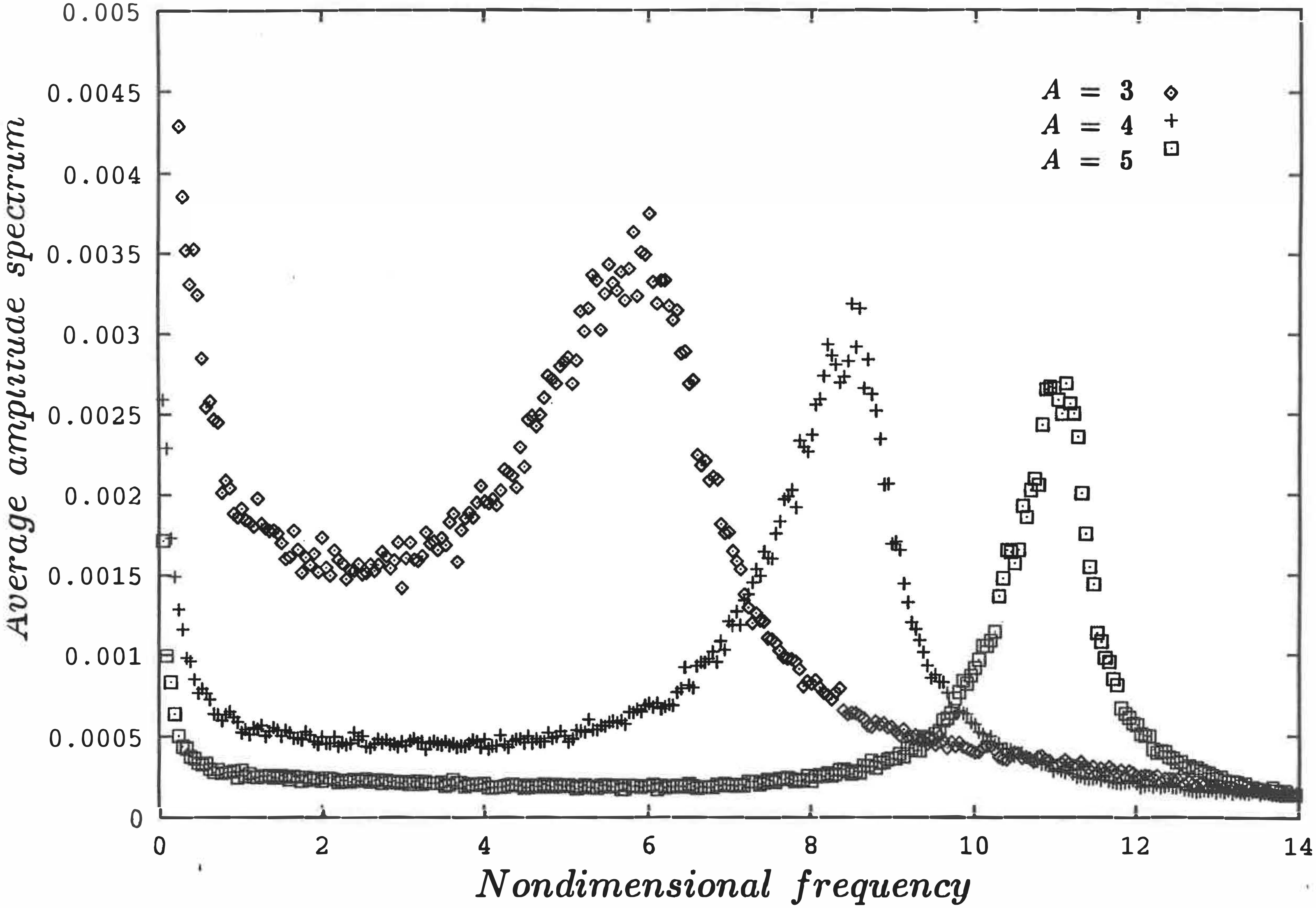

0.005
0.0045
0.004
0.0035
0.003
0.0025
0.002
0.0015
0.001
0.0005
0
Average amplitude spectrum
0
2
4
6
8
10
12
14
Nondimensional frequency
$A = 3$ ◇
$A = 4$ +
$A = 5$ ⊡

Figure 2c

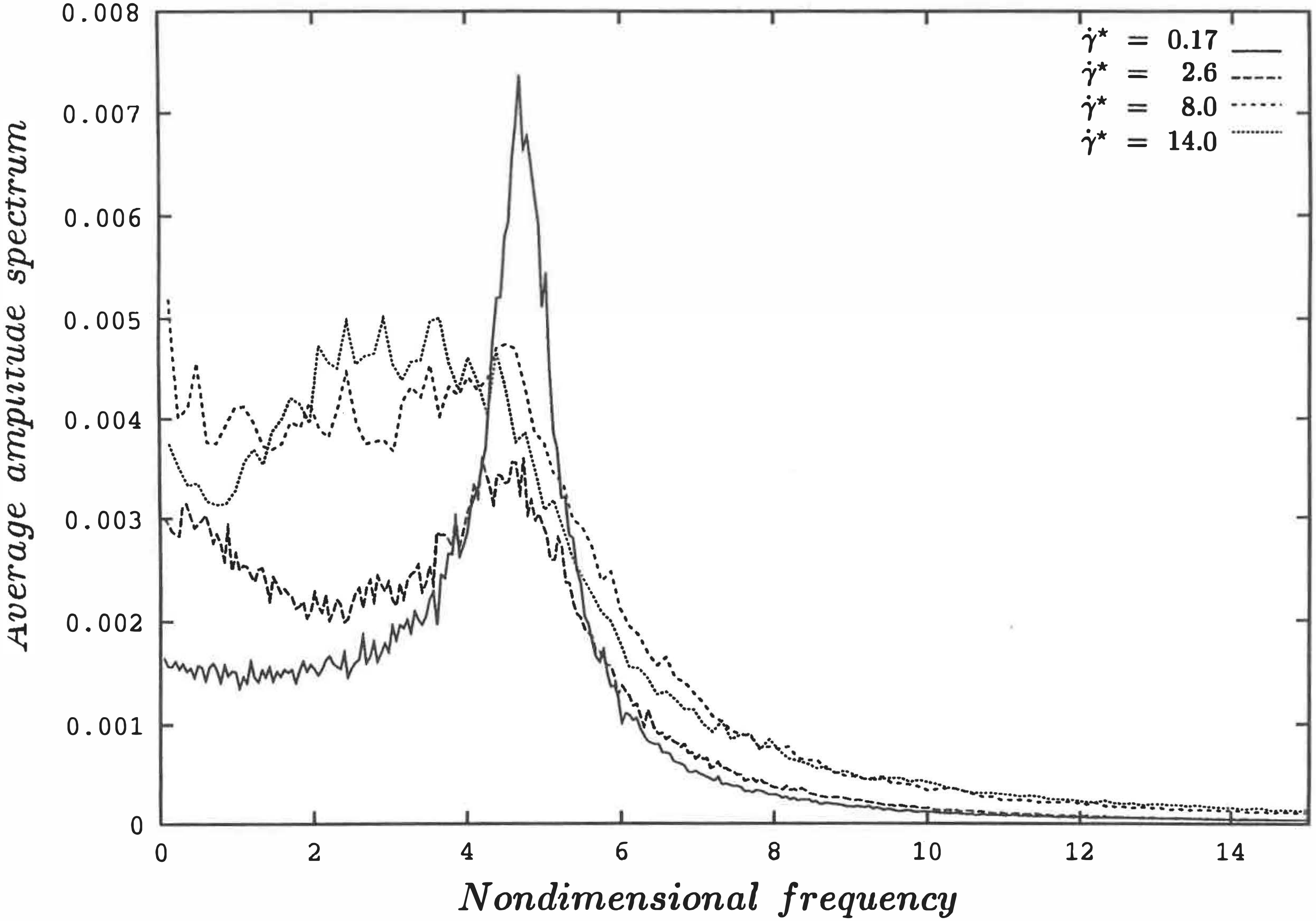

$\dot{\gamma}^* = 0.17$
$\dot{\gamma}^* = 2.6$
$\dot{\gamma}^* = 8.0$
$\dot{\gamma}^* = 14.0$
Average amplitude spectrum
Nondimensional frequency
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
0
2
4
6
8
10
12
14

Figure 3

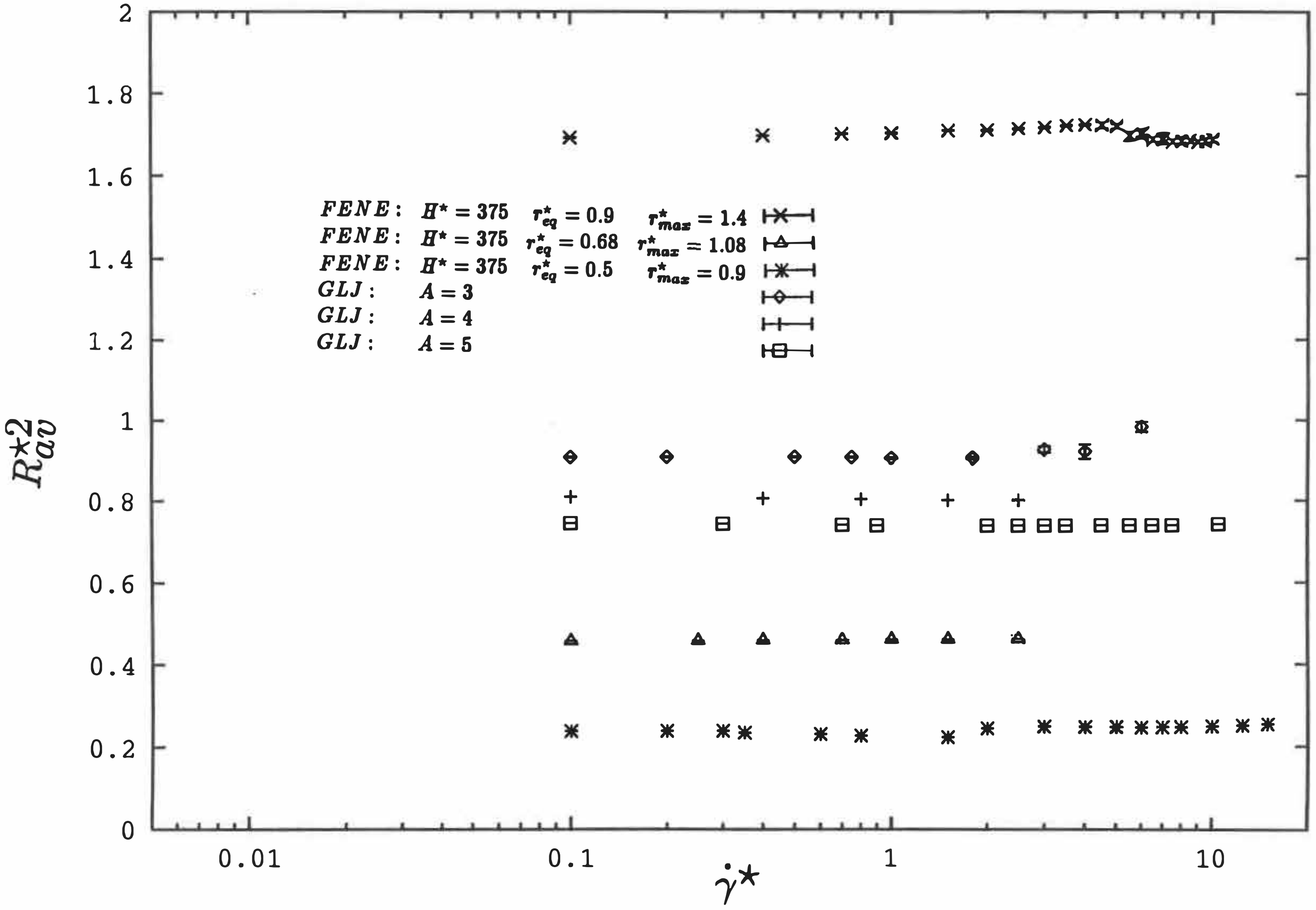
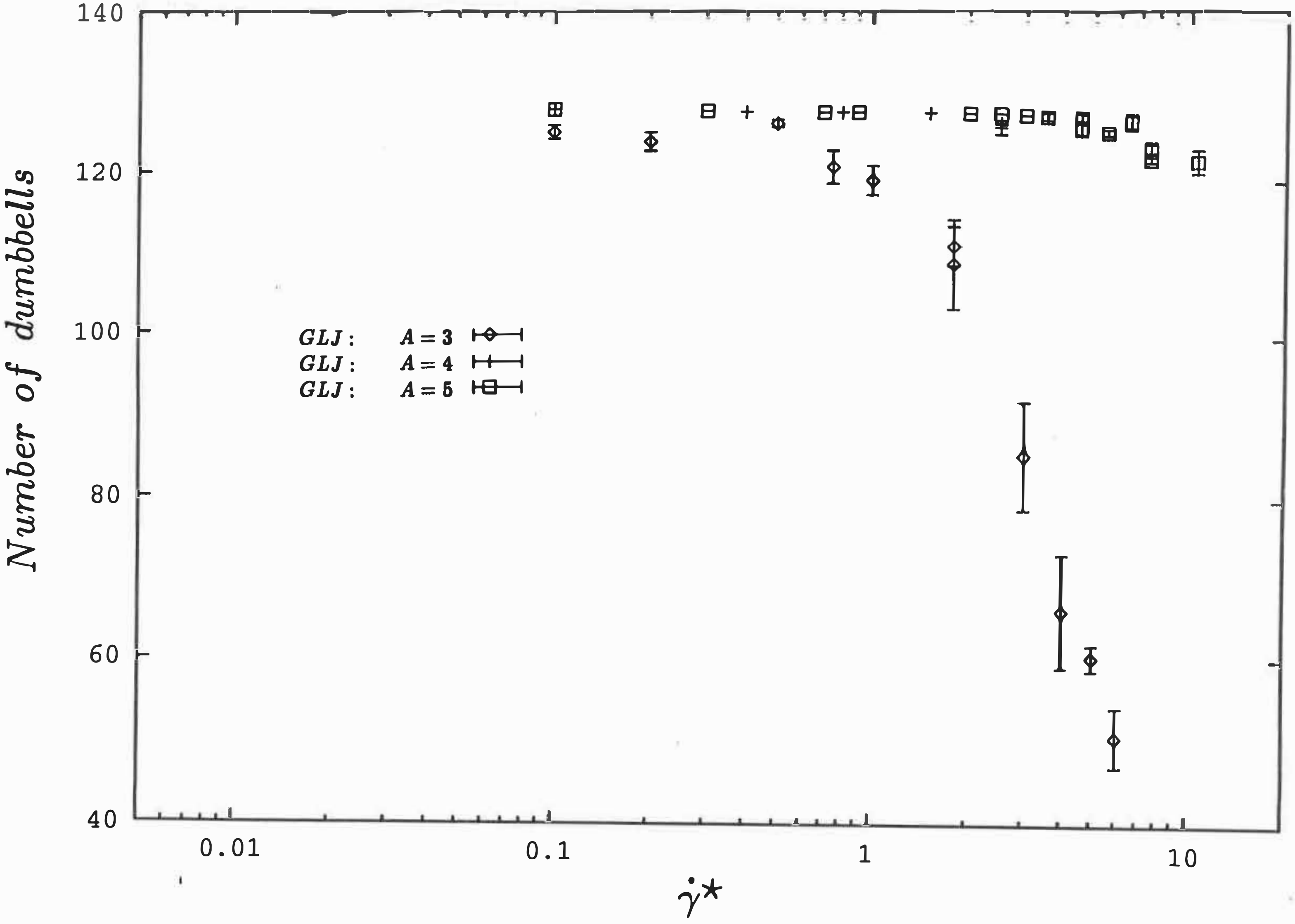
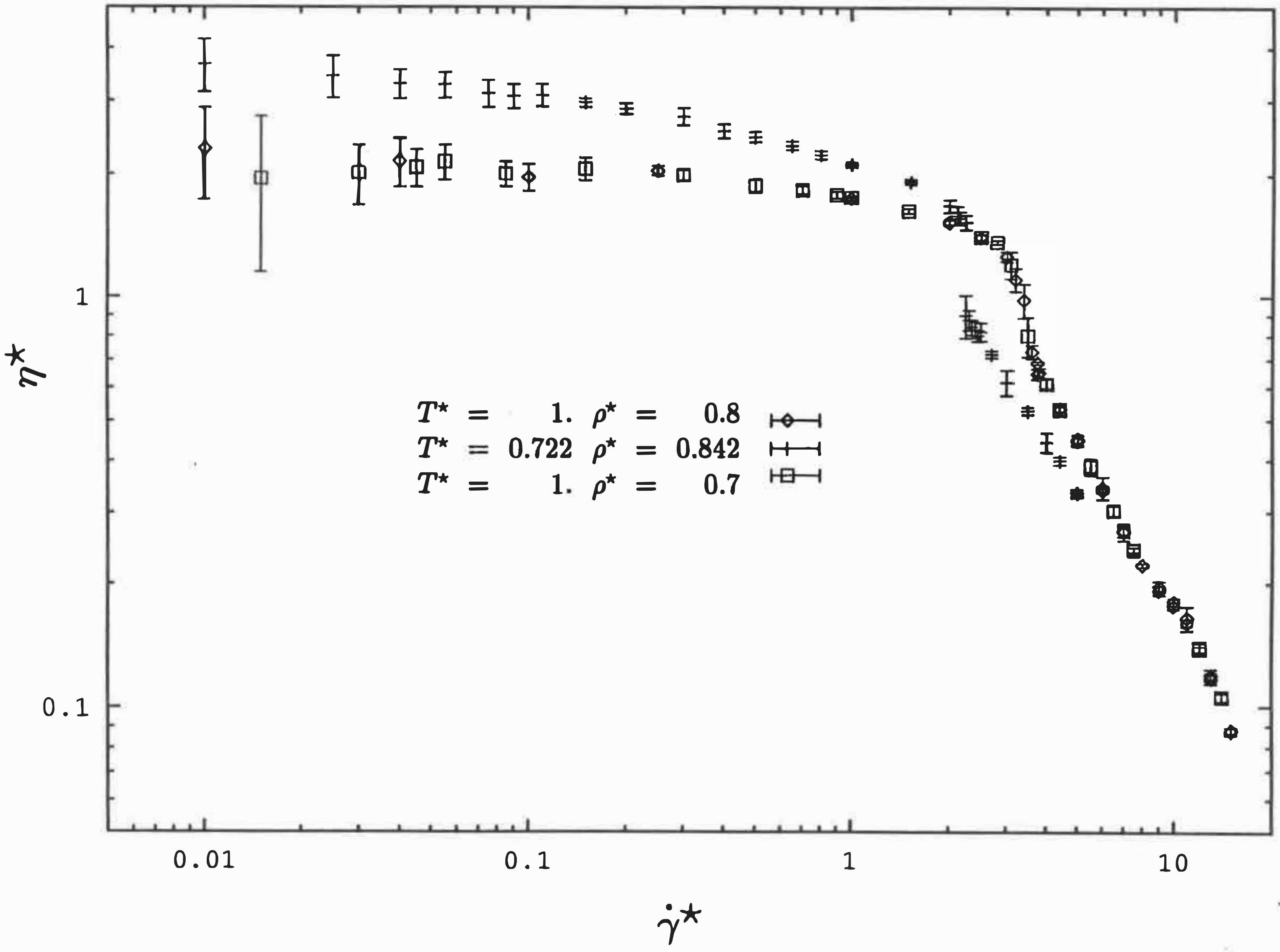

FENE: $H^* = 375$ $r^*_{eq} = 0.9$ $r^*_{max} = 1.4$
FENE: $H^* = 375$ $r^*_{eq} = 0.68$ $r^*_{max} = 1.08$
FENE: $H^* = 375$ $r^*_{eq} = 0.5$ $r^*_{max} = 0.9$
GLJ: $A = 3$
GLJ: $A = 4$
GLJ: $A = 5$
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.01
0.1
1
10
$R^{\star 2}_{av}$
$\dot{\gamma}^\star$

Figure 4



140
120
100
80
60
40
Number of dumbbells
0.01
0.1
1
10
$\dot{\gamma}^{\star}$
GLJ : A = 3
GLJ : A = 4
GLJ : A = 5

Figure 5a



$\eta^\star$
1
0.1
0.01
0.1
1
10
$\dot{\gamma}^\star$
$T^\star$ = 1. $\rho^\star$ = 0.8
$T^\star$ = 0.722 $\rho^\star$ = 0.842
$T^\star$ = 1. $\rho^\star$ = 0.7

Figure 5b

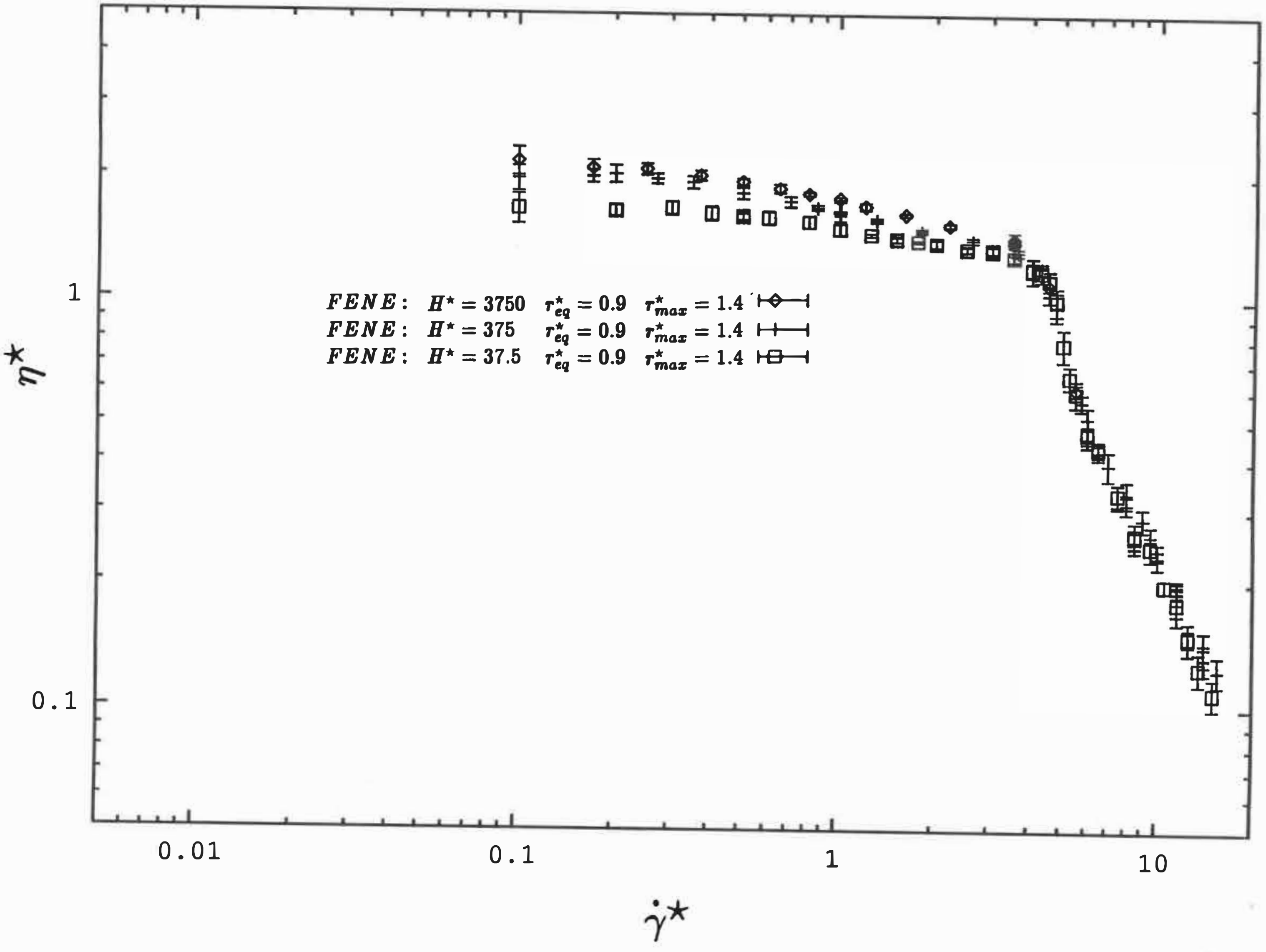

$FENE$: $H^\star = 3750$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$
$FENE$: $H^\star = 375$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$
$FENE$: $H^\star = 37.5$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$
$\eta^\star$
1
0.1
0.01
0.1
1
10
$\dot{\gamma}^\star$

Figure 5c

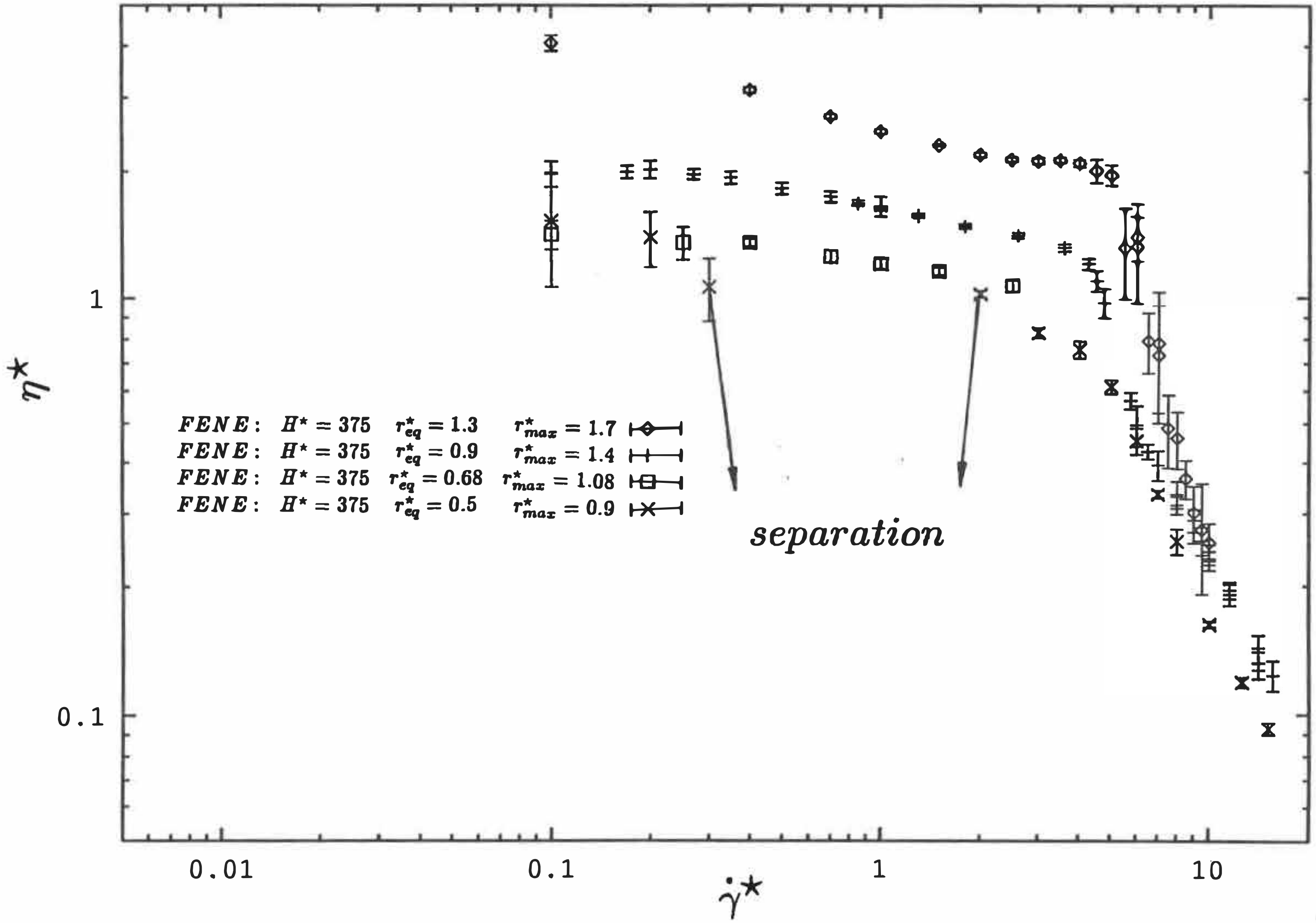

$\eta^\star$
1
0.1
0.01
0.1
1
10
$\dot{\gamma}^\star$
$FENE$: $H^\star = 375$ $r^\star_{eq} = 1.3$ $r^\star_{max} = 1.7$
$FENE$: $H^\star = 375$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$
$FENE$: $H^\star = 375$ $r^\star_{eq} = 0.68$ $r^\star_{max} = 1.08$
$FENE$: $H^\star = 375$ $r^\star_{eq} = 0.5$ $r^\star_{max} = 0.9$
separation

Figure 5d

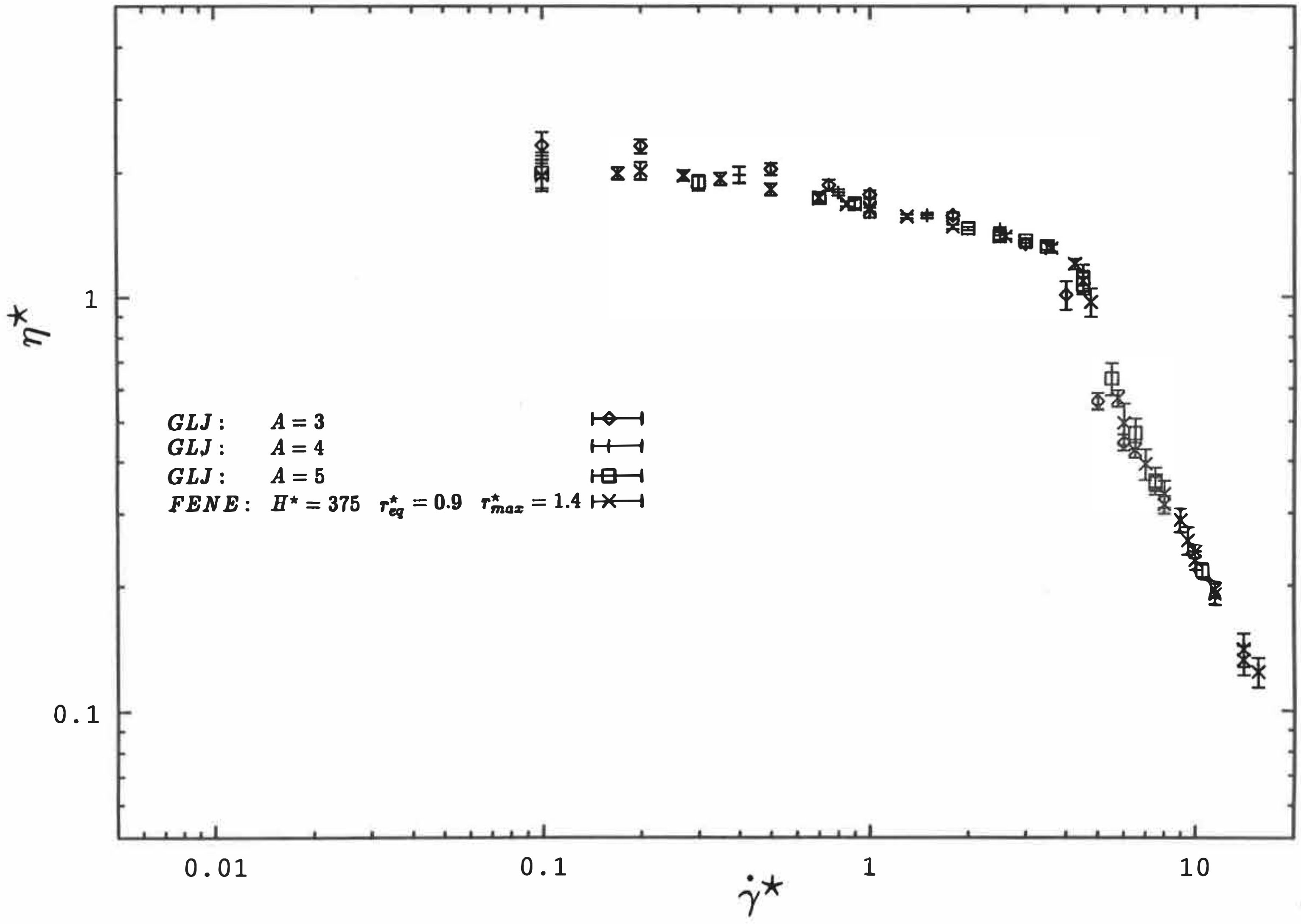
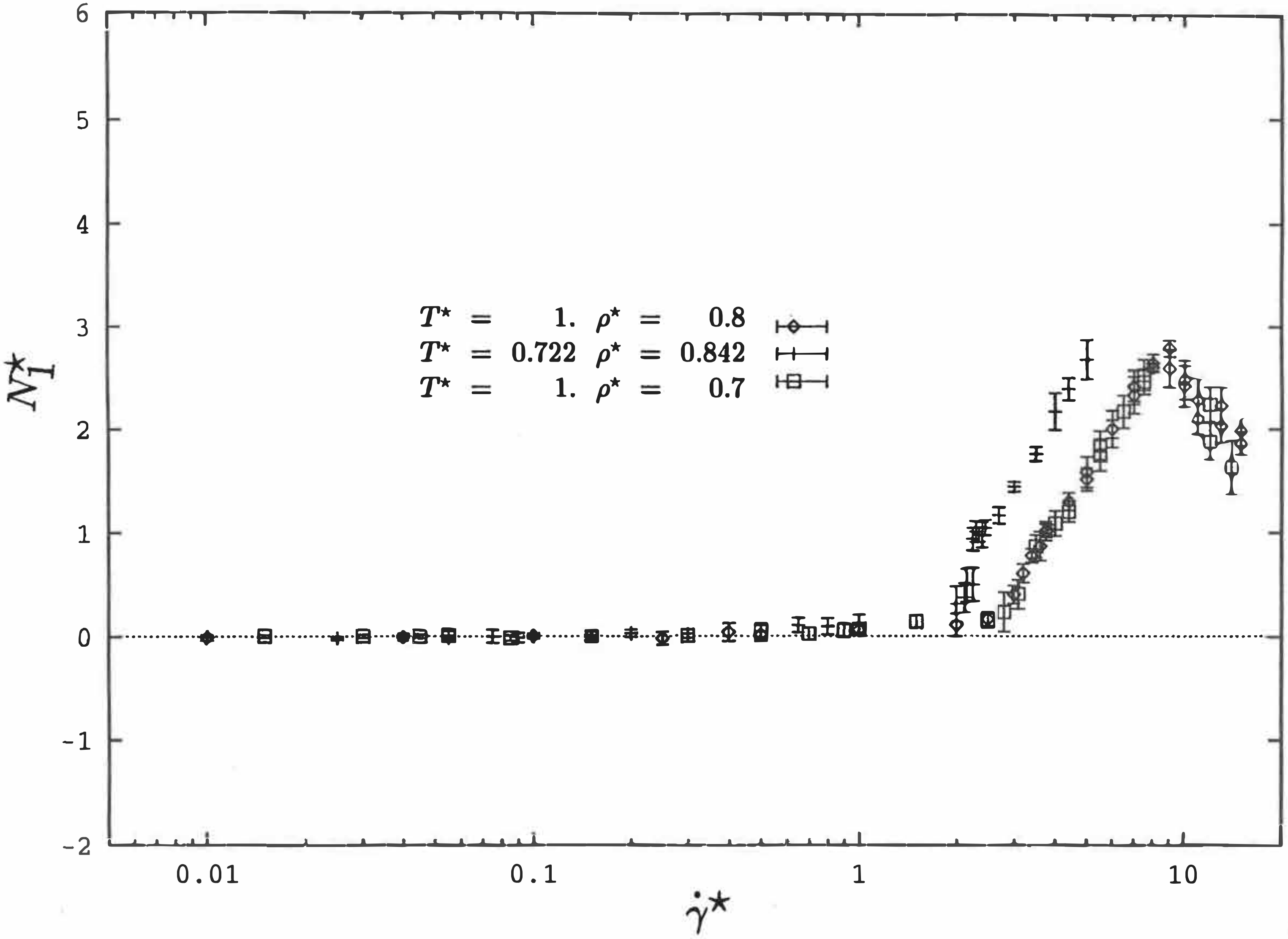

$\eta^\star$
1
0.1
$GLJ:$ $A=3$
$GLJ:$ $A=4$
$GLJ:$ $A=5$
$FENE:$ $H^\star=375$ $r^\star_{eq}=0.9$ $r^\star_{max}=1.4$
0.01
0.1
1
10
$\dot{\gamma}^\star$

Figure 6a



6
5
4
3
2
1
0
-1
-2
$N_1^\star$
0.01
0.1
1
10
$\dot{\gamma}^\star$
$T^\star$ = 1. $\rho^\star$ = 0.8
$T^\star$ = 0.722 $\rho^\star$ = 0.842
$T^\star$ = 1. $\rho^\star$ = 0.7

Figure 6b

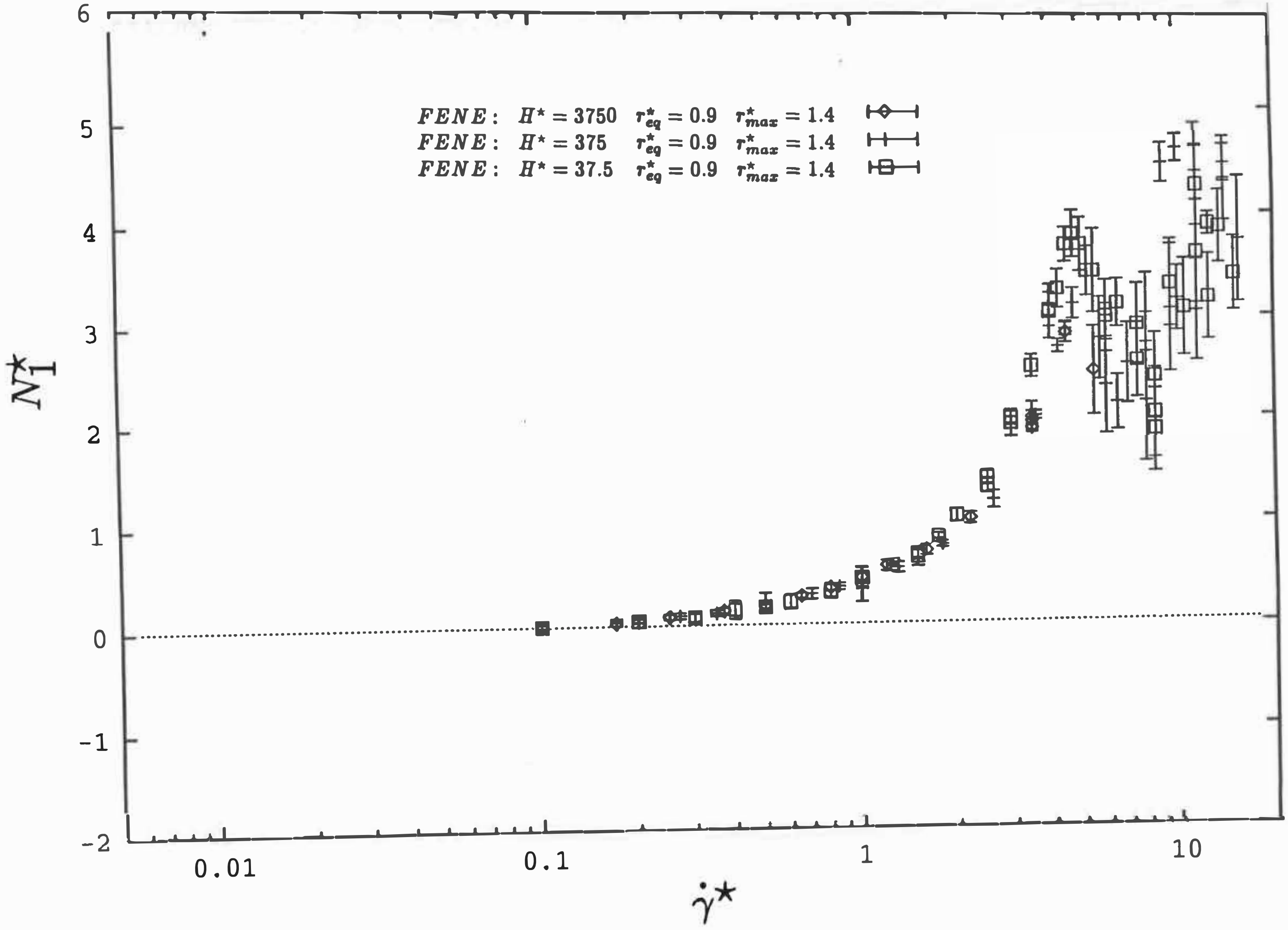
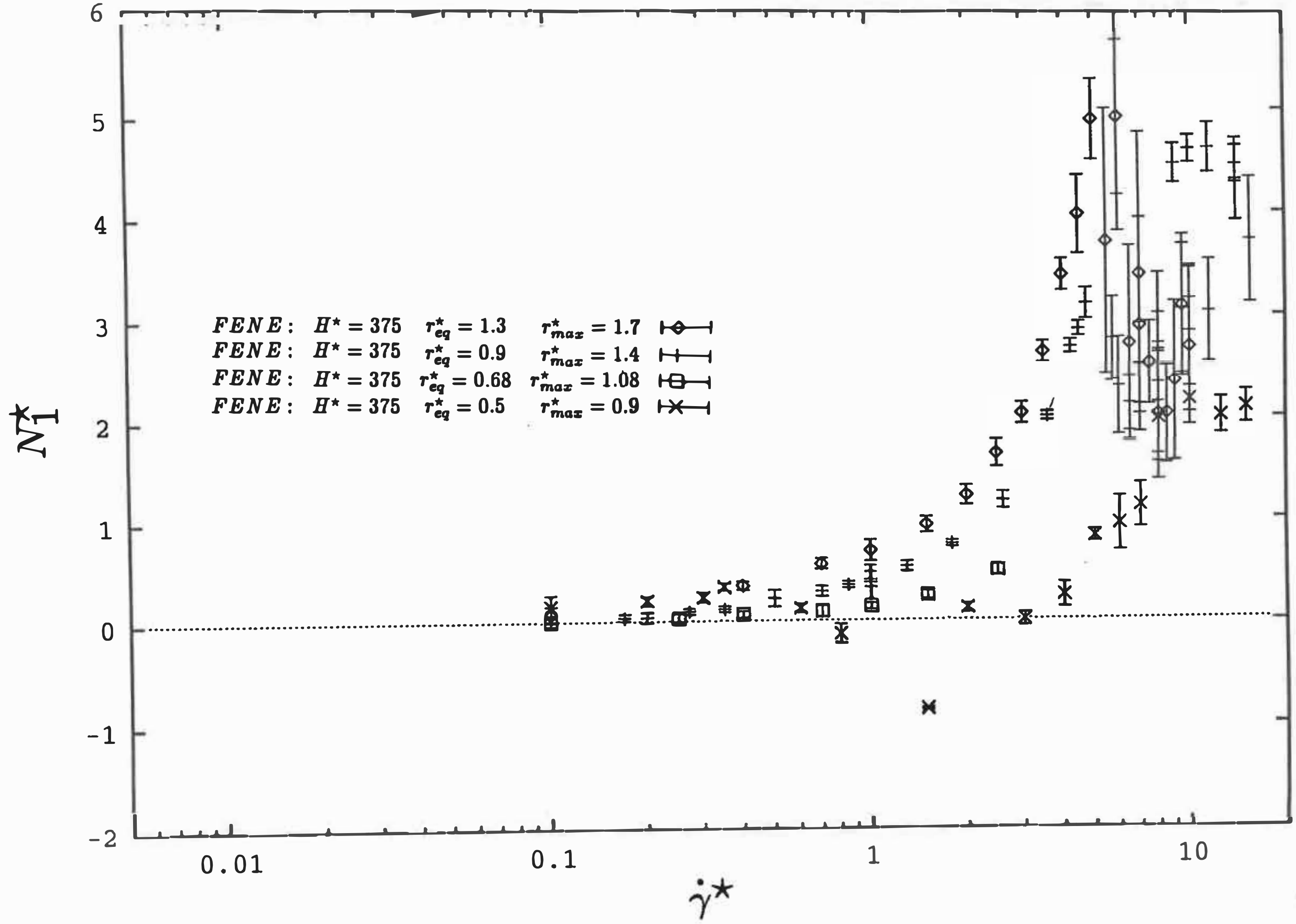

FENE: H* = 3750 r*_eq = 0.9 r*_max = 1.4
FENE: H* = 375 r*_eq = 0.9 r*_max = 1.4
FENE: H* = 37.5 r*_eq = 0.9 r*_max = 1.4
6
5
4
3
2
1
0
-1
-2
N*_1
0.01
0.1
1
10
γ̇*

Figure 6c



$N_1^\star$
6
5
4
3
2
1
0
-1
-2
0.01
0.1
1
10
$\dot{\gamma}^\star$
$FENE: \quad H^\star = 375 \quad r^\star_{eq} = 1.3 \quad r^\star_{max} = 1.7$
$FENE: \quad H^\star = 375 \quad r^\star_{eq} = 0.9 \quad r^\star_{max} = 1.4$
$FENE: \quad H^\star = 375 \quad r^\star_{eq} = 0.68 \quad r^\star_{max} = 1.08$
$FENE: \quad H^\star = 375 \quad r^\star_{eq} = 0.5 \quad r^\star_{max} = 0.9$

Figure 6d

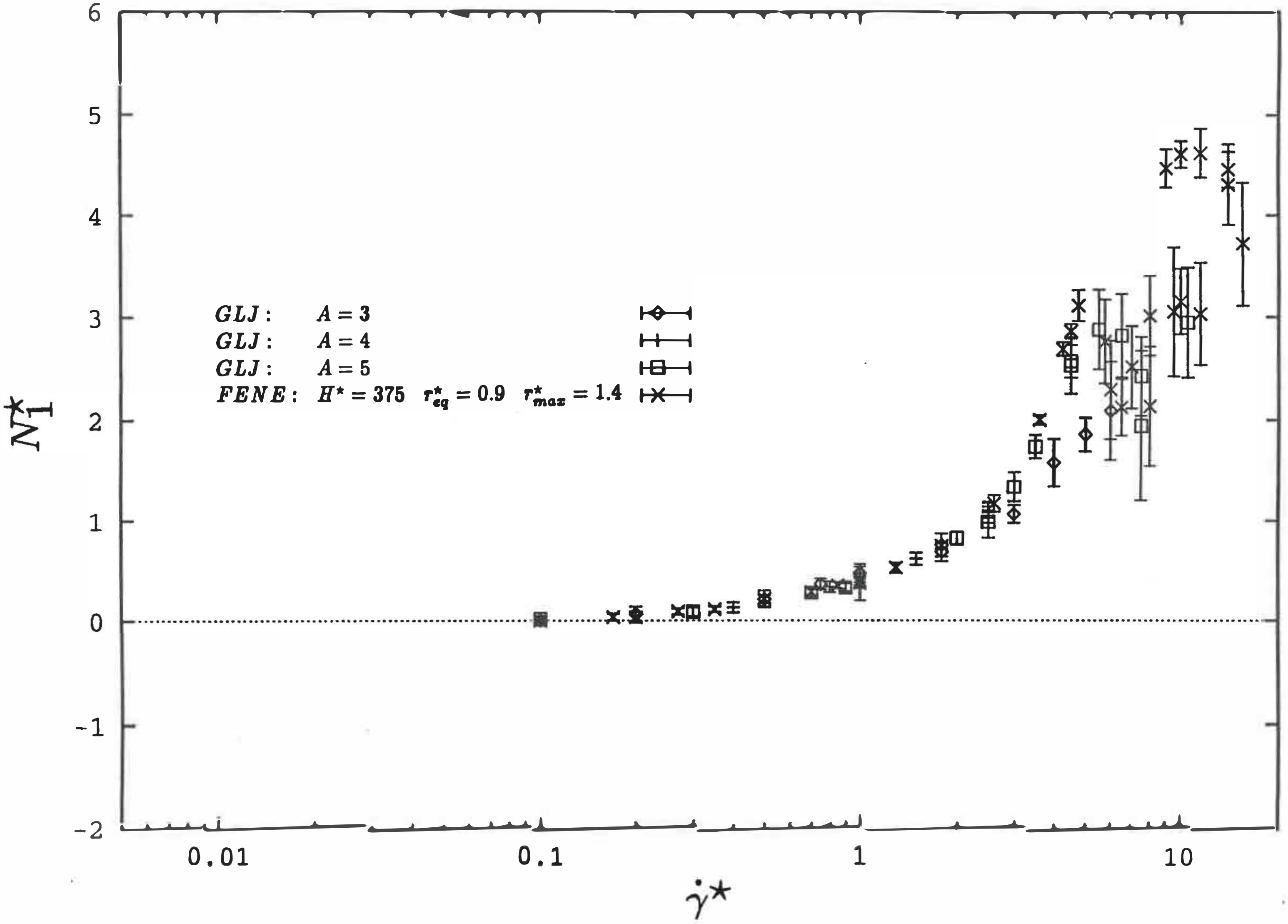

6
5
4
3
2
1
0
-1
-2
$N_1^\star$
0.01
0.1
1
10
$\dot{\gamma}^\star$
GLJ : $A = 3$
GLJ : $A = 4$
GLJ : $A = 5$
$FENE$: $H^* = 375$ $r_{eq}^* = 0.9$ $r_{max}^* = 1.4$

Figure 7a

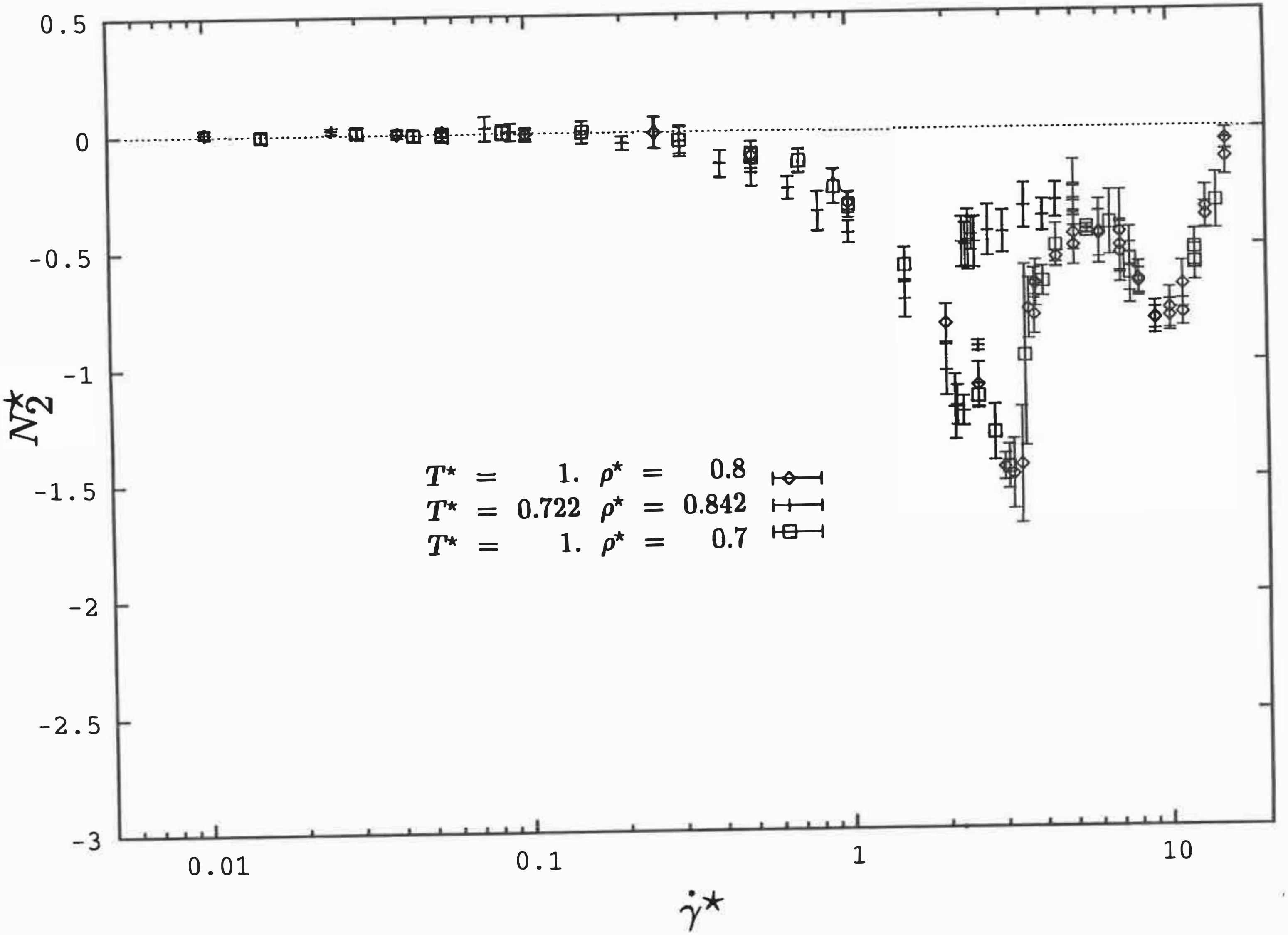
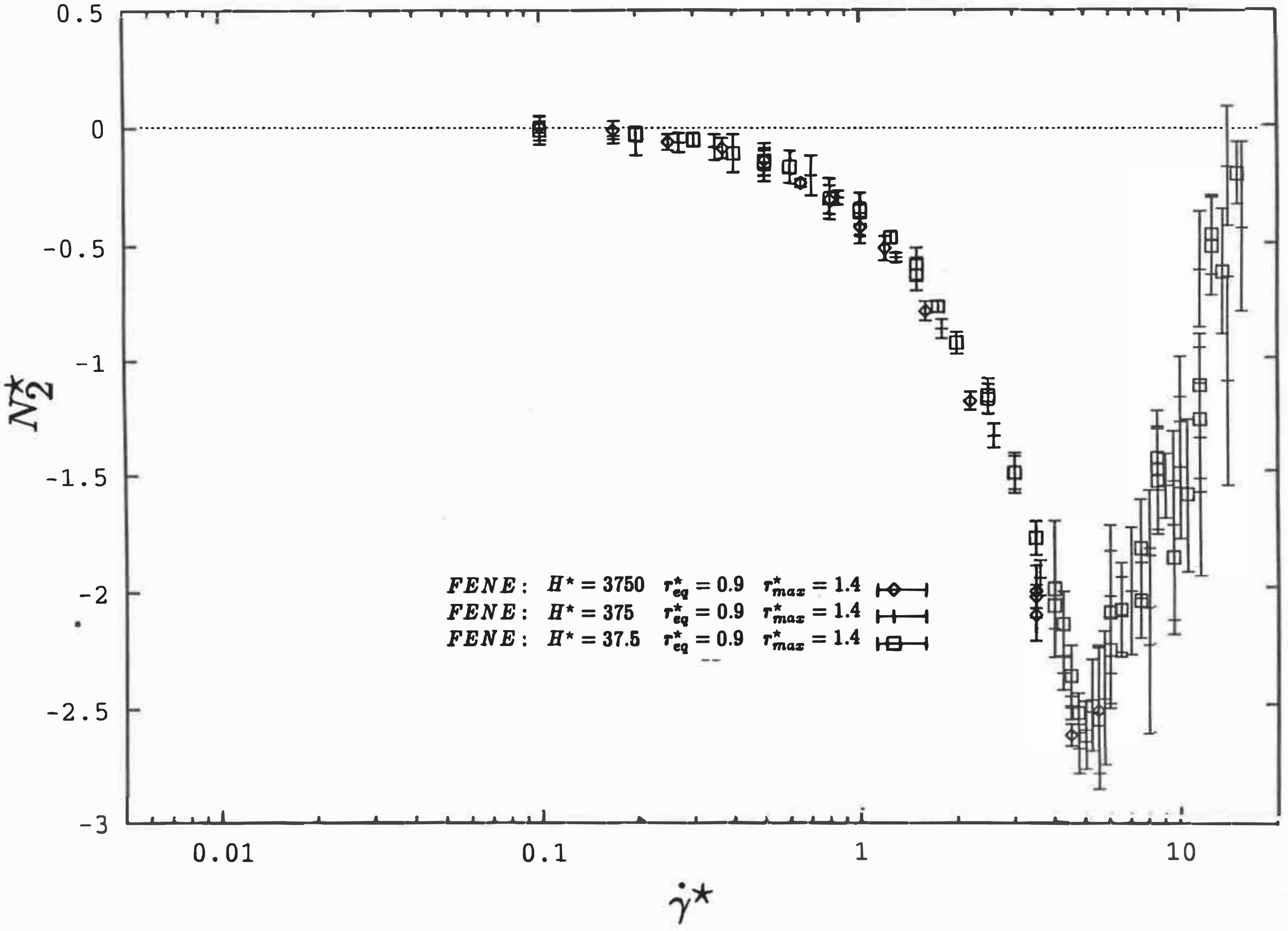

0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
$N_2^\star$
0.01
0.1
1
10
$\dot{\gamma}^\star$
$T^\star$ = 1. $\rho^\star$ = 0.8
$T^\star$ = 0.722 $\rho^\star$ = 0.842
$T^\star$ = 1. $\rho^\star$ = 0.7

Figure 7b



0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0.01
0.1
1
10
$N_2^\star$
$\dot{\gamma}^\star$
$FENE$: $H^\star = 3750$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$
$FENE$: $H^\star = 375$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$
$FENE$: $H^\star = 37.5$ $r^\star_{eq} = 0.9$ $r^\star_{max} = 1.4$

Figure 7c

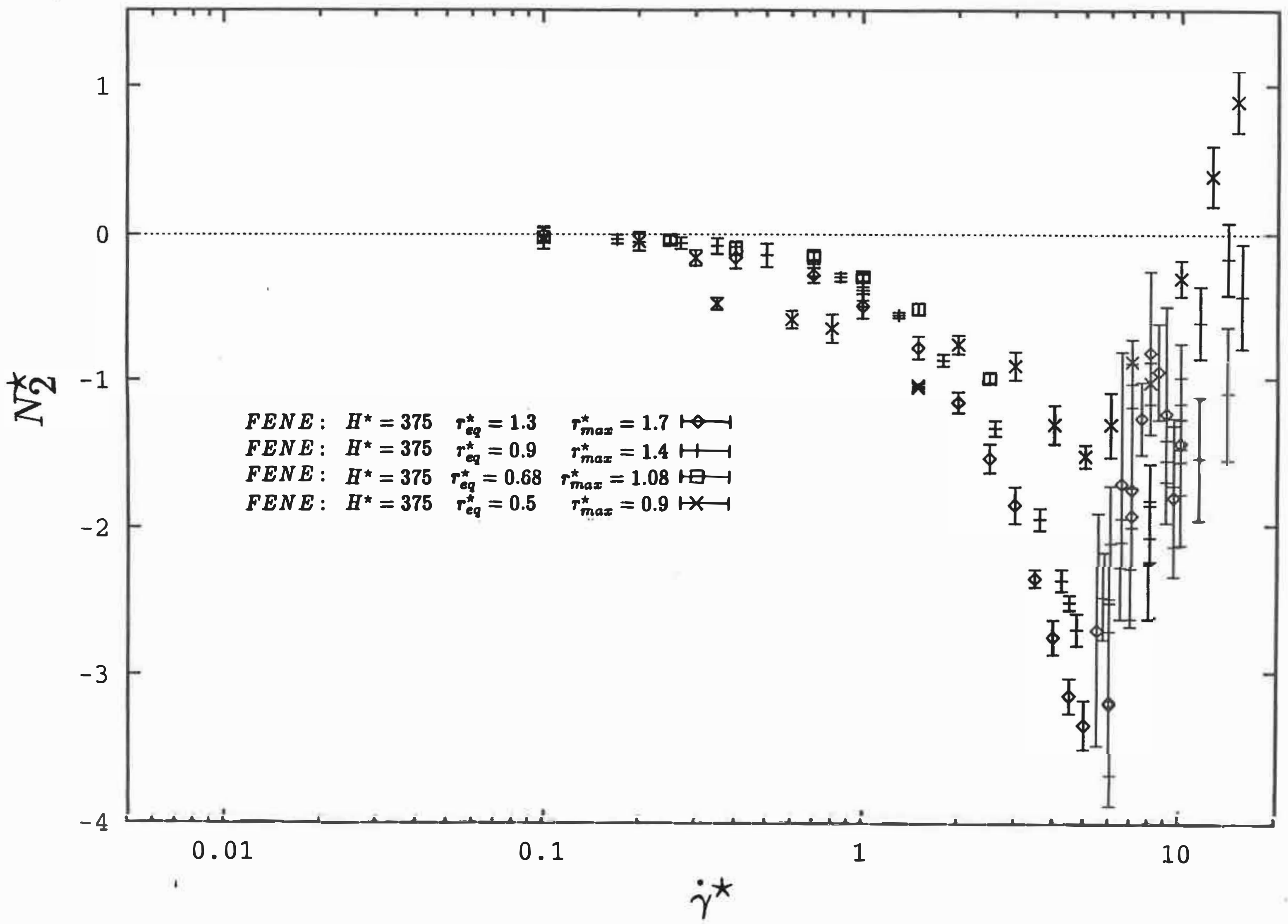

1
0
-1
-2
-3
-4
$N_2^\star$
0.01
0.1
1
10
$\dot{\gamma}^\star$
$FENE: \; H^\star = 375 \quad r^\star_{eq} = 1.3 \quad r^\star_{max} = 1.7$
$FENE: \; H^\star = 375 \quad r^\star_{eq} = 0.9 \quad r^\star_{max} = 1.4$
$FENE: \; H^\star = 375 \quad r^\star_{eq} = 0.68 \quad r^\star_{max} = 1.08$
$FENE: \; H^\star = 375 \quad r^\star_{eq} = 0.5 \quad r^\star_{max} = 0.9$

Figure 7d

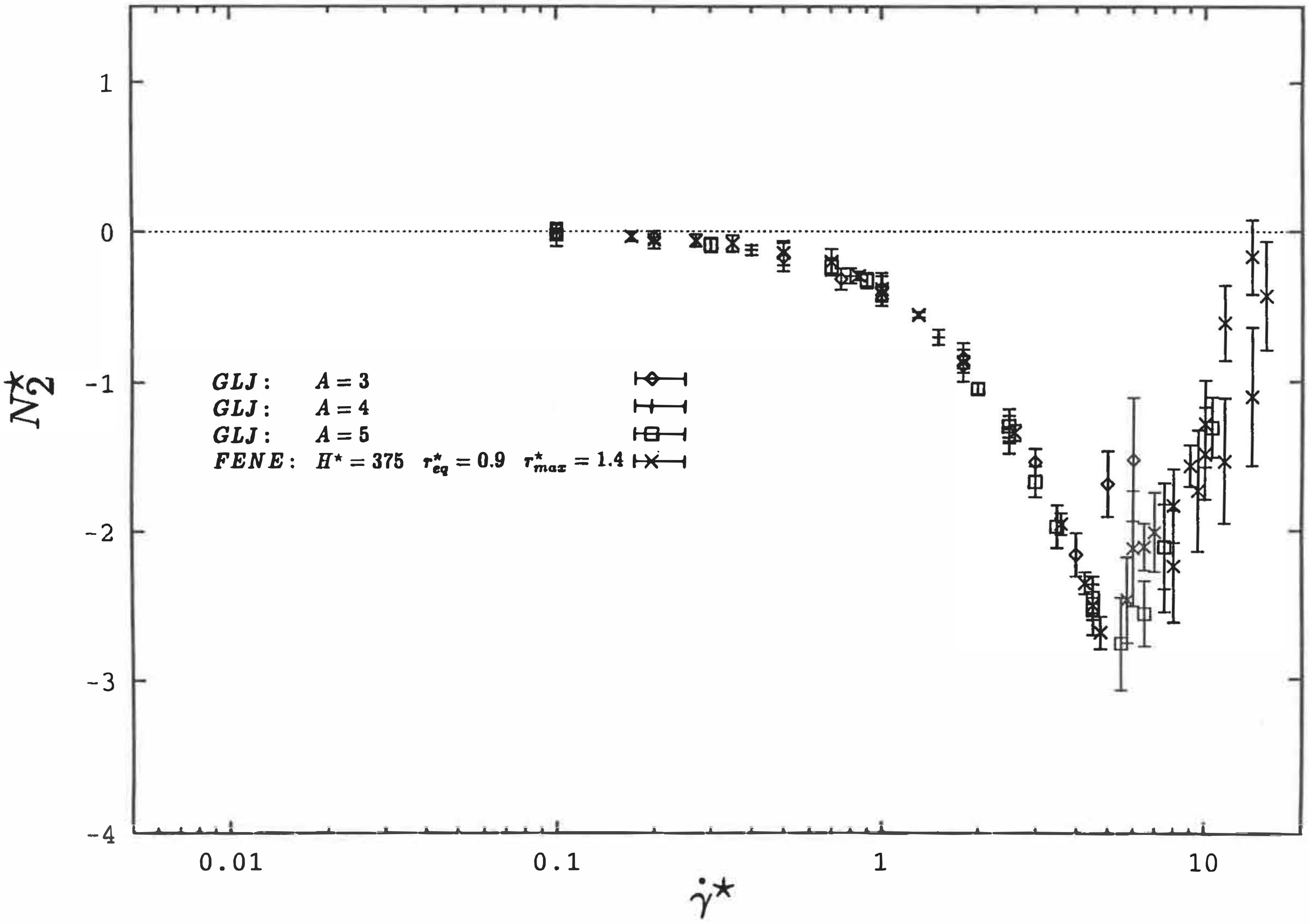

1
0
-1
-2
-3
-4
$N_2^\star$
GLJ : $A = 3$
GLJ : $A = 4$
GLJ : $A = 5$
$FENE$: $H^\star = 375$ $r_{eq}^\star = 0.9$ $r_{max}^\star = 1.4$
0.01
0.1
1
10
$\dot{\gamma}^\star$

Figure 8a

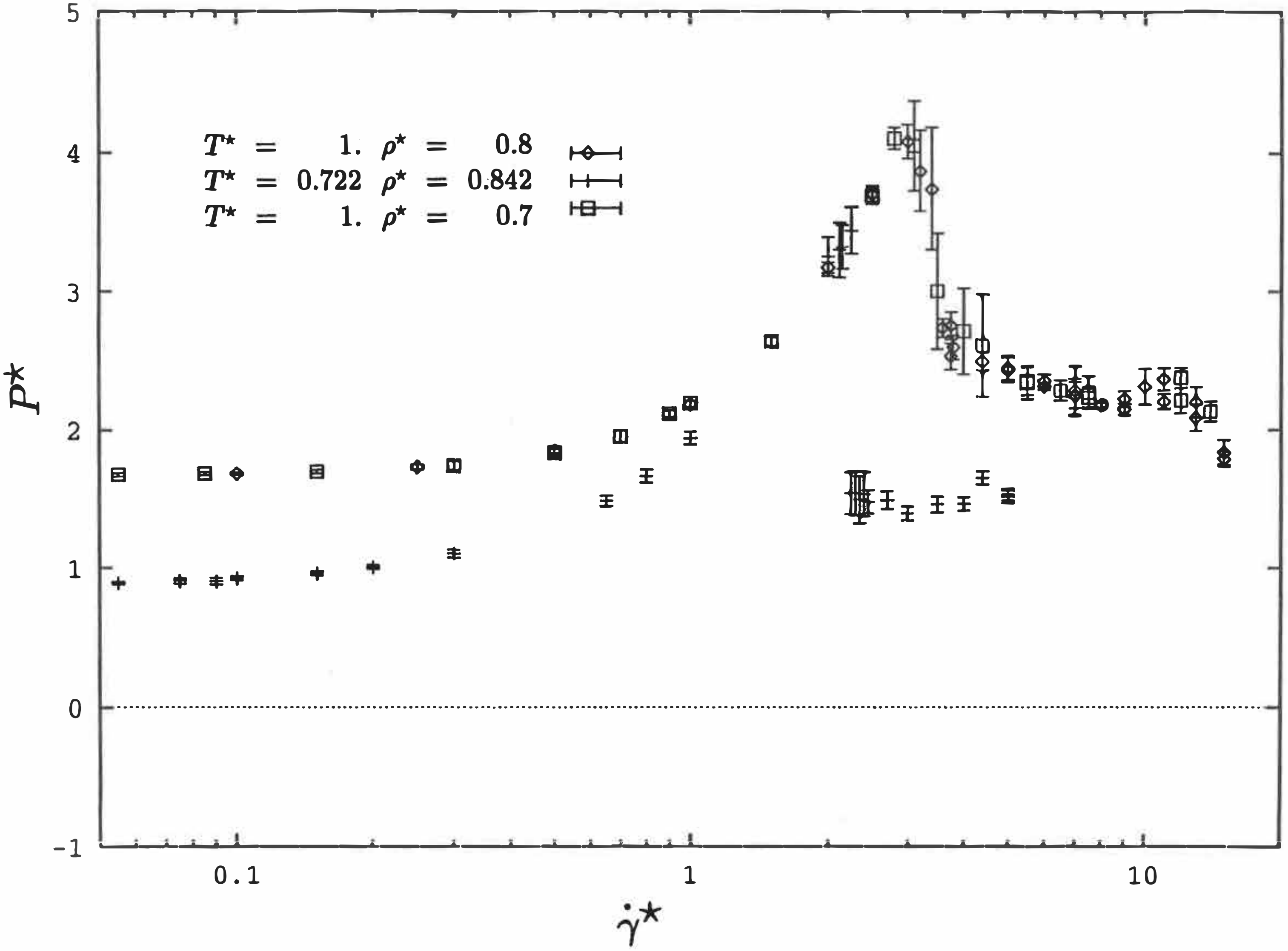
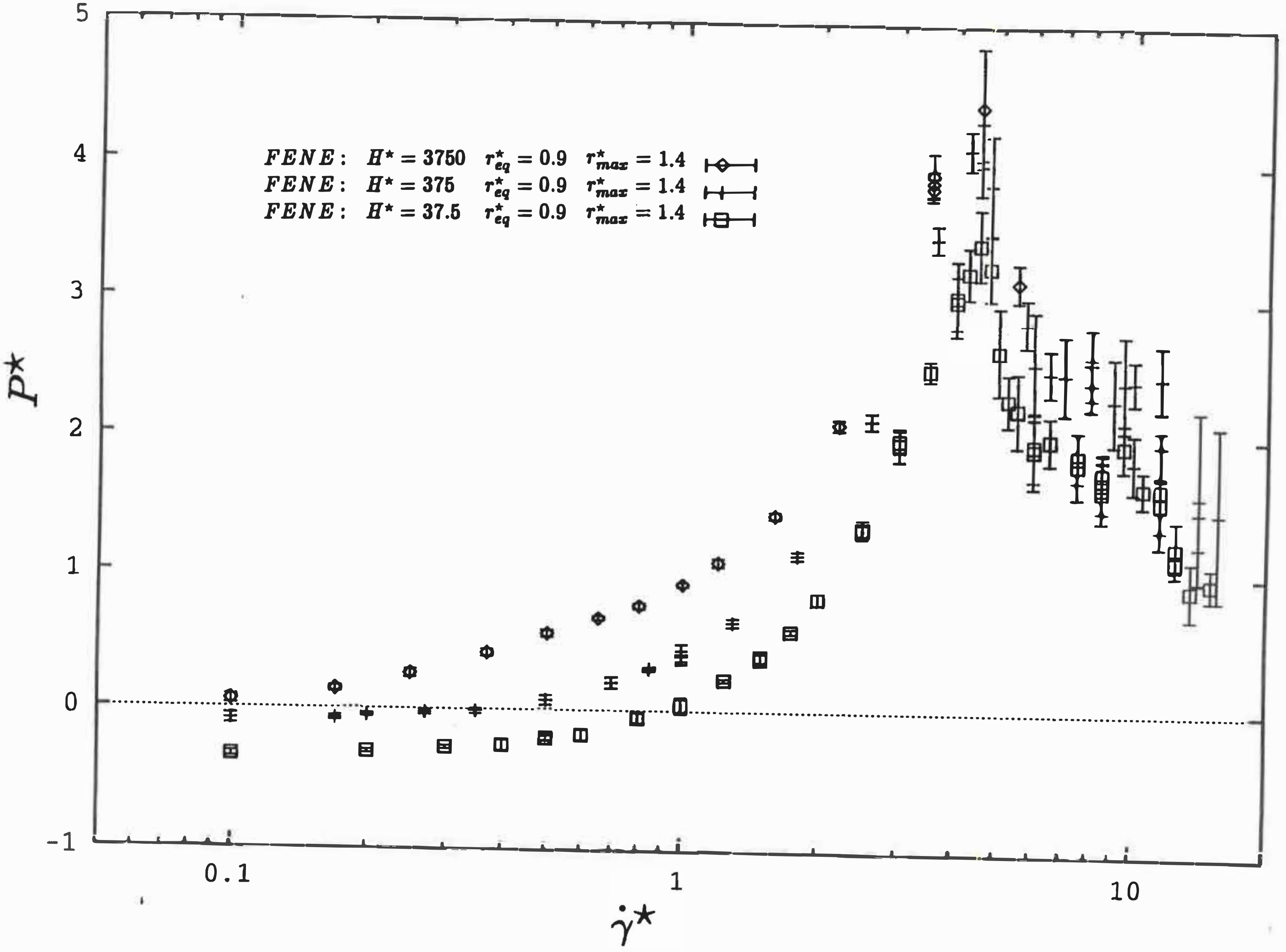
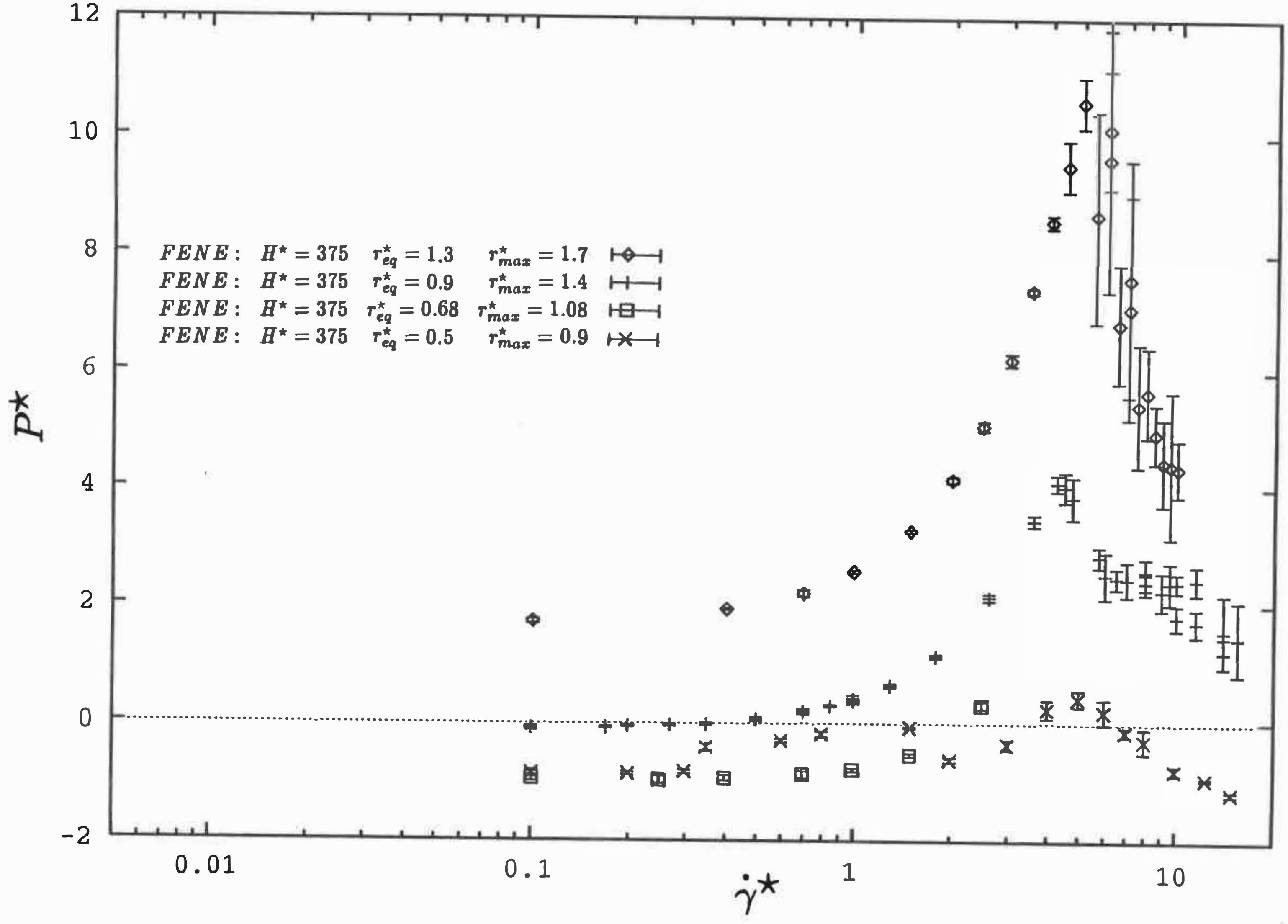

$P^\star$
5
4
3
2
1
0
-1
0.1
1
10
$\dot{\gamma}^\star$
$T^\star$ = 1. $\rho^\star$ = 0.8
$T^\star$ = 0.722 $\rho^\star$ = 0.842
$T^\star$ = 1. $\rho^\star$ = 0.7

Figure 8b



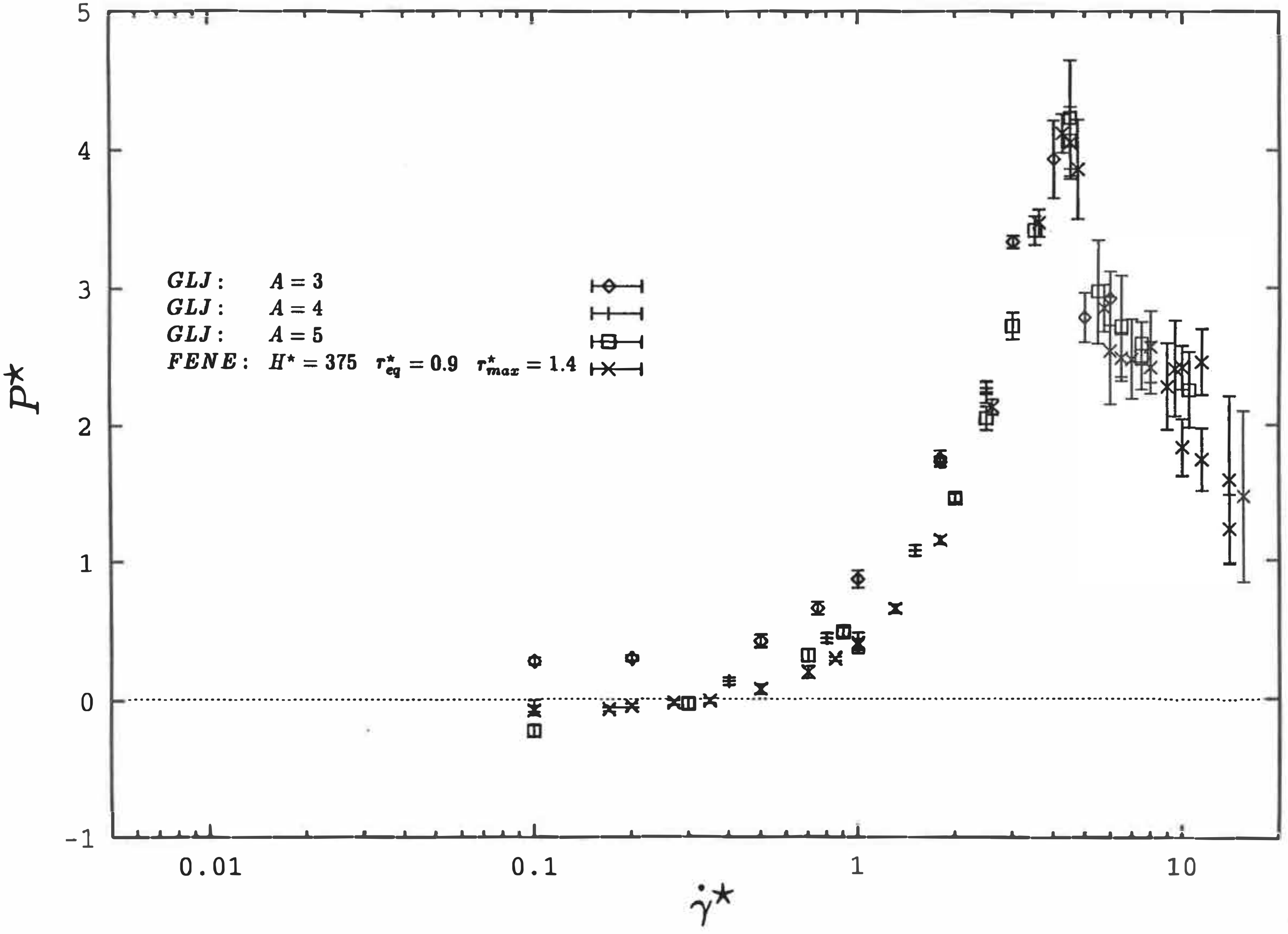
FENE: $H^* = 3750$ $r^*_{eq} = 0.9$ $r^*_{max} = 1.4$
FENE: $H^* = 375$ $r^*_{eq} = 0.9$ $r^*_{max} = 1.4$
FENE: $H^* = 37.5$ $r^*_{eq} = 0.9$ $r^*_{max} = 1.4$
P^*
$\dot{\gamma}^*$
5
4
3
2
1
0
-1
0.1
1
10

Figure 8c



$FENE: H^* = 375 \quad r_{eq}^* = 1.3 \quad r_{max}^* = 1.7$
$FENE: H^* = 375 \quad r_{eq}^* = 0.9 \quad r_{max}^* = 1.4$
$FENE: H^* = 375 \quad r_{eq}^* = 0.68 \quad r_{max}^* = 1.08$
$FENE: H^* = 375 \quad r_{eq}^* = 0.5 \quad r_{max}^* = 0.9$
P^*
12
10
8
6
4
2
0
-2
0.01
0.1
1
10
$\dot{\gamma}^*$

Figure 8d



GLJ : A = 3
GLJ : A = 4
GLJ : A = 5
FENE : $H^* = 375$ $r^*_{eq} = 0.9$ $r^*_{max} = 1.4$
$P^\star$
5
4
3
2
1
0
-1
0.01
0.1
1
10
$\dot{\gamma}^\star$

Microscopic and Mesoscopic Results from Non-Equilibrium Molecular Dynamics Modeling of FENE Dumbbell Liquids

Bogdan Z. Dlugogorski, Miroslav Grmela and Pierre J. Carreau

Département de génie chimique
Centre de recherche appliquée sur les polymères (CRASP)
École Polytechnique
Case postale 6079, succursale A
Montréal, Québec H3C 3A7, CANADA.

- *Journal of Non-Newtonian Fluid Mechanics* (in press)

Summary

The microscopic and the mesoscopic results are presented for several finitely extensible nonlinear elastic (FENE) dumbbell fluids investigated under imposed flow according to SLLOD dynamics. The contracted distribution functions are calculated in both position and velocity spaces, and the fluid structure is probed by two conformation tensors. It is observed that dumbbells form a variety of short and long range structures depending on the imposed shear rate and the size of a single dumbbell. The assumption of Maxwellian distributed bead velocities, which is often used in the elastic dumbbell theories, is shown not to be satisfied but at low shear rates. Under shear, the distribution of the end-to-end distances is similar to the Gibbs equilibrium distribution function in configuration space if intra-molecular interactions are much stronger then inter-molecular forces. On the average, the longest dumbbells are found at between 30° and 50° to the direction of flow, and the shortest at between $-50°$ and $-30°$.

1 Introduction

According to the rheological theories formulated at the mesoscopic level of description, the state of the complex polymeric fluid under flow is determined by the five fundamental hydrodynamic fields of density, momenta and energy, and additional fields called internal state variables describing the internal structure of the fluid. All other physical quantities can be found from the knowledge of these fields. It has been recently shown [1] that the SLLOD [2] equations of motion that were originally introduced at the microscopic level of description within the context of non-equilibrium molecular dynamics (NEMD), may be reinterpreted in the spirit of mesoscopic theories, as equations governing the time evolution of the internal state variables. These variables are position and peculiar velocities of SLLOD (effective, renormalized) particles. Thus, SLLOD can be seen as yet another standard rheological theory. In this paper we apply it to investigate dumbbell fluids.

The physical insight gained from studying the dynamic behavior of fluids by micro-rheological simulations, such as SLLOD, can be put to practical use in the formulation of more macroscopic models. This is especially important for mesoscopic theories that make explicit assumptions about statistical quantities which describe model fluids, for example about distribution of bead velocities in case of the mesoscopic elastic dumbbell theories [3]. Sometimes, additional corrections are incorporated in mesoscopic theories to improve predictions; corrections for hydrodynamic interactions or excluded volume effects are typical examples. Molecular dynamics simulations, as those presented in this paper, have potential to examine the implications involved in selection of critical assumptions, and to estimate the significance of additional corrections.

Another motivation to investigate liquids which are defined at the molecular level, is based on the notion that all liquids have common properties (*e.g.* finite expansion, flow, *etc.*) and hence at the microscopic level all liquids must be similar. It is recognized that many polymeric liquids exhibit similar viscometric behavior. We recall a widely accepted practice in experimental rheology to lump results for several polymers into one master curve. In addition, results from NEMD calculations conducted at shear rates too high to be attained in the laboratory when normalized, show surprisingly good resemblance to those obtained from experiments for viscoelastic liquids. For example, when the Carreau macroscopic four parameter viscosity model was used to fit computational NEMD results for liquid argon, it predicted the zero-shear viscosity to be exactly the same as determined from experiments [4]. Clearly, there is more than a coincidental similarity that pervades all liquids, making the traditional distinction between simple and complex liquids less important at the micro-rheological level. In other words, starting from the molecular dynamics simulation of dumbbell liquids, one may extract information that is also pertinent to polymeric liquids.

It has been often stressed, especially in the context of computational physics and chemistry, that the microscopic simulations have many similarities to laboratory experiments, and hence the term "computer experiments" was coined. Indeed, one can distinguish between the actual simulation (experiments) when the trajectory of the system of particles is calculated by means of ordinary differential equations and the "post-processing" (measurements) stage when the average properties are extracted from positions and velocities of beads along the trajectory. In analogy to real experiments, the average behavior of beads may be sampled by a variety of probes that yield information not only about macroscopic quantities such as pressure, but also about mesoscopic variables such as distribution functions. The beauty of micro-rheological simulations is related to the fact that the macroscopic phenomena may be directly related to the underlying microstructure. Furthermore, as opposed to laboratory experiments where only a part of data is recorded by instruments, in molecular dynamics calculations, everything can be determined. There exists a parallelism between problems encountered in experimental rheology and in micro-rheological simulations. Every experimentalist is well aware of the difficulties in collecting data in low and high shear rate regimes. Similarly, in NEMD calculations which are performed at low shear rates, the external force is small in comparison with the fluctuations. Consequently, long running times are necessary to minimize errors. In addition, at high shear rates the state of the microscopic system is often "arrested" in a small volume of the phase space for long times. Again, long runs are required to allow the system to sample the entire phase space.

In Sec. 2 we emphasize some features of the FENE dumbbell models and of the numerical technique, the thorough description of the simulations has been given in detail elsewhere [5]. We investigate the FENE dumbbells only upto a certain shear rate to minimize computational artifacts that may appear as a result of the thermostating mechanism that is incorporated in the equations of motion. One of them is formation of the so-called string phase, a process which is not well understood theoretically. Hence, in Sec. 3 we examine the issues which relate to the emergence of the long range order for both atomic and dumbbell liquids. In Secs. 4.1-4 we describe a contraction process that leads from the N-particle probability density function to viscometric functions, and give a set of results at different level of physical detail. These include photographs of instantaneous configurations at the molecular level, distribution functions in contracted configuration, and velocity spaces as well as the conformation tensor results at the mesoscopic level. We also compare the molecular dynamics and mesoscopic elastic dumbbell models and review their predictions. In Sec. 5, we summarize the major findings of this work.

2 Numerical technique

The "computer" fluid consists of 128 dumbbells or 256 beads (atoms) that are enclosed in a cubic (primitive) cell. The periodic boundary conditions (PBC) are constructed by populating the entire space by images of the primitive cube. In the flow (x) and neutral (z) directions, standard orthogonal PBC are implemented, whereas in the direction of the velocity gradient (y), the Lees-Edwards "shearing blocks" PBC are used [6]. The system evolves under SLLOD dynamics [2], as described in more detail in [5],

$$\begin{aligned}
\dot{r}_{i\beta} &= \frac{p_{i\beta}}{m} + r_{i\gamma}\chi_{\gamma\beta}, && (1)\\
\dot{p}_{i\beta} &= -\sum_{j=1, j\neq i}^{N} \frac{\partial\varphi(r_{ij})}{\partial r_{ij\beta}} - p_{i\gamma}\chi_{\gamma\beta} - \alpha p_{i\beta}, && (2)\\
\alpha &= \frac{\sum_{i=1}^{N}(-p_{i\beta}\sum_{j=1,j\neq i}^{N}\frac{\partial\varphi(r_{ij})}{\partial r_{ij\beta}} - p_{i\beta}\chi_{\beta\gamma}p_{i\gamma})}{\sum_{i=1}^{N} p_{i\beta}p_{i\beta}}. && (3)
\end{aligned}$$

For Couette flow, the tensor $\underline{\underline{\chi}}$ ($\chi_{\beta\gamma} = \frac{\partial u_\beta}{\partial r_\gamma}$, $\mathbf{u}$ is the streaming velocity) has only one non-zero component that signifies the imposed shear rate, namely $\chi_{xy} = \dot{\gamma}$. $N = 256$, $\mathbf{r_i}$ and $\mathbf{p_i}$ denote position and thermal momenta coordinates of each bead, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$, and m stands for mass of a single bead. α is a Lagrangian like multiplier, often called Gaussian thermostat, whose function is to remove heat generated by the viscous dissipation [7]. The usual summation convention is used in Equations 1-3. Lastly, φ denotes either inter-molecular (Lennard-Jones type, LJ) or intra-molecular (finitely extensible nonlinear elastic type, FENE [8]) potentials,

$$\varphi_{ij}^{LJ}(r_{ij}) = \begin{cases} 4\epsilon[(\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^{6}] & \frac{r_{ij}}{\sigma} < 2.5 \\ 0 & \frac{r_{ij}}{\sigma} \geq 2.5, \end{cases} \qquad (4)$$

and,

$$\varphi_{ij}^{FENE}(r_{ij}) = \begin{cases} -\frac{H(r_{max}-r_{eq})^2}{2}\, ln\,[1 - (\frac{r_{ij}-r_{eq}}{r_{max}-r_{eq}})^2] & r_{min} < r_{ij} < r_{max} \\ \infty & r_{ij} \leq r_{min} \text{ } or \text{ } r_{ij} \geq r_{max}, \end{cases} \qquad (5)$$

ϵ and σ are the Lennard-Jones parameters that denote the depth of the potential well and the collision diameter, respectively. H stands for a spring constant, r_{eq} corresponds to the minimum in the FENE potential, r_{min} and r_{max} are the minimum and the maximum extensions of a single FENE dumbbell. In subsequent sections, the standard nondimensional variables (denoted by *) are used (see also [5]). Tab. 1 summarizes the conversion factors between the nondimensional unit quantities and the SI units.

To be considered as an alternative to the traditional rheological equations of state, the microscopic dumbbell models must be able to predict a range of behavior observed for viscoelastic liquids. To this end, we investigate three types of dumbbell fluids composed of either small ($r^*_{eq} = 0.5$, $r^*_{max} = 0.9$), medium ($r^*_{eq} = 0.5$, $r^*_{max} = 0.9$), or large ($r^*_{eq} = 0.5$, $r^*_{max} = 0.9$) dumbbells. Moreover, the medium dumbbells are studied for three values of the spring constant: $H^* = 37.5, 375$, and 3750, to determine the effect of frequency of vibration on results at different levels of description. A larger spring constant leads to faster vibrations.

Equations 1-3 are solved in the atomic representation [5, 9] by the Verlet [10] half-step leap-frog method that implies that the velocities are calculated at $t + \frac{\Delta t}{2}$, in the middle of each time step. Initially, however, it is necessary to take a half-step in velocities, from $t - \frac{\Delta t}{2}$ to t, and to iterate between Equations 1 and 3 to ensure consistency of these equations [11, 12]. Convergence ($\frac{\Delta\alpha}{\alpha} < 10^{-5}$) is reached rapidly in 3 to 5 iterations but it may take longer at high shear rates. The convergence criterion should not be set lower than 10^{-5}, particularly for calculations in single precision numbers, since α may oscillate between two distinct values. The velocities at $t + \frac{\Delta t}{2}$ are then computed from new (α) at time t as well as from the old velocities at time $t - \frac{\Delta t}{2}$. Finally, the positions at time $t + \Delta t$ are calculated.

We have also solved the SLLOD equations of motion by means of the Gear predictor-corrector method [13]. This technique is easily applicable to different types of equations of motion, since it does not require the reformulation of the algorithm once the equations of motion are altered. The iterations, similar to those encountered in the leap-frog method, are avoided but at the price of a smaller time step. Moreover, occasional resettings of velocities are required to eliminate the effect of the energy drift. Another disadvantage of using the Gear method comes from the fact that Equation 1 is really discontinuous for particles that leave the primitive cube, and are reintroduced at the opposite wall; several past positions are taken into account by the Gear algorithm in predicting the new position. This problem, which results in the aggravated energy drift, is especially important in case of x coordinate where the second term on the right hand side, due to the shear rate, changes its sign. Thus the boundary conditions must be applied in such a way as to alter not only the very last position, as it is done for the Verlet methods, but also positions prior to it. Alternatively, one may introduce two sets of coordinates: the physical, in which a trajectory is computed, and the virtual, that are confined to the primitive cube and utilized in calculation of forces, stresses, etc. We feel that unless there is a need to reproduce exact trajectories, the Verlet leap-frog algorithm offers more advantages, and consequently is employed in all production runs.

3 Long range order

Since the averaged motion of all particles is expected to conform to the imposed linear velocity profile, the SLLOD equations lead to a biased thermostat [14] for atomic liquids at high shear rates. In other words, any disturbance in the flow field is taken as the thermal motion and suppressed. In this section, we would like to make a few comments about the appearance of the Erpenbeck [15] string (ordered) phase whose physical existence has never been proved, nor disproved, for that matter, to our satisfaction. Specifically, it is not clear if formation of the string phase is independent of thermostating, or if the string phase is a transient phenomenon with respect to time and shear rate.

It should be emphasized that formation of the string phase has been reported only for atomic liquids. A recent study that describes rheological properties of n-

hexadecane subjected to very high shear rates by means of the SLLOD algorithm, makes no reference to the string phase [16]. In this investigation of dumbbell liquids, small domains of the string phase within the primitive cell may be identified (see Figs. 1 c,d). However, the ordered phase has been never observed to persist throughout the entire domain of computation, but at the highest shear rates. These observations lead to a conclusion that additional degrees of freedom present in more complex systems, such as in a liquid composed of short polymeric chains, in effect impede formation of highly organized linear structures. The discussion that follows refers only to relatively simple systems of atomic and dumbbell fluids, and is relevant to the so-called structural stability of the FENE micro-rheological models.

Considering a two dimensional system, Evans and Morriss [14] attempted to show that the formation of the string phase is a computational artifact which stems from the enforced linear velocity profile; incidentally, the string phase may appear in two, as well as in three dimensional systems [17]. Loose and Hess put forward an interpretation [18], that Evans and Morriss' results were perhaps effected by poor statistics due to the insufficient number of molecules. In brief, the initial implementation of the profile unbiased thermostat (PUT) required that the computational domain be divided into small cells containing two particles. The streaming velocities would then be calculated in every cell and used to find the peculiar kinetic energy and the kinetic temperature; the peculiar kinetic energy of a particle is calculated on the basis of the peculiar velocity $\mathbf{v}$ which, in turn is obtained by subtracting the average local (streaming) velocity of the fluid $\mathbf{u}$ from the particle laboratory velocity $\mathbf{c}$. Eventually, all velocities would be scaled accordingly, to maintain the total thermal kinetic energy constant. We observe, however, that the cell-averaged instantaneous velocity, to be employed in calculations of the kinetic temperature, may be taken over cells which contain more than 30 molecules [19]. Lastly, it has been said [14] that it is not possible to have simple shear flow of two coexisting phases that exhibit different thermodynamic and transport properties, yet such flows were simulated using the SLLOD equations of motion. We accept this argument but would like to point out that oscillations in the dynamic structure tend to occur in the transitional region between flows of purely amorphous (chaotic) and ordered phases. For example, for the atomic liquid undergoing shearing, at $T^{\star} = 0.722$ and $\rho^{\star} = 0.842$, the oscillations occur only in a narrow shear rate interval around $\dot{\gamma}^{\star} = 2.25$. At a time, the liquid may remain in one phase for tens of thousand of time steps; macroscopically, such behavior corresponds to two distinct values assumed by the viscometric functions, (see Fig. 13a in [5]). According to our computer experiments, no two phase liquid structures could persist unchanged in the shear flow throughout an entire simulation run. At the microscopic level this is shown in two photographs (Figs. 1 a,b) which represent instantaneous configurations of particles taken during the same run but at different time steps. In Fig. 1b, the long range order permeates the whole primitive cube. Although the region lying to the right is slightly distorted it may not be regarded as an amorphous phase. It is also possible that the structural arrangements reflect the finite size effects since the coexistence of chaotic and ordered phases was observed for systems with few thousand molecules. For example, Heyes [20] showed that for a system consisting of 2048 molecules, two phases may coexist not only at $\dot{\gamma}^{\star} = 2.5$ but also at $\dot{\gamma}^{\star} = 5.0$. Similar observations were reported by Yamada and Nosé [21] and by Heyes [22].

We consider the Evans and Morriss [14] objection to the model to be a very serious one. As a matter of fact, the objection threatens the so-called structural stability of the model at high shear rates where strings permeate the flow. Unfortunately, we do not have the necessary computational resources to investigate the formation of the ordered phase for systems with large number of particles, for which PUT could be implemented with sufficient statistics. The structural stability implies that there is no singularity in a model. In other words, if a single parameter is changed, for example a method of thermostating, the prediction of a model should not be significantly affected. However, the lack of the string phase, which is perhaps brought about by PUT, leads to a typical singularity, that is to the disappearance of power law like shear thinning [14]. Although, it has been customary to establish a relationship between the macroscopic phenomena and the formation of the microstructure by using molecular simulations, in the context of NEMD modeling of atomic and dumbbell fluids the existence of the long range order in the form of the string phase is yet to be thoroughly investigated.

4 Results and discussion

4.1 Three levels of description

Models that describe the dynamic behavior of viscoelastic fluids may be conceived at three levels of detail: 1) fully microscopic models that employ computational techniques of molecular [23, 8] and Brownian [24] dynamics, 2) mesoscopic models that have roots in the kinetic theory [3, 25], and 3) macroscopic models that originate in the continuum theory [26]. In order to formulate mesoscopic or macroscopic models it is sufficient to use the physical understanding and hence the variables which are available at these levels of description. For example, the mesoscopic models use internal state variables such as one molecule distribution function, configuration space distribution function, and a conformation tensor [27]. In this section, we give an account of the passage between FENE micro-rheological models, and results at the three levels of description.

In the most detailed manner, the state of a system consisting of N identical beads ($\frac{N}{2}$ dumbbells), is represented as a point in $6N$ dimensional phase ($\mathbf{\Gamma}$) space which is spanned by coordinates that are represented by positions and momenta or velocities of beads. In statistical mechanics, it is customary to introduce the N-

particle probability density function $f(\mathbf{\Gamma}, t)$ that describe the distribution of points in the phase space; $\mathbf{\Gamma} \equiv \mathbf{r}_1^1, \mathbf{r}_2^1, \mathbf{r}_3^2, \mathbf{r}_4^2, \ldots \mathbf{c}_{N-1}^{\frac{N}{2}}, \mathbf{c}_N^{\frac{N}{2}}$, superscripts and subscripts denote dumbbells and beads respectively, and $\mathbf{c}$'s signify laboratory velocities. Specifically, $f(\mathbf{\Gamma}, t)d\mathbf{\Gamma}$ is the probability of finding the system in a given microstate, in the infinitesimal volume element of the phase space $d\mathbf{\Gamma}$ around $\mathbf{\Gamma}$ at time t. The convention of using superscripts for dumbbells and subscripts for beads will be followed later for other variables.

The most obvious result from the SLLOD simulation is a trajectory that portrays the path followed by a system in the phase space. The set of trajectories passing through all points in $\mathbf{\Gamma}$ can be converted to the N-particle nonequilibrium distribution function in the phase space defined in terms of bead positions and their peculiar (thermal) velocities. In this article we report only the steady state distribution functions. To emphasize that from now on the distribution function is based on the peculiar rather than on the laboratory velocities we define f_v,

$$f_v(\mathbf{r}_1^1, \mathbf{r}_2^1, \ldots, \mathbf{v}_{N-1}^{\frac{N}{2}}, \mathbf{v}_N^{\frac{N}{2}}) = f(\mathbf{r}_1^1, \mathbf{r}_2^1, \ldots, \mathbf{c}_{N-1}^{\frac{N}{2}} - \mathbf{r}_{N-1}^{\frac{N}{2}} \cdot \underline{\underline{\chi}}, \mathbf{c}_{N-1}^{\frac{N}{2}} - \mathbf{r}_{N-1}^{\frac{N}{2}} \cdot \underline{\underline{\chi}}). \quad (6)$$

In a similar manner the subscript v is added to $\mathbf{\Gamma}$ to denote the newly defined phase space. The explicit functional dependence of f_v on t is dropped to reflect the steady state nature of the distribution function.

The function f_v represents the most detailed description of the dynamic state of the fluid, but at the same time is not immediately useful. Even the hardware necessary to calculate and store f_v for 256 beads is outside of reach of today's technology. The alternative is to take pictures or make movies to represent a state of the fluid at an instant and then to rely on the human capability of pattern recognition to gain a physical insight. Such an analysis will be given in Sec. 4.2.

At the mesoscopic level, the microscopic FENE model yields a set of predictions that are derived from Equation 6 either by calculating statistical averages over

f_v or by reducing the N-particle distribution function to more tractable single particle distribution functions in velocity and configurational spaces. In the next few paragraphs, we formally introduce variables and functions whose importance will become apparent in Sec. 4.3. Firstly, we define two distribution function which are obtained by contracting the phase space.

$$\Psi(\mathbf{r},\mathbf{R}) = \int f_v(\mathbf{\Gamma}_v)\delta(\mathbf{r} - \frac{\mathbf{r}_1+\mathbf{r}_2}{2})\delta(\mathbf{R}+\mathbf{r}_2-\mathbf{r}_1)d\mathbf{\Gamma}_v \tag{7}$$

is termed a contracted single dumbbell configuration distribution function, whereas

$$\Xi(\mathbf{v}_1,\mathbf{v}_2) = \int f_v(\mathbf{\Gamma}_v)\, d\mathbf{r}_1 \dots d\mathbf{r}_N\; d\mathbf{v}_3 \dots d\mathbf{v}_N \tag{8}$$

denotes contracted two bead peculiar velocity distribution function in a homogeneous fluid.

Arbitrarily, but motivated by the physical significance we contract $\Psi(\mathbf{r},\mathbf{R})$ further to obtain as the final result distributions of end-to-end distances $\Psi(R)$ in Fig. 2, and alignment angles $\Psi(\phi)$ in Fig. 3, where for i^{th} dumbbell, ϕ is calculated as,

$$\phi^i = tan^{-1}(\frac{R^i_y}{R^i_x}). \tag{9}$$

In addition, we compute the correlation between the average end-to-end distance with the alignment angle, as shown in Fig. 4.

The analogous procedure in case of the velocity distribution function $\Xi(\mathbf{v}_1,\mathbf{v}_2)$ leads to a single bead distribution $\Xi(v_1)$, as well as the center of mass ($\mathbf{V} = \frac{\mathbf{v}_1+\mathbf{v}_2}{2}$) and the relative ($\mathbf{v} = \mathbf{v}_1 - \mathbf{v}_2$) velocity distributions, $\Xi(V)$ and $\Xi(v)$ respectively (see Figs. 5 and 6). Furthermore, additional velocity distribution functions are constructed by observing that $\mathbf{v}$ may be decomposed into the vibrational ($\mathbf{v}_{||}$) and the rotational ($\mathbf{v}_{\perp}$) components (see Fig. 7).

It is possible to calculate $\Xi(v_1)$, $\Xi(V)$ and $\Xi(v)$ in the limit of small shear rates on the assumption that bead peculiar velocities are arranged according to the Maxwellian distribution,

$$\Xi(\mathbf{v}_1,\mathbf{v}_2) = (\frac{m}{2\pi kT})^3\, exp(-\frac{m(\mathbf{v}_1^2+\mathbf{v}_2^2)}{2kT}). \tag{10}$$

The integration is performed in spherical coordinates by using the appropriate coordinate transformation,

$$\Xi(V) = \int_0^\infty \int_0^{2\pi} \int_0^{2\pi} \int_0^\pi \int_0^\pi V^2 v^2 sin(\theta_V) sin(\theta_v) (\frac{m}{2\pi kT})^3 \, exp(-\frac{m}{kT}(V^2+\frac{v^2}{4})) d\theta_V d\theta_v d\phi_V d\phi_v dv, \tag{11}$$

which leads to,

$$\Xi(V) = \frac{4}{\sqrt{\pi}}(\frac{m}{kT})^{3/2} \, exp(-\frac{mV^2}{kT})V^2. \tag{12}$$

A similar integration gives,

$$\Xi(v) = \frac{1}{2\sqrt{\pi}}(\frac{m}{kT})^{3/2} \, exp(-\frac{mv^2}{4kT})v^2. \tag{13}$$

For completeness,

$$\Xi(v_1) = \frac{2}{\sqrt{\pi}}(\frac{m}{kT})^{3/2} \, exp(-\frac{mv_1^2}{2kT})v_1^2. \tag{14}$$

The equilibrium velocity distribution functions are shown as dashed lines in Figs. 5 and 6.

In the same spirit, the equilibrium distribution of the end-to-end distances may be obtained from the Gibbs expression by neglecting the inter-particle interactions and integrating out velocity and most of the position coordinates. This is correct for $H \to \infty$, but fails for small values of the spring constant. Using φ^{FENE} defined in Equation 5,

$$\Psi(R) = \frac{4\,\pi\, R^2 exp(-\frac{\varphi^{FENE}(R)}{kT})}{\int exp(-\frac{\varphi^{FENE}(R)}{kT})\, d\mathbf{R}} = \frac{R^2[1-(\frac{R-r_{eq}}{r_{max}-r_{eq}})^2]^{\frac{b}{2}}}{\int_{r_{min}}^{r_{max}} R^2[1-(\frac{R-r_{eq}}{r_{max}-r_{eq}})^2]^{\frac{b}{2}}\, dR}, \tag{15}$$

where

$$b = \frac{H\,(r_{max}-r_{eq})^2}{k_B\,T}. \tag{16}$$

k_B denotes the Boltzmann constant. The above expression may be derived also from the dumbbell diffusion equation [28].

The changes that take place in sheared fluid may be observed and qualified at the microscopic level. Although useful, this type of characterization is not adequate to reveal the general nature of collective structural modes in sheared fluid in a way that would be compatible with neutron and light scattering experiments, and that could aid in calculation of thermodynamic properties. In this paper, the structure of the FENE fluids is probed by two complementary techniques at the mesoscopic level. Configuration of a single dumbbell in space is investigated in terms of the first conformation tensor $\underline{\underline{c}}$ whereas the spatial structure formed by a system of dumbbells is analyzed by using the second conformation tensor $\underline{\underline{C}}$, to be defined later.

The conformation tensor $\underline{\underline{c}}$ is especially convenient to probe the average alignment of FENE dumbbells under shear,

$$c_{\alpha\beta}(\mathbf{r},\dot{\gamma}) = \int R^i_\alpha R^i_\beta \Psi(\mathbf{r},\mathbf{R}) d\mathbf{R}. \tag{17}$$

The functional dependence of $\underline{\underline{c}}$ on $\mathbf{r}$ disappears since the Couette flow field implies a homogeneous structure. The mean square end-to-end distance of dumbbells is trivially related to the radius of gyration and is given by the trace of the conformation tensor,

$$R^2(\dot{\gamma}) = c_{xx}(\dot{\gamma}) + c_{yy}(\dot{\gamma}) + c_{zz}(\dot{\gamma}). \tag{18}$$

In case of dumbbells, the traceless part of the normalized conformation tensor is the same as the alignment tensor whose largest eigenvalue is termed the birefringence parameter. The conformation tensor may be illustrated graphically as an ellipsoid in three dimensional space with its axes equal to two times the eigenvalues of $\underline{\underline{c}}$. Conversely, as in this section, each ellipsoid may be represented by three ellipses which are obtained as traces of cross-sections of the ellipsoid with $z = 0$, $y = 0$, and $x = 0$ planes. Directions of the ellipses are defined by three alignment angles, denoted as χ^{al}_{xy}, χ^{al}_{xz}, χ^{al}_{yz}, that indicate the angles between the long axes of the ellipses and the abscissas in xy, xz, and yz Cartesian coordinates; see Fig. 8 and Tab. 2. Clearly, χ^{al}_{xy} may be considered as an approximation to the birefringence extinction angle.

In general, it is possible to define a second conformation tensor by a straightforward generalization of the normalized first conformation tensor to include elements of the global structure of the FENE fluids. The new tensor is a function of a distance r that separates centers of masses of two arbitrary dumbbells, which are located at $\mathbf{R}^i$, $\mathbf{R}^j$, and similarly to $\underline{\underline{c}}$ is obtained by taking an average, this time over the N-particle configuration distribution function. Replacing the long integration by the ensemble average $< \ldots >$ to facilitate abbreviate notation, and normalizing by the single dumbbell density ρ_d which is equal to $\frac{\rho}{2}$, we write $\underline{\underline{C}}$ as,

$$C_{\alpha\beta}(r,\dot{\gamma}) = \frac{1}{\rho_d} < \frac{2}{N} \sum_{i}^{\frac{N}{2}-1} \sum_{j>i}^{\frac{N}{2}} \frac{(R^i_\alpha - R^j_\alpha)(R^i_\beta - R^j_\beta)}{(\mathbf{R}^i - \mathbf{R}^j)^2} \delta(\mathbf{r} + \mathbf{R}^j - \mathbf{R}^i) > . \qquad (19)$$

It is easy to observe that $\underline{\underline{C}}$ is reminiscent on the dumbbell pair distribution function g, but incorporates the anisotropy which is characteristic of fluids under shear,

$$g(r,\dot{\gamma}) = C_{xx}(r,\dot{\gamma}) + C_{yy}(r,\dot{\gamma}) + C_{zz}(r,\dot{\gamma}). \qquad (20)$$

Heyes [29, 20] introduced the pair angular distribution function for atomic fluids, that differs by a constant from the second conformation tensor,

$$g_{\alpha\beta}(r,\dot{\gamma}) = 15\, C_{\alpha\beta}(r,\dot{\gamma}). \qquad (21)$$

g is plotted in Fig. 9 whereas the diagonal and off-diagonal components of $\underline{\underline{C}}$ are shown in Figs. 10 and 11, respectively.

Lastly, the components of the potential and peculiar kinetic energies are considered as mesoscopic variables since they represent averages of microscopic quantities which may not be measured by macroscopic instruments. The graphs in Fig. 12 display the variation of translational (center-of-mass), rotational and vibrational thermal energies with shear rate. In this study, the total kinetic energy of a system consisting of 128 dumbbells at $T^\star = 1$,

$$K(\dot{\gamma}) = K^x_{cm}(\dot{\gamma}) + K^y_{cm}(\dot{\gamma}) + K^z_{cm}(\dot{\gamma}) + K_{vib}(\dot{\gamma}) + K_{rot}(\dot{\gamma}), \qquad (22)$$

is maintained constant at 384ϵ, thus at small shear rates,

$$< K_{cm}^{x} >=< K_{cm}^{y} >=< K_{cm}^{z} >=< \frac{1}{2}K_{rot} >=< K_{vib} >= 64\epsilon. \tag{23}$$

The inter-particle and bonded potentials are shown in Fig. 13, as obtained from,

$$\Phi(\dot{\gamma}) = \Phi_{inter}(\dot{\gamma}) + \Phi_{intra}(\dot{\gamma}) = \sum_{i}^{N-1}\sum_{j>i}^{N}\varphi_{ij}^{LJ} + \sum_{i=1}^{\frac{N}{2}}\varphi^{i,FENE}. \tag{24}$$

We add that for homonuclear molecules the total and reduced masses which are associated with translational and relative velocities are $2m$ and $\frac{1}{2}m$, respectively [30]. The consistency of the velocity and the energy data (Figs. 7 and 12 c,d) may be verified by integrating square of velocities over velocity distributions to obtain kinetic energies; see Tab. 3. For example,

$$K_{vib}(\dot{\gamma}) = \frac{1}{2}\int_{0}^{\infty}\frac{m}{2}v_{\|}^{2}\Xi(v_{\|})dv_{\|}, \tag{25}$$

$$K_{rot}(\dot{\gamma}) = \frac{1}{2}\int_{0}^{\infty}\frac{m}{2}v_{\perp}^{2}\Xi(v_{\perp})dv_{\perp}. \tag{26}$$

It is important to realize that at high shear rates, the tails of the rotational and the vibrational velocity distributions decay slower than at equilibrium. Thus the integration should be carried out until $v_{\perp}^{*} = v_{\|}^{*} \simeq 10$ to avoid underestimating of the kinetic energies.

At the most general macroscopic level of description, the dynamic state of a polymeric fluid is characterized by a set of functions derived from viscometric experiments conducted usually in plate-plate, cone-plate or capillary viscometers. These functions could also be recovered from the results of molecular simulations as the final stage of the reduction process, from the definition of the total stress tensor,

$$\sigma_{\alpha\beta}(\dot{\gamma}) = -\frac{1}{V}(\sum_{i=1}^{N}\frac{p_{i\alpha}p_{i\beta}}{m_{i}} + \sum_{i=1}^{N-1}\sum_{j>i}^{N}r_{ij\alpha}F_{ij\beta}), \tag{27}$$

where the forces ($F_{ij\beta}$) are calculated from,

$$F_{ij\beta} = -\frac{\partial\varphi(r_{ij})}{\partial r_{ij\beta}}. \tag{28}$$

Actually, $F_{ij\beta}$ takes two different forms for intra- and inter-molecular interactions, as shown in Sec. 2. Here, V denotes the total volume of the primitive cube. In the standard rheological notation [31], the viscosity (η), the first (N_1) and the second (N_2) normal stress differences are,

$$\eta(\dot{\gamma}) = \frac{\sigma_{xy}(\dot{\gamma})}{\dot{\gamma}}, \tag{29}$$

$$N_1(\dot{\gamma}) = \sigma_{xx}(\dot{\gamma}) - \sigma_{yy}(\dot{\gamma}), \tag{30}$$

$$N_2(\dot{\gamma}) = \sigma_{yy}(\dot{\gamma}) - \sigma_{zz}(\dot{\gamma}), \tag{31}$$

and in addition the hydrostatic pressure P is,

$$P(\dot{\gamma}) = -\frac{1}{3}(\sigma_{xx}(\dot{\gamma}) + \sigma_{yy}(\dot{\gamma}) + \sigma_{zz}(\dot{\gamma})). \tag{32}$$

Although this level of description is devoid of the richness of detail of the original formulation, it is actually here that predictions of the model may be verified with the experimental data. The viscometric functions of dumbbell fluids are described in the complementary publication [5].

The distribution functions Ψ, Ξ and the both second conformation tensors $\underline{\underline{c}}$ and $\underline{\underline{C}}$ were obtained by sampling the configuration and velocity spaces every 25 steps, during 100,000 steps runs, to avoid the correlation between values at the successive time steps. Histograms were constructed, for Ψ and Ξ to account for the distributions, and the total area of all vertical rectangles was then normalized. From the same runs, the correlation between the average end-to-end distance and the alignment angle was found by counting the number and adding up the extensions of dumbbells that fell within small intervals ($\Delta\phi = 1.8°$) and then averaging end-to-end distances within each interval. The errors associated with kinetic and potential energies (Figs. 12 and 13) denote one standard deviation and were calculated on the basis of subaverages, each taken over 10% of a simulation run [5].

We summarize this section by stating that the insistence on formulating a rheological model at the most detailed level has two consequences. Firstly, more macroscopic predictions are found through the gradual contraction of the original N-particle distribution function. Numerically, this corresponds to computation of time averages. Secondly, the complex mathematics that must be used in order to obtain predictions from mesoscopic and macroscopic models of viscoelastic fluids is avoided.

4.2 Microscopic results - visual observations

Within the context of molecular simulations, results are considered reproducible at all levels of description, as allowed by statistical uncertainties, if they do not depend on initial conditions. For atomic liquids, even at the same $\dot{\gamma}^*$, not all string phases are alike ! In photographs similar to Fig. 1b, we have observed that a type of the string phase depends on the pre-shearing history of a sample. If two string structures generated at the same shear rate, are projected onto the yz plane and differ by a certain angle of rotation, the macroscopic results would not the same. This is especially true for the first and second normal stress differences, whereas viscosities are less sensitive to details in the structure of atomic fluids, above the phase transition. To ensure the reproducibility at intermediate and high shear rates, fresh samples were always pre-sheared, at $\dot{\gamma}^*$ corresponding to the phase transition (between amorphous and ordered phases), before imposing the final shear rate. Curiously enough, smaller time steps were not found to lead to improved reproducibility. In case of dumbbell fluids, the viscometric functions are easier to reproduce, provided that computer runs are sufficiently long. However, fresh samples of dumbbell fluids must be always pre-sheared for $\dot{\gamma}^* \geq 10$, to avoid so-called arrested states, where the system's trajectory is confined to a small volume of the phase space.

We now describe new long and short ranged structures observed at the microscopic level, by a means of three dimensional computer visualization. If the equilibrium bead-bead extension of a single dumbbell (r^*_{eq}) is chosen to be less that 0.67 then under moderately low shear rates ($0.2 < \dot{\gamma}^* < 2.0$), the liquid separates into a void-like space and a horizontal layer that contains all dumbbells; such an arrangement is called the separated structure. This leads not only to distinct macroscopic and mesoscopic results [5] than for larger dumbbell fluids.

As Figs. 1c and 1d demonstrate, medium and also large dumbbells may form small domains consisting of two strings. Within each domain dumbbells are aligned perpendicular rather than parallel to the direction of flow. This structure is local and transient with respect to shear rate. It disappears for $\dot{\gamma}^* > 10$ where more global structure such as string phase becomes more important.

Small peaks displayed by the normal components of the second conformation tensor (as defined in Equation 19) and by the dumbbell pair distribution function at small r^* (Fig. 8c) indicate that it is in fact possible for two big dumbbells to pass through each other. Sometimes, this passing cannot be completed as the result of changes in end-to-end distances of dumbbells due to vibration. Consequently, a local structure composed of two dumbbells is formed. This structure is stabilized for small shear rates but disintegrates into two separate dumbbells in the non-Newtonian regime.

4.3 Mesoscopic results - statistical averaging

Since for large values of the spring constant (H^*), the intra-molecular interactions dominate the inter-molecular forces, it is possible to neglect the latter and consider single dumbbells as if there were isolated. This assumption proved to be very successful in the estimation of the dominant frequency of vibrations [5]. The total potential energy may then be approximated by the sum of intra-molecular contributions due to all dumbbells. Subsequently, the contracted single dumbbell equilibrium distribution function $\Psi(R)$ is found explicitly as shown in Equation 15. For $H^\star = 3750$, Fig. 2 clearly indicates that the equilibrium distribution function of the end-to-end distances corresponds very well to similar distributions for slightly sheared dumbbell fluids, especially for longer dumbbells. As a matter of fact, the fit is even better for $H^\star = 3750$, but is poor for $H^\star = 37.5$, when the assumption of isolated dumbbells is no longer valid for estimation of $\Psi(R)$. In case of $H^\star = 37.5$, $\Psi(R^\star)$ may even exhibit several local extrema.

Three additional observations follow from Fig. 2. (1) At higher shear rates, due to the formation of the anisotropic structure that impedes free vibration of dumbbells, $\Psi(R^\star)$ becomes sharper except for the phase transition region (see $\Psi(R^\star)$ for $\dot{\gamma}^\star = 5$, $r^\star_{eq} = 1.3$, and $r^\star_{max} = 1.7$) where dumbbells start to form a highly oriented spatial structure and hence are more likely to become shorter or longer than in fully developed flows of chaotic and ordered phases. (2) The theoretical equilibrium distribution functions are off-set to the right with respect to the computed distributions at low shear rates, since dumbbells in the liquid tend to be slightly smaller than freely vibrating dumbbells. (3) Dumbbells that form the separated structure (Sec. 4.2) are smaller on the average than in the absence of separation. At the mesoscopic level of description, this phenomenon is observed most clearly in Fig. 4b but it can also be noted in Fig. 2 for $\dot{\gamma}^\star = 0.8$, $r^\star_{eq} = 0.5$, and $r^\star_{max} = 0.9$) and in Fig. 8a (center and bottom).

The fact that NEMD models include inter-molecular potentials should be recognized as the most significant difference between our models and the mesoscopic elastic dumbbell theories, for which the inter-particle potentials are not taken into account and the behavior of a polymeric fluid is determined by the interplay between intra-molecular and solvent-polymer interactions.

The radial one dumbbell distribution function $\Psi(\phi)$ (Fig. 3) and the first conformation tensor (Fig. 8) convey complementary messages. The alignment of dumbbells at a certain angle to the direction of flow is noticeable even at the smallest shear rates. In the separated structure, dumbbells are perpendicular to the direction of flow, on the average (Fig. 3a). Since ϕ is defined in the xy plane, the neutral direction z has been integrated out in $\Psi(\phi)$. This implies that dumbbells that are approximately perpendicular to flow, as shown in Figs. 1 c,d, are not accounted for correctly in $\Psi(\phi)$. However, at the intermediate shear rates $\dot{\gamma}^* \simeq 8-10$, their presence may be inferred from the shapes of ellipses that represent the first conformation tensor in xz and yz planes (Figs. 8 b,c). Above $\dot{\gamma}^* \simeq 10$, the characteristic structure that is related to dumbbells parallel to the neutral direction disappears, as corroborated by a sudden decrease of the long axis of the $\dot{\gamma}^* = 14$ ellipsis (Fig. 8b, bottom). This phenomenon contributes to the so-called secondary structural rearrangements that affect the macroscopic results at $\dot{\gamma}^* \simeq 10$, such as a minor change of slopes of the viscosity curves [5]. We conclude this paragraph by noting that only for $\dot{\gamma} \rightarrow 0$ there is no correlation between the average end-to-end distance and the alignment (Fig. 4).

In the zero-shear limit, the diagonal components of the second conformation tensor are essentially the same, but in the non-Newtonian regime the fluid becomes rapidly anisotropic. In case of medium size dumbbells, the first peaks in C_{xx}, C_{yy}, and C_{zz} signify that at the elevated shear rates the centers of masses of dumbbells are located closer to each other in the y and z ($\simeq 1.1$) than in x ($\simeq 1.9-2.0$) direction (Figs. 10 a-c). In this light, g represents average characteristics of the diagonal components of the second conformation tensor (Fig. 9), analogously to the hydrostatic pressure that gives an average of the normal stresses; both g and P are not suitable to describe the state of a highly sheared fluid. The off-diagonal elements of $\underline{\underline{C}}$ show that even at small shear rates, dumbbell fluids may be anisotropic, especially in the xy plane (see Fig. 11). In the spirit of modern thermodynamics [32], we feel that new meso-rheological theories for complex viscoelastic fluids ought to include a second state variable, such as $\underline{\underline{C}}$, that would perhaps convey the global structure of a polymeric liquid.

The formation of the string phase leads to a solid-like behavior in the direction of the velocity gradient and in the neutral direction z, as observed by Heyes [20, 11], who measured the force and the stress autocorrelation function as well as the directional self diffusion coefficients for atomic liquids. A fluid-like behavior is preserved in the direction of flow. Similarly, for dumbbell fluids the fluid and solid resembling properties are observed in x and z directions, respectively. However, as opposed to atomic systems, the liquid like behavior is noted in the direction of the velocity gradient, for the intermediate shear rates. Dumbbells tend to be aligned within the strings in the direction of flow, but it is actually possible to have dumbbells with beads belonging to two different strings (Figs. 1 c,d). Such a structure extends throughout the primitive cube and is stabilized in the horizontal direction (z) by the fact that both strings are subjected to the same shear. The structural constrains and the lack of gradients within horizontal planes create an impediment to the dumbbell self-diffusion in the z direction. This is corroborated by the translational z kinetic energy which monotonically decreases above $\dot{\gamma}^{\star} \simeq 1$ for all values of $H^{\star}$ (Figs. 12 a,b).

If a bead leaves its place in the original string and diffuses in the y direction, it eventually assumes a position in a string that moves with a different streaming velocity. The energy barrier at $r^{\star}_{max}$, within a single FENE dumbbell, prevents the infinite extension of the bead-bead bond and exerts a force on the dumbbell to place itself in the horizontal plane either in the direction of flow or perpendicular to it. This aids in the self-diffusion of dumbbells in the y direction. The crossover of the translational kinetic energies in the x and y directions (Figs. 12 a,b) is related to the phase transition at $\dot{\gamma}^{\star} \simeq 5$. However, at larger values of $H^{\star}$ the kinetic energy of translation in the direction of the velocity gradient tends to overshoot the kinetic energy of the center of mass, in the direction of flow. At high shear rates, both energies seem to diverge indicating that the fluid becomes more solid-loke in the y

direction (Figs. 12 a,b); see also Figs. 10 b,c for $\dot{\gamma}^* = 14$. Finally, a jump in the translational z kinetic energy (Figs. 12a), around $\dot{\gamma}^* \simeq 10$, may be related to the same structural rearrangements which are also reflected in the change of slopes of the viscosity curves and caused by the destruction of small two-strings structures containing dumbbells normal to the direction of flow.

We terminate the discussion of the kinetic energies by making three comments. (1) The sum of the translational energies is approximately constant, for the whole range of shear rates, indicating that thermostating either total or the center of mass energy should not significantly influence the results. The total kinetic energy is constant during simulations, thus the sum of the vibrational and the rotational energies remains nearly unchanged (Figs. 12 a-d). (2) At larger values of the spring constant, especially for $H^* = 3750$, the bead-bead distance in dumbbells becomes constrained around the equilibrium position, with increasing shear rates (see also Fig. 2). Since the equilibrium position corresponds to the minimum in the trough of the FENE potential, it follows that the intra-particle potential energy of dumbbells decreases, as shown in Fig. 13a. This also explains, to a certain extend, the profound decrease in the vibrational kinetic energy for $H^* = 3750$ observed in Fig. 12d. (3) Although, at high shear rates the peaks in the velocity distributions come to be slender, taller, and are displaced towards the low velocities (Figs. 5-7), the tails of the distributions store significant parts of the kinetic energies.

4.4 Comparison to the elastic dumbbell theory

Unlike NEMD dumbbell models, mesoscopic elastic dumbbell theories have been developed for diluted polymeric solutions by neglecting inter-molecular interaction (see Sec. 4.3). Therefore, it comes as no surprise that both types of models exhibit many distinct features. For example, the excluded volume effects and hydrodynamic interactions are treated exactly for micro-rheological models, but only approximately for classical dumbbells [3], At the microscopic level, the microscopic FENE potentials take into account physical dimensions of beads (Equation 5), whereas at the mesoscopic level, beads are considered to be points. It is customary for mesoscopic dumbbell models to rely on the assumption of the Maxwellian velocity distribution, which has been shown not to be satisfied even at relatively low shear rates (Figs. 5 and 6). There have been attempts, notably by Bird and his students [3], to introduce anisotropic velocity distributions by redefining Equation 10. Although this procedure is legitimate, it emphasizes the fact that velocity distribution itself is part of a mesoscopic model, rather than a prediction as for micro-rheological simulations. In addition, both types of models consider temperature to be defined in the kinetic rather than in the thermodynamic sense.

Nonequilibrium one dumbbell configuration space distribution functions were calculated numerically within the context of the mesoscopic theory by Fan [33] who concluded that the direction of the maximum extension of dumbbells becomes aligned to the direction of flow, at high shear rates. Our results indicate that on the average the longest dumbbells are approximately at between 30° and 50° to the direction of flow regardless of shear rate (Figs. 4 a-c). However, it should be noted that at high shear rates a relatively small number of dumbbells is aligned at large angles to the direction of flow, as corroborated by a large scatter in data in Figs. 4 a-c, and in addition by the dominant angle of alignment as shown in Figs. 3 a-c and 8 a-c.

We have already noted that at high shear rates dumbbells tend to remain closer to the equilibrium position $r^{\star}_{eq}$ (Fig. 2). The opposite effect was reported by Fan [33]. These observations may be reconciled, however, by noticing that in case of concentrated solutions the conformation of dumbbells is controlled by the shear depended spatial structure which is unlikely to be formed in diluted solutions.

5 Conclusions

We have shown how direct simulations of the time evolution of a model macromolecular fluid yield a complete set of microscopic, mesoscopic and macroscopic results. A major advantage of the model is the fact that it is formulated in terms of ordinary, rather than partial differential equations. It leads directly to the results and avoids complex numerical mathematics that is essential in most rheological models based on the kinetic theory. In addition, we have pointed out similarities and differences in the development and in predictions of the microscopic and the mesoscopic elastic dumbbell models.

The major accomplishments of this work may be summarized as follows:

1. The string phase that pervades atomic liquids at the higher shear rates is shown to be less important for dumbbell fluids. It is believed that the ordered phase would not occur for model liquids which incorporate more degrees of freedom than dumbbell fluids. In this light, further research on model fluids should be concentrated either on investigation of problems related to the formation of the string phase as outlined in Sec. 3 and in [5], or on studying multi-bead fluids.

2. At the molecular level, several new structures are identified for the first time. For $0.2 < \dot{\gamma}^{\star} < 2.0$, very small FENE dumbbells ($r^{\star}_{eq} < 0.67$) form the separated structure. It is conceivable that such a structure is stabilized by the small physical size of the primitive cube. On the other hand, two very large dumbbells may lock themselves into a cross-like arrangement that seems to break at high shear rates. This structure forms when two big dumbbells attempt to pass through each other. In case of medium size dumbbells ($r^{\star}_{eq} = 0.9$, $r^{\star}_{max} = 1.4$), a distinct crosslinked arrangement is observed that consists of two parallel strings that lie in the horizontal plane; each string contains only one bead of each dumbbell. This structure ceases to be seen for $\dot{\gamma}^{\star} >$

10. At high shear rates, neither atomic nor dumbbell data are reproducible, unless fresh samples are pre-sheared. In case of atomic fluids, oscillations are observed between amorphous and ordered phases, in the transitional regime.

3. Original mesoscopic results are calculated directly for dumbbell fluids in terms of configurational and velocity distribution functions as well as in the form of the first and second conformation stress tensors. Even at small shear rates, the anisotropic effects are observed in the system, such as dumbbells aligning themselves at 45° to the direction of flow. At higher shear rates, the alignment increases to approach 0°. Dumbbells that are at between 30° and 50° to the direction of flow are the longest, and those at around −40° are the shortest, regardless of the shear rate and the spring constant. With the exception of the Newtonian limit, the alignment angle and the average end-to-end distance are correlated. Only at low shear rates, velocities of beads and dumbbells are arranged according to the Maxwellian distribution. In the complementary publication [5], it was found that the viscometric functions of dumbbell fluids depend strongly on the size, but not on the frequency of vibrations of a single dumbbell. Similar conclusions may be drawn for mesoscopic, and to a certain extend for microscopic results.

6 Acknowledgments

We would like to express our gratitude to Joel Welling of the Pittsburgh Supercomputing Center for making available a three dimensional visualization software package called P3D, which was indispensable in preparing the photographs. Financial support for this project was provided by the *Natural Sciences and Engineering Research Council of Canada* and *Fonds pour la formation de chercheurs et l'aide à la recherche* of the Province of Québec.

References

[1] M. Grmela. Coupling between microscopic and macroscopic dynamics in NEMD. *Phys. Lett. A*, 174:59–65, 1993.

[2] D. J. Evans. Nonequilibrium molecular dynamics. In Ciccotti G. and W.G. Hoover, editors, *Molecular-Dynamics Simulation of Statistical-Mechanical Systems*, pages 221–240. North-Holland, Oxford, 1986.

[3] R. B. Bird, C. F. Curtiss, R. C. Armstrong, and O. Hassager. *Dynamics of Polymeric Liquids*, volume 2. John Wiley & Sons, New York, 2 edition, 1987.

[4] D. M. Heyes. Non-Newtonian behavior of simple liquids. *J. Non-Newt. Fl. Mech.*, 21:137–155, 1986.

[5] B. Z. Dlugogorski, M. Grmela, and P. J. Carreau. Viscometric functions for FENE and generalized Lennard-Jones dumbbell liquids in Couette flow: Molecular dynamics study. *J. Non-Newt. Fl. Mech. (in press)*, 1993.

[6] A. W. Lees and S. F. Edwards. The computer study of transport processes under extreme conditions. *J. Phys. C: Solid State Phys.*, 5:1921–1929, 1972.

[7] D. J. Evans and G. P. Morriss. Nonequilibrium molecular-dynamics simulation of Couette flow in two-dimensional fluids. *Phys. Rev. Lett.*, 51(19):1776–1779, 1983.

[8] J. W. Rudisill and P. T. Cummings. The contribution of internal degrees of freedom to the non-Newtonian rheology of model polymer fluids. *Rheol. Acta*, 30:33–43, 1991.

[9] A. J. C. Ladd. Equations of motion for non-equilibrium molecular dynamics simulations of viscous flow in molecular fluids. *Mol. Phys.*, 53(2):459–463, 1984.

[10] L. Verlet. Computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.*, 159(1):98–103, 1967.

[11] D. M. Heyes. The nature of extreme shear thinning in simple liquids. *Mol. Phys.*, 57(6):1265–1382, 1986.

[12] D. M. Heyes. Transport coefficients of Lennard-Jones fluids: A molecular-dynamics and effective hard-sphere treatment. *Phys. Rev. B*, 37(10):5677–5696, 1988.

[13] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids.* Clarendon Press, Oxford, 1987.

[14] D. J. Evans and G. P. Morriss. Shear thickening and turbulence in simple fluids. *Phys. Rev. Lett.*, 56(20):2172–2175, 1986.

[15] J. J. Erpenbeck. Shear viscosity of the hard-sphere fluid *via* nonequilibrium molecular dynamics. *Phys. Rev. Lett.*, 52(15):1333–1335, 1984.

[16] A. Berker, S. Chynoweth, U. C. Klomp, and Y. Michopoulos. Non-equilibrium molecular dynamics (NEMD) simulations and the rheological properties of liquid n-hexadecane. *J. Chem. Soc., Faraday Trans.*, 88(13):1719–1725, 1992.

[17] L. V. Woodcock. Origins of shear dilatancy and shear thickening phenomena. *Chem. Phys. Lett.*, 111(4,5):455–461, 1984.

[18] W. Loose and S. Hess. Rheology of dense model fluids *via* nonequilibrium molecular dynamics: shear thinning and ordering transition. *Rheol. Acta*, 28:91–101, 1989.

[19] B. Z. Dlugogorski, M. Grmela, and P. J. Carreau. Direct numerical studies of viscous flow of two dimensional liquids. *In preparation, École Polytechnique*, 1993.

[20] D. M. Heyes. Shear thinning and thickening of the Lennard-Jones liquid. *J. Chem. Soc., Faraday Trans. 2*, 82:1365–1383, 1986.

[21] T. Yamada and S. Nosé. Two-phase coexistence of string and liquid phases: Nonequilibrium molecular-dynamics simulation of Couette flow. *Phys. Rev. A*, 42(10):6282–6291, 1990.

[22] D. M. Heyes. Some physical consequences of large shear rates on simple liquids. *J. Chem. Phys.*, 85(2):997–1009, 1986.

[23] S. Hess. Rheological properties *via* nonequilibrium molecular dynamics: From simple towards polymeric liquids. *J. Non-Newt. Fl. Mech.*, 23:305–319, 1987.

[24] J. W. Rudisill and P. T. Cummings. Brownian dynamics simulation of model polymer fluids in shear flow. I. Dumbbell models. *J. Non-Newt. Fl. Mech.*, 41:275–288, 1992.

[25] M. Grmela. Thermodynamic and rheological modeling: Polymeric liquid crystals. In A.A. Collyer and L.A. Utracki, editors, *Polymer Rheology and Processing.* Elsevier, London, 1990.

[26] R. B. Bird, R. C. Armstrong, and O. Hassager. *Dynamics of Polymeric Liquids*, volume 1. John Wiley & Sons, New York, 2 edition, 1987.

[27] M. Grmela and P. J. Carreau. Conformation tensor rheological models. *J. Non-Newt. Fl. Mech.*, 23:271–294, 1987.

[28] R. C. Armstrong. Kinetic theory and rheology of dilute solutions of flexible macromolecules. I Steady state behavior. *J. Chem. Phys.*, 60(3):724–728, 1974.

[29] D. M. Heyes, J. J. Kim, C. J. Montrose, and T. A. Litovitz. Time dependent nonlinear shear stress effects in simple liquids: A molecular dynamics study. *J. Chem. Phys.*, 73(8):3987–3996, 1980.

[30] J. D. Johnson, M. S. Show, and B. L. Holian. The thermodynamics of dense fluid nitrogen by molecular dynamics. *J. Chem. Phys.*, 80(3):1279–1294, 1984.

[31] J. M. Dealy. Official nomenclature for material functions describing the response of a viscoelastic fluid to various shearing and extensional deformations. *J. Rheol.*, 28(1):181–195, 1984.

[32] F. M. Dennery, D. Fargue, and A. Truyol. Contemporary thermodynamics. *Intl. Chem. Eng.*, 32(1):73–81, 1992.

[33] X.-J. Fan. Viscosity, first normal-stress coefficient, and molecular stretching in dilute polymer solutions. *J. Non-Newt. Fl. Mech.*, 17:125–144, 1984.

Figure Captions

- Figure 1. Instantaneous configurations of 256 beads enclosed in the primitive cube under shear. The cube is centered at (0,0,0) and has sides of 6.7245σ for atomic, and 6.8399σ for FENE fluids, respectively. The green arrow denotes the direction of flow (x), normal to the photograph, towards the reader, whereas the red arrow points in the direction of the velocity gradient (y). Bead diameter is 0.25 of the collision diameter. a,b) Atomic fluid, typified by liquid argon close to its triple point ($T^\star = 0.722$, $\rho^\star = 0.842$), is sheared at $\dot{\gamma}^\star = 2.25$. The photographs are taken at two different time steps by placing the camera at (80σ,0,0). Note the oscillations between the amorphous and ordered phases during the same run. c,d) FENE fluid at $T^\star = 1$, $\rho^\star = 0.8$, $r^\star_{eq} = 0.9$, and $r^\star_{max} = 1.4$, under shear at $\dot{\gamma}^\star = 10$. The photographs represent two exposures of the same instantaneous configuration taken by a camera located at (80σ,0,0), and at (80σ,-20σ,0). Note the dumbbells with beads belonging to different strings.

- Figure 2. Reduced one dumbbell configuration distribution function in terms of bead-bead separation; $T^\star = 1$, $\rho^\star = 0.8$, and $H^\star = 375$. The dashed lines represent equilibrium distribution obtained by neglecting the inter-molecular interactions. The tails of the distributions are not shown, with the exception of those lying at the extreme left and right in the figure to preserve clarity. i) $r^\star_{eq} = 0.5$, $r^\star_{max} = 0.9$, ii) $r^\star_{eq} = 0.9$, $r^\star_{max} = 1.4$, iii) $r^\star_{eq} = 1.3$, $r^\star_{max} = 1.7$.

- Figure 3. Distribution of dumbbell alignment angles as a function of shear rate; $T^\star = 1$, $\rho^\star = 0.8$, and $H^\star = 375$. a) $r^\star_{eq} = 0.5$, $r^\star_{max} = 0.9$, b) $r^\star_{eq} = 0.9$, $r^\star_{max} = 1.4$, c) $r^\star_{eq} = 1.3$, $r^\star_{max} = 1.7$. Note difference in scales of the ordinates. Distributions for $H^\star = 37.5$ and $H^\star = 3750$ ($r^\star_{eq} = 0.9$,$r^\star_{max} = 1.4$) have similar widths and heights with respect to $\dot{\gamma}^\star$ as in b).

- Figure 4. The correlation between the average end-to-end distance and the alignment angle; $T^\star = 1$, $\rho^\star = 0.8$, and $H^\star = 375$. a) $r^\star_{eq} = 0.5$, $r^\star_{max} = 0.9$, b) $r^\star_{eq} = 0.9$, $r^\star_{max} = 1.4$, c) $r^\star_{eq} = 1.3$, $r^\star_{max} = 1.7$. The scatter at high and low angles is a result of small sample size.

- Figure 5. Distribution of bead peculiar velocities; $T^\star = 1$, $\rho^\star = 0.8$, $H^\star = 375$, $r^\star_{eq} = 0.5$, and $r^\star_{max} = 0.9$. The dashed line corresponds to the equilibrium (Maxwellian) distribution.

- Figure 6. Reduced one dumbbell velocity distribution function in terms of the peculiar center of mass and relative (insets) velocities; $T^\star = 1$, $\rho^\star = 0.8$, and $H^\star = 375$. a) $r^\star_{eq} = 0.5$, $r^\star_{max} = 0.9$, b) $r^\star_{eq} = 0.9$, $r^\star_{max} = 1.4$, c) $r^\star_{eq} = 1.3$, $r^\star_{max} = 1.7$. The dashed lines, are derived on the assumption that bead peculiar velocities are Maxwellian distributed. The equilibrium and the small shear rate results are essentially the same.

- Figure 7. Distribution of dumbbell vibrational and rotational (insets) velocities; $T^\star = 1$, $\rho^\star = 0.8$, and $H^\star = 375$. a) $r^\star_{eq} = 0.5$, $r^\star_{max} = 0.9$, b) $r^\star_{eq} = 0.9$, $r^\star_{max} = 1.4$, c) $r^\star_{eq} = 1.3$, $r^\star_{max} = 1.7$. Distributions at the higher shear rates are significantly different than in the Newtonian limit.

- Figure 8. Pictorial representation of the conformation tensor in terms of orientation ellipses in xy, xz, and yz planes; $T^\star = 1$, $\rho^\star = 0.8$, $H^\star = 375$. a) $r^\star_{eq} = 0.5$, $r^\star_{max} = 0.9$, b) $r^\star_{eq} = 0.9$, $r^\star_{max} = 1.4$, c) $r^\star_{eq} = 1.3$, $r^\star_{max} = 1.7$. The axes are plotted in nondimensional quantities that have physical significance of square of dimensionless distance. In symbols such as xy, the first letter denotes the abscissa, whereas the second signifies the ordinate.

- Figure 9. Dumbbell pair radial distribution function dependence on shear rate; $T^\star = 1$, $\rho^\star = 0.8$, $H^\star = 375$. a) $r^\star_{eq} = 0.5$, $r^\star_{max} = 0.9$, b) $r^\star_{eq} = 0.9$, $r^\star_{max} = 1.4$, c) $r^\star_{eq} = 1.3$, $r^\star_{max} = 1.7$. In case of small dumbbells, the diagonal components of $\underline{\underline{C}}$ are almost indistinguishable, for $\dot{\gamma}^\star < 2.5$. For large dumbbells, the differences among C_{xx}, $C_{yy} \sim C_{zz}$, and g are small but easy to notice, as shown in the inset in c). Normal components of $\underline{\underline{C}}$ for medium dumbbells, are however remarkably different (see Figure 10).

- Figure 10. Normal components of the second conformation tensor for different shear rates; $T^\star = 1$, $\rho^\star = 0.8$, $H^\star = 375$, $r^\star_{eq} = 0.9$, and $r^\star_{max} = 1.4$. a) C_{xx}, b) C_{yy}, c) C_{zz}.

- Figure 11. Variation of the radial elements of the second conformation tensor with shear rate; $T^\star = 1$, $\rho^\star = 0.8$, $H^\star = 375$, $r^\star_{eq} = 0.9$, and $r^\star_{max} = 1.4$. Note different scales of the ordinates.

- Figure 12. Distribution of kinetic energies among translational (a for $H^\star = 37.5$, and b for $H^\star = 3750$), rotational (c), and vibrational (d) reservoirs as a function of strain rate; FENE fluids at $T^\star = 1$, $\rho^\star = 0.8$, $r^\star_{eq} = 0.9$, and $r^\star_{max} = 1.4$.

- Figure 13. Variation of the intra- (a) and inter-particle (b) potential energies with shear rate; FENE fluids at $T^\star = 1$, $\rho^\star = 0.8$, $r^\star_{eq} = 0.9$, and $r^\star_{max} = 1.4$.

Table 1: Nondimensional unit quantities used in this work.

Nondimensional unit quantity	SI conversion factor
mass	6.63×10^{-26} kg
length	3.405×10^{-10} m
time	2.156×10^{-12} s
temperature	119.8 K
shear rate	$4.64 \times 10^{11} s^{-1}$
velocity	157.9 m/s
density	1679.4 kg/m^3
spring constant	8.36×10^{-2} N/m
energy	1.65×10^{-21} J
pressure	41.9×10^{6} Pa
viscosity	9.03×10^{-5} $Pa\ s$

Table 2: Alignment angles and average of the square of the end-to-end distance from the conformation tensor analysis for the FENE fluids at $T^\star = 1$ and $\rho^\star = 0.8$; the numbers in subscript denote one standard deviation. With the exception of the largest dumbbells, χ^{al}_{xy} decreases with the shear rate.

$H^\star$	$r^\star_{eq}$	$r^\star_{max}$	$\dot{\gamma}^\star$	χ^{al}_{xy} [*deg*]	χ^{al}_{xz} [*deg*]	χ^{al}_{yz} [*deg*]	$R^{\star 2}$
37.5	0.9	1.4	4.5	13.0	0.6	88.7	0.878_{18}
37.5	0.9	1.4	8.5	4.3	9.8	86.0	0.756_{12}
37.5	0.9	1.4	15	1.2	-0.7	-74.0	0.7759_{50}
375	0.5	0.9	0.2	33.7	1.4	-86.8	0.2383_{35}
375	0.5	0.9	0.8	4.8	0.0	-89.8	0.2267_{30}
375	0.5	0.9	10	0.6	-1.4	90.0	0.24746_{41}
375	0.9	1.4	0.17	40.4	-5.9	1.6	0.80365_{31}
375	0.9	1.4	2.6	17.1	0.3	89.6	0.80988_{56}
375	0.9	1.4	8	2.9	-5.5	-86.0	0.8044_{16}
375	0.9	1.4	14	1.1	0.6	82.7	0.81085_{71}
375	1.3	1.7	0.1	46.6	50.0	-26.3	1.6915_{55}
375	1.3	1.7	5	-88.0	29.9	-2.1	1.726_{54}
375	1.3	1.7	10	3.1	0.1	88.3	1.683_{61}
3750	0.9	1.4	0.1	39.5	-33.0	-88.3	0.809247_{20}
3750	0.9	1.4	1	25.9	-1.7	87.0	0.809358_{38}
3750	0.9	1.4	3.5	12.6	-0.2	89.9	0.80963_{12}

Table 3: Check of consistency in the velocity data. The rotational and the vibrational energies were calculated either by integration over the velocity distributions or directly in the simulations; the numbers in subscript correspond to one standard deviation.

$H^\star$	$\dot{\gamma}^\star$	$K^\star_{vib}$	$K^\star_{rot}$	$K^\star_{vib}$	$K^\star_{rot}$
		(integrated)	(integrated)		
37.5	4.5	67.70	133.13	67.90_{94}	132.18_{98}
37.5	8.5	64.2	133.7	$63.7_{1.1}$	$135.3_{1.6}$
37.5	15	68.1	122.6	$73.6_{1.4}$	$121.7_{1.8}$
375	0.17	63.21	127.97	63.64_{39}	128.16_{56}
375	2.6	47.19	138.81	47.16_{49}	138.86_{53}
375	8	62.09	135.9	61.60_{82}	$137.5_{1.5}$
375	14	75.0	122.1	$75.1_{1.4}$	$125.1_{2.9}$
3750	0.1	53.2	132.1	$54.2_{3.0}$	$132.0_{1.4}$
3750	1	10.44	150.31	10.93_{53}	150.46_{59}
3750	3.5	23.68	150.7	24.11_{54}	$150.7_{1.0}$

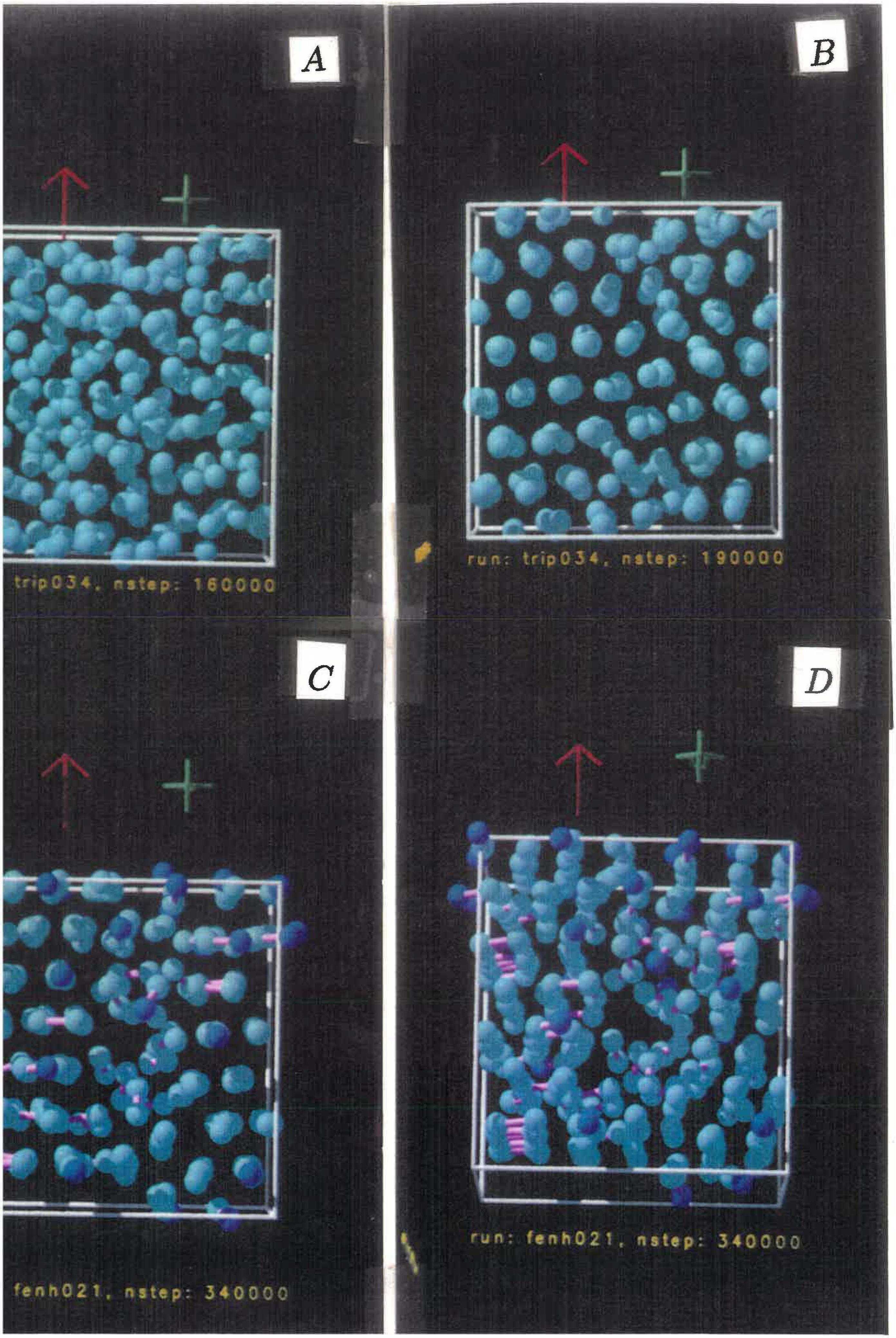
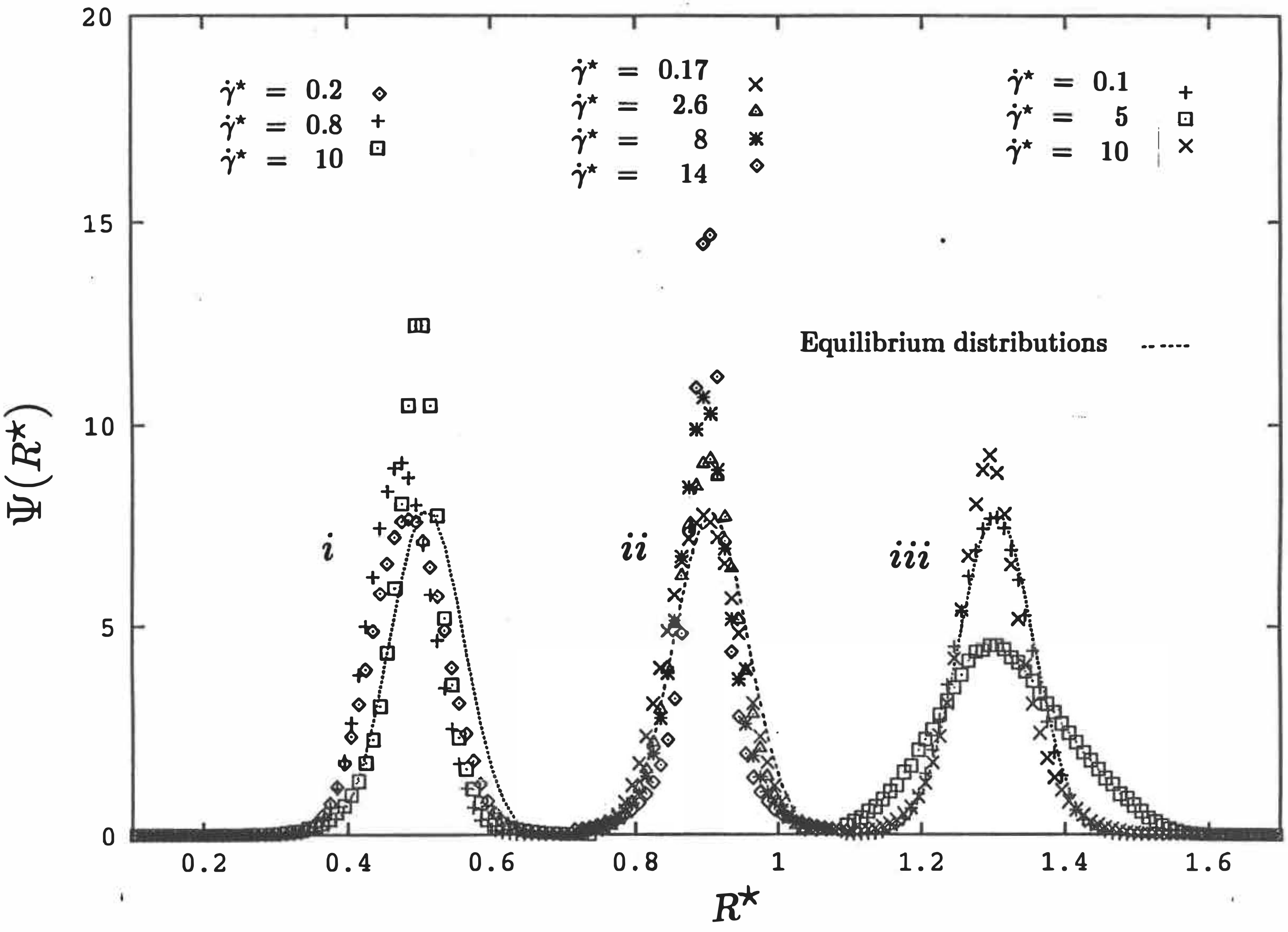

A
trip034, nstep: 160000
B
run: trip034, nstep: 190000
C
fenh021, nstep: 340000
D
run: fenh021, nstep: 340000

Figure 2



$\dot{\gamma}^\star = 0.2$ ◇
$\dot{\gamma}^\star = 0.8$ +
$\dot{\gamma}^\star = 10$ ⊡
$\dot{\gamma}^\star = 0.17$ ×
$\dot{\gamma}^\star = 2.6$ △
$\dot{\gamma}^\star = 8$ ✱
$\dot{\gamma}^\star = 14$ ◇
$\dot{\gamma}^\star = 0.1$ +
$\dot{\gamma}^\star = 5$ ⊡
$\dot{\gamma}^\star = 10$ ×
Equilibrium distributions
i
ii
iii
$\Psi(R^\star)$
20
15
10
5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
$R^\star$

Figure 3a

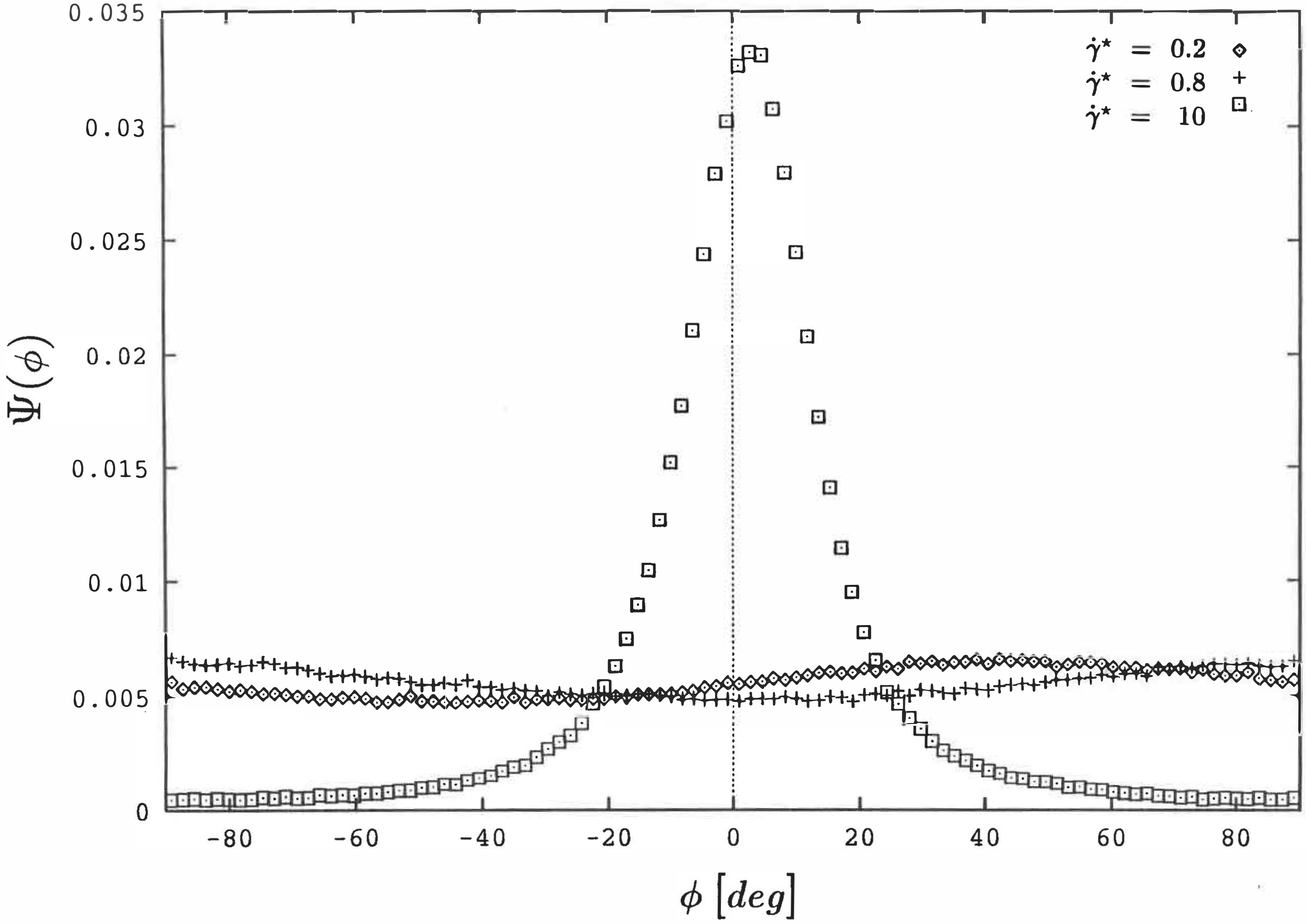
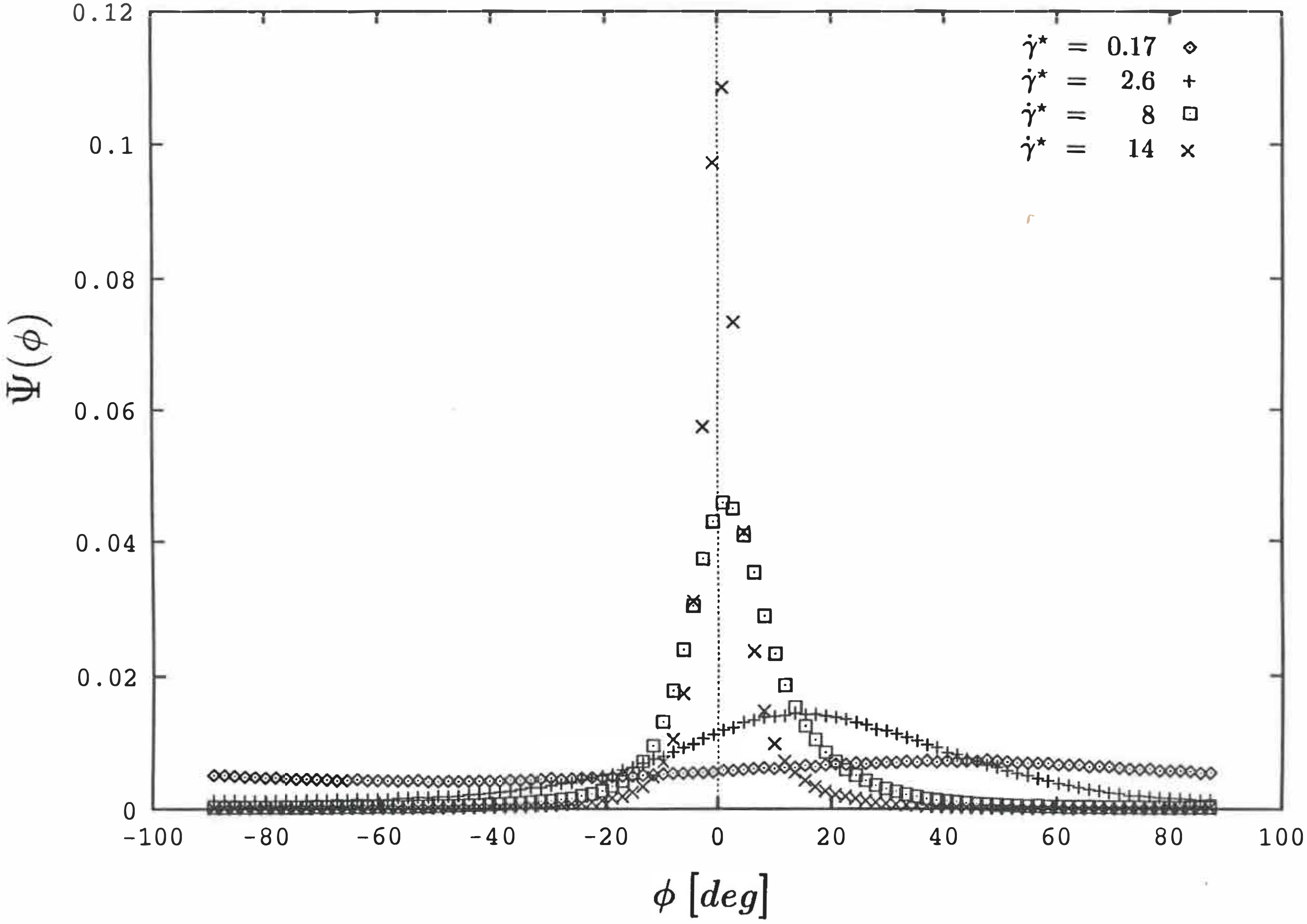
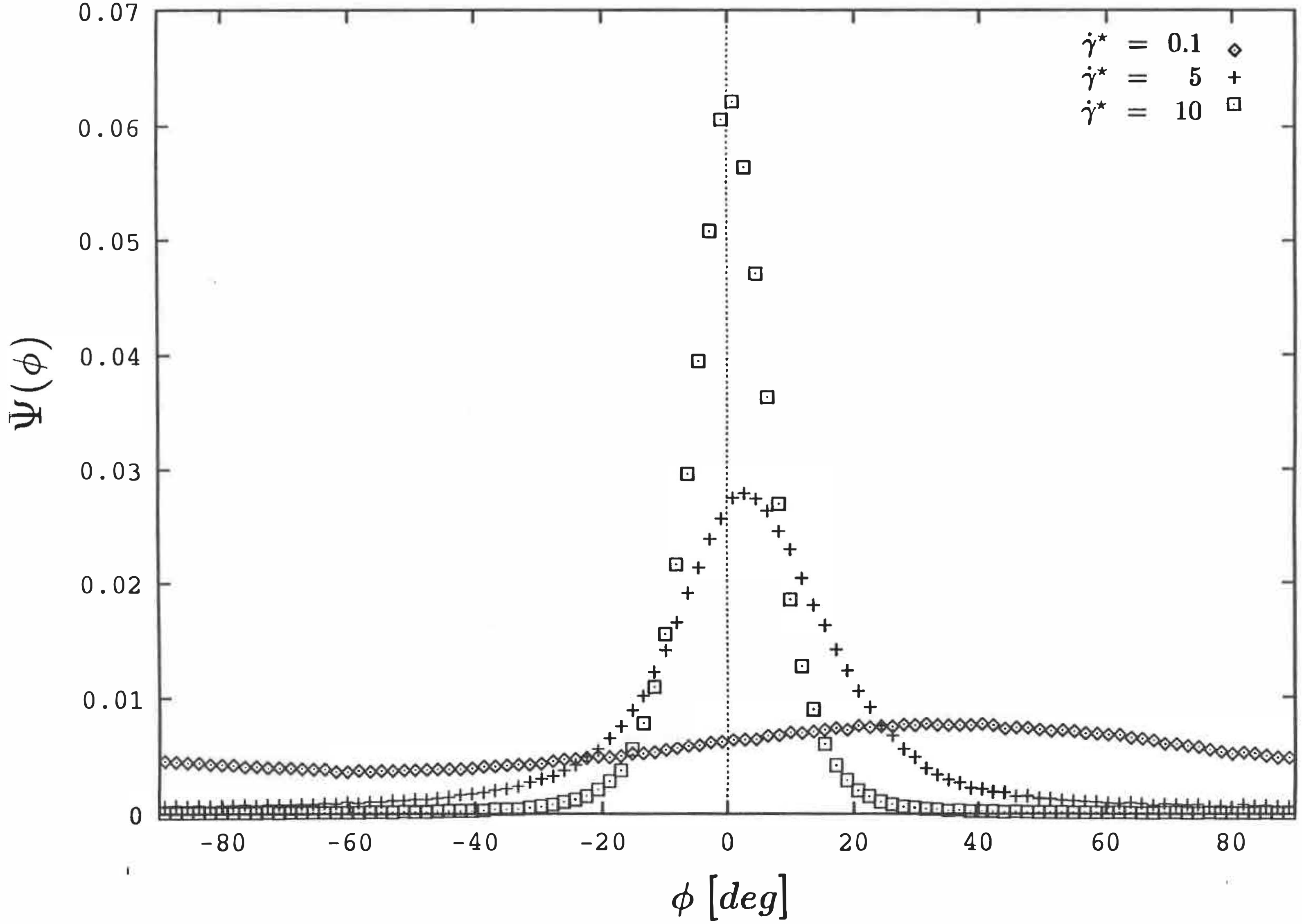

0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
$\Psi(\phi)$
-80
-60
-40
-20
0
20
40
60
80
$\phi\ [deg]$
$\dot{\gamma}^* = 0.2$ ◇
$\dot{\gamma}^* = 0.8$ +
$\dot{\gamma}^* = 10$ □

Figure 3b



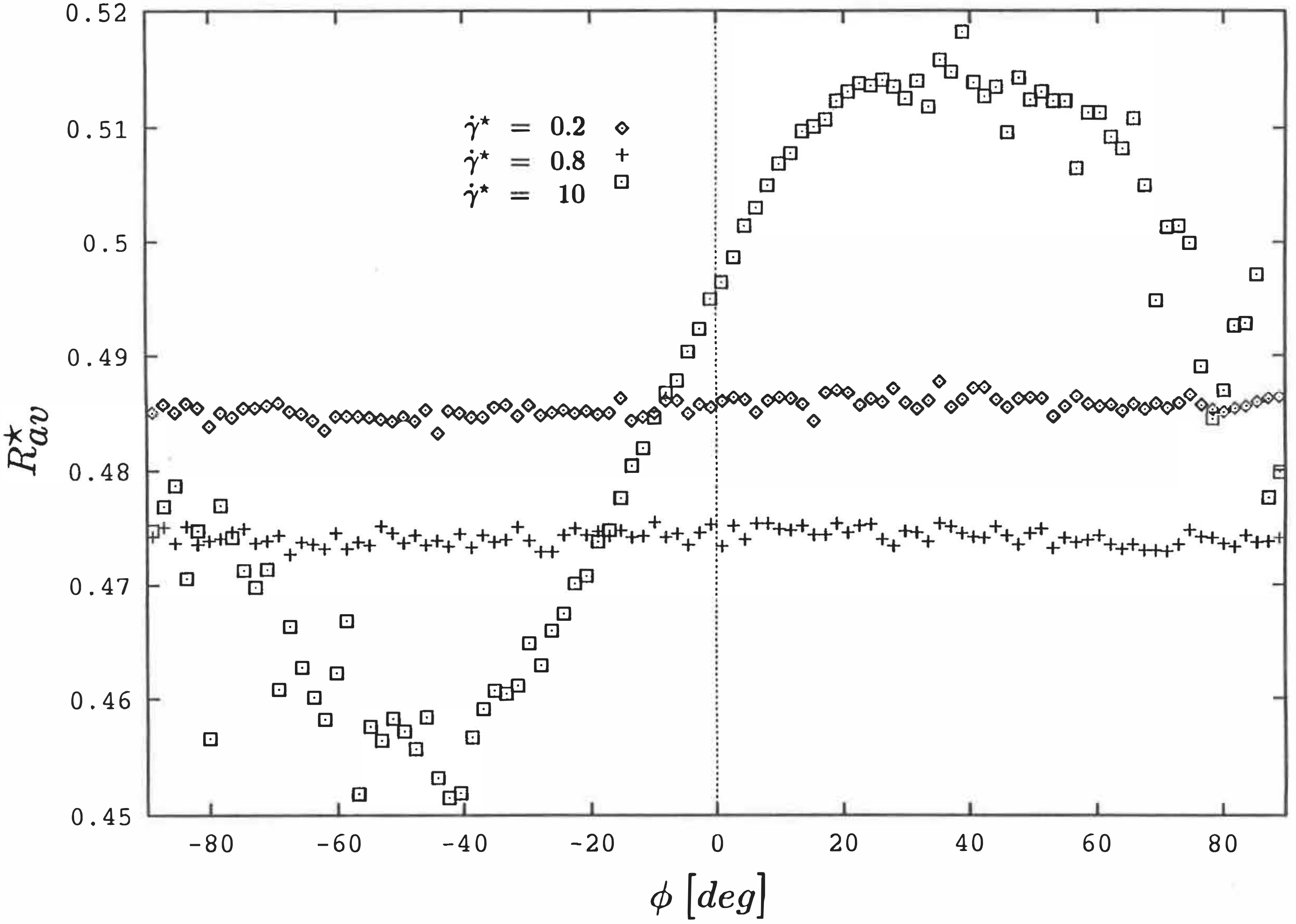
0.12
0.1
0.08
0.06
0.04
0.02
0
-100
-80
-60
-40
-20
0
20
40
60
80
100
$\Psi(\phi)$
ϕ [deg]
$\dot{\gamma}^* = 0.17$ ◇
$\dot{\gamma}^* = 2.6$ +
$\dot{\gamma}^* = 8$ ⊡
$\dot{\gamma}^* = 14$ ×

Figure 3c



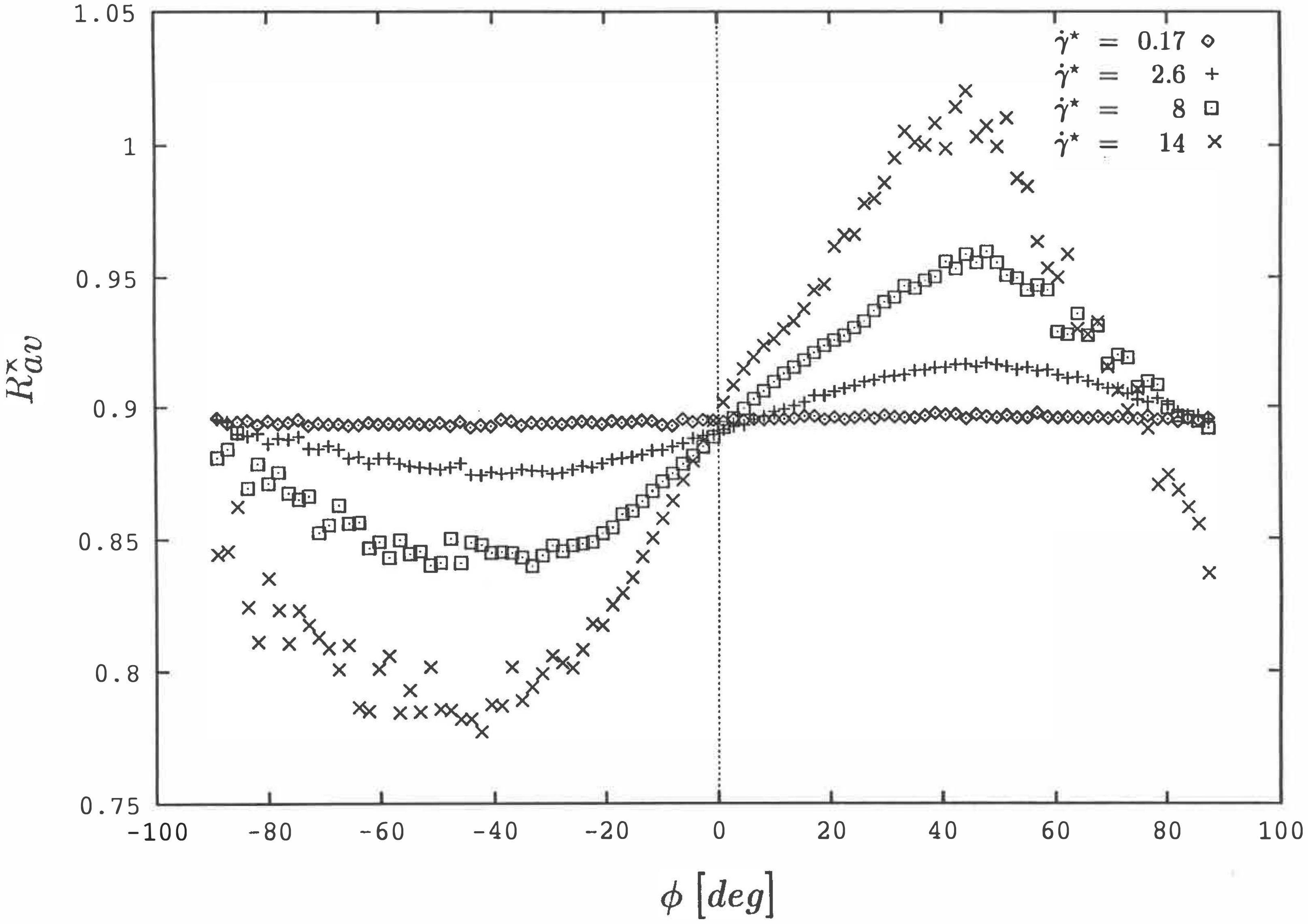
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
-80
-60
-40
-20
0
20
40
60
80
$\Psi(\phi)$
$\phi\ [deg]$
$\dot{\gamma}^{\star}\ =\ 0.1$ ◇
$\dot{\gamma}^{\star}\ =\ 5$ +
$\dot{\gamma}^{\star}\ =\ 10$ ⊡

Figure 4a



$\dot{\gamma}^{\star}$ = 0.2 ◇
$\dot{\gamma}^{\star}$ = 0.8 +
$\dot{\gamma}^{\star}$ = 10 ⊡
$R^{\star}_{av}$
0.52
0.51
0.5
0.49
0.48
0.47
0.46
0.45
-80
-60
-40
-20
0
20
40
60
80
ϕ [deg]

Figure 4b.



$R^{\star}_{av}$
1.05
1
0.95
0.9
0.85
0.8
0.75
-100
-80
-60
-40
-20
0
20
40
60
80
100
ϕ [deg]
$\dot{\gamma}^{\star}$ = 0.17 ◇
$\dot{\gamma}^{\star}$ = 2.6 +
$\dot{\gamma}^{\star}$ = 8 ⊡
$\dot{\gamma}^{\star}$ = 14 ×

Figure 4c

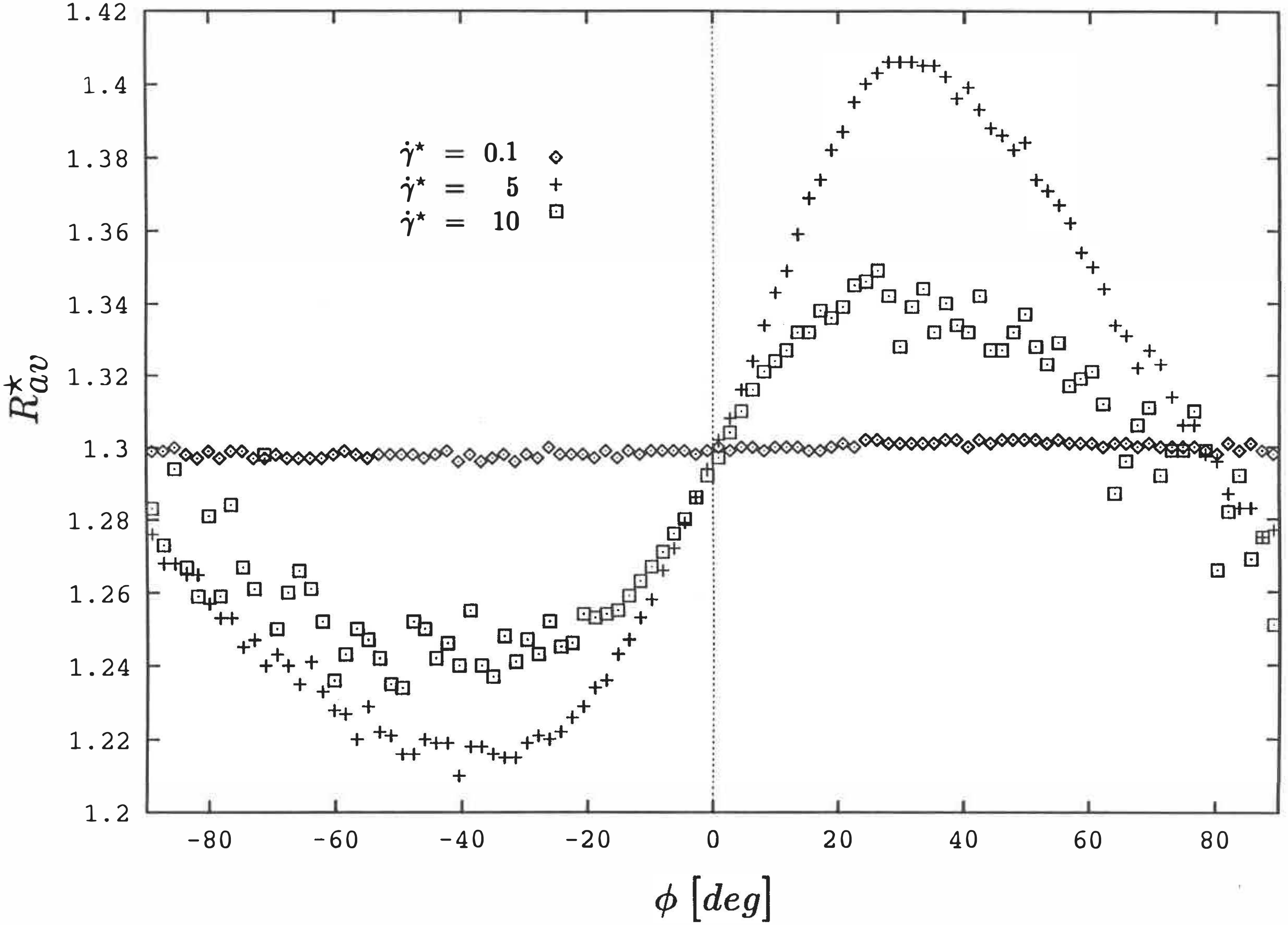
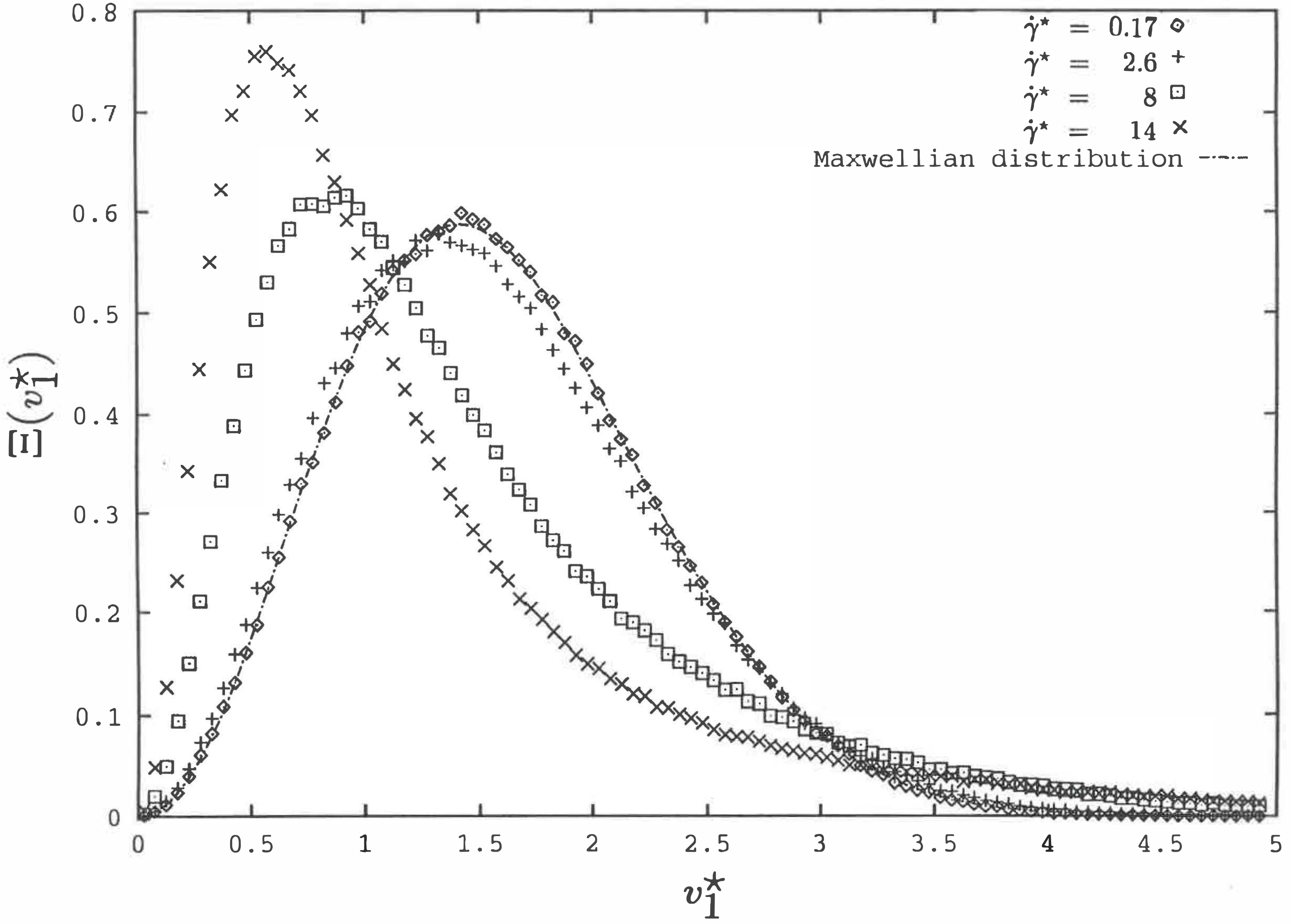

$R^{\star}_{av}$
1.42
1.4
1.38
1.36
1.34
1.32
1.3
1.28
1.26
1.24
1.22
1.2
-80
-60
-40
-20
0
20
40
60
80
ϕ [deg]
$\dot{\gamma}^{\star}$ = 0.1 ◇
$\dot{\gamma}^{\star}$ = 5 +
$\dot{\gamma}^{\star}$ = 10 ⊡

Figure 5



$\dot{\gamma}^\star = 0.17$ ◇
$\dot{\gamma}^\star = 2.6$ +
$\dot{\gamma}^\star = 8$ ⊡
$\dot{\gamma}^\star = 14$ ×
Maxwellian distribution -·-·-
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
$\Xi(v_1^\star)$
$v_1^\star$

Figure 6a

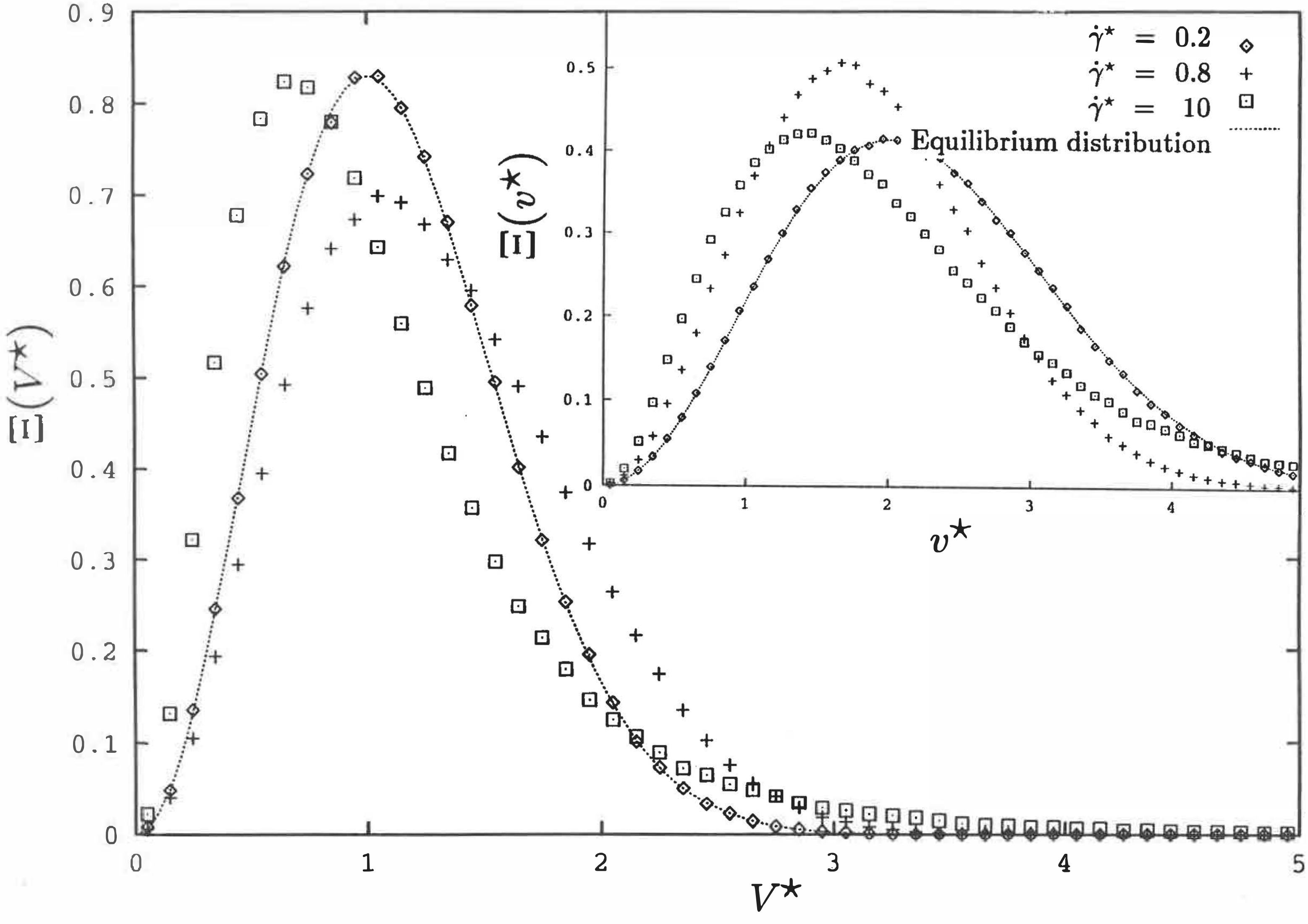

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
$\Xi(V^\star)$
0
1
2
3
4
5
$V^\star$
0.5
0.4
0.3
0.2
0.1
0
$\Xi(v^\star)$
0
1
2
3
4
$v^\star$
$\dot{\gamma}^\star = 0.2$ ◇
$\dot{\gamma}^\star = 0.8$ +
$\dot{\gamma}^\star = 10$ ⊡
Equilibrium distribution

Figure 6b.

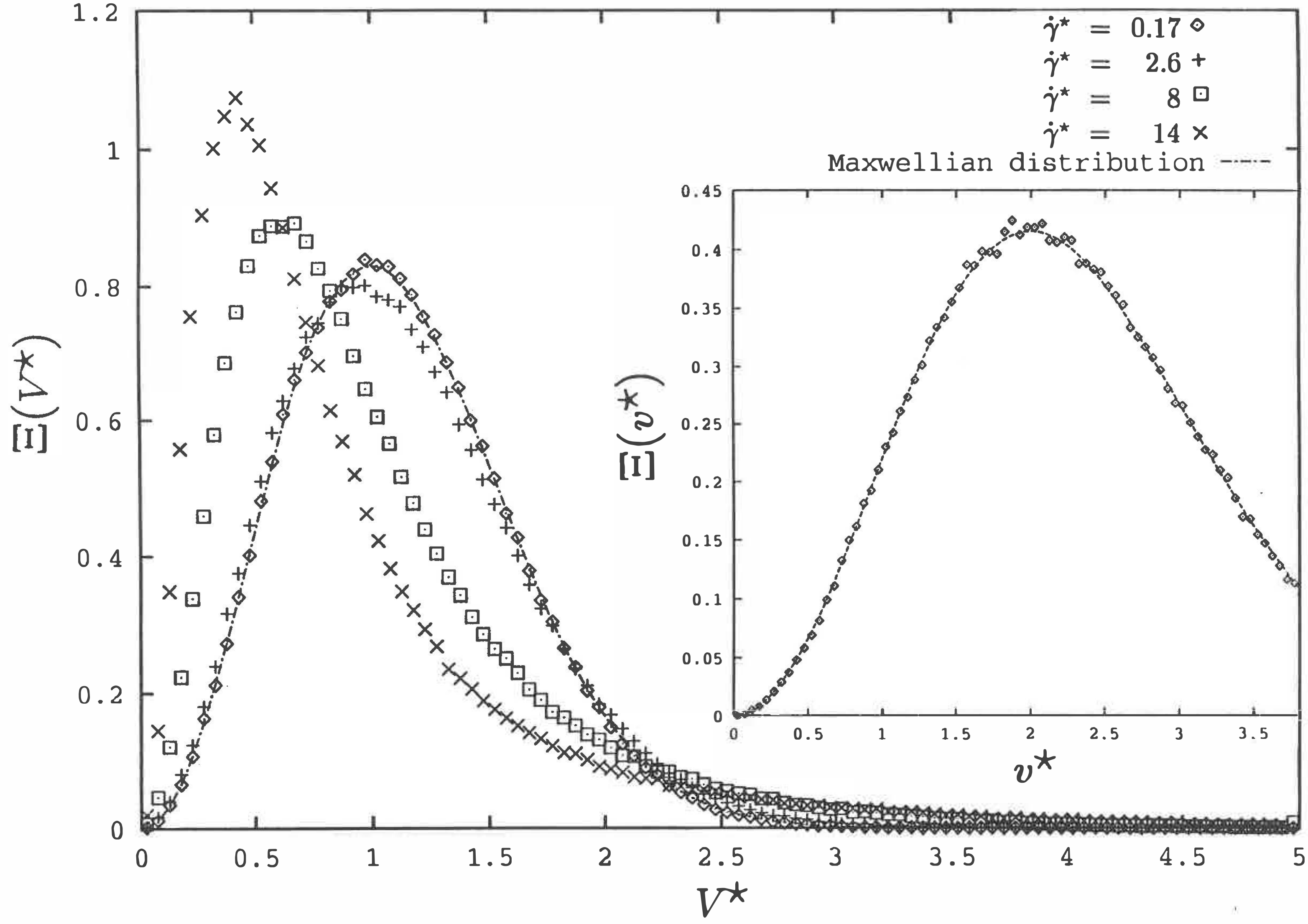
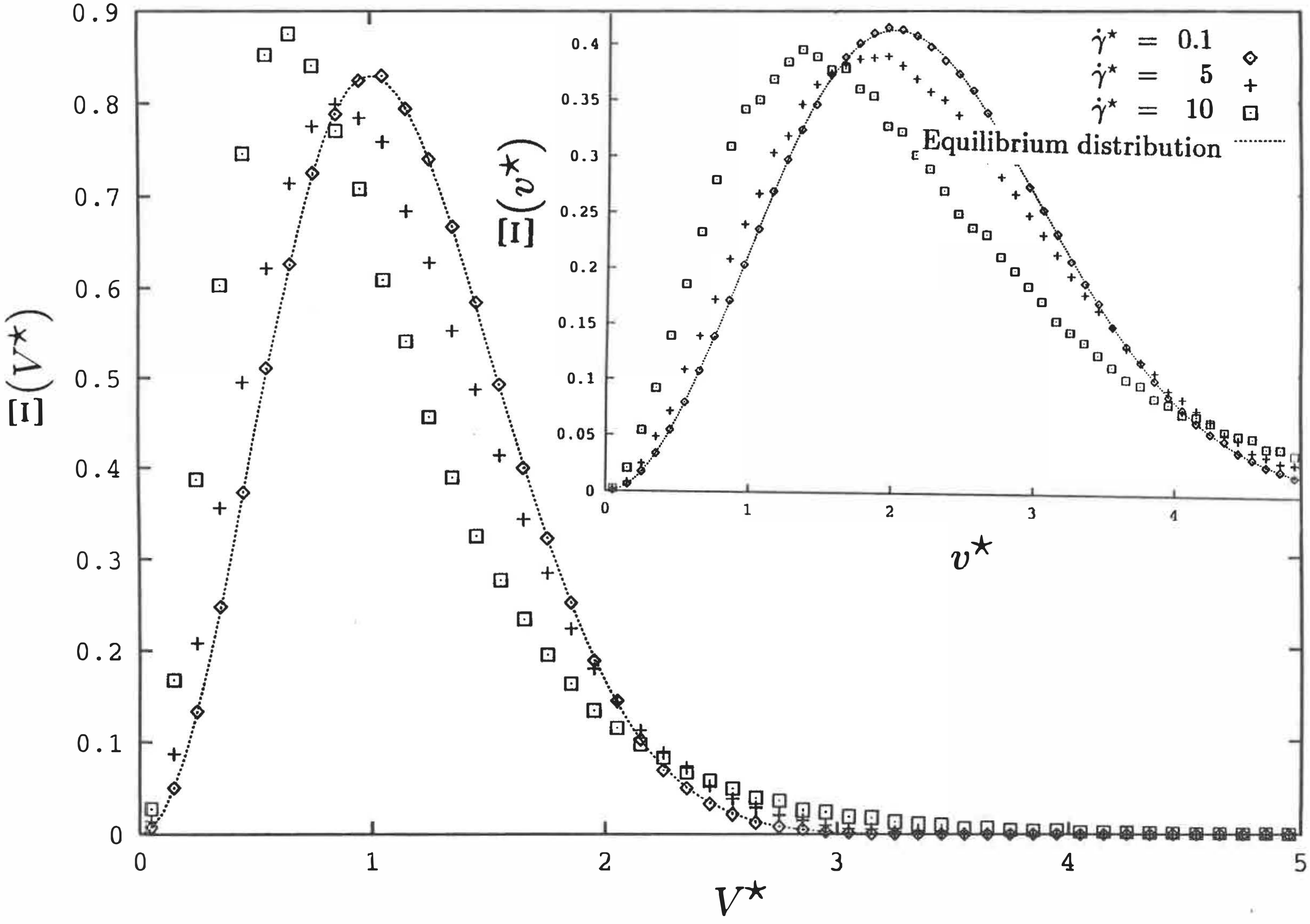

$\dot{\gamma}^* = 0.17$ ◇
$\dot{\gamma}^* = 2.6$ +
$\dot{\gamma}^* = 8$ ⊡
$\dot{\gamma}^* = 14$ ×
Maxwellian distribution
$\Xi(V^\star)$
$V^\star$
1.2
1
0.8
0.6
0.4
0.2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
$\Xi(v^\star)$
$v^\star$
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.5
1
1.5
2
2.5
3
3.5

Figure 6c



0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
$V^\star$
$\Xi(V^\star)$
$\dot{\gamma}^\star = 0.1$ ◇
$\dot{\gamma}^\star = 5$ +
$\dot{\gamma}^\star = 10$ ⊡
Equilibrium distribution
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1
2
3
4
$v^\star$
$\Xi(v^\star)$

Figure 7a

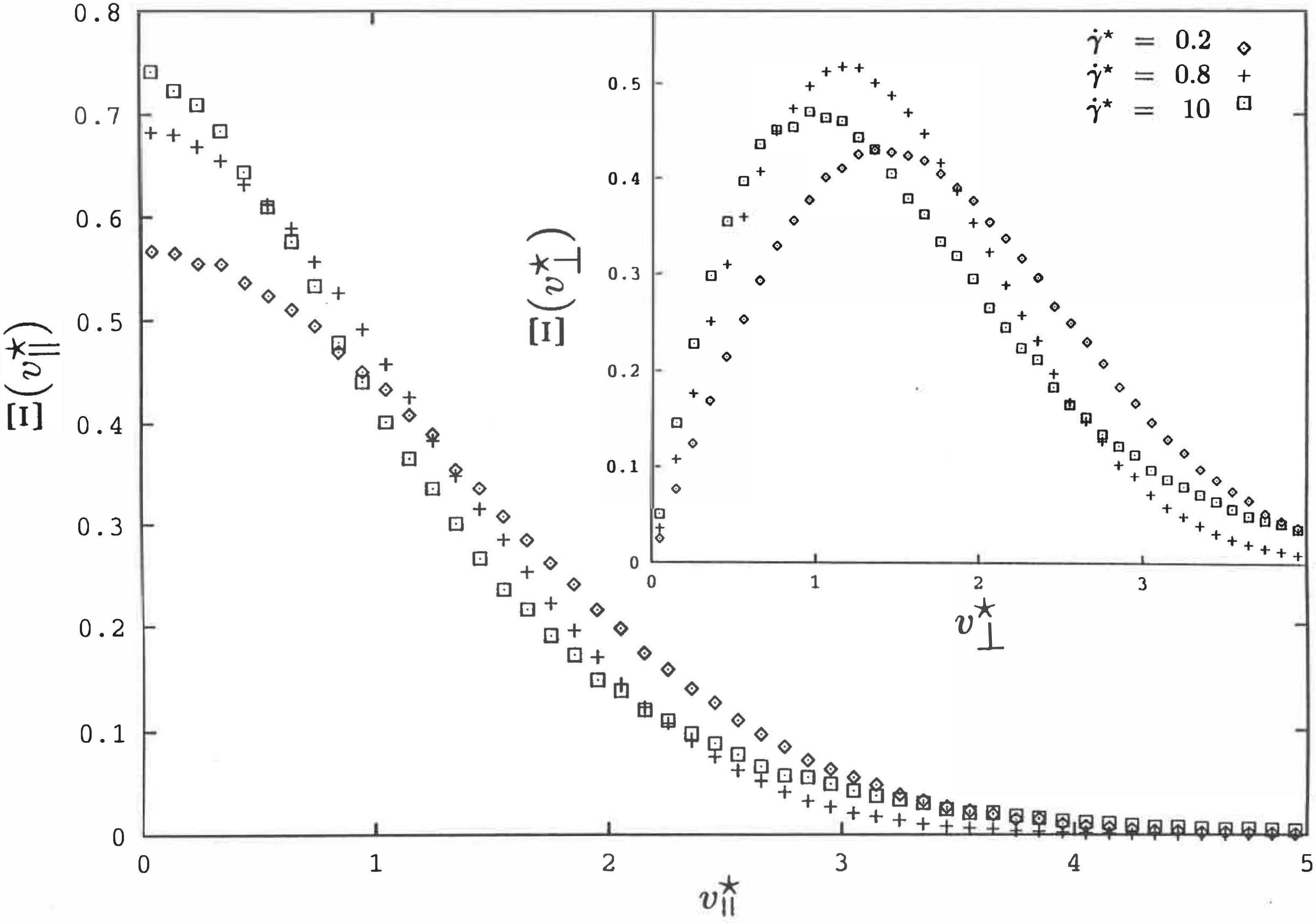
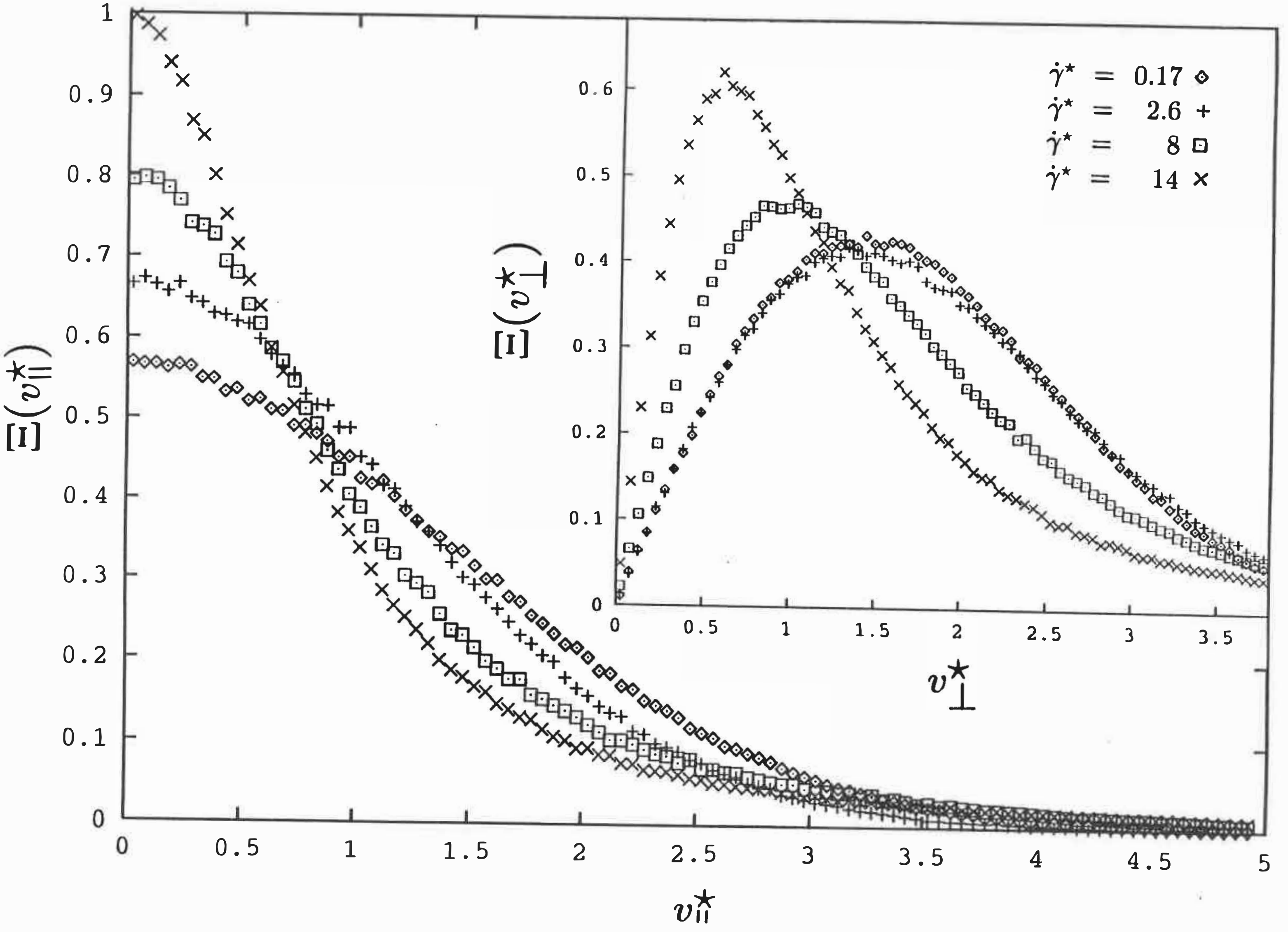
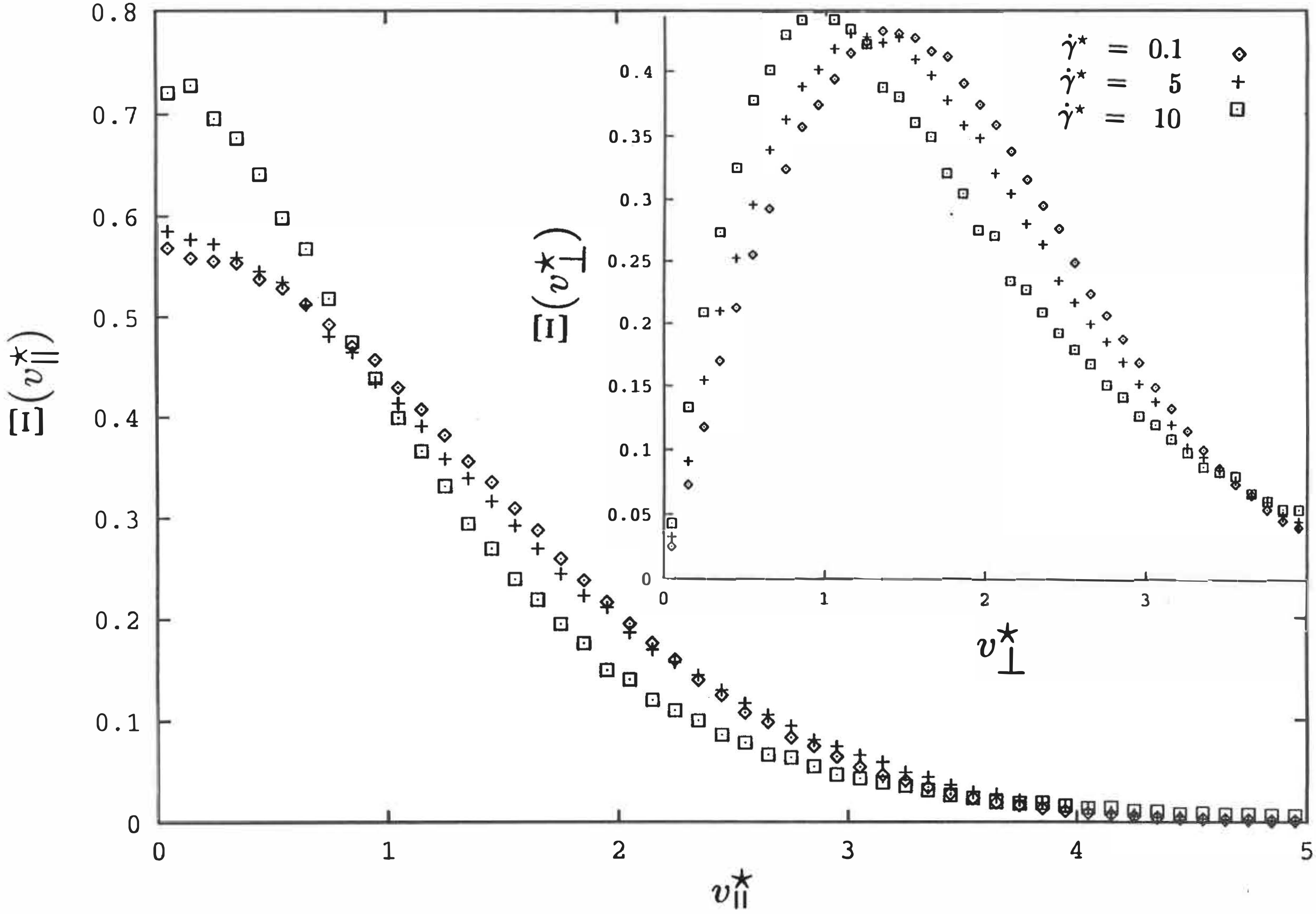

0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
$\Xi(v_{\|}^{\star})$
0
1
2
3
4
5
$v_{\|}^{\star}$
0.5
0.4
0.3
0.2
0.1
0
$\Xi(v_{\perp}^{\star})$
0
1
2
3
$v_{\perp}^{\star}$
$\dot{\gamma}^{\star} = 0.2$ ◇
$\dot{\gamma}^{\star} = 0.8$ +
$\dot{\gamma}^{\star} = 10$ ⊡

Figure 7b



$\Xi(v_{\parallel}^{\star})$
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
$v_{\parallel}^{\star}$
$\Xi(v_{\perp}^{\star})$
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3
3.5
$v_{\perp}^{\star}$
$\dot{\gamma}^{\star} = 0.17$ ◇
$\dot{\gamma}^{\star} = 2.6$ +
$\dot{\gamma}^{\star} = 8$ ⊡
$\dot{\gamma}^{\star} = 14$ ×

Figure 7c



0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
$v^\star_{\|}$
$[\mathrm{I}](v^\star_{\|})$
$\dot{\gamma}^\star = 0.1$ ◇
$\dot{\gamma}^\star = 5$ +
$\dot{\gamma}^\star = 10$ ⊡
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1
2
3
$v^\star_{\perp}$
$[\mathrm{I}](v^\star_{\perp})$

Figure 8a

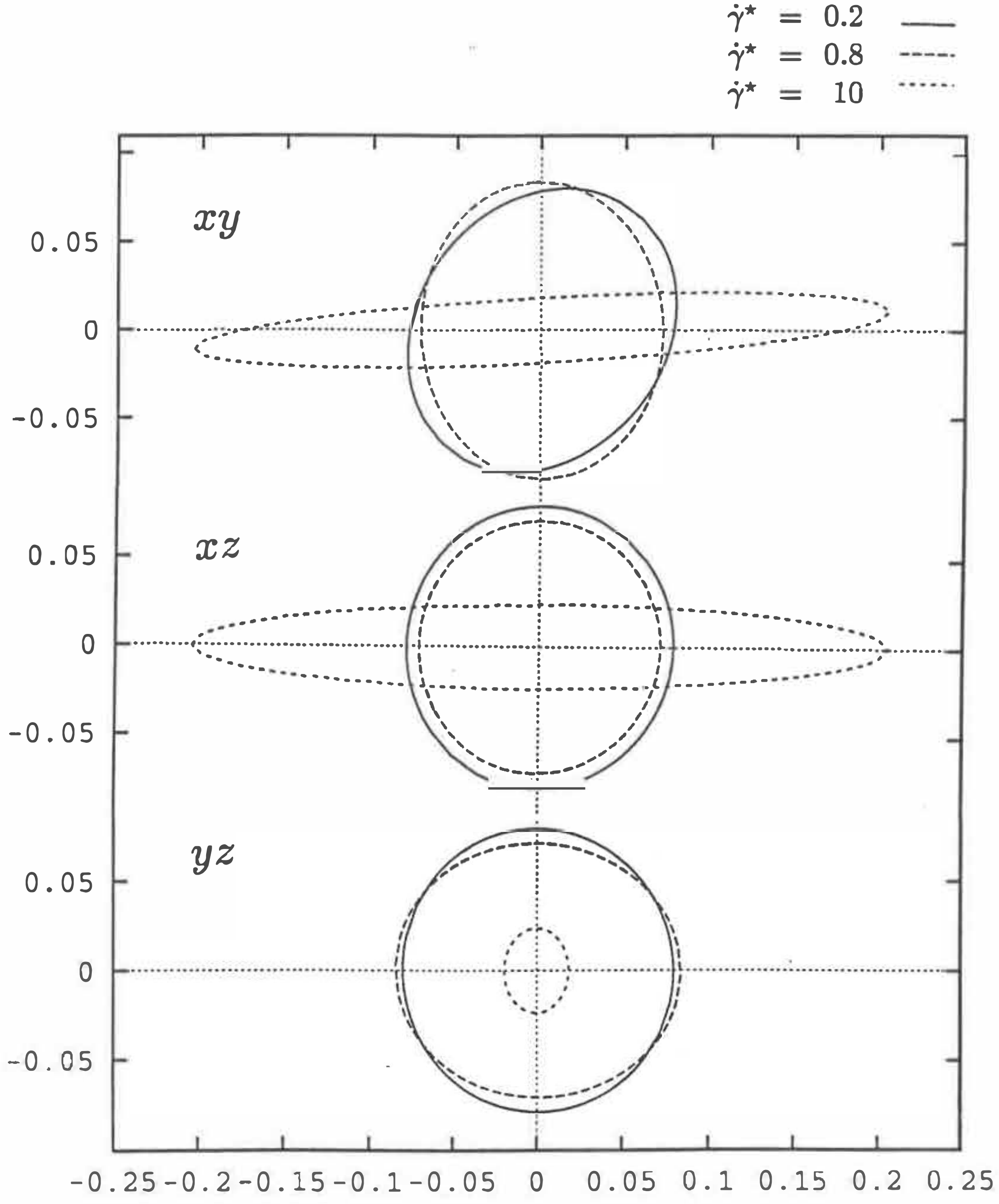

$\dot{\gamma}^* = 0.2$
$\dot{\gamma}^* = 0.8$
$\dot{\gamma}^* = 10$
xy
xz
yz
0.05
0
-0.05
0.05
0
-0.05
0.05
0
-0.05
-0.25
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2
0.25

Figure 8b

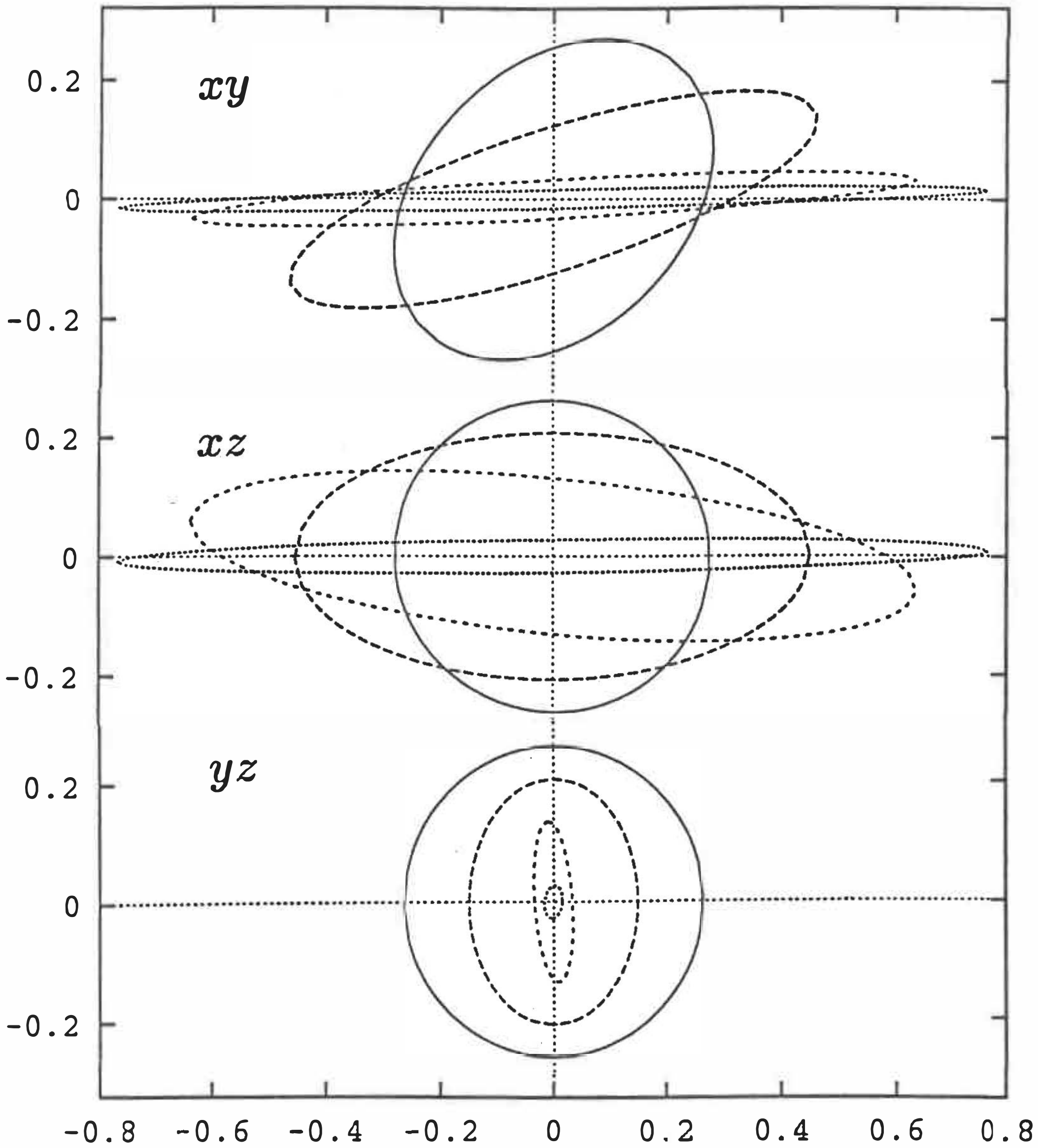

$\dot{\gamma}^{*}$ = 0.17
$\dot{\gamma}^{*}$ = 2.6
$\dot{\gamma}^{*}$ = 8
$\dot{\gamma}^{*}$ = 14
xy
xz
yz
0.2
0
-0.2
0.2
0
-0.2
0.2
0
-0.2
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8

Figure 8c

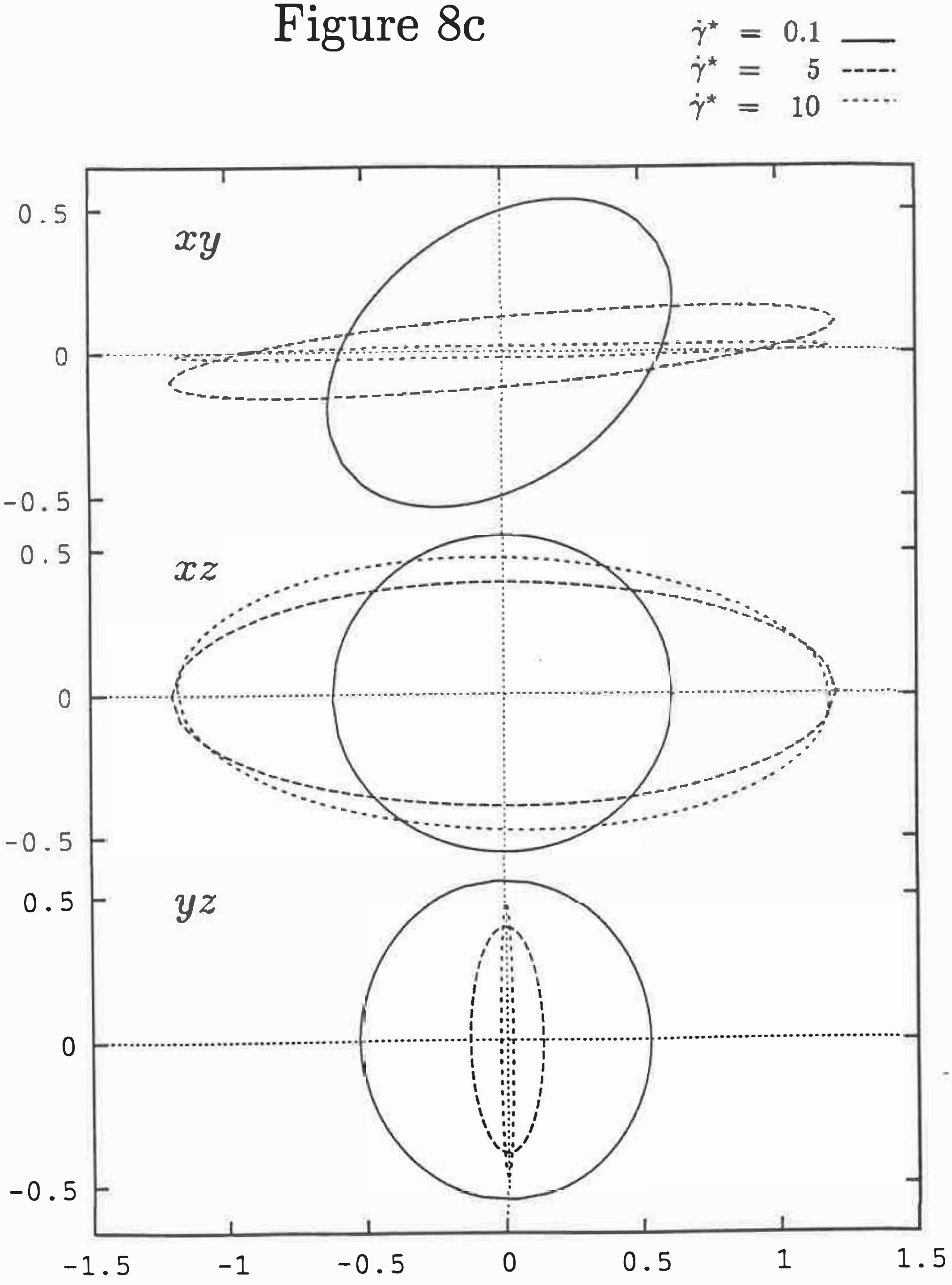
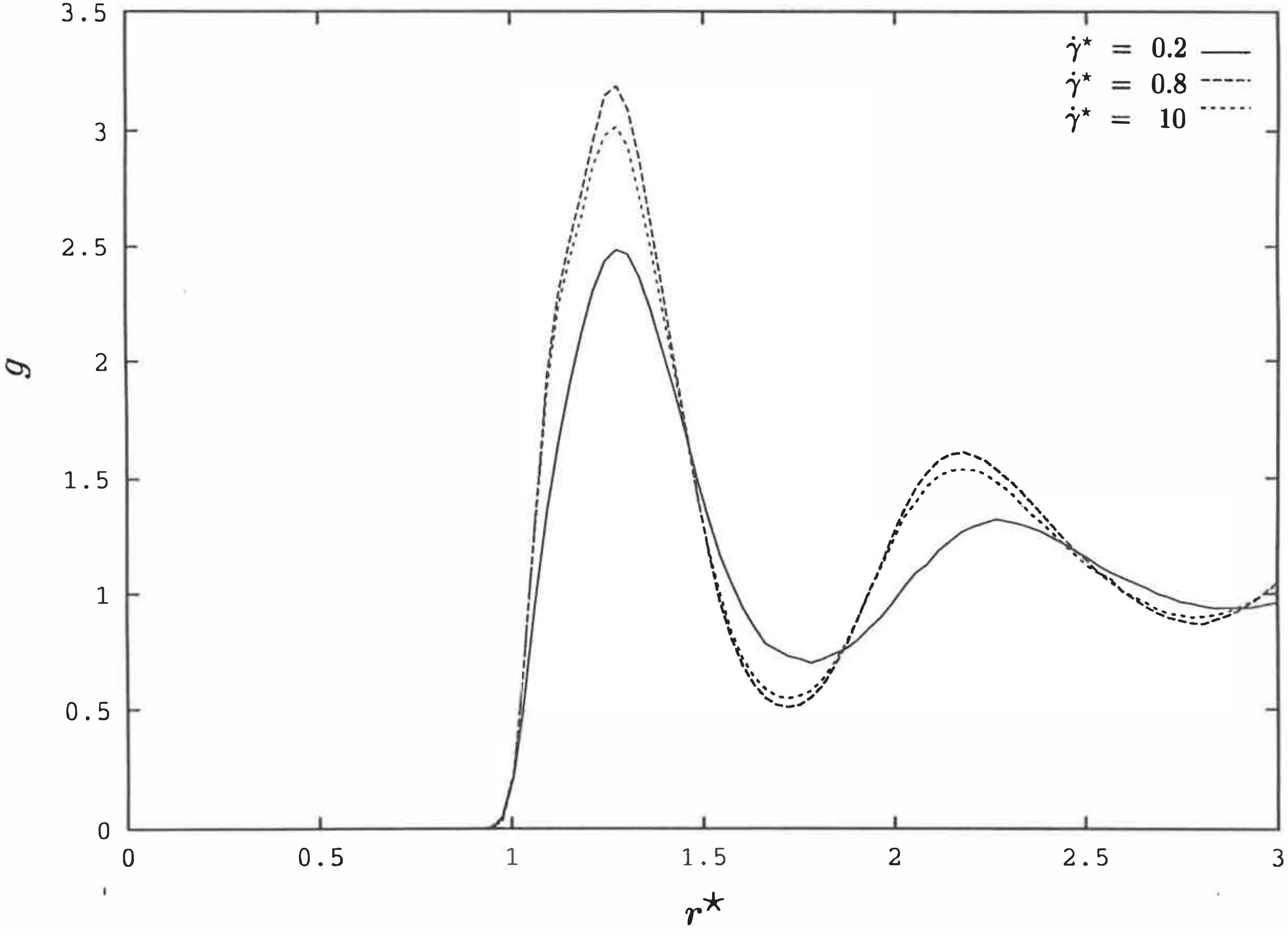

$\dot{\gamma}^* = 0.1$
$\dot{\gamma}^* = 5$
$\dot{\gamma}^* = 10$
xy
xz
yz
0.5
0
-0.5
0.5
0
-0.5
0.5
0
-0.5
-1.5
-1
-0.5
0
0.5
1
1.5

Figure 9a



3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
g
$r^\star$
$\dot\gamma^\star = 0.2$
$\dot\gamma^\star = 0.8$
$\dot\gamma^\star = 10$

Figure 9b

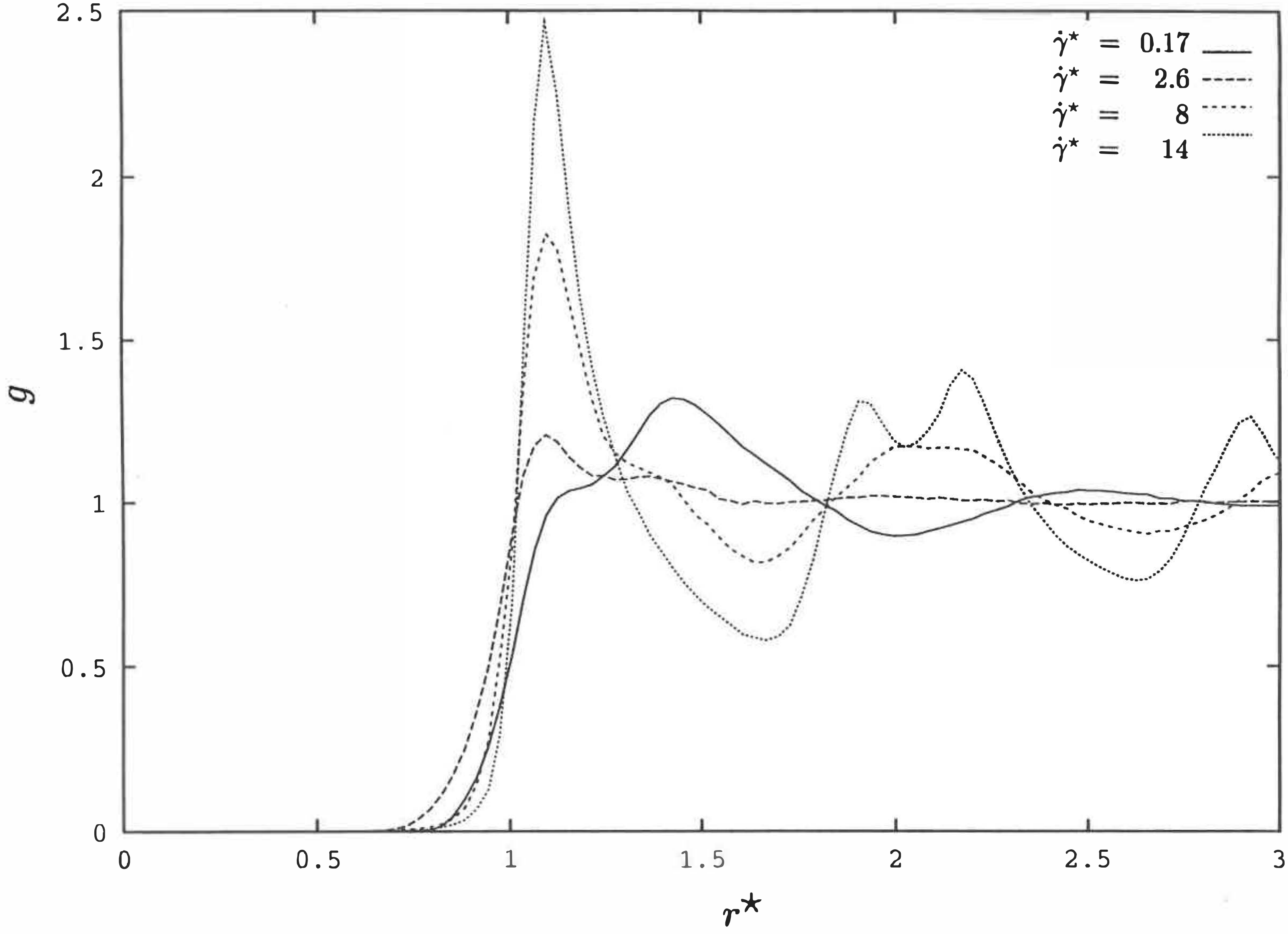

$\dot{\gamma}^\star = 0.17$
$\dot{\gamma}^\star = 2.6$
$\dot{\gamma}^\star = 8$
$\dot{\gamma}^\star = 14$
g
$r^\star$
0
0.5
1
1.5
2
2.5
3

Figure 9c

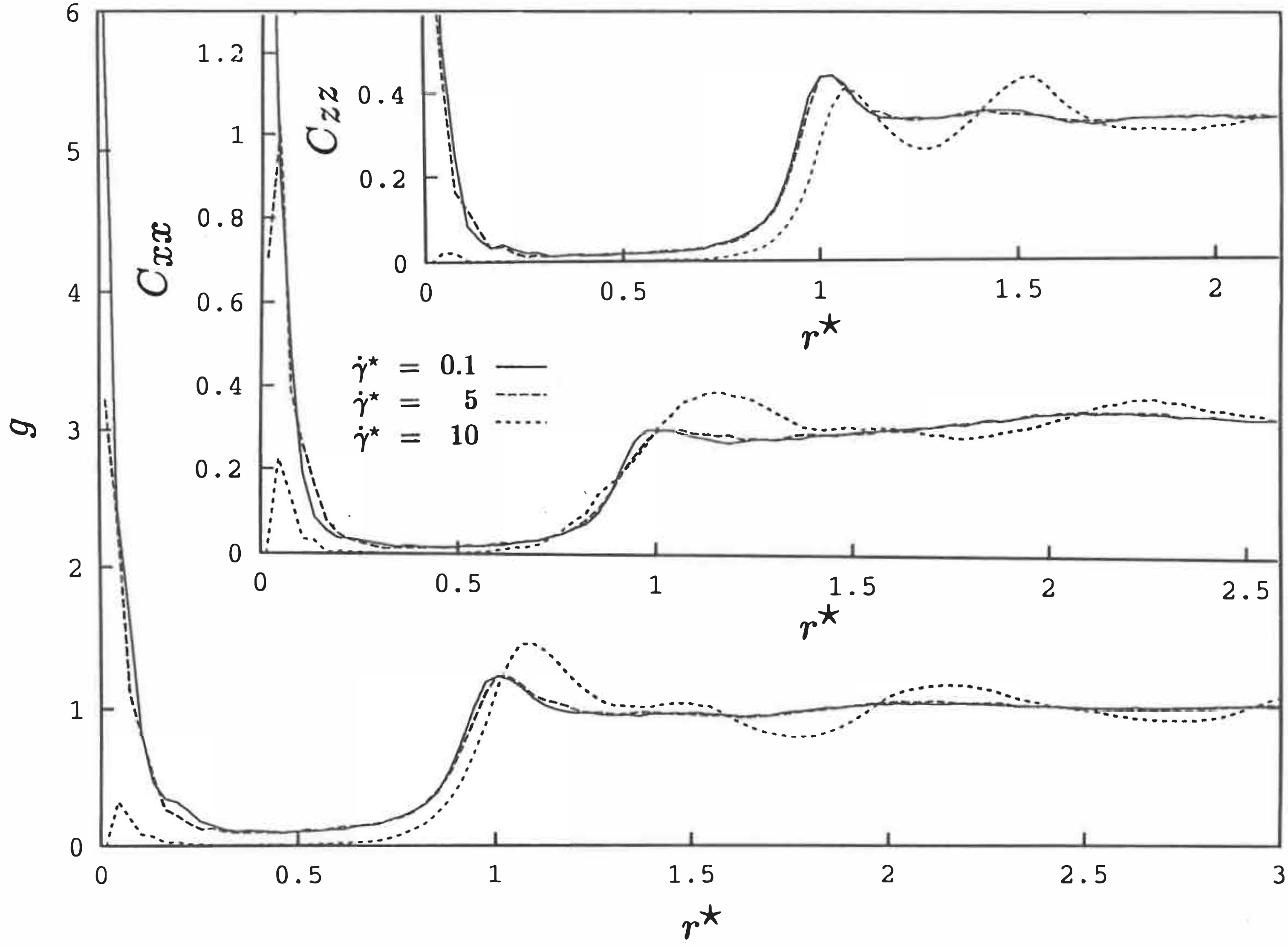
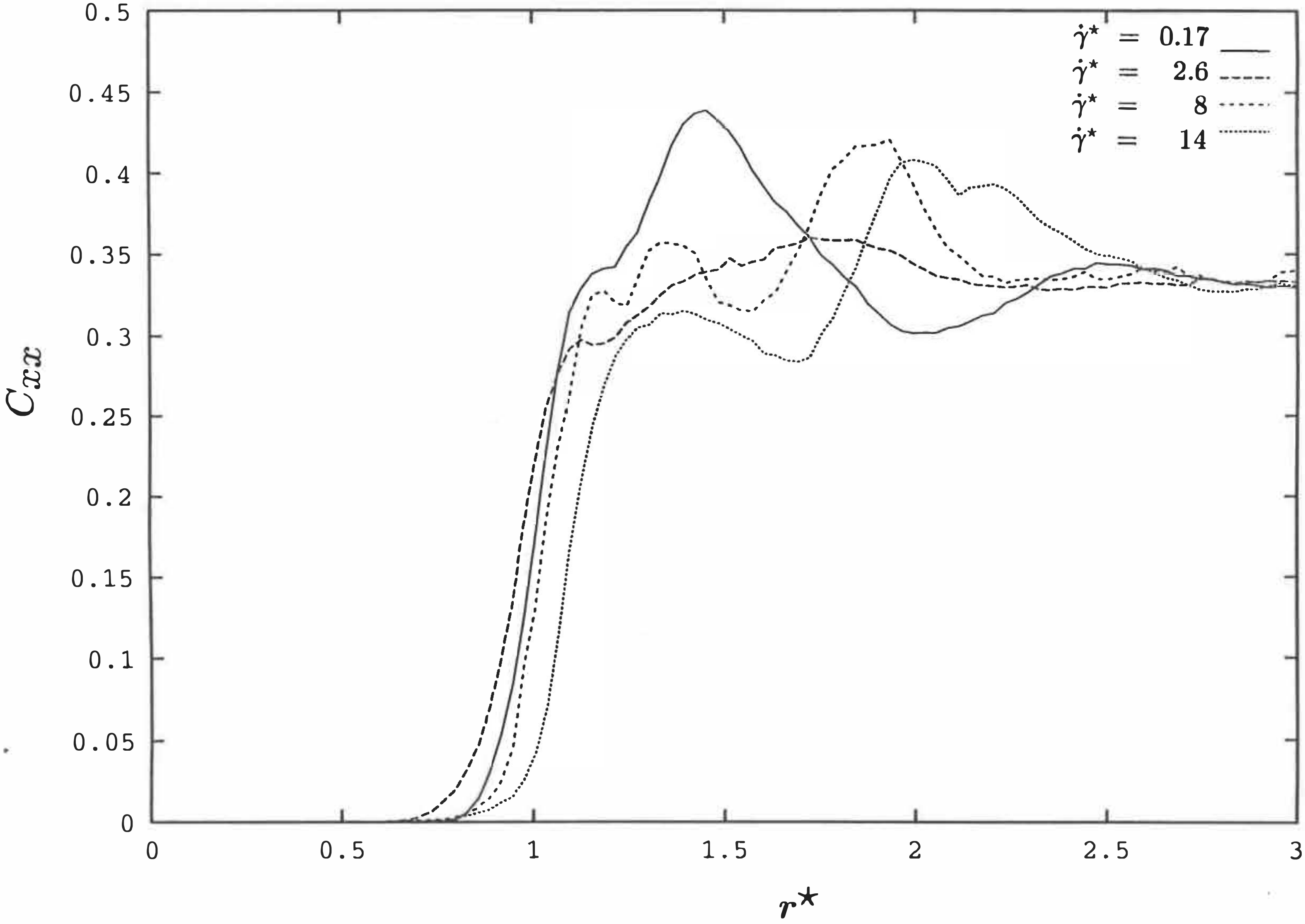

g
6
5
4
3
2
1
0
0
0.5
1
1.5
2
2.5
3
$r^\star$
C_{xx}
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
$r^\star$
C_{zz}
0.4
0.2
0
0
0.5
1
1.5
2
$r^\star$
$\dot{\gamma}^\star$ = 0.1
$\dot{\gamma}^\star$ = 5
$\dot{\gamma}^\star$ = 10

Figure 10a



$\dot{\gamma}^\star$ = 0.17
$\dot{\gamma}^\star$ = 2.6
$\dot{\gamma}^\star$ = 8
$\dot{\gamma}^\star$ = 14
C_{xx}
$r^\star$
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.5
1
1.5
2
2.5
3

Figure 10b

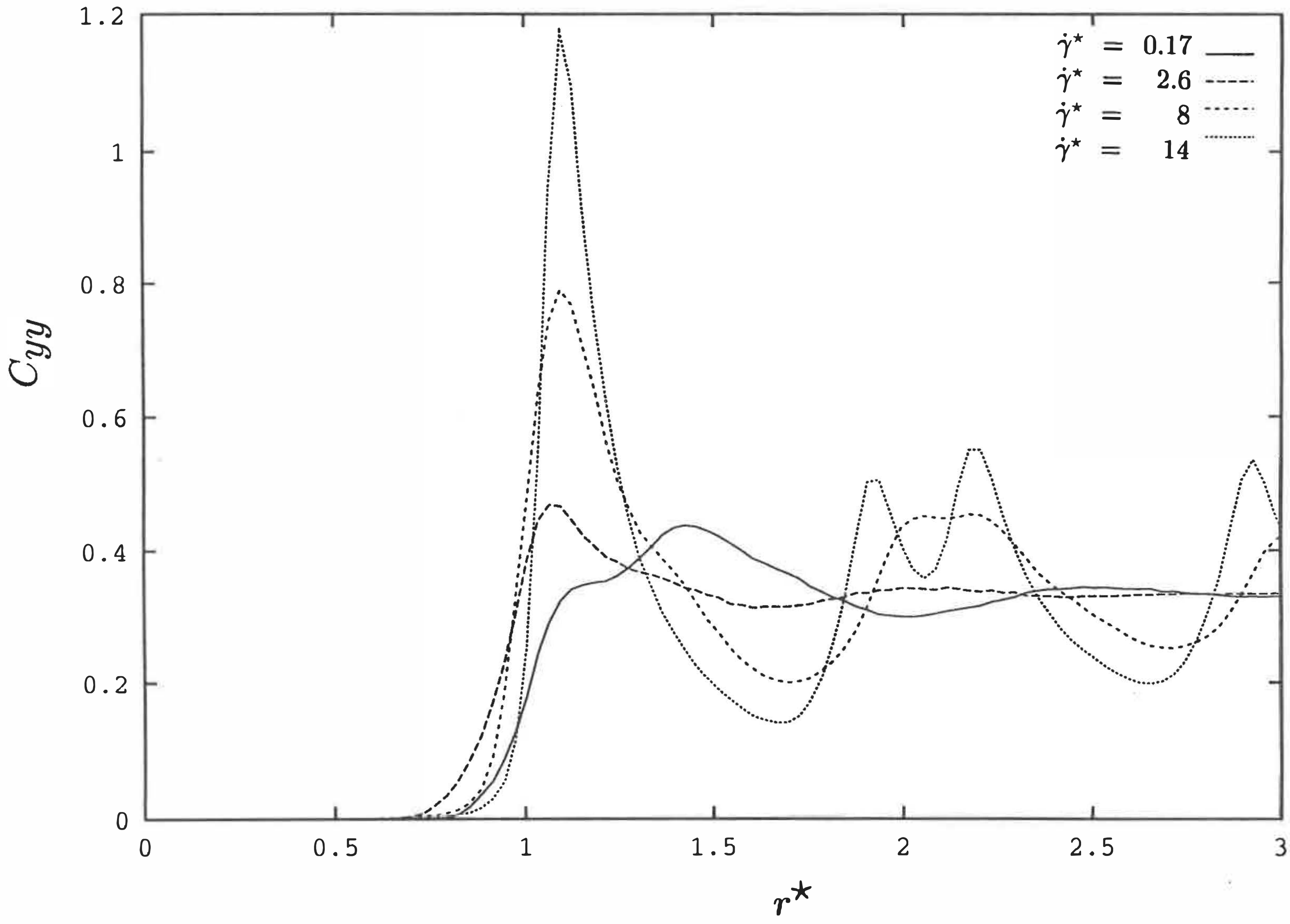
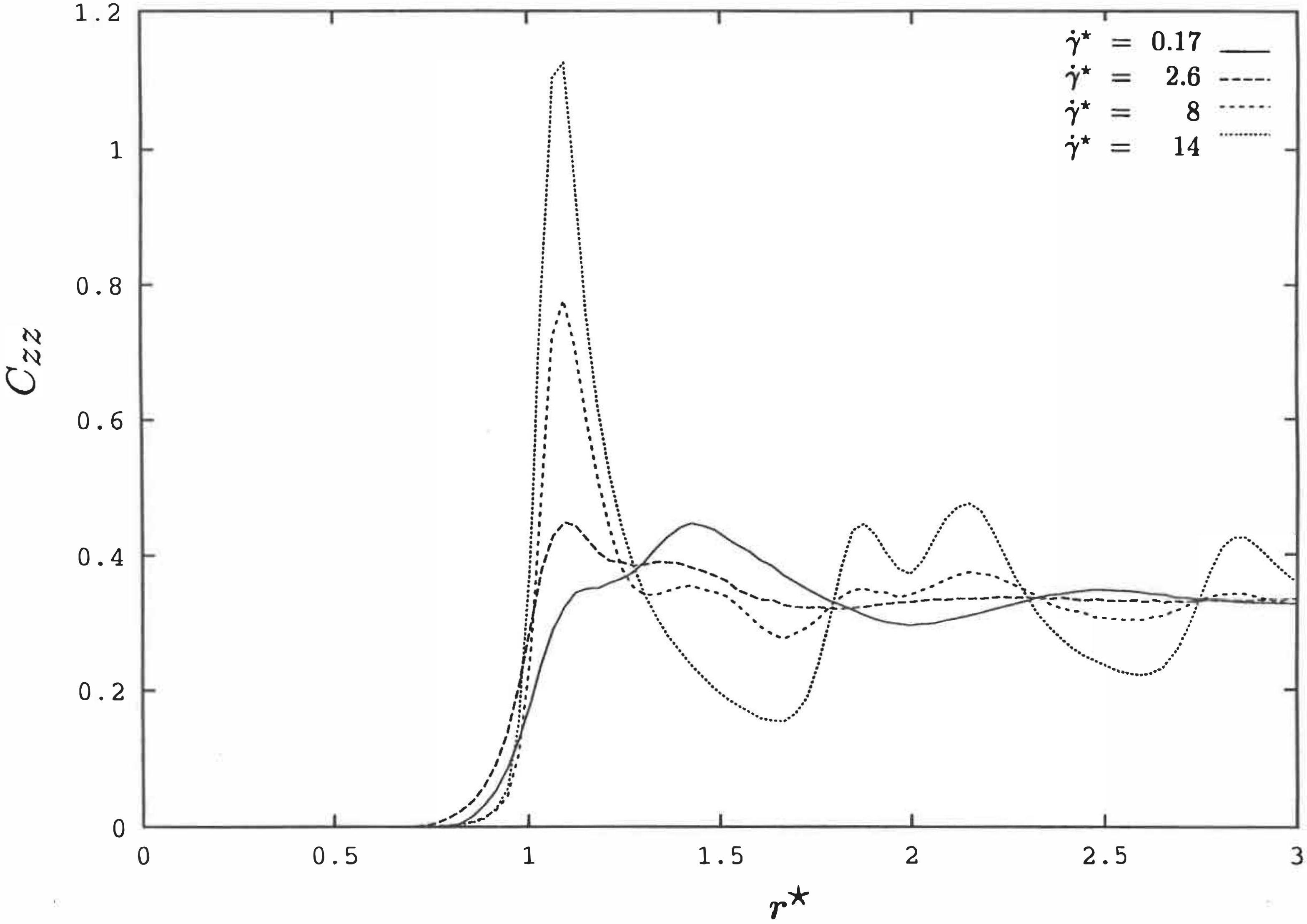

$\dot{\gamma}^\star$ = 0.17
$\dot{\gamma}^\star$ = 2.6
$\dot{\gamma}^\star$ = 8
$\dot{\gamma}^\star$ = 14
C_{yy}
$r^\star$
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
3

Figure 10c



$\dot{\gamma}^\star$ = 0.17
$\dot{\gamma}^\star$ = 2.6
$\dot{\gamma}^\star$ = 8
$\dot{\gamma}^\star$ = 14
C_{zz}
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
3
$r^\star$

Figure 11

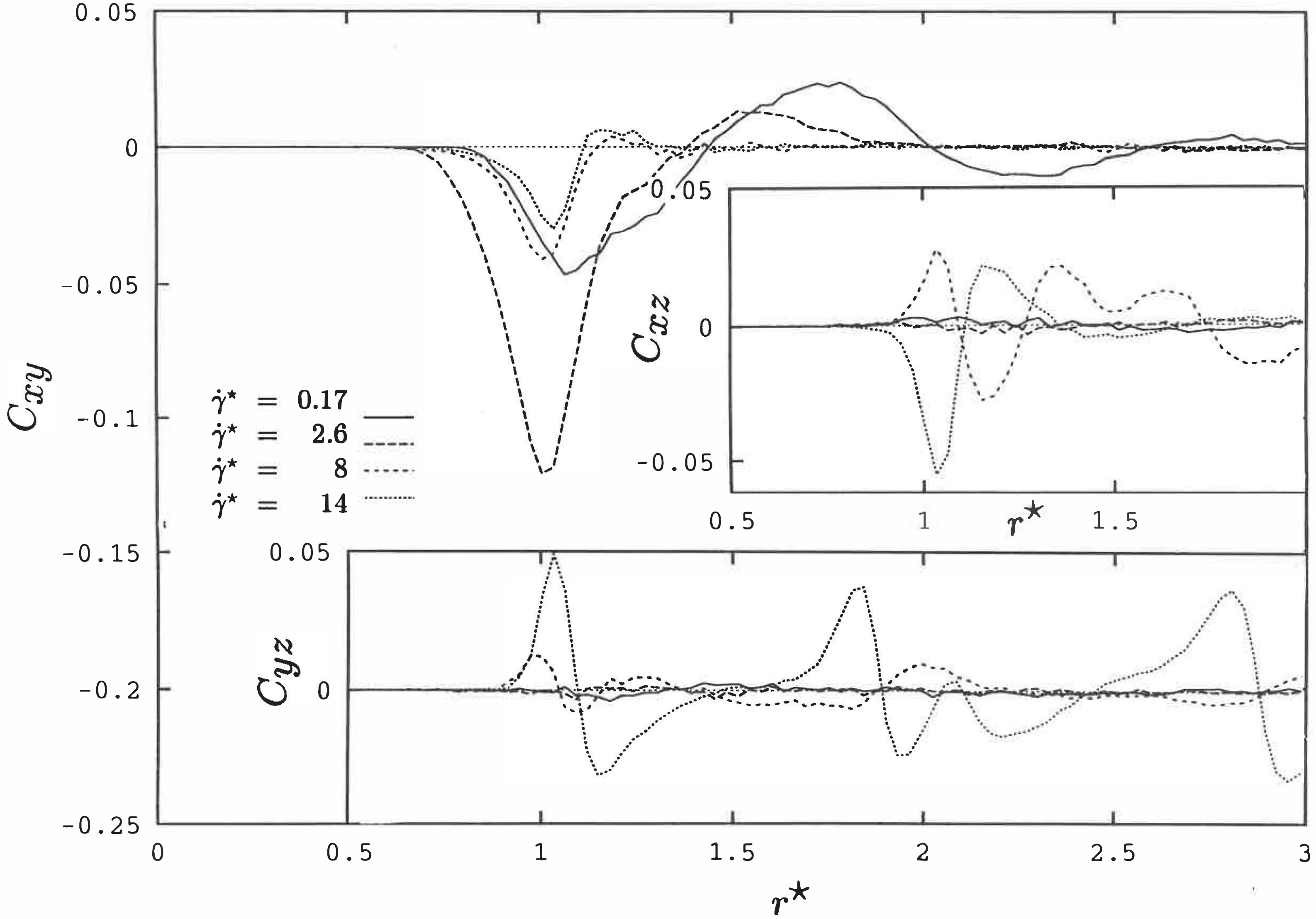

0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
C_{xy}
0
0.5
1
1.5
2
2.5
3
$r^\star$
$\dot{\gamma}^\star$ = 0.17
$\dot{\gamma}^\star$ = 2.6
$\dot{\gamma}^\star$ = 8
$\dot{\gamma}^\star$ = 14
0.05
0
-0.05
C_{xz}
0.5
1
1.5
$r^\star$
0.05
0
C_{yz}

Figure 12a

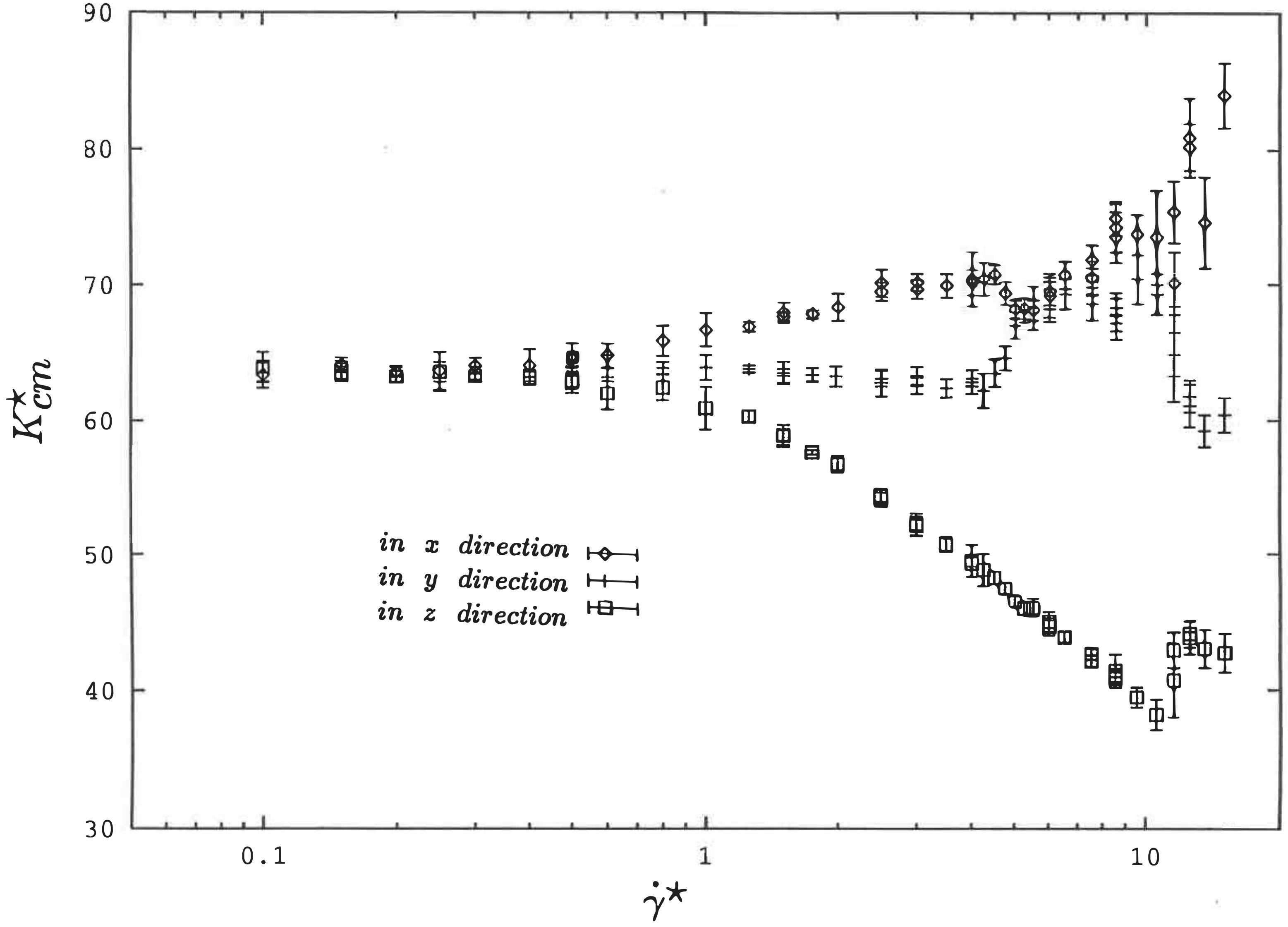

90
80
70
60
50
40
30
$K^\star_{cm}$
0.1
1
10
$\dot{\gamma}^\star$
in x direction
in y direction
in z direction

Figure 12b

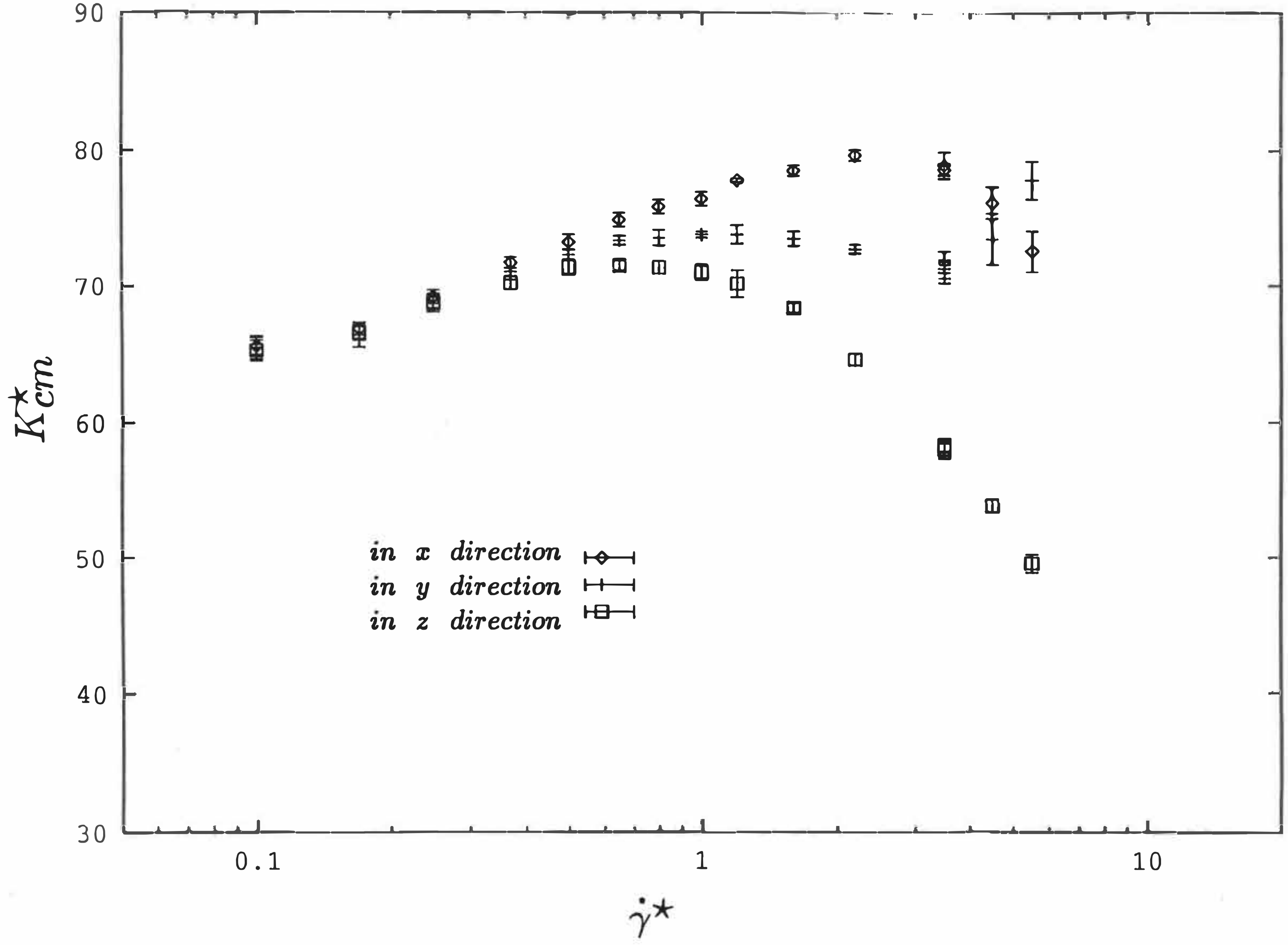
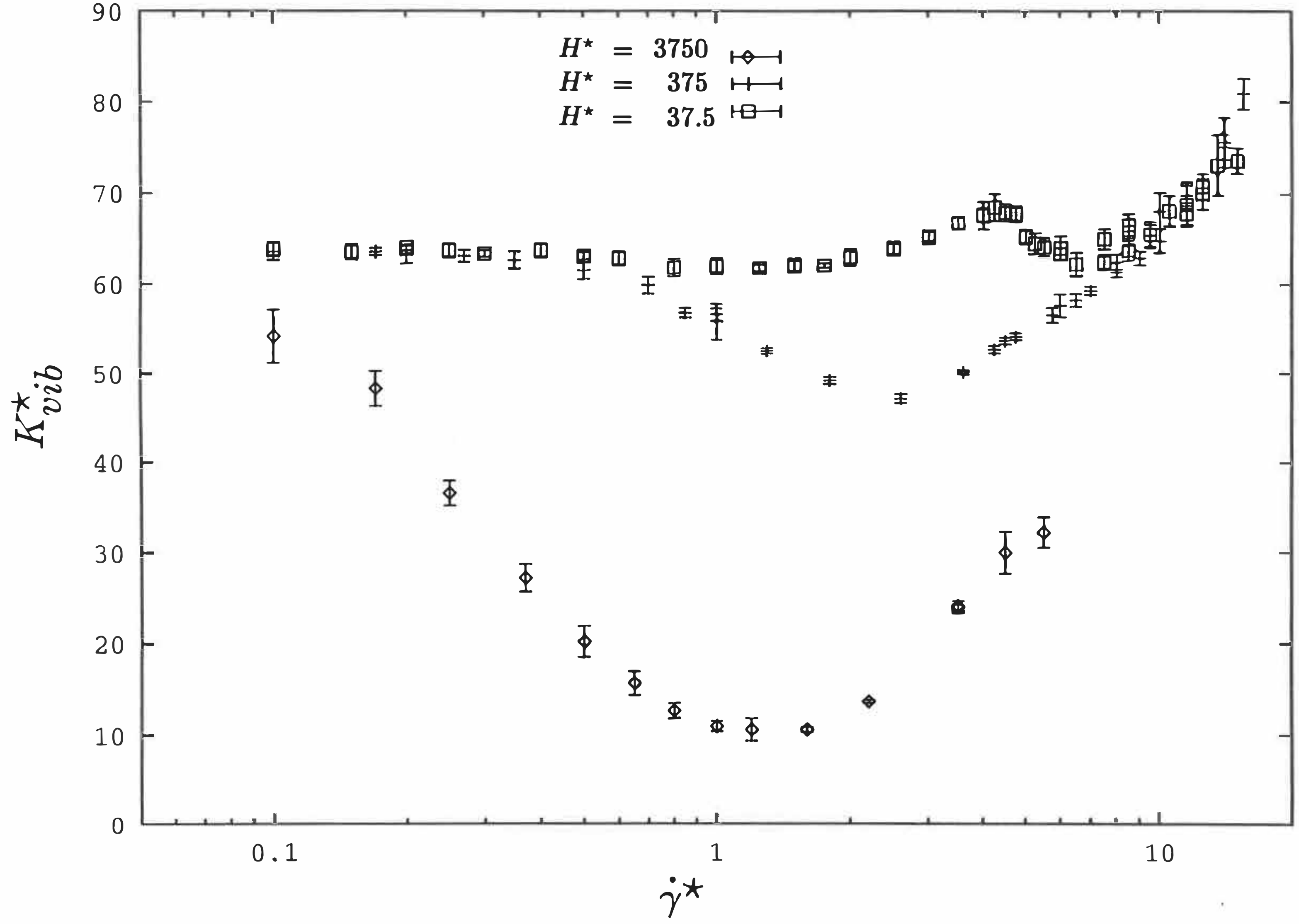

90
80
70
60
50
40
30
$K^\star_{cm}$
0.1
1
10
$\dot{\gamma}^\star$
in x direction
in y direction
in z direction

Figure 12c



$H^\star$ = 3750
$H^\star$ = 375
$H^\star$ = 37.5
$K^\star_{vib}$
0
10
20
30
40
50
60
70
80
90
0.1
1
10
$\dot{\gamma}^\star$

Figure 12d

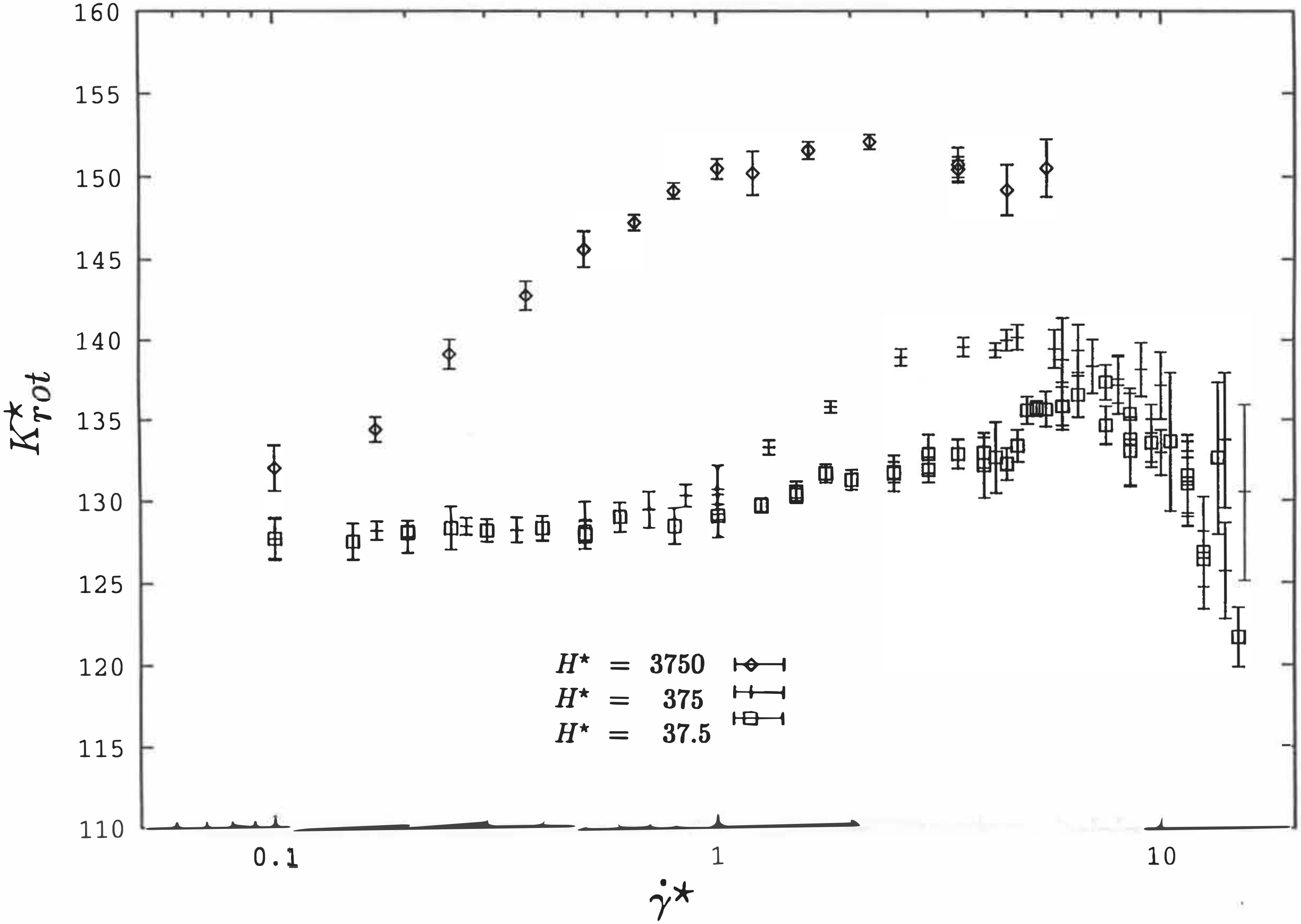

160
155
150
145
140
135
130
125
120
115
110
$K^{\star}_{rot}$
0.1
1
10
$\dot{\gamma}^{\star}$
$H^{\star}$ = 3750
$H^{\star}$ = 375
$H^{\star}$ = 37.5

Figure 13a

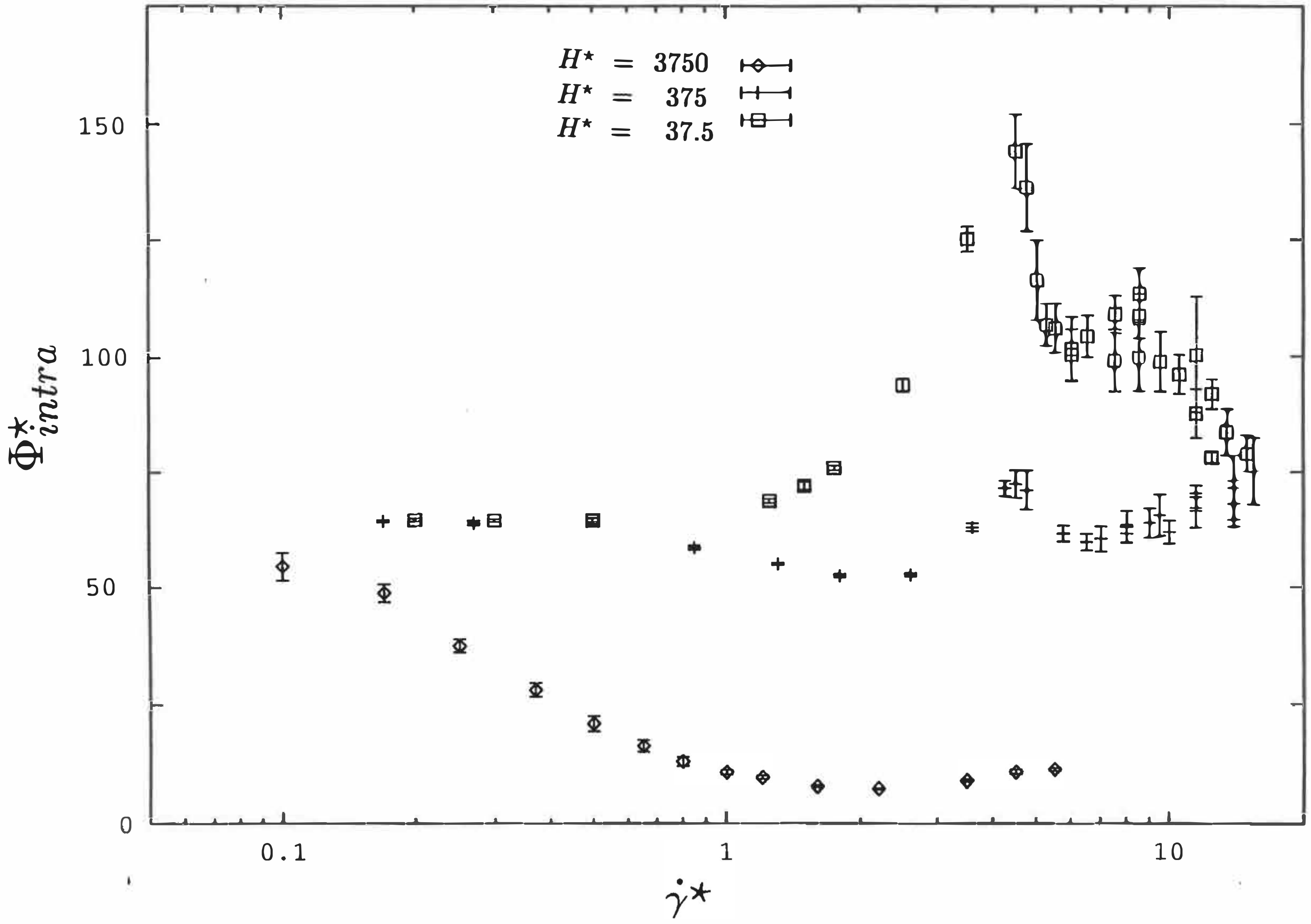
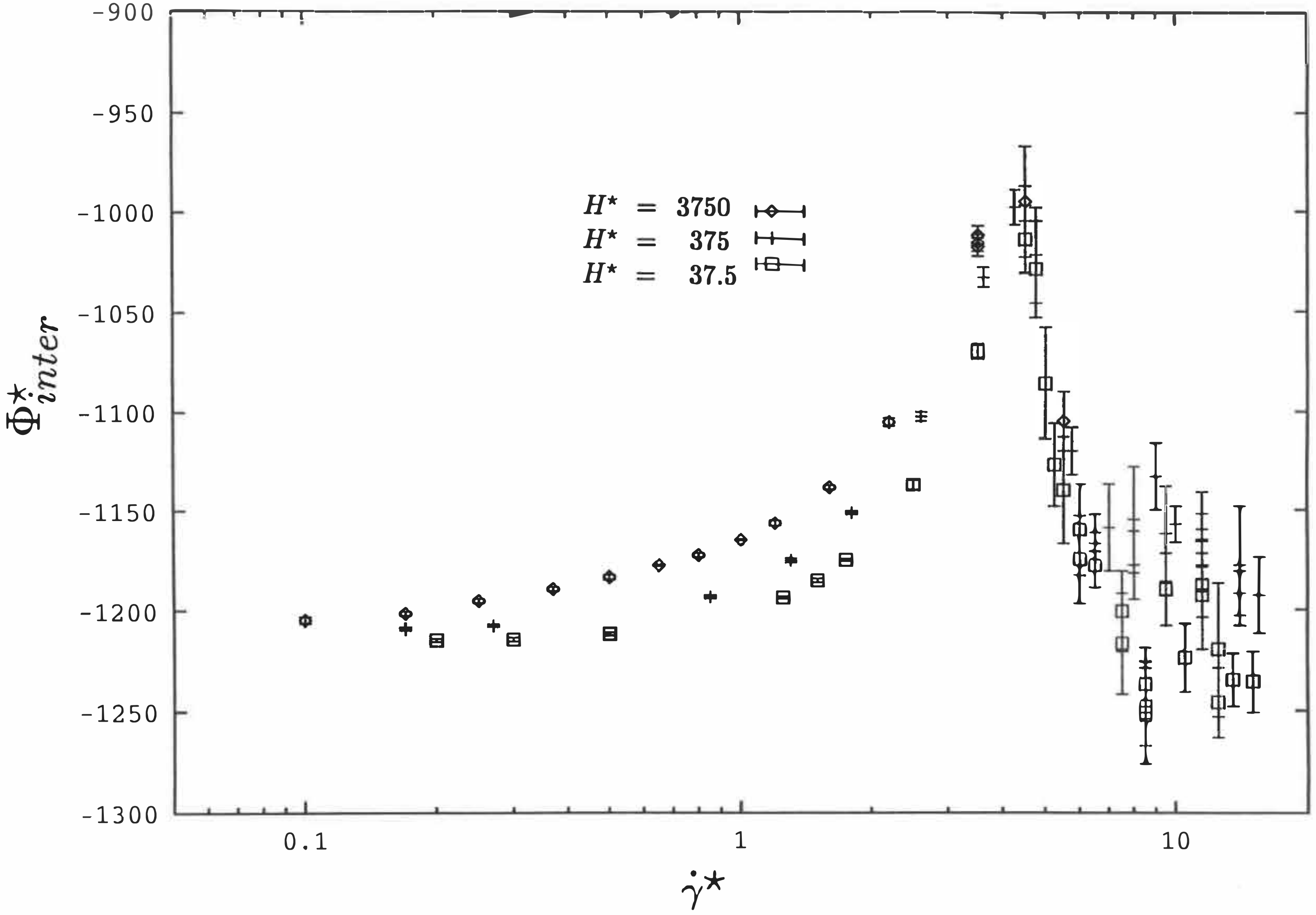


Figure 13b


-900
-950
-1000
-1050
-1100
-1150
-1200
-1250
-1300
$\Phi^{\star}_{inter}$
$H^{\star}$ = 3750
$H^{\star}$ = 375
$H^{\star}$ = 37.5
0.1
1
10
$\dot{\gamma}^{\star}$

3 RHEOLOGY OF RIGID BODIES

Rheology of Several Hundred Rigid Bodies

Bogdan Z. Dlugogorski*, Miroslav Grmela, Pierre J. Carreau
Département de génie chimique
Centre de recherche appliquée sur les polymères (CRASP)
École Polytechnique
Case postale 6079, succursale A
Montréal, Québec H3C 3A7, CANADA,

and

Georgy Lebon
Département de physique
Université de Liège
B5 - Sart Tilman
B-4000 Liège, BELGIUM.

Present address:

Conseil national de recherches
Institut de recherche en construction
Laboratoire nationale de l'incendie
Ottawa, Ontario K1A 0R6, CANADA.

Summary

A novel nonequilibrium molecular dynamics, originating in mesoscopic theory of suspensions, is introduced to investigate the rheology of several hundred ellipsoids of revolution, interacting *via* the Gay-Berne potential. The ratio of the strength parameters for two particles in side-by-side and end-to-end configurations is taken as 2.5. The microcanonical equations of motion for the translational and angular momenta as well as for mass-centers and orientational unit vectors are derived from a Hamiltonian. These expressions are then augmented by SLLOD-like and Gaussian thermostat terms added consistently to equations for both the rotational and translational degrees of freedom. The thermodynamic data are generated along one isotherm (nondimensional temperature maintained at unity). Rheology is investigated for two state points (*viz.* particle number density - ρ - equal to 0.25 and 0.4) that lie well inside the isotropic phase if no external flow is applied. As indicated by snapshots of molecular configurations, at the intermediate shear rates (nondimensional shear rate approximately 1-2), ellipsoids become aligned at a certain angle to the direction of flow and the stress tensor begins to be nonsymmetric. At even higher shear rates, this configuration breaks down leading to the formation of a transitory isotropic-type fluid, and then to the build-up of a highly ordered structure exhibiting global orientation of particles in the direction of the vorticity axis. For $\rho = 0.4$, the first (N_1) and the second (N_2) normal stress differences are positive and negative respectively, but at low densities ($\rho = 0.25$), N_1 becomes slightly negative. In addition to the stress tensor, we compute the conformation tensor, the order parameter and the components of the pair radial distribution function. At high shear rates the radial distribution functions become significantly anisotropic. Furthermore, we investigate the phenomenon of the stress overshoot at the inception of the simple shear flow from a molecular perspective, and study the evolution of the distribution of translational velocities as a function of the shear

rate.

1 Introduction

Non-equilibrium molecular dynamics (NEMD) is now a standard tool for collecting rheological data for short and medium sized alkanes [1, 2, 3, 4], and for model polymeric liquids [5, 6, 7]. Although in many respect similar, these two approaches to molecular modeling of rheology have different objectives. The former is concerned with reproducing exact experimental data by attempting to mirror the physical world, as much as it is computationally possible. In this approach, it is common to replace methyl or methylene groups of alkanes by a single computer particle (site) and to account for removed hydrogen atoms by selecting appropriate dihedral potentials. It is obvious that the number of sites that may be included in a single molecule is limited, even on the fastest supercomputers, and realistic polymer molecules cannot be modeled.

On the other hand, the objective of NEMD investigations of model polymeric particles, such as vibrating or rigid dumbbells (rigid rotors), is to formulate microrheological models that could ultimately be implemented in polymer processing operations yielding predictions (*e.g.* on surface quality) that are outside of reach of more traditional partial differential equations based techniques. The viscoelastic properties of vibrating dumbbell liquids (Newtonian viscosity, shear thinning, stress overshoot, *etc.*) derived from NEMD simulations are similar to those observed for real polymeric liquids [5, 6, 7]. On the other hand, the rigid rotor fluids are yet to be thoroughly investigated at the microrheological level.

In this work we approach rigid dumbbells from a slightly different point of view. Rather than considering them as a particular implementation of vibrating dumbbells we recognize that they are in fact a group of rigid bodies. In this context, our goal is to develop a novel NEMD algorithm for rigid bodies and to apply it to generate rheological data.

At equilibrium, the nature of rigid rotor fluids was studied by a way of two algorithms. These algorithms require that the atomic positions of beads be transformed into the mass-center and relative coordinates and the components of forces along molecular axis be removed to satisfy the constraint of constant bead-bead length. This is accomplished either by following an ingenious geometric construction [8, 9] or by introducing spherical polar angles [10] in case of the first algorithm, or alternatively by using angular velocities [11]. The former technique has been popular in collecting thermodynamical data for systems of diatomic homonuclear molecules, such as N_2 [9], whereas the latter has been employed for generating some microrheological results for rigid dumbbell fluids [5], despite the fact that this algorithm requires the evaluation of higher order derivatives of potential leading to complicated expressions. The simulation procedure becomes considerably simplified if the computation is done for rigid bodies rather than for rigid rotors. In addition, for rigid bodies the intermolecular interactions are modeled directly by interbody forces in place of a sum of four bead-bead potentials.

The SLLOD [12] equations of motion were originally developed for atomic liquids by incorporating the macroscopic Couette velocity field in the equations of motion for single particles. The extension of these equations to rigid rotors was given by Rudisill and Cummings [5], but only for the translational equations of motion; the equations for rotation of molecules were microcanonical. In this work, we propose a new form of the SLLOD algorithm with extra terms due to the explicit coupling between the velocity field and the rotational motion of particles.

In the next section, we briefly review our implementation of the intermolecular Gay-Berne potential. Elsewhere [6, 7], we have investigated the microscopic and mesoscopic properties, as well as the viscometric functions of fluids consisting of finitely extensible nonlinear elastic (FENE) and generalized Lennard-Jones (GLJ) dumbbells. Consequently, this work may be considered as an extension of the prior studies since the vibrating but stiff dumbbells, characterized by the average nondimensional bead-bead displacement of 0.9, are approximated by similarly sized rigid ellipsoids of revolution.

Since our goals were to study the rheology of systems composed of rigid bodies, we have confined ourselves to the investigation of the isotropic phase only; the rheology of smectic and nematic phases will be a subject of a subsequent publication. In Sec. 3.2, we verify by a means of thermodynamic (equilibrium) calculations that the system of rigid bodies, before the imposition of shear, is indeed well within the isotropic phase. The results presented in Sec. 3.2 were obtained for the isokinetic ensemble ($NVE_{rot,trans}$, both the rotational and the translational energies were thermostated separately) using the equations of motion derived in Sec. 3.1. We modify these equation in Sec. 4.1 in order to make them compatible with the macroscopic Couette velocity field, by adding a few supplementary terms. In case of the translational equations of motion, we keep the classical SLLOD expressions normally employed for atomic liquids. For the rotational equations of motion, we introduce new terms. In Sec. 4.2 we discuss the viscometric functions, and in addition the structural results. In the final section we assemble the major findings of this work.

2 Potential

A convenient means to simulate interaction of rigid molecules, each consisting of several single atoms or beads, is to replace site-site interactions by a single intermolecular potential. A potential of this type was first described by Corner [13] and later cast into an elegant analytical form by Berne and co-workers [14, 15, 16]. Of particular interest to us is the so-called Gay-Berne [16] potential that may be equally well applied to study long eccentric molecules comprising of several atoms and to short rigid rotors consisting of two beads.

Long molecules simplified as ellipsoids of revolution have been extensively used in equilibrium modeling of liquid crystals [17, 18, 19, 20, 21]. Equilibrium molecular simulations revealed the appearance of the vapor-liquid phase transition and also the isotropic-nematic, nematic-smectic and isotropic-smectic liquid crystal phase changes [20]. In these investigations, one type of the Gay-Berne potential was studied for molecules whose ratio of major ($\sigma_{\leftrightarrow}$) and minor ($\sigma_{\updownarrow}$) axes was set equal to 3 [16, 17]. This potential was deemed to represent the interaction between two linear molecules, each consisting of four Lennard-Jones atoms separated by a distance of 2/3 σ; σ is the collision radius or a distance at which the attractive and repulsive terms of the classical Lennard-Jones 12-6 potential cancel each other. It should be added for completeness that systems composed of hard ellipsoids of revolution [22, 23] and of molecules interacting *via* site-site potentials [24] also exhibit the phase transition between the isotropic and nematic phases.

Our main objective, is to investigate rheology of viscoelastic materials. Since we are interested in comparing the rheology of rigid bodies with the behavior of FENE and GLJ fluids we assume $\frac{\sigma_{\leftrightarrow}}{\sigma_{\updownarrow}} = 1.9$. In other words, we are pursuing the physics of less anisotropic molecules.

The interaction anisotropy between two cylindrically symmetric molecules whose orientation in space is defined by two unit vectors $\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j$ along their respective major axes and by the vector $\mathbf{r}_{ij}$ that connects their mass-centers, enters into the Gay-Berne potential by a means of two anisotropy parameters κ and $\mathcal{K}$,

$$\kappa = \frac{(\frac{\sigma_{\leftrightarrow}}{\sigma_{\updownarrow}})^{\frac{1}{2}} - 1}{(\frac{\sigma_{\leftrightarrow}}{\sigma_{\updownarrow}})^{\frac{1}{2}} + 1}, \quad \frac{\sigma_{\leftrightarrow}}{\sigma_{\updownarrow}} = 1.9, \tag{1}$$

$$\mathcal{K} = \frac{(\frac{\epsilon_{=}}{\epsilon_{--}})^{\frac{1}{2}} - 1}{(\frac{\epsilon_{=}}{\epsilon_{--}})^{\frac{1}{2}} + 1}, \quad \frac{\epsilon_{=}}{\epsilon_{--}} = 2.5. \tag{2}$$

It is obvious that the geometric anisotropy of a single molecule is embodied in the definition of κ, whereas the purpose of $\mathcal{K}$ is to correct the discrepancy in the depth of the potential energy well between the site-site and the simple Gaussian overlap model potential [14], which was a precursor to the Gay-Berne potential.

The expression $\frac{\epsilon_{=}}{\epsilon_{--}}$ in Eq. (2), considered normally as a fitted quantity, is the ratio of depths of the potential well for side-by-side $_{=}$ and end-to-end $_{--}$ configurations, in case of atom-atom interactions. The exponent $\frac{1}{2}$ in Eq. (2) is another parameter that could be adjusted, although historically it has been always set to $\frac{1}{2}$; we are not aware of any publication in which this term has been changed. A similar comment is also applicable to yet another additional adjustable parameter, called ν, originally introduced by Gay and Berne [16], but never used in simulations; it is always equal to one. Finally, in statistical thermodynamics investigations of mesogenic molecules $\frac{\epsilon_{=}}{\epsilon_{--}} = 5$ [17, 18, 19, 20, 21].

For simplicity we introduce nondimensional quantities, denoted in this section by an asterisk which would be dropped thereafter, based on the parameters of the Lennard-Jones potential, such as σ and ϵ; for a complete list of nondimensional quantities see [6] and App. B of [25]. Using these quantities the Gay-Berne equations are enunciated as follows,

$$\phi^{\star}_{ij}(\hat{\mathbf{p}}_i,\hat{\mathbf{p}}_j,\mathbf{r}^{\star}_{ij}) = 4\;\epsilon^{\star}_{ij}(\hat{\mathbf{p}}_i,\hat{\mathbf{p}}_j,\hat{\mathbf{r}}_{ij})\;[(\frac{1}{r^{\star}_{ij}-\sigma^{\star}_{ij}(\hat{\mathbf{p}}_i,\hat{\mathbf{p}}_j,\hat{\mathbf{r}}_{ij})+1})^{12} - (\frac{1}{r^{\star}_{ij}-\sigma^{\star}_{ij}(\hat{\mathbf{p}}_i,\hat{\mathbf{p}}_j,\hat{\mathbf{r}}_{ij})+1})^{6}], \quad (3)$$

$$\epsilon^{\star}_{ij}(\hat{\mathbf{p}}_i,\hat{\mathbf{p}}_j,\hat{\mathbf{r}}_{ij}) = (\frac{1-\kappa^2}{1-\kappa^2(\hat{\mathbf{p}}_i\cdot\hat{\mathbf{p}}_j)^2})^{\frac{1}{2}}\;\{1-\frac{\mathcal{K}}{2}[\frac{(\hat{\mathbf{r}}_{ij}\cdot(\hat{\mathbf{p}}_i+\hat{\mathbf{p}}_j))^2}{1+\mathcal{K}(\hat{\mathbf{p}}_i\cdot\hat{\mathbf{p}}_j)} + \frac{(\hat{\mathbf{r}}_{ij}\cdot(\hat{\mathbf{p}}_i-\hat{\mathbf{p}}_j))^2}{1-\mathcal{K}(\hat{\mathbf{p}}_i\cdot\hat{\mathbf{p}}_j)}]\}^2, \quad (4)$$

$$\sigma^{\star}_{ij}(\hat{\mathbf{p}}_i,\hat{\mathbf{p}}_j,\hat{\mathbf{r}}_{ij}) = \{1-\frac{\kappa}{2}[\frac{(\hat{\mathbf{r}}_{ij}\cdot(\hat{\mathbf{p}}_i+\hat{\mathbf{p}}_j))^2}{1+\kappa(\hat{\mathbf{p}}_i\cdot\hat{\mathbf{p}}_j)} + \frac{(\hat{\mathbf{r}}_{ij}\cdot(\hat{\mathbf{p}}_i-\hat{\mathbf{p}}_j))^2}{1-\kappa(\hat{\mathbf{p}}_i\cdot\hat{\mathbf{p}}_j)}]\}^{-\frac{1}{2}}. \quad (5)$$

Where,

$$\mathbf{r}^{\star}_{ij} = \mathbf{r}^{\star}_{i} - \mathbf{r}^{\star}_{j}, \quad r^{\star}_{ij} = \mid \mathbf{r}^{\star}_{ij} \mid . \qquad (6)$$

In Eq. (4), we have introduced a normalization factor $(1-\kappa^2)^{\frac{1}{2}}$ to make certain that $\epsilon^{\star}$ for side-by-side configuration is unity. A comparison between the Gay-Berne and the sums of four Lennard-Jones atom-atom potentials for both side-by-side and end-to-end configuration is presented in Fig. 1. For these configurations, especially for the end-to-end geometry, the two potentials are almost indistinguishable from each other. As shown in Fig. 2, the Gay-Berne potential tends to smooth the inhomogeneities introduced by interactions of single beads. After examining a number of two-dumbbell geometries, similar to these in Fig. 2, we conclude that the Gay-Berne potential approximates closely intermolecular bead-bead potential.

Although Eqs. (3)-(5) lead to rather complex expressions for forces,

$$\mathbf{F}_i^r = -\mathbf{F}_j^r = -\frac{\partial}{\partial \mathbf{r}_i}\sum_{i\neq j,j=1}^{N}\phi_{ij}, \quad \mathbf{F}_i^p = -\frac{\partial}{\partial \hat{\mathbf{p}}_i}\sum_{i\neq j,j=1}^{N}\phi_{ij}, \tag{7}$$

from the computational point of view it is 10 times faster to calculate forces from the analytical formulae than by taking gradients of the Gay-Berne potential numerically.

3 Equilibrium Calculations

3.1 Details of the model, algorithm and computational procedure

The Hamiltonian for a system of rigid bodies interacting through a Gay-Berne pair potential is expressed in terms of position $\mathbf{r}$ and momenta $\mathbf{M}$ of mass-centers, orientational unit vectors $\hat{\mathbf{p}}$, and angular momenta $\mathbf{m}$,

$$\mathcal{H}(\mathbf{r},\mathbf{P},\hat{\mathbf{p}},\mathbf{M}) = \frac{1}{2}\sum_{i=1}^{N}\frac{P_{i\alpha}P_{i\alpha}}{m_i} + \frac{1}{2}\sum_{i=1}^{N} I_{i\alpha\alpha}^{-1}M_{i\alpha}M_{i\alpha} + \sum_{i=1}^{N-1}\sum_{j>i}^{N}\phi_{ij}(\hat{\mathbf{p}}_i,\hat{\mathbf{p}}_j,\mathbf{r}_{ij}). \tag{8}$$

The summation convention is used in Eq. (8). This Hamiltonian leads to the following equations of motion,

$$\dot{\mathbf{r}}_i = \frac{\partial\mathcal{H}}{\partial\mathbf{P}_i} = \frac{\mathbf{P}_i}{m_i}, \tag{9}$$

$$\dot{\mathbf{P}}_i = -\frac{\partial\mathcal{H}}{\partial\mathbf{r}_i} = -\frac{\partial}{\partial\mathbf{r}_i}\sum_{i\neq j,j=1}^{N}\phi_{ij}, \tag{10}$$

$$\dot{\hat{\mathbf{p}}}_i = -\hat{\mathbf{p}}_i\times\frac{\partial\mathcal{H}}{\partial\mathbf{M}_i} = -\hat{\mathbf{p}}_i\times\underline{\omega}_i, \tag{11}$$

$$\dot{\mathbf{M}}_i = -\mathbf{M}_i \times \frac{\partial \mathcal{H}}{\partial \mathbf{M}_i} - \hat{\mathbf{p}}_i \times \frac{\partial \mathcal{H}}{\partial \hat{\mathbf{p}}_i} = -\mathbf{M}_i \times \underline{\omega}_i - \hat{\mathbf{p}}_i \times \frac{\partial}{\partial \hat{\mathbf{p}}_i} \sum_{i \neq j, j=1}^{N} \phi_{ij}. \qquad (12)$$

The angular velocity $\underline{\omega}$ is related to the angular momentum $\mathbf{M}$ by $M_\alpha = I_{\alpha\alpha}\,\omega_\alpha$, in the system of coordinates fixed in the body; $I_{\alpha\alpha}$ are diagonal elements of the inertia tensor $\underline{\underline{I}}$. In this system, off-diagonal elements of both the inertia $\underline{\underline{I}}$ and the inverse inertia tensor $\underline{\underline{I}}^{-1}$ disappear. In addition to constant total energy and vanishing translational momenta, these equations of motion also preserve two other quantities, namely $\sum_{i=1}^{N} \mathbf{M}_i \cdot \hat{\mathbf{p}}_i$, and for each particle $\hat{\mathbf{p}}_i \cdot \hat{\mathbf{p}}_i$ [26]. The angular velocity and momenta are in the body frame. On the other hand, Eq. (11) describes the motion of the unit vector $\hat{\mathbf{p}}$ as seen from the system; see also Eq. (2) of ref. [27]. Eqs. (8)-(12) are N-particle extension of equations introduced by Sudarshan and Mukunda [28] (see also [26]). We use × and · for the vector and scalar products, respectively.

In order to guarantee that the norm of $\hat{\mathbf{p}}_i$ $i = 1, \ldots, N$ is constant also in numerical calculations (the equations are solved in Cartesian rather than in spherical coordinates), we shall replace (11) by,

$$\dot{\hat{\mathbf{p}}}_i = -\hat{\mathbf{p}}_i \times \underline{\omega}_i, -\lambda_i \hat{\mathbf{p}}_i, \qquad (13)$$

where λ_i's are Lagrange multipliers

$$\lambda_i = -\frac{\hat{\mathbf{p}}_i \cdot (\hat{\mathbf{p}}_i \times \underline{\omega}_i)}{\hat{\mathbf{p}}_i \cdot \hat{\mathbf{p}}_i} \qquad (14)$$

(in numerical calculations $\hat{\mathbf{p}}_i \cdot (\hat{\mathbf{p}}_i \times \underline{\omega}_i)$ is not strictly zero).

Gauss's principle of least constraint, first introduced within the context of molecular dynamics by Evans *et al.* [29] is applied to Eqs. (10&12) to give,

$$\dot{\mathbf{P}}_i = \mathbf{F}_i^r - \alpha_1 \mathbf{P}, \qquad (15)$$

$$\dot{\mathbf{M}}_i = -\mathbf{M}_i \times \underline{\omega}_i + \hat{\mathbf{p}}_i \times \mathbf{F}_i^p - \alpha_2 \mathbf{M}_i. \qquad (16)$$

Where,

$$\alpha_1 = \frac{\sum_{i=1}^{N} \mathbf{P}_i \cdot \mathbf{F}_i^r}{\sum_{i=1}^{N} \mathbf{P}_i \cdot \mathbf{P}_i}, \tag{17}$$

$$\alpha_2 = \frac{\sum_{i=1}^{N} \underline{\omega}_i \cdot (\hat{\mathbf{p}}_i \times \mathbf{F}_i^p - \mathbf{M}_i \times \underline{\omega}_i)}{\sum_{i=1}^{N} \underline{\omega}_i \cdot \mathbf{M}_i}. \tag{18}$$

The Lagrangian multiplier α_2 is obtained by setting the total rotational kinetic energy constant,

$$\frac{d}{dt} \sum_{i=1}^{N} I_{i\alpha\alpha}^{-1} M_{i\alpha} M_{i\alpha} = \sum_{i=1}^{N} \underline{\omega} \cdot \dot{\mathbf{M}}_i = 0, \tag{19}$$

and by substituting (19) into (16). Subsequently, α_2 is found by a straightforward algebraic manipulation.

For the sake of numerical accuracy we also include the second term in the numerator in the equation for α_2. During shearing, α_1 and α_2 serve as energy sinks taking away the energy generated by the viscous dissipation.

We have performed calculations for 256 rigid bodies which were originally placed on the faced-centered-cubic (FCC) lattice and the standard periodic boundary conditions were applied in three orthogonal directions. In the course of equilibration, the translational velocities were rescaled whereas the rotational velocities were left to equilibrate by themselves. After two-three thousand time steps, although the entire equilibration lasted 10,000 time steps, we have always observed that $E_{trans} = \frac{3}{2}E_{rot}$, as expected from statistical mechanics. The Verlet order parameter [30] and Haile's implementation of the Boltzmann H-function [31] were computed to verify the melting of the FCC lattice and the emergence of the Maxwellian distribution.

The mass of a single ellipsoid of revolution was taken as nondimensional two to ensure that the dimensional density of fluids composed of rigid bodies would in fact correspond to density of fluids built from vibrating dumbbells. For example, in nondimensional quantities $\rho = 0.4$ as used in this paper becomes $\rho = 0.8$ in refs [6, 7]. The moments of inertia were calculated on the basis of the ellipsoids' repulsive cores (see Fig. 2), taking major and minor axes as 0.95 and 0.5 respectively, thus $I_{zz} = 0.2$, and $I_{xx} = I_{yy} = 0.461$. The static properties are not effected by these choices, however the dynamic properties do depend on them.

For all runs, the nondimensional temperature was maintained constant and equal to unity. The particle number density, defined as a number of particles in a cubic box of unit sides, was varied between 0.05 and 0.725 to generate the thermodynamic data along one isotherm ($T = 1$). On the other hand, the rheological calculations presented in Sec. 4 were performed for two particle number densities *viz.*, 0.25 and 0.4 as a function of the shear rate.

The Gear five-value predictor-corrector method (App. E of ref. [25]) was employed to integrate Eqs. (9)-(12) using a time step of 0.0025. In order to minimize the total energy drift (for microcanonical runs) as well as the translational and rotational temperature drifts (for isokinetic runs), and thus avoiding frequent rescaling of velocities, no cut-off radius was used, at the cost of longer execution (2.3 sec. per integration, including various overheads, on RISC 320). The temperature drifts are similar for the translational and rotational temperatures and depend on the fluid density. For example, for $T = 1$ and a time step of 0.0025 these drifts are (per integration), 2.3 10^{-9} for $\rho = 0.05$, 4.1 10^{-8} for $\rho = 0.4$, and 2.2 10^{-7} for $\rho = 0.6$. After reaching steady state, a process that might have taken up to 100,000 times steps for points within the nematic phase, for each run 10 subaverages were calculated over 1000 time steps periods. Subsequently, these subaverages were considered as independent observation and used to calculate an overall average and its standard deviation.

3.2 Equilibrium Results

As the density of the liquid is increased the molecules are not able to rotate freely and are forced to assume structurally preferred configurations. A very sensitive probe into the density induced evolutionary changes within the fluid is provided by the total potential energy Φ. As demonstrated in Fig. 3, when the density of model molecules is sufficiently low, the configurational energy decreases proportionally to the particle number density ρ. This proportionality relation follows from the fact that at low densities molecules are separated by long distances and the intermolecular interaction is mostly due to the attractive Gay-Berne forces (Fig. 1). For this reason, the velocity autocorrelation function is always positive for low density fluids. Since there is no predominant direction exhibited by the ensemble of molecules and the fluid is above its critical point but closer to its vapor region (see the caption of Fig. 4) this initial part of the $\Phi(\rho)$ curve defines the field of existence of the vapor-like isotropic phase, also denoted by isotropic phase region I.

Before carrying on with the discussion we make a short digression devoted to computational techniques designed to probe the liquid's structure at the molecular level. For example, the directional global order is examined by the conformation tensor $\underline{\underline{c}}$:

$$c_{\alpha\beta} = \frac{1}{N} < \sum_{i=1}^{N} \hat{p}_{i\alpha}\hat{p}_{i\beta} >_t . \tag{20}$$

In 3-D space, $\underline{\underline{c}}$ is often visualized as an ellipsoid (*e.g.* [7]) whose axes correspond to eigenvectors of the conformation tensor. The largest eigenvector, normalized to unity and stripped of its arrow to become headless, is called director ($\hat{\mathbf{n}}$). It points along the ellipsoid's major axis. Distribution of orientational unit vectors around $\hat{\mathbf{n}}$ is conveniently measured either by specifying the lengths of the two other minor axes or by defining the order parameter S,

$$S_t = \frac{3}{2N} < \sum_{i=1}^{N} (\hat{\mathbf{n}} \cdot \hat{\mathbf{p}}_i)^2 >_t - \frac{1}{2}. \tag{21}$$

The notation $< \ldots >_t$ stands for time average. By the same token, S_t denotes a time averaged order parameter. An instantaneous order parameter is indicated simply by S. According, to their definitions, S_t and S may vary between zero for a perfectly isotropic phase, for which $< c_{xx} >_t=< c_{yy} >_t=< c_{zz} >_t$, and unity in the limit of a faultless alignment. The trace of the conformation tensor is unity since molecules are rigid and their direction is space is specified in terms of orientational unit vectors.

The orientation independent pair radial distribution function for monatomic liquids,

$$g(r) = \frac{2}{N\,\rho} < \sum_{i}^{N-1} \sum_{j>i}^{N} \delta(r - r_{ij}) >_t = \frac{2}{N\,\rho} \frac{< N(r, \Delta r) >_t}{4\pi r^2 \Delta r}, \tag{22}$$

conveys two types of information about the local liquid structure. First, the overall appearance of $g(r)$ allows to identify the state of matter of the simulated material [31]. On the other hand, peaks in g indicate location and the relative significance of coordination spheres. $N(r, \Delta r)$ denotes number of molecules (js) in a concentric spherical shell of thickness Δr around molecule i; here, we follow the nomenclature used in App. A of ref. [31]. Unfortunately, g is not immediately useful for cylindrically symmetric particles due to its averaging action. For molecular liquids exhibiting directional ordering, it is necessary to resolve g into tensorial components. In general, the resolution of g may be carried out in relation to the system coordinates,

$$g_{\alpha\beta}(r) = \frac{2}{N\,\rho} \frac{< \sum_{i=1}^{N-1} \sum_{j>i}^{N} \frac{r_{ij\alpha} r_{ij\beta}}{r_{ij} r_{ij}} >_t}{4\pi r^2 \Delta r}, \tag{23}$$

leading to a symmetric tensor $\underline{\underline{g}}$. The summation $\sum_{i=1}^{N-1} \sum_{j>i}^{N}$ has a similar meaning as $N(r, \Delta r)$ except that it may assume real values since the addition is done over real numbers rather than integers as before. Alternatively, the pair distribution function may be resolved into elements of a 2 × 2 tensor, according to the instantaneous director. The diagonal components of such a tensor are called parallel $g_{||}(r)$ and perpendicular $g_{\perp}(r)$ pair radial distribution functions,

$$g_{||}(r) = \frac{2}{N\,\rho} \frac{< \sum_{i=1}^{N-1} \sum_{j>i}^{N} \frac{r_{ij||} r_{ij||}}{r_{ij} r_{ij}} >_t}{4\pi r^2 \Delta r}, \tag{24}$$

$$g_{\perp}(r) = \frac{2}{N\,\rho} \frac{< \sum_{i=1}^{N-1} \sum_{j>i}^{N} \frac{r_{ij\perp} r_{ij\perp}}{r_{ij} r_{ij}} >_t}{4\pi r^2 \Delta r}. \tag{25}$$

Where $r_{ij\|}$ and $r_{ij\perp}$ are the components of $\mathbf{r}_{ij}$ along and normal to $\hat{\mathbf{n}}$.

For simulations of liquid crystals, it is convenient to introduce two other functions, namely the longitudinal $g_{\|}(r_{\|})$ and transverse $g_{\perp}(r_{\perp})$ pair distribution functions, to help distinguish among several liquid phases,

$$g_{\|}(r_{\|}) = \frac{2}{N\rho}\frac{< N(r_{\|}, \Delta r_{\|}) >_t}{2L^2\Delta r_{\|}}, \tag{26}$$

$$g_{\perp}(r_{\perp}) = \frac{2}{N\rho}\frac{< N(r_{\perp}, \Delta r_{\perp}) >_t}{2\pi L r_{\perp}\Delta r_{\perp}}. \tag{27}$$

Where, L denotes side length of the primitive cube. In case of the longitudinal pair distribution function, the space around each molecule is divided into rectangular parallelepipeds of thickness $\Delta r_{\|}$ perpendicular to $\hat{\mathbf{n}}$. For the transverse pair distribution function, the concentric cylindrical shells of thickness $\Delta r_{\perp}$ and length L are taken.

Upon further compression, the isotropic phase becomes more liquid-like (region II) as one can infer from the fact that the total potential energy of the ensemble of molecules begins to rise (Fig. 3). On the average, ellipsoids are closer together, and the repulsive forces among them come to be more significant. A sudden increase in pressure in region II (Fig. 4) confirms lower compressibility that is characteristic of the liquid phase. In this region, it is possible for the liquid to form local directionally-ordered clusters that are promptly destroyed as indirectly indicated by significant fluctuations in the structure factor (Fig. 5). Perhaps the most complete justification for the evolution in the nature of the fluid comes from the parallel ($g_{\|}(r)$) and perpendicular ($g_{\perp}(r)$) pair radial distribution functions (Figs 6 a&b). For $\rho = 0.25$, both functions are structureless with a small but noticeable local minimum at $r \simeq 2$. The complete absence of such a minimum is an evidence for a low-density gas phase [31]. On the other hand, well defined but not deep troughs at the same location signify liquid phase, as shown for $\rho = 0.5$ and 0.625 in Figs 6 a&b.

The isotropic/nematic phase transition is similar in appearance as the one observed elsewhere, for $T = 0.95$ for mesogenic molecules; compare Figs 3-5 with ref. [20]. This transition is reflected by discontinuities in pressure and potential energy and an important increase in the order parameter. The exact location of the two phase region, or in other words the location of the horizontal line connecting two branches of the pressure and potential curves may be determined by the free energy calculations. However, this was not attempted here.

According to the conclusions drawn by de Miguel *et al.* [20] for their type of the Gay-Berne fluid, above the triple point temperature (isotropic/nematic/smectic) of $T = 0.8$, isotropic phase evolves to a stable nematic phase. At the lower temperatures, Miguel *et al.* observed a direct phase transition between the isotropic and smectic phases. We have no sufficient thermodynamic data to estimate the triple point temperatures but may verify the build-up of the nematic phase. This will be done next.

The longitudinal pair distribution function is structureless for the isotropic phase, but displays symmetric peaks for the nematic phase (Fig. 6c). Similar peaks, though of much larger amplitude, indicate smectic phase (Fig. 11 of ref. [20]). However, there is one important difference that allows one to differentiate between the phases. For the isotropic phase, the distance between two neighboring peaks of $g_{||}(r_{||})$ pair distribution function is slightly less than half of the molecular end-to-end distance (0.9σ), as seen in Fig. 6c, whereas in case of the smectic phase the peak-to-peak separation is closer to the entire end-to-end distance (Fig. 11 of ref. [20]). In other words, each plane within the smectic phase consists of rows of directionally ordered molecules. This alignment is preserved for the isotropic phase, but no rows are present. Similar configurational information as presented by $g_{||}(r_{||})$ is shown by the transverse distribution function (Fig. 6d), which is structureless for the isotropic phase, both for regions I&II, but displays cylindrical coordination shells for the nematic phase. Finally, snapshots of the isotropic and nematic phases are given in Fig.7.

4 Nonequilibrium Calculations

4.1 Extension of the algorithm to simple shear flow

The dynamical equations introduced in Sec. 3.1 are applicable only if the system composed of the model molecules is not subjected to external force. If the model molecules are subjected to a flow (playing a role of an external force) then Eqs. (10)-(18) have to be modified. In particular, it has to be specified how the flow influences the motion of the model molecules and how the macroscopic quantities of interest (*e.g.* the stress tensor) are expressed in terms of trajectories of the model molecules.

In the case of molecules modeled as Lennard-Jones-like particles or as chains of Lennard-Jones-like particles, the modification is known under the name of SLLOD dynamics [12]. The SLLOD dynamics combined with hydrodynamics (*i.e.* an infinitely dimensional dynamical system) has been shown to be Hamiltonian [32]. The Hamiltonian function that generates the combined dynamics is the total energy. This new result can be used for example to derive systematically the expressions for the stress tensor that are compatible with the dynamics of model molecules and to provide new simple profs of some well known results about SLLOD dynamics. It has been shown [12], for example, that in a simple shear flow (*i.e.* $\frac{\partial u_x}{\partial r_y} = \dot{\gamma} = const$, $\mathbf{u}$ is the field of the fluid velocity) $\frac{d\mathcal{H}}{dt} = \dot{\gamma}\sigma_{xy}V$, where $\underline{\underline{\sigma}}$ is the stress tensor (the stress tensor used in this paper equals minus the stress tensor used in [32]), V is the total volume and $\mathcal{H}$ is the Hamiltonian of the system composed of the model molecules. This result comes from the Hamiltonian structure proven in [32] as follows. The total Hamiltonian that is conserved during the time evolution is $\mathcal{H}+\int d\,\mathbf{r}\frac{1}{2}\rho u^2$ where ρ is the mass density field of the fluid. We have thus $\frac{d\mathcal{H}}{dt} = -\frac{d}{dt}\int d\,\mathbf{r}\frac{1}{2}\rho u^2 = \dot{\gamma}\sigma_{xy}V$. The second equality follows immediately from $\frac{\partial(\rho\mathbf{u})}{\partial t} = \mathbf{div}(-\rho\mathbf{u}\mathbf{u} + \underline{\underline{\sigma}})$ and from the integration by parts. Similarly, the so-called adiabatic incompressibility of the phase space [12] can be proven from the Liouville theorem for Hamiltonian systems.

In this paper, the molecules are not modeled as Lennard-Jones-like particles but as Gay-Berne-like ellipsoids of revolution. The question arises of how the SLLOD-like modification of particle dynamics extends to the rigid body dynamics. This question has been addressed for a particular case of rigid body dynamics in [33, 34] and for the general case in [35]. The total system of the time evolution equations composed of Eqs. (10)-(18) coupled to extended hydrodynamical equations is shown again to represent a Hamiltonian system. The stress tensor arising in the analysis (*i.e.* the stress tensor that is compatible with the molecular dynamics (10)-(18) - for other type of consideration of the stress tensor see [36, 37]) is given by the standard formula; see Eq. (35). The state variables of extended hydrodynamics are $\rho(\mathbf{r})$, $\mathbf{u}(\mathbf{r})$, $\underline{\mu}(\mathbf{r})$, and $\underline{\underline{c}}(\mathbf{r})$, denoting respectively the mass density field, velocity field, angular momentum field and conformation tensor field. If it is assumed that the two last fields $\underline{\mu}$, $\underline{\underline{c}}$ evolve in time much faster than the classical hydrodynamics fields ρ, $\mathbf{u}$ then one finds that $\underline{\mu} = K\underline{\Omega}$, where $\underline{\Omega} = \mathbf{rot}\,\mathbf{u}$ is the vorticity field and K is a constant. In this paper we put $K = 1$; a detailed analysis of this coefficient will be given elsewhere. The SLLOD-like extension of Eqs. (10)-(18) becomes then,

$$\dot{\mathbf{r}}_i = \frac{\mathbf{P}_i}{m_i} + \mathbf{r}_i \cdot \underline{\underline{\chi}}, \tag{28}$$

$$\dot{\mathbf{P}}_i = \mathbf{F}_i^r - \mathbf{P}_i \cdot \underline{\underline{\chi}} - \alpha_1 \mathbf{P}_i, \tag{29}$$

$$\dot{\hat{\mathbf{p}}}_i = -\hat{\mathbf{p}}_i \times \underline{\omega}_i - \hat{\mathbf{p}}_i \times \underline{\Omega} - \lambda_i \hat{\mathbf{p}}_i, \tag{30}$$

$$\dot{\mathbf{M}}_i = -\mathbf{M}_i \times \underline{\omega}_i + \hat{\mathbf{p}}_i \times \mathbf{F}_i^p - \mathbf{M}_i \times \underline{\Omega} - \alpha_2 \mathbf{M}_i, \tag{31}$$

$$\alpha_1 = \frac{\sum_{i=1}^N \mathbf{P}_i \cdot (\mathbf{F}_i^r - \mathbf{P}_i \cdot \underline{\underline{\chi}})}{\sum_{i=1}^N \mathbf{P}_i \cdot \mathbf{P}_i}, \tag{32}$$

$$\alpha_2 = \frac{\sum_{i=1}^N \underline{\omega}_i \cdot (\hat{\mathbf{p}}_i \times \mathbf{F}_i^p - \mathbf{M}_i \times \underline{\omega}_i - \mathbf{M}_i \times \underline{\Omega})}{\sum_{i=1}^N \underline{\omega}_i \cdot \mathbf{M}_i}, \tag{33}$$

$$\lambda_i = -\frac{\hat{\mathbf{p}}_i \cdot (\hat{\mathbf{p}}_i \times \underline{\omega}_i + \hat{\mathbf{p}}_i \times \underline{\Omega})}{\hat{\mathbf{p}}_i \cdot \hat{\mathbf{p}}_i}. \tag{34}$$

The Hamiltonian nature of this dynamics proven in [35] guarantees again that all the well known results about SLLOD dynamics (see the previous paragraphs) are valid. It can be also verified immediately that if $\dot{\gamma} = 0$ then Eqs. (28)-(34) reduce to Eqs. (10)-(18).

The equations are solved by the Gear 5 value method using time steps of between 0.00125 for high and 0.0025 for low shear rates. The original SLLOD equations for spherically symmetric particles are a particular case of Eqs. (28)-(34), by omitting the variables $\hat{\mathbf{p}}$ and $\mathbf{M}$. Similarly to the classical SLLOD/LE/GAUSS algorithm, the Lees-Edwards (LE) boundary conditions are applied in the direction of the velocity gradient.

The results were easily reproduced at low shear rates, but less so in the viscoelastic regime, where it was necessary either to pre-shear liquid samples or to execute long runs (100,000-200,000 time steps) before taking statistics. As for equilibrium computations, every production run was divided into 10 bins, each 1000-2000 time steps long. Bin averages were calculated and used to obtain an overall average and its standard deviation. Several points were reproduced (see figures in the next section) by imposing shear on fluid samples having different initial molecular configurations.

4.2 Rheological and microstructural results

The viscometric functions, such as the two shear viscosities (η_1, η_2), the normal stress differences (N_1, N_2), and the hydrostatic pressure (P) are extracted from the stress tensor $\underline{\underline{\sigma}}$ [38],

$$\sigma_{\alpha\beta}(\dot{\gamma}) = -\frac{1}{V} < \sum_{i=1}^{N} \frac{P_{i\alpha}P_{i\beta}}{m_i} + \sum_{i=1}^{N-1} \sum_{j>i}^{N} r_{ij\alpha}F^{r}_{ij\beta} >_t, \tag{35}$$

according to,

$$\eta_1(\dot{\gamma}) \quad = \quad \frac{\sigma_{xy}(\dot{\gamma})}{\dot{\gamma}}, \tag{36}$$

$$\eta_2(\dot{\gamma}) = \frac{\sigma_{yx}(\dot{\gamma})}{\dot{\gamma}}, \quad (37)$$
$$N_1(\dot{\gamma}) = \sigma_{xx}(\dot{\gamma}) - \sigma_{yy}(\dot{\gamma}), \quad (38)$$
$$N_2(\dot{\gamma}) = \sigma_{yy}(\dot{\gamma}) - \sigma_{zz}(\dot{\gamma}), \quad (39)$$
$$P(\dot{\gamma}) = -\frac{1}{3}(\sigma_{xx}(\dot{\gamma}) + \sigma_{yy}(\dot{\gamma}) + \sigma_{zz}(\dot{\gamma})). \quad (40)$$

Where V denotes the total volume of the primitive cube.

The stress tensor is nonsymmetric in general for molecular fluids when molecules interact according to binary molecule-molecule potential rather than according to bead-bead intermolecular potentials; for the latter, $\underline{\underline{\sigma}}$ is strictly symmetric. We recall that the stress tensor can be asymmetric if the total angular momentum of the fluid is a sum of $\mathbf{r} \times \mathbf{u}$ and the angular momentum of the particles suspended in the fluid (see for example [39, 40]). We observe that a degree of asymmetry is related to the magnitude of the imposed shear rate; in the limit of small shear rates $\underline{\underline{\sigma}}$ is always symmetric on the average.

In Tab. 1 we have collected the conformation tensor results for the two fluid densities. The ellipses, defined by the largest and the second largest eigenvectors of $\underline{\underline{c}}$ are drawn below the table to facilitate the perusal of the conformation stress data. In this sketch, inclination of each ellipse corresponds to an angle between the major axis of an ellipse and the direction of flow (x). This angle is acute at low shear rates. At around $\dot{\gamma} = 4.5$, the major and minor axes of the ellipses become similar in length, indicating that an isotropic phase, transitory with respect to the shear rate, has been formed; see $\dot{\gamma} = 4.5$ for $\rho = 0.25$ and $\dot{\gamma} = 5$ for $\rho = 0.4$ below Tab. 1. At even higher shear rates, the ellipses become preferentially oriented normally to flow along the neutral direction z, which is also the direction of the vorticity.

Other indicators, such as the total potential energy (Fig. 8a), hydrostatic pressure (Fig. 8b), and the structure factor (Fig. 9) exhibit features that point to the same structural rearrangements as the conformation tensor ellipses. As a result of the initial alignment of ellipsoids (for $0.2 < \dot{\gamma} < 2$) Φ, P and S_t monotonically increase but only the structure factor reaches a maximum for $\dot{\gamma} \simeq 2 - 3$, depending on the fluid density. If the shear rate is increased the structure factor declines substantially indicating the formation of an isotropic-like phase (S_t does not reach the truly isotropic plateau). Neither Φ nor P passes through a maximum, rather their increase is accelerated with the shear rate. The fluid is able to support such a rapid ascent in the total potential energy and in the hydrostatic pressure only until $\dot{\gamma} = 5$ before being forced to reorganize. Initially, this reorganization leads to lower Φ and P but soon both variables resume their upward trends that coincide with the shear thinning region of the viscosity curves (Fig. 10a).

The Newtonian viscosity of the liquid crystal polymers in solutions increases with the concentration of particles to reach a maximum at the phase transition between that isotropic and nematic phases. Our NEMD results collected only for two particle number density points tend to support this laboratory observation. Namely, the Newtonian viscosity for the higher density fluid $\rho = 0.4$ is three times higher than for the lower density fluid $\rho = 0.25$ (Fig. 10a). At around $\dot{\gamma} = 1$, the stress tensor becomes nonsymmetric, and the viscosity curves for the two densities bifurcate with respect to the shear rate. The difference between the upper (η_1) and the lower (η_2) branches of two viscosity curves is most significant during the initial structure formation. In the viscoelastic region, the distance between η_1 and η_2 appears not to depend on the shear rate. Both N_1 and N_2 are not zero, even for slightly anisotropic liquids (Figs. 10 b&c). The fact the N_1 for $\rho = 0.25$ is initially negative may be related to the particle number density. We have also observed the negative first normal stress difference for very small vibrating dumbbells [7].

In addition to the conformation tensor, the spatial ordering of the rigid bodies is probed by the diagonal components of the resolved (tensorial) pair radial distribution functions (Figs. 11 a-c; see Eq. 23), and by the longitudinal as well as the transverse pair radial distribution functions (Figs. 11 d-e; see Eqs. 24&25). As oppose to the dumbbell fluids [7], where the solid-like phase is formed in the neutral direction z and in the direction of the velocity gradient x, for rigid bodies all three diagonal components of the pair radial distribution function are fluid-like. At the low shear rates and for both fluid densities, the radial distributions functions are structureless. However, all three radial components of $\underline{\underline{g}}$ have well defined peaks at the position of the first coordination neighbors, when the microstructure oriented along the vorticity axis is built, This microstructure has features very unlike the nematic phase as it follows from comparison between Figs. 12 d&e and **Figs.** 6 c&d. As a matter of fact the distance between peaks for two computer experiment ($\rho = 0.4$, $\dot{\gamma} = 9$ and $\rho = 0.25$, $\dot{\gamma} = 10$) is close to 1.7 what would indicate the appearance of the shear-induced smectic-like phase (see also snapshots 12 d&c).

If we compare the microstructural results for rigid bodies (Fig. 12) with those for vibrating dumbbells (Fig. 1 of ref. [7]) it is immediately evident that for large shear rates dumbbells are not aligned on the average in the direction of vorticity. There is a small tendency on the part of dumbbells to build local cross-string spanning structure which consists of two strings with each bead of every dumbbell belonging to a separate string. The structure is transient with respect to the shear rate, it appears for $\dot{\gamma} \simeq 8$, never pervades the entire primitive cube and disappears for $\dot{\gamma} > 10$. It is possible that the vibrational degree of freedom may contribute sufficiently to the nature of vibrating dumbbells to induce their more disorderly behavior. It is also important to realize that rigid bodies that the dynamics of rigid bodies includes rotations around the molecular axis, whereas for dumbbells the rotation around the bead-bead axis is not taken into account. It is know that at the macroscopic level, rigid fibers may in fact align themselves parallel to the vorticity axis.

In case of atomic liquids, smaller assemblies (*e.g.* 256) of particles exhibit well defined phase transition zones between isotropic and ordered phases [41]. No strings are built for rigid ellipsoids of revolution neither in the direction of flow (Figs. 12 f-j) nor in the direction of vorticity (Figs. 12 a-d). A limited volume of such systems, allows no simultaneous existence of the two phases, leading to oscillations between purely amorphous and string arrangements [6]. The dramatic change in the fluid structure around $\dot{\gamma} = 5$ for the higher density fluid manifested by a sudden decrease in η_1 and η_2 may reflect not only the physical response of the system to shearing but also small particle number and periodic boundary effects.

It has been known for some time [42] that in case of dilute gases the macroscopic transport processes are related to the changes in the velocity distribution function. For fluids of rigid bodies and at increasing shear rates, distributions of the translational velocities display three distinct trends, shown with arrows in Fig. 13. Firstly, more translational energy is stored in tails of the distributions at the higher shear rates than at the equilibrium. Since within the shear induced microstructure molecules are not as free to move as within the isotropic fluid, maxima of the distribution curves are shifted to low velocities. We have observed similar distortion in the translational velocity distribution, marked by the gradual disappearance of the Maxwellian distribution (solid line in Fig. 13), for vibrating dumbbells [7].

In addition to viscometric functions that show shear thinning and large normal stress differences, the viscoelasticity of fluid composed of rigid bodies is manifested by the stress overshoot (Fig. 14a). The stress overshoot is a transient effect related to the imposition of the Heaviside shear rate function on the fluid sample which was initially at equilibrium; stress overshoot is normally plotted in terms of the transient viscosity η^+, for constant shear rate flows. According to our results the overshoot is more pronounced for σ_{xy} than σ_{yx} and exists only for shear rates within the shear thinning region; for example, there is no shear overshoot for $\dot{\gamma} = 1$, as shown in inset in Fig. 14a. The diagonal elements of the stress tensor (Fig. 14b) also display an initial overshoot. The fact that both σ_{xx} and σ_{zz} equilibrate to the same value is reflected in approximate equality: $N_1 \simeq -N_2$ which holds at the higher shear rates (see Fig. 10b). The overshoot in the diagonal and off-diagonal elements of $\underline{\underline{\sigma}}$ is the macroscopic manifestation of microstructural rearrangement, which are evidenced in plots of the instantaneous structure factor S and components of the conformation tensor $\underline{\underline{c}}$ (Fig. 14c). Only when the ordering of the liquid structure has been accomplished the viscosity begins to fluctuate around its steady state value.

5 Conclusions

This work presented a nonequilibrium molecular dynamics algorithm obtained from a particular implementation of the nonlinear Onsager-Casimir type equations introduced by Grmela [35] within the context of mesoscopic hydrodynamics of suspensions. The original system of coupled equations was reduced to result in a novel SLLOD-type dynamics of rigid particles.

Together with the data for more elongated molecules [18, 20] the thermodynamic results indicate that the liquid-vapor phase transition for Gay-Berne fluids takes place around $\rho = 0.1$ and below $T = 1$. At this temperature but at much higher densities, the nematic phase is stable. At the lower temperatures, the isotropic phase evolves directly to the smectic phase. Similarly to rigid bodies, dumbbells display the transition between isotropic and nematic phases [24]; in general, the thermodynamic data for dumbbell liquids is very incomplete. At the mesoscopic level of description, the nematic and smectic phases may be conveniently differentiated by probing the fluid structure with longitudinal pair correlation function $g_{||}(r_{||})$.

When a fluid composed of rigid bodies is subjected to an imposed shear rate an oriented microstructure is formed in the direction of the fluid flow. At the intermediate shear rates, this microstructure is destroyed and replaced by an isotropic-like phase. At even higher shear rates, ellipsoids become oriented along the vorticity axis. These structural reorganizations have not been observed by us [7] (for vibrating dumbbells) nor by Rudisill and Cummings [5] (for rigid rotors and vibrating dumbbells).

Trends in macroscopic (viscometric functions, pressure) and mesoscopic (*e.g.* pair distribution functions) variables are all related to changes in the underlying microstructure. Even the transient phenomena, such as the stress overshoot, are traced to rearrangements taking place among the rigid bodies. Once the new order is established at the microscopic level, the transient effects die out; no stress overshoot is observed at low shear rates when the shear induced orientation of particles is small.

As oppose to the other investigation [5], in our studies all molecular liquids exhibit shear thinning. The onset of shear thinning appears for the same shear rate, for both types of fluids (compare Fig. 5b of ref. [6] with Fig. 10a for $\rho = 0.4$). However, the shapes of viscometric function are unlike for rigid bodies and vibrating dumbbells. Finally, the stress tensor is nonsymmetric for systems of rotating rigid bodies.

The microstructural and rheological predictions for ellipsoids of revolution are quite different from those obtained for vibrating dumbbells in spite of the fact that parameters in the Gay-Berne potential have been chosen in such a way that the potential used in the dumbbell dynamics [6, 7] and the Gay-Berne potential are very similar (Figs. 1&2). The difference in rheological predictions reflects the difference in dynamics between dumbbells and ellipsoids of revolution. Also thermostating has been introduced differently for dumbbells and for ellipsoids of revolution. The total kinetic energy is thermostated for vibrating dumbbells, whereas for rigid bodies the translational and rotational kinetic energies are thermostated separately [7]. The method of thermostating is perhaps not very important since for thermostated vibrating dumbbell liquids the kinetic energy is partitioned approximately equally between the translational and internal energy reservoirs [6]; there is no equilibration among different modes in either reservoir, especially at the higher shear rates.

6 Acknowledgments

This investigation would not be possible without the financial support from the *Natural Sciences and Engineering Research Council of Canada* and *Fonds pour la formation de chercheurs et l'aide à la recherche* of the Province of Québec. Additional funding for the project has been provided by the Governments of Belgium and Québec within the program *Coopération universitaire Communauté française de Belgique/Québec.* Discussions with Claude Cohen of Cornell University are acknowledged with gratitude.

References

[1] R. Edberg, G. P. Morriss, and D. J. Evans, J. Chem. Phys. **86**, 4555 (1987).

[2] G. P. Morriss, P. J. Daivis, and D. J. Evans, J. Chem. Phys. **94**, 7420 (1991).

[3] R. Rowley and J. F. Ely, Mol. Phys. **72**, 831 (1991).

[4] A. Berker, S. Chynoweth, U. C. Klomp, and Y. Michopoulos, J. Chem. Soc., Faraday Trans. **88**, 1719 (1992).

[5] J. W. Rudisill and P. T. Cummings, Rheol. Acta **30**, 33 (1991).

[6] B. Z. Dlugogorski, M. Grmela, and P. J. Carreau, J. Non-Newt. Fl. Mech. (in press) (1993).

[7] B. Z. Dlugogorski, M. Grmela, and P. J. Carreau, J. Non-Newt. Fl. Mech. (in press) (1993).

[8] J. W. Rudisill and P. T. Cummings, Mol. Phys. **33**, 1757 (1977).

[9] J. D. Johnson, M. S. Show, and B. L. Holian, J. Chem. Phys. **80**, 1279 (1984).

[10] J. Barojas, D. Levesque, and B. Quentrec, Phys. Rev. A **7**, 1092 (1973).

[11] P. S. Y. Cheung and J. G. Powles, Mol. Phys. **30**, 921 (1975).

[12] D. J. Evans, Nonequilibrium molecular dynamics, in *Molecular-Dynamics Simulation of Statistical-Mechanical Systems*, edited by C. G. and W. Hoover, pages 221–240, North-Holland, Oxford, 1986.

[13] J. Corner, Proc. Royal Soc. Lond. **A192**, 275 (1948).

[14] B. J. Berne and P. Pechukas, J. Chem. Phys. **56**, 4213 (1972).

[15] J. Kushick and B. J. Berne, J. Chem. Phys. **64**, 1362 (1976).

[16] J. G. Gay and B. J. Berne, J. Chem. Phys. **74**, 3316 (1981).

[17] D. J. Adams, G. R. Luckhurst, and R. W. Phippen, Mol. Phys. **61**, 1575 (1987).

[18] E. de Miguel, L. F. Rull, M. K. Chalam, and K. E. Gubbins, Mol. Phys. **71**, 1223 (1990).

[19] E. de Miguel, L. F. Rull, M. K. Chalam, and K. E. Gubbins, Mol. Phys. **72**, 593 (1991).

[20] E. de Miguel, L. F. Rull, M. K. Chalam, and K. E. Gubbins, Mol. Phys. **74**, 405 (1991).

[21] E. de Miguel, L. F. Rull, M. K. Chalam, and K. E. Gubbins, Phys. Rev. A **45**, 3813 (1992).

[22] S. Hess, D. Frenkel, and M. P. Allen, Mol. Phys. **74**, 765 (1991).

[23] M. P. Allen and D. Frenkel, Phys. Rev. Lett. **58**, 1748 (1987).

[24] S. H. Lee, H. S. Kim, and H. Pak, J. Chem. Phys. **97**, 6933 (1992).

[25] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford, 1987.

[26] D. D. Holm, J. E. Marsden, T. Ratiu, and A. Weinstein, Phys. Rep. **123**, 1 (1985).

[27] S. Hess, Flow alignment of a colloidal solution which can undergo a transition from the isotropic to the nematic phase (liquid crystal), in *Electro-Optics and Dielectrics of Macromolecules and Colloids*, edited by B. R. Jennings, pages 321–327, Plenum, 1979.

[28] M. P. G. Sudarshan and N. Mukunda, *Classical Dynamics: A Modern Review*, John Wiley & Sons, New York, 1971.

[29] D. J. Evans, W. G. Hoover, B. C. Failor, B. Moran, and A. J. C. Ladd, Phys. Rev. A **28**, 1016 (1983).

[30] L. Verlet, Phys. Rev. **159**, 98 (1967).

[31] M. P. Haile, *Molecular Dynamics Simulations: Elementary Methods*, John Wiley & Sons, New York, 1992.

[32] M. Grmela, Phys. Lett. A **174**, 59 (1993).

[33] D. J. Evans, Mol. Phys. **34**, 317 (1977).

[34] R. Edberg, D. J. Evans, and G. P. Morriss, Mol. Phys. **62**, 1357 (1987).

[35] M. Grmela, Proc. Royal Soc. Lond. A (submitted) (1993).

[36] M. P. Allen, Mol. Phys. **52**, 705 (1984).

[37] A. J. C. Ladd, Mol. Phys. **53**, 459 (1984).

[38] J. H. Irving and J. G. Kirkwood, J. Chem. Phys. **18**, 817 (1950).

[39] J. S. Dahler and L. E. Scriven, Nature **192**, 36 (1961).

[40] A. C. Eringen, *Mechanics of Continua*, Wiley, New York, 1967.

[41] D. M. Heyes, J. Chem. Soc., Faraday Trans. 2 **82**, 1365 (1986).

[42] W. Loose and S. Hess, Physica A **174**, 47 (1991).

Table 1: Components of the conformation tensor; numbers in subscript indicate one standard deviation. The diagonal elements of $\underline{\underline{c}}$, namely c_{xx}, c_{yy} and c_{zz} define the average orientation of particles, whereas trends in these variables are indicative of structural rearrangements taking place in the fluid at the molecular level.

ρ	$\dot{\gamma}$	c_{xx}	c_{xy}	c_{xz}	c_{yy}	c_{yz}	c_{zz}
0.25	1.0	0.3311_{93}	0.0547_{77}	-0.0028_{99}	0.3066_{70}	0.0002_{63}	0.3623_{82}
0.25	2.0	0.378_{12}	0.0090_{84}	0.000_{13}	0.2887_{91}	0.0002_{88}	0.3330_{96}
0.25	3.0	0.442_{12}	0.008_{62}	-0.003_{17}	0.2419_{99}	-0.0007_{37}	0.316_{18}
0.25	4.5	0.384_{16}	0.0227_{53}	0.002_{11}	0.215_{12}	0.0015_{68}	0.401_{27}
0.25	5.5	0.306_{94}	-0.062_{43}	-0.027_{62}	0.1873_{92}	-0.0000_{45}	0.506_{18}
0.25	7	0.226_{11}	-0.039_{23}	0.026_{42}	0.1551_{79}	0.0004_{45}	0.618_{19}
0.25	10	0.134_{12}	-0.065_{20}	0.027_{30}	0.1056_{96}	-0.0011_{32}	0.760_{21}
0.4	1	0.4090_{87}	0.1502_{74}	-0.0006_{99}	0.2761_{40}	0.002_{66}	0.3150_{96}
0.4	1.5	0.483_{16}	0.1814_{80}	0.005_{13}	0.2495_{57}	0.007_{11}	0.268_{15}
0.4	2	0.5423_{91}	0.182_{58}	-0.003_{14}	0.2188_{40}	0.0024_{63}	0.239_{87}
0.4	3	0.545_{17}	0.087_{10}	-0.015_{23}	0.1790_{70}	0.001_{10}	0.284_{21}
0.4	4	0.415_{15}	0.030_{50}	-0.0035_{87}	0.1641_{97}	0.0023_{59}	0.421_{24}
0.4	5	0.3143_{60}	0.0124_{39}	0.0044_{59}	0.1499_{55}	-0.0007_{72}	0.536_{11}
0.4	6	0.1744_{64}	-0.0049_{26}	0.031_{12}	0.0856_{49}	-0.0044_{52}	0.740_{11}
0.4	7	0.1343_{70}	-0.0079_{20}	-0.0251_{54}	0.0712_{53}	0.0039_{26}	0.794_{12}
0.4	9	0.0726_{48}	-0.0124_{17}	0.0362_{85}	0.0401_{41}	-0.0204_{50}	0.8873_{84}

Figure Captions

- Figure 1. Comparison between the site-site potential for rigid rotor molecules, whose bead-bead extensions were set to 0.9σ, and the Gay-Berne potential for ellipsoids of revolution (Eqs 1-6). All figures are plotted using nondimensional quantities; the asterisks are dropped.

- Figure 2. Graphical visualization of the site-site (a,b) and Gay-Berne (c,d) potentials between two molecules: a,c) both molecules lie on the same plane and their orientational unit vectors, along their respective major axes, are parallel, b,d) same as (a,e) but the unit vectors are perpendicular. In generating the figures, the first molecule was stationary, whereas the second was moved around by translation only. The isolines were projected onto the xy plane to contour the repulsive core.

- Figure 3. The total configurational energy - $\Phi = \sum_{i=1}^{N-1} \sum_{j>i}^{N} \phi_{ij}$ - for a system of 256 ellipsoids as a function of density. Every other point has been replicated, but due to the closeness of the data some replicates are indistinguishable. In this and in the subsequent figures, the error bars are drawn between (an average value minus one standard deviation) and (an average value plus one standard deviation).

- Figure 4. An isotherm ($T = 1$) for a Gay-Berne fluid ($\kappa = 1.9$, $\mathcal{K} = 2.5$). In the inset, de Miguel *et al.* data collected for another Gay-Berne fluid ($\kappa = 3$, $\mathcal{K} = 5$), in the vicinity of the liquid-vapor phase transition; see Fig. 1 in ref. [18]. The pressure was calculated as minus one third of the trace of the stress tensor, exactly as defined in Sec. 4.2. When plotting $P(\rho)$ in the inset ($T = 1$), we have made no adjustment for the fact that our molecules are smaller in size than those in ref. [18]. Such an adjustment, would translate the pressure curve by a certain distance to the right. In any case, there is no indication that the isotherm ($T = 1$) passes through the two phase region. At the higher temperatures, such as $T = 1$, the isotropic phase evolves first to the nematic phase which later, for higher densities, goes to the smectic phase (not shown); compare also with Fig. 15 in ref. [20].

- Figure 5. The order parameter plotted against the density. At the intermediate densities, a line has been drawn approximatively as a guide to the eye.

- Figure 6. (a,b) The diagonal elements of the angular resolution of the pair radial distribution function ($g_{||}(r)$, $g_{\perp}(r)$) according to the instantaneous director $\hat{\mathbf{n}}$. (c,d) The longitudinal ($g_{||}(r_{||})$) and transverse ($g_{\perp}(r_{\perp})$) pair distribution functions. Both $g_{||}(r_{||})$ and $g_{\perp}(r_{\perp})$ are functions of the components of the center-to-center headless vector, which are parallel and perpendicular to $\hat{\mathbf{n}}$ respectively. Thus, in general neither $g_{||}(r_{||})$ nor $g_{\perp}(r_{\perp})$ vanish at zero. Note that in Figs c&d data for $\rho = 0.7$ are plotted in place of $\rho = 0.5$.

- Figure 7. Snapshots of the spatial arrangement of molecules in the isotropic (a, $\rho = 0.4$) and nematic (b, $\rho = 0.675$) phases. The ordinate ($\hat{n}_{||}$) is parallel to the instantaneous director, whereas the abscissa is arbitrary but perpendicular to $\hat{\mathbf{n}}$ ($\hat{n}_{\perp}$). The axes are drawn in units of σ, and the number of molecules is the same (256) for both densities. In 3-D, before projecting onto $\hat{n}_{||}\hat{n}_{\perp}$ plane, each line segment corresponds to the molecular end-to-end distance (1.9σ). For reasons of clarity, in Fig. 12 line segments of length 0.9σ are plotted.

- Figure 8. The total potential energy Φ (a) and pressure P (b) follow the same trends if plotted against the shear rate ($\dot{\gamma}$). For convenience, the equilibrium results for the two densities are drawn on the ordinates; compare with Figs

3&4.

- Figure 9. The average order parameter (S) against the shear rate ($\dot{\gamma}$). The lines have been drawn as guides to the eye. The equilibrium results are shown on the vertical axis.

- Figure 10. The viscometric functions *viz.* the viscosity η (a), the first N_1 (b) and second N_2 (c) normal stress differences plotted against the shear rate $\dot{\gamma}$. In the limit of small shear rates, N_1 and N_2 tend to zero, but the Newtonian viscosity cannot be unequivocally determined.

- Figure 11. The diagonal elements of the tensorial pair radial distribution function $g_{xx}(r)$, $g_{yy}(r)$, $g_{zz}(r)$ (a-c), as well as longitudinal $g_{||}(r_{||})$ (d) and transverse $g_{\perp}(r_{\perp})$ (e) pair distribution functions plotted against the shear rate $\dot{\gamma}$.

- Figure 12. Graphical representation of evolution of the molecular configuration under increasing shear rate. The ordinates are always along the instantaneous direction of $\hat{\mathbf{n}}$, which in most cases (b-i) seems to be very close to the orientation of the system z-axis. The abscissae are perpendicular to $\hat{\mathbf{n}}$ pointing approximatively along either y (b-d) or x (f-j) axes. In (j), the director is normal to the figure. The orientations of $\hat{n}_{||}$ and $\hat{n}_{\perp}$ in space in relation to the system coordinates are indicated at the upper right corner of each graph.

- Figure 13. Distribution (Ξ) of the translational velocities V. The arrows indicate the evolution of the distribution function at increasing shear rates. The equilibrium curve was obtained by plotting

$$\Xi(V) = \frac{4}{\sqrt{\pi}}\, T^{-\frac{3}{2}}\, exp(-\frac{V^2}{T})\, V^2, \tag{41}$$

for nondimensional mass equal to 2 and by setting $T = 1$.

- Figure 14. Formation of the molecular structure and structural relaxation after the imposition of a step shear rate $\dot{\gamma}^+ = 7$ (a-c) and $\dot{\gamma}^+ = 1$ (inset in a) at time zero, in terms of viscosity η (a), normal components of the stress tensor $\underline{\underline{\sigma}}$ (b), and the diagonal elements of the conformation tensor $\underline{\underline{c}}$ (c). Data in Figs b&c were generated for $\rho = 0.4$ and $\dot{\gamma}^+ = 7$. The instantaneous order parameter S is also given in Fig. c.

Figure 1

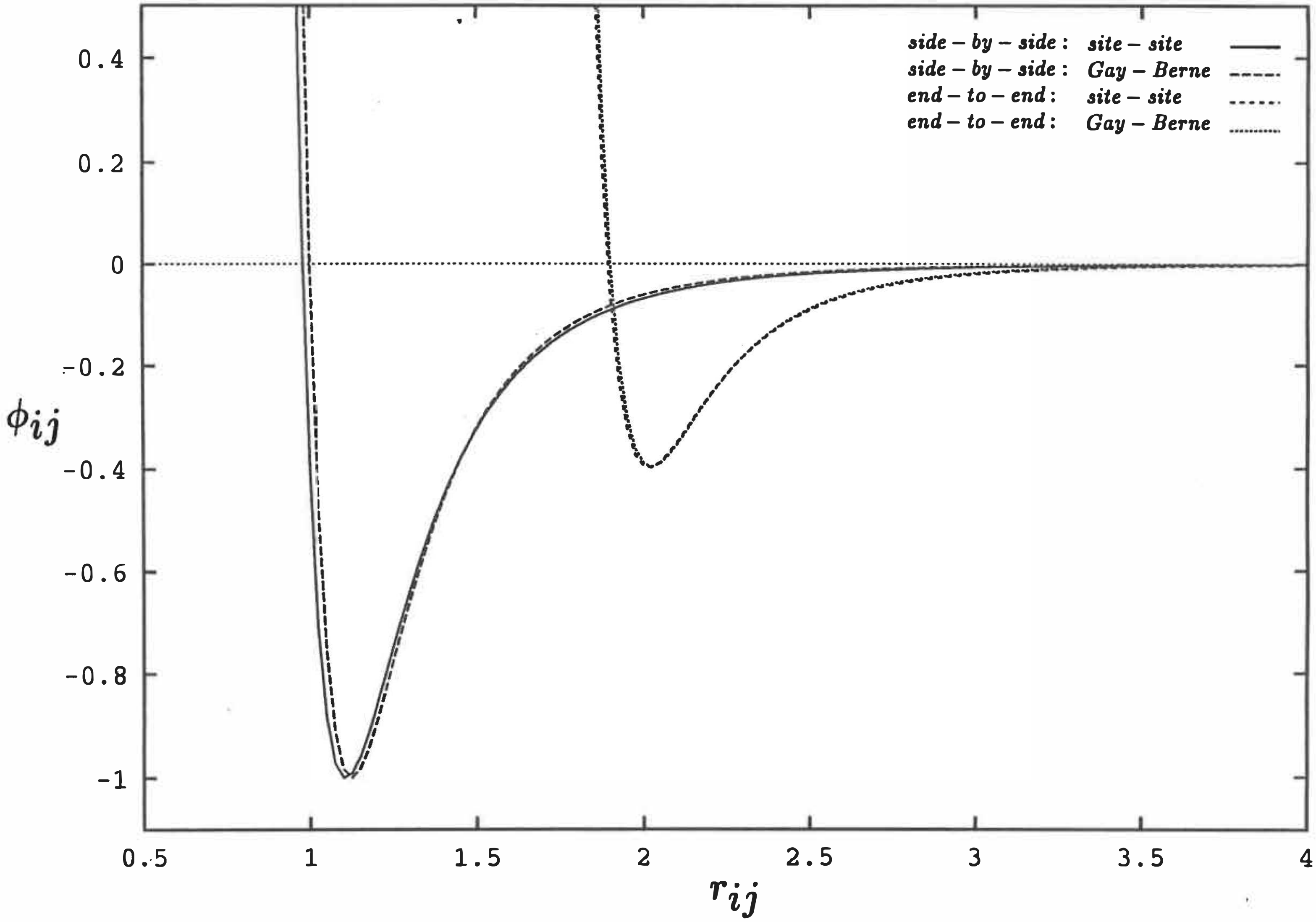

side – by – side : site – site
side – by – side : Gay – Berne
end – to – end : site – site
end – to – end : Gay – Berne
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
ϕ_{ij}
0.5
1
1.5
2
2.5
3
3.5
4
r_{ij}

Figure 2a

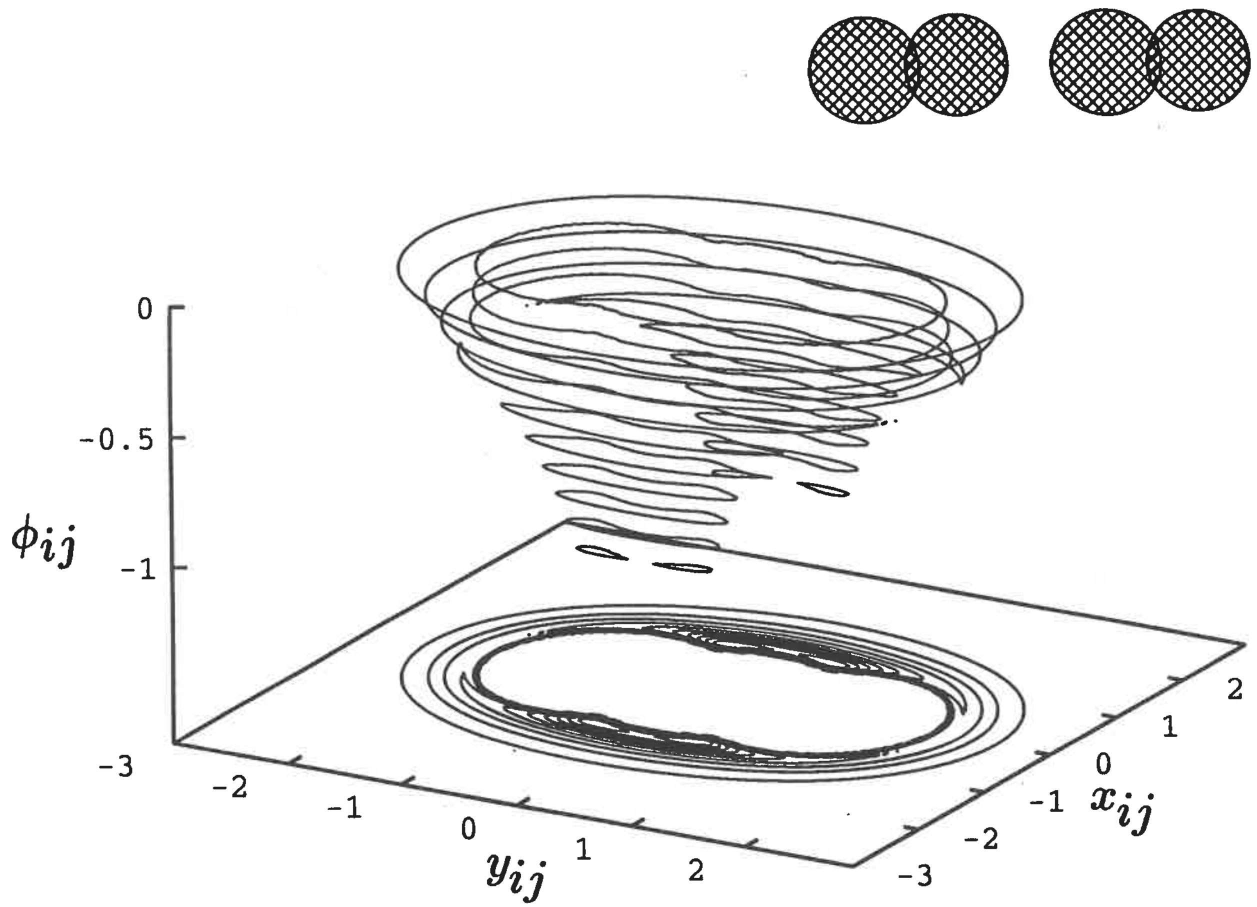


Figure 2b

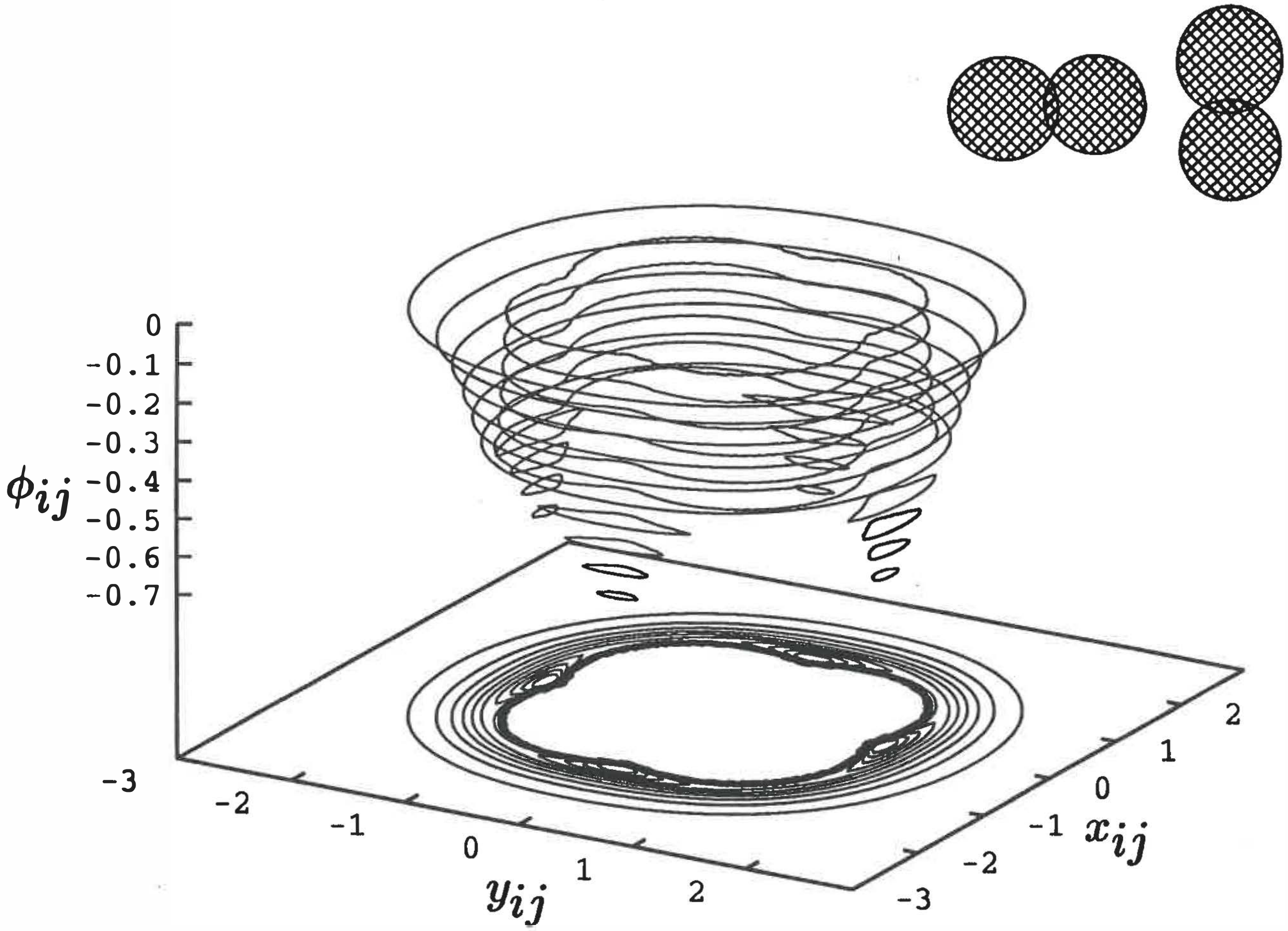


Figure 2c

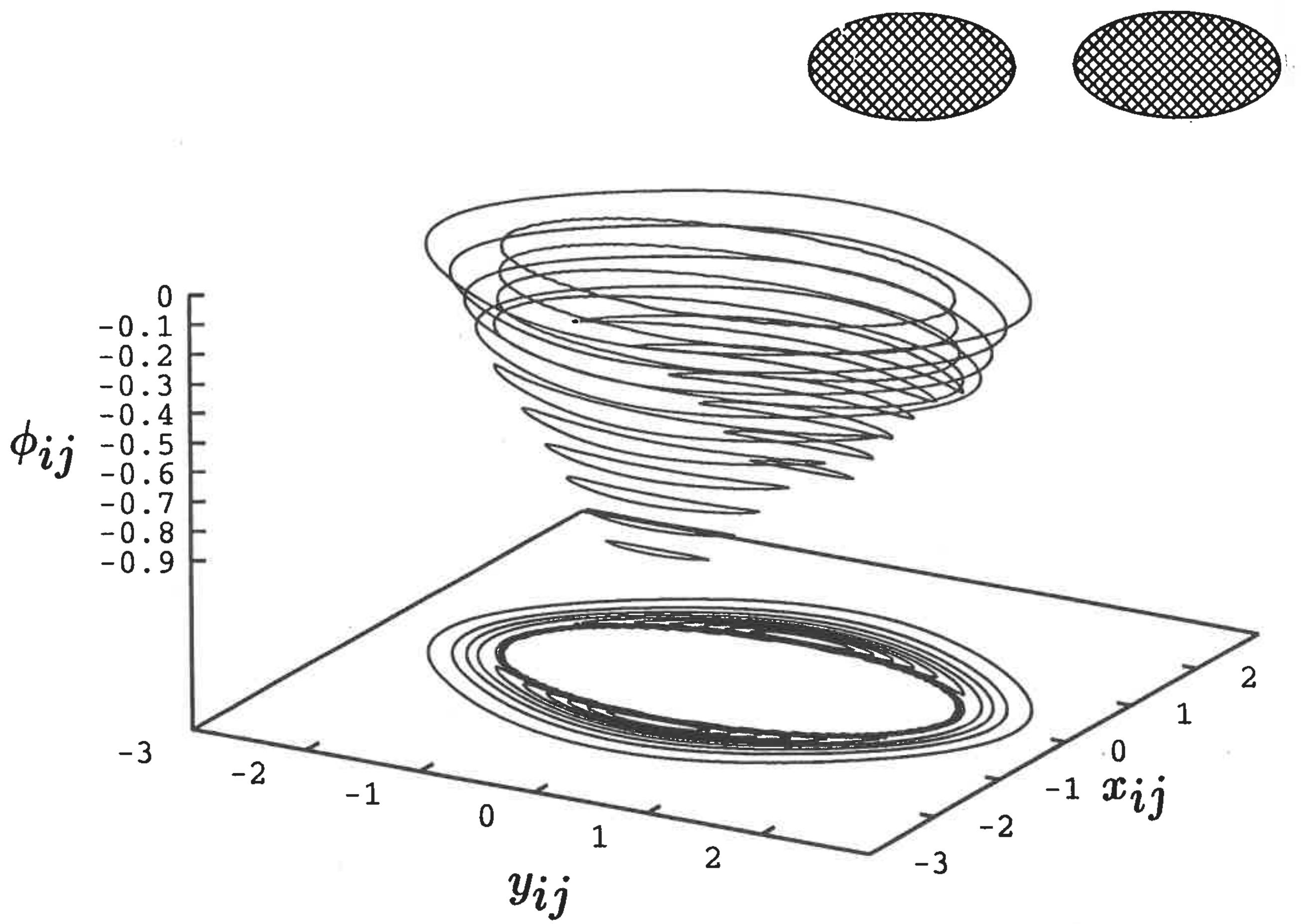


Figure 2d

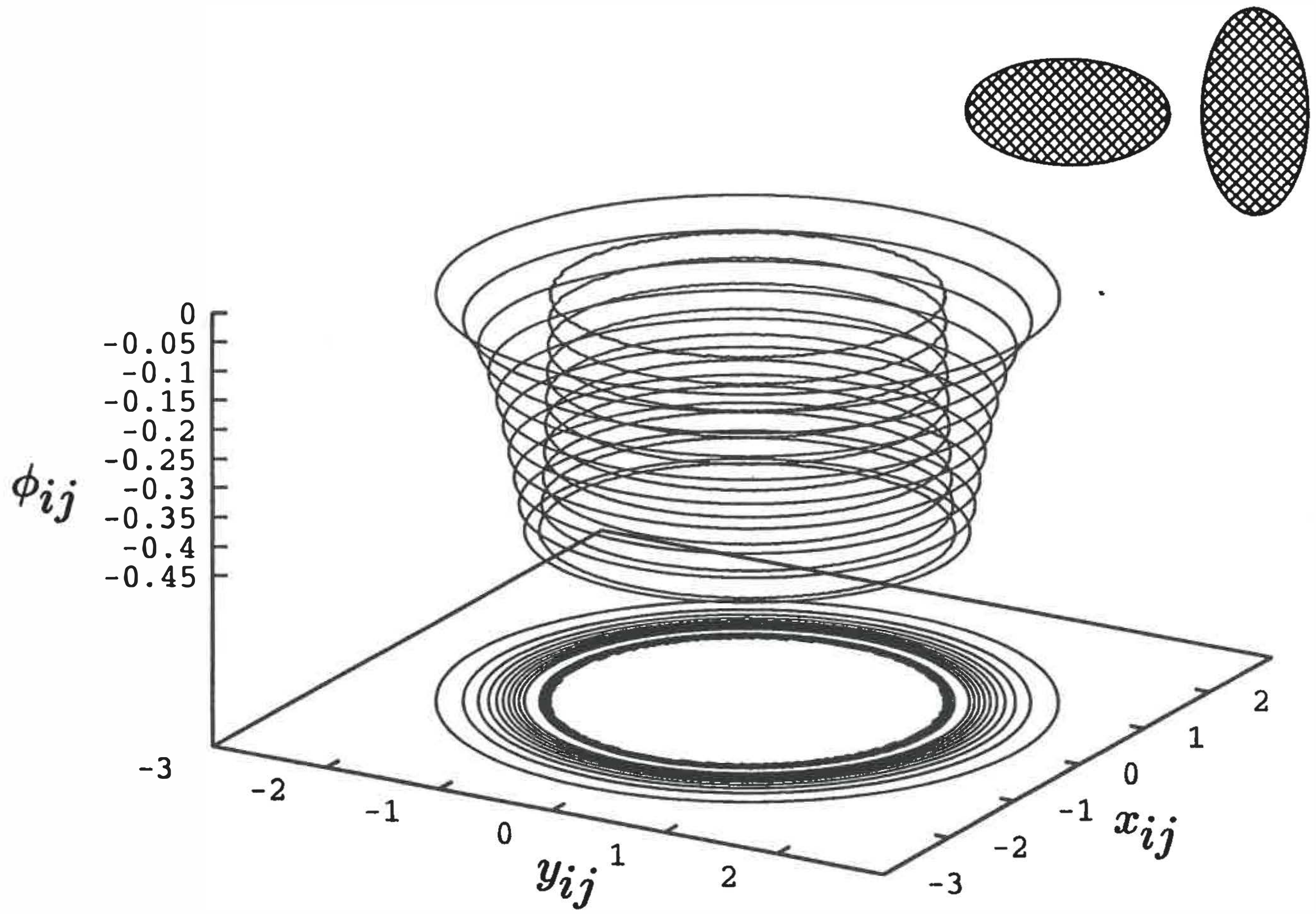


Figure 3

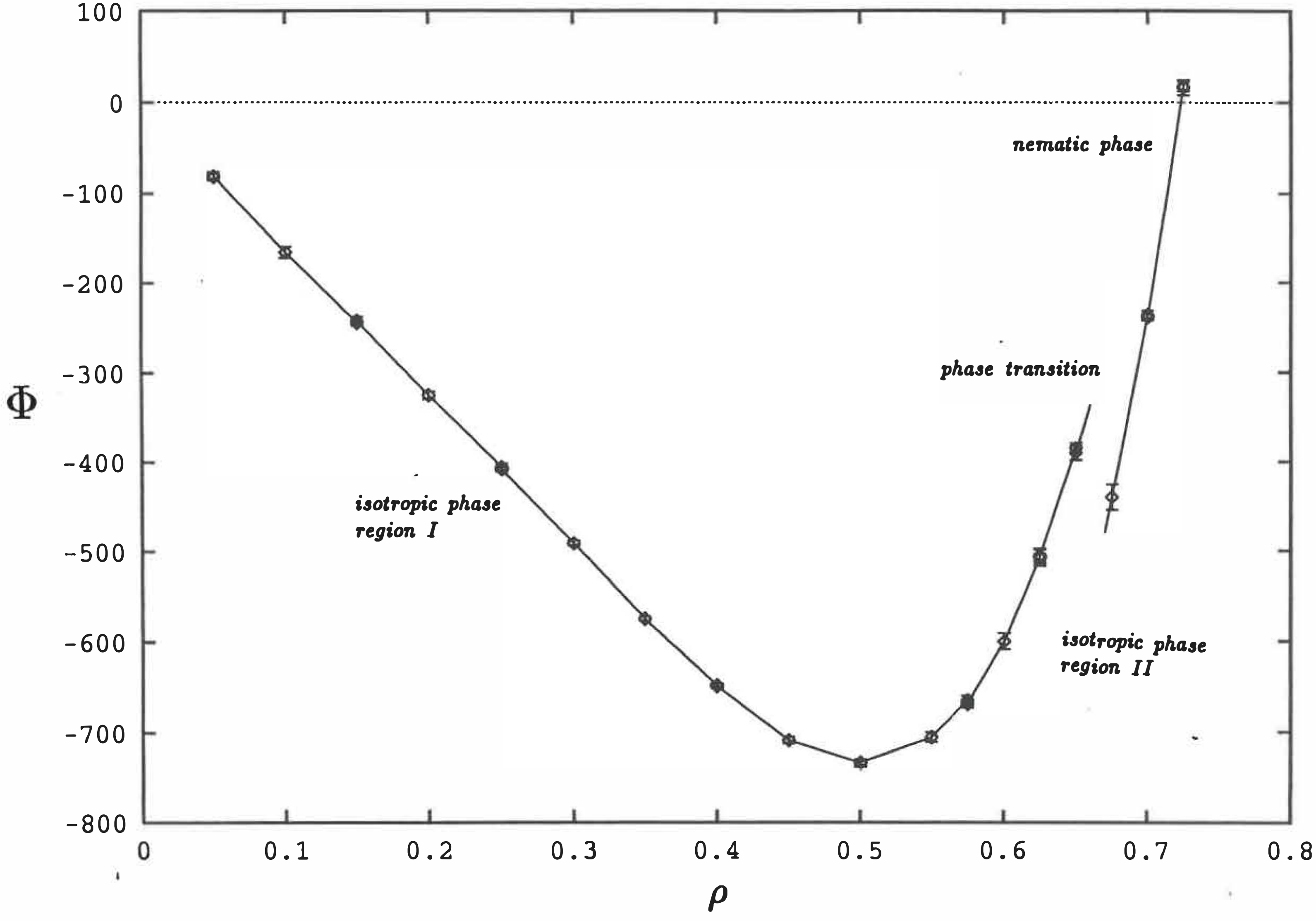
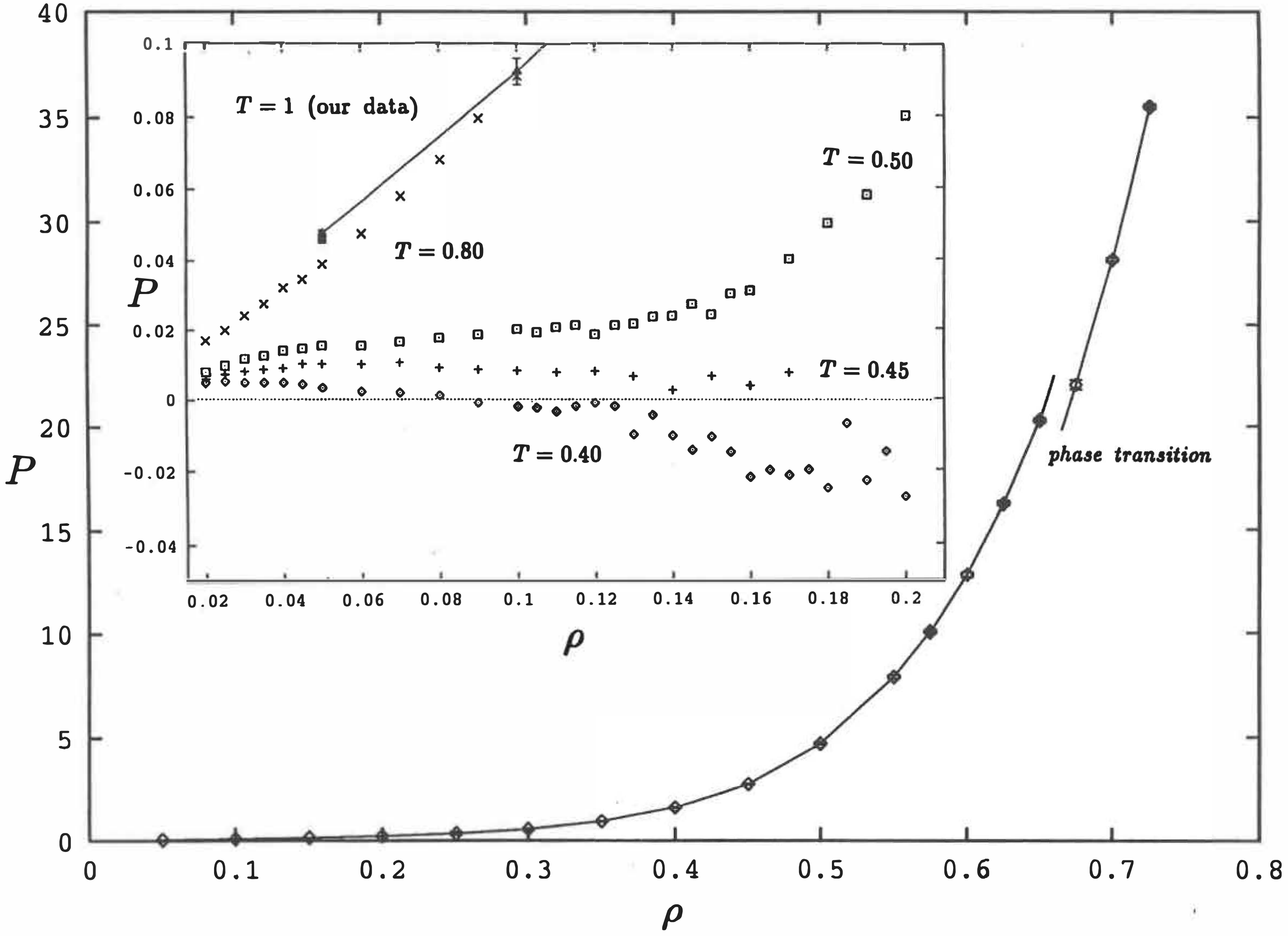

100
0
-100
-200
-300
-400
-500
-600
-700
-800
Φ
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
ρ
nematic phase
phase transition
isotropic phase
region I
isotropic phase
region II

Figure 4



40
35
30
25
20
15
10
5
0
P
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
ρ
phase transition
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
P
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
ρ
$T = 1$ (our data)
$T = 0.80$
$T = 0.50$
$T = 0.45$
$T = 0.40$

Figure 5

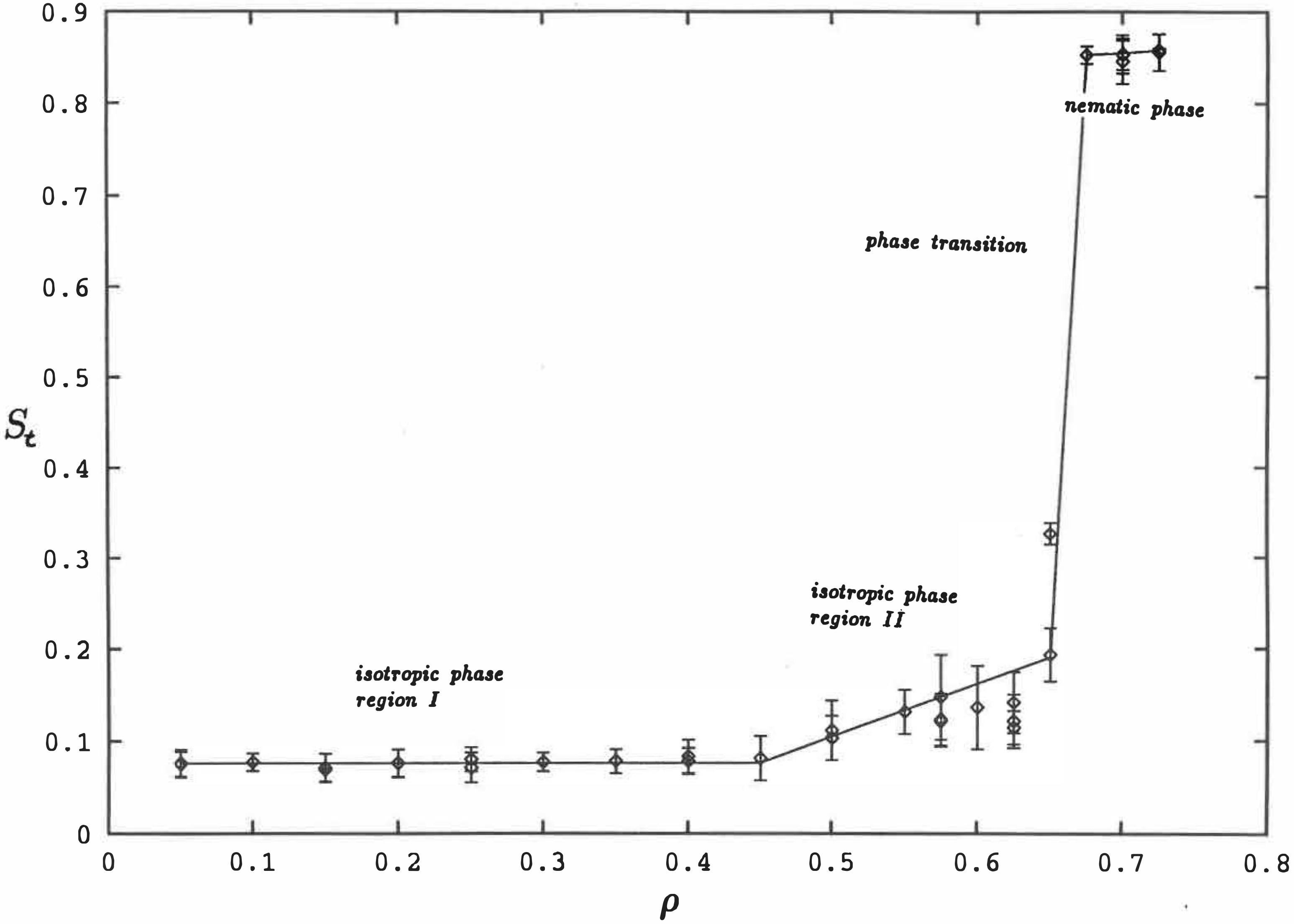

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
S_t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
ρ
nematic phase
phase transition
isotropic phase
region II
isotropic phase
region I

Figure 6a

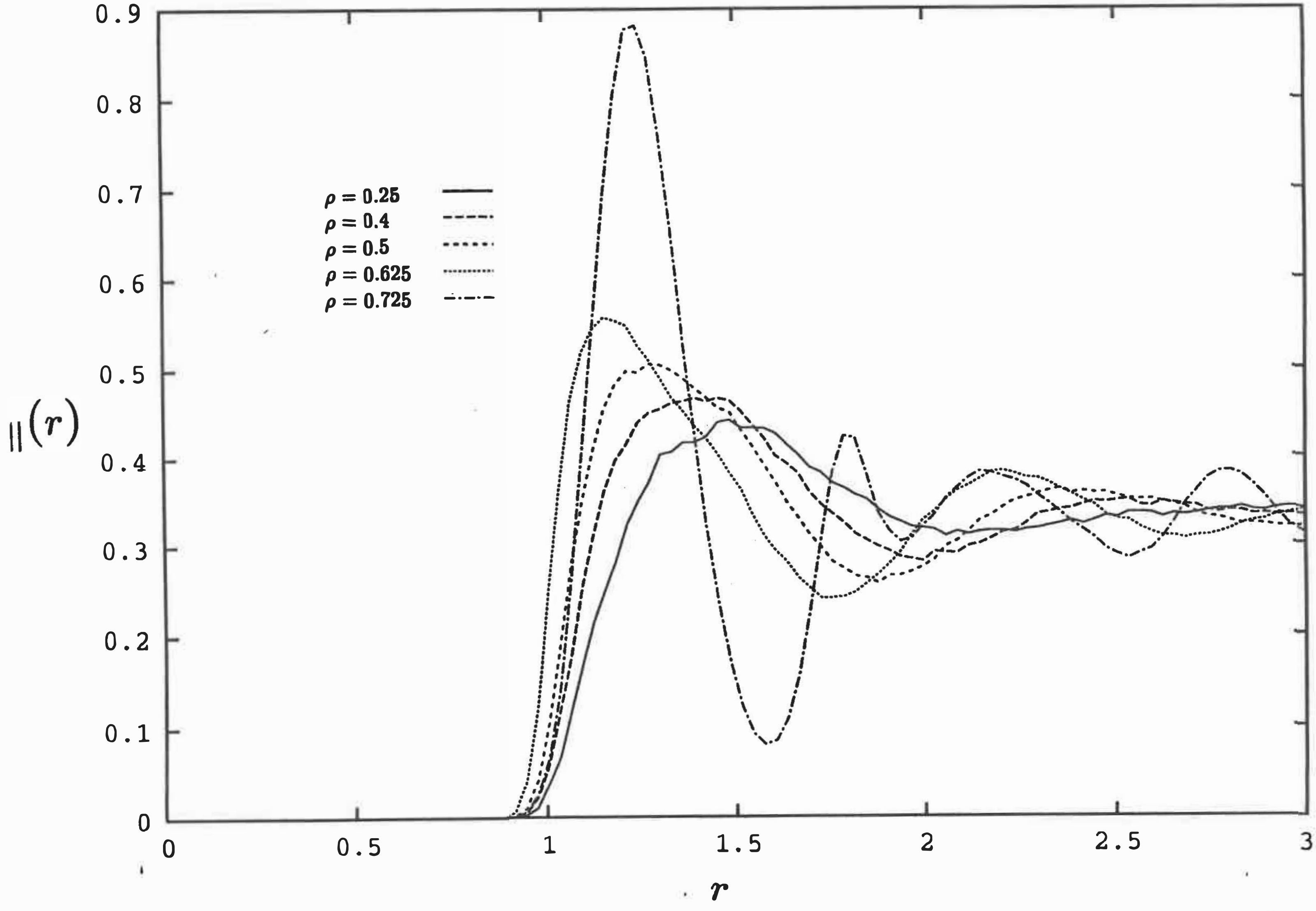

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
$_{||}(r)$
$\rho = 0.25$
$\rho = 0.4$
$\rho = 0.5$
$\rho = 0.625$
$\rho = 0.725$
0
0.5
1
1.5
2
2.5
3
r

Figure 6b

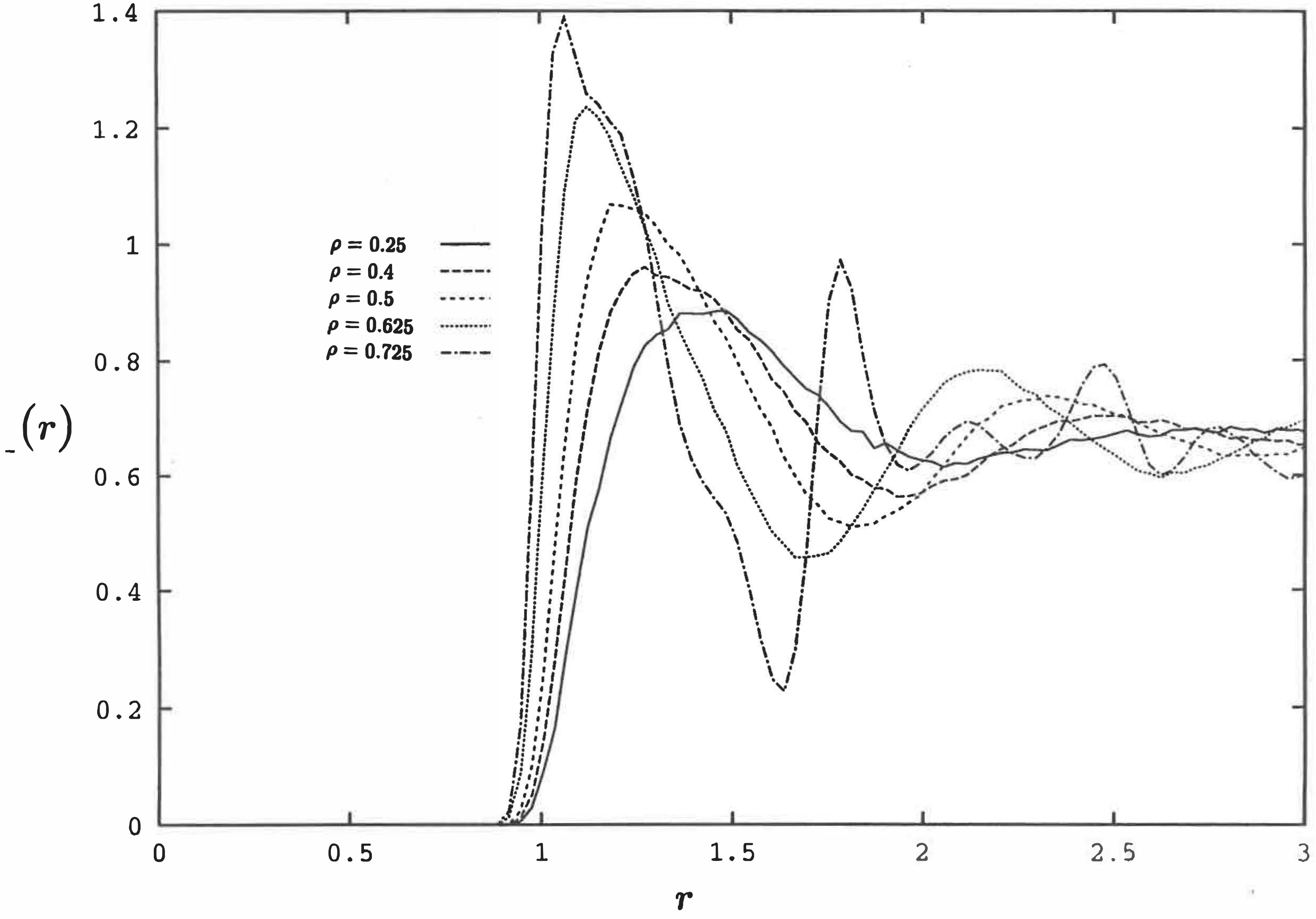
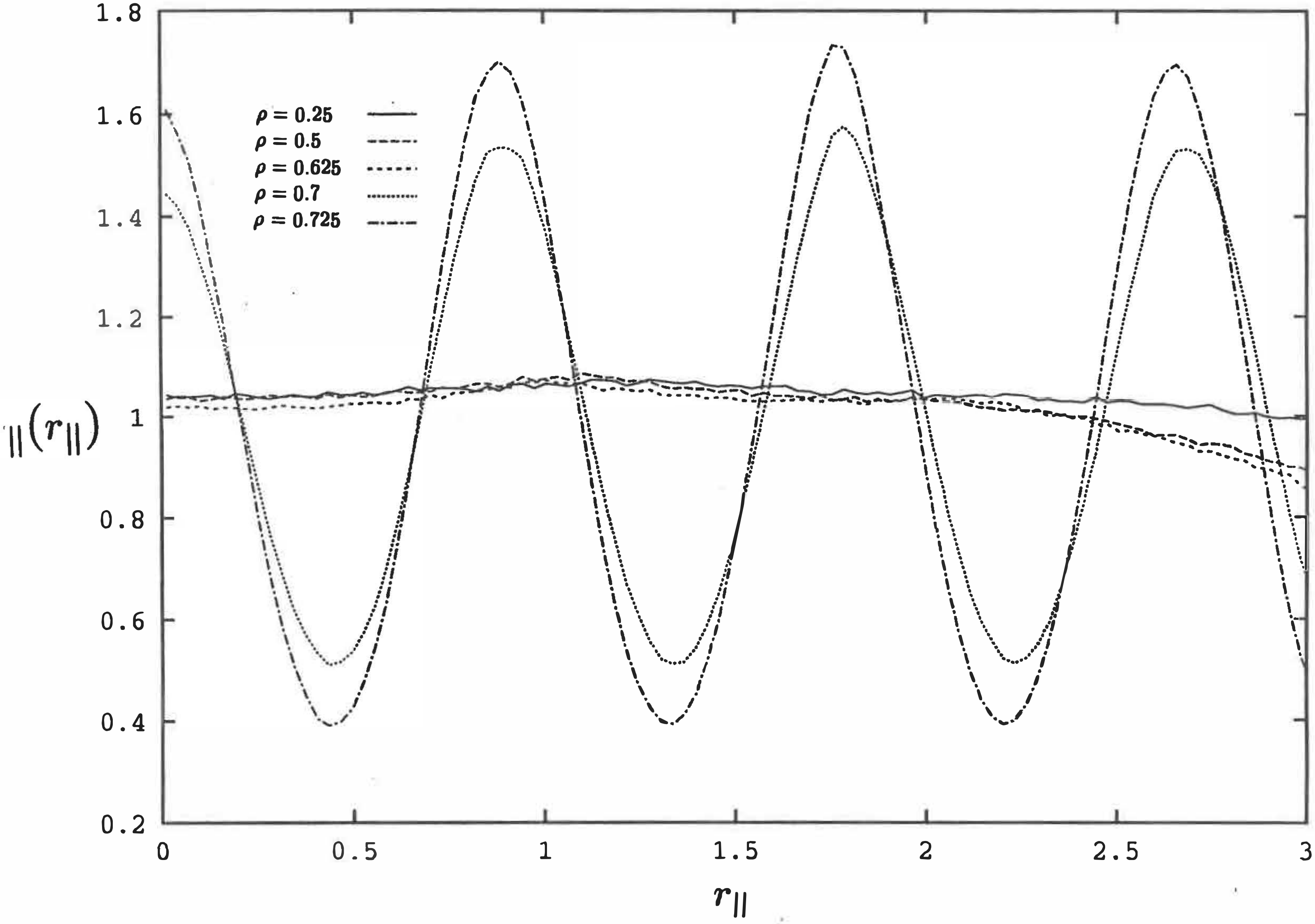


Figure 6c



$\rho = 0.25$
$\rho = 0.5$
$\rho = 0.625$
$\rho = 0.7$
$\rho = 0.725$
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
$_{||}(r_{||})$
0
0.5
1
1.5
2
2.5
3
$r_{||}$

Figure 6d

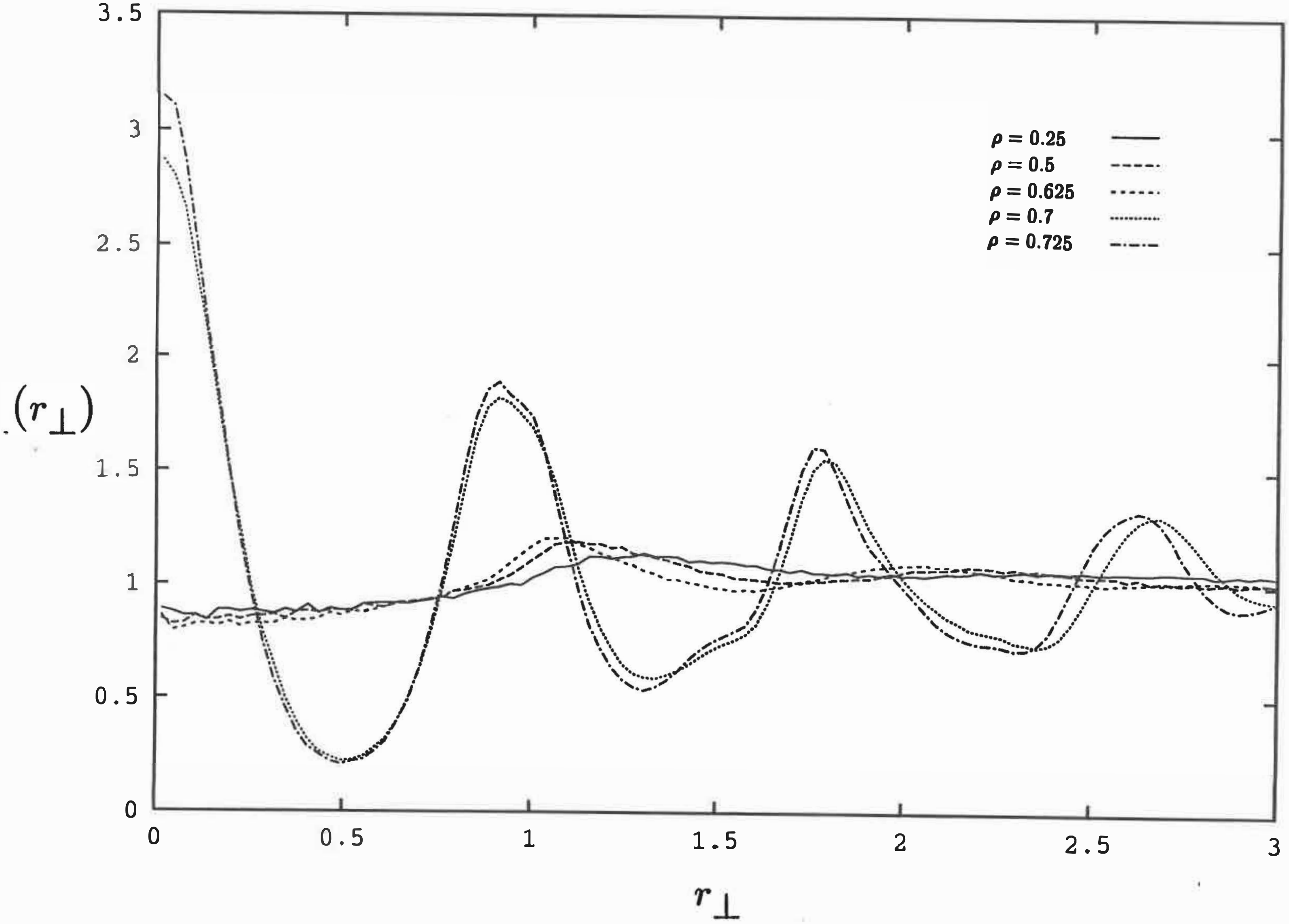

3.5
3
2.5
2
1.5
1
0.5
0
$(r_\perp)$
0
0.5
1
1.5
2
2.5
3
$r_\perp$
$\rho = 0.25$
$\rho = 0.5$
$\rho = 0.625$
$\rho = 0.7$
$\rho = 0.725$

Figure 7a

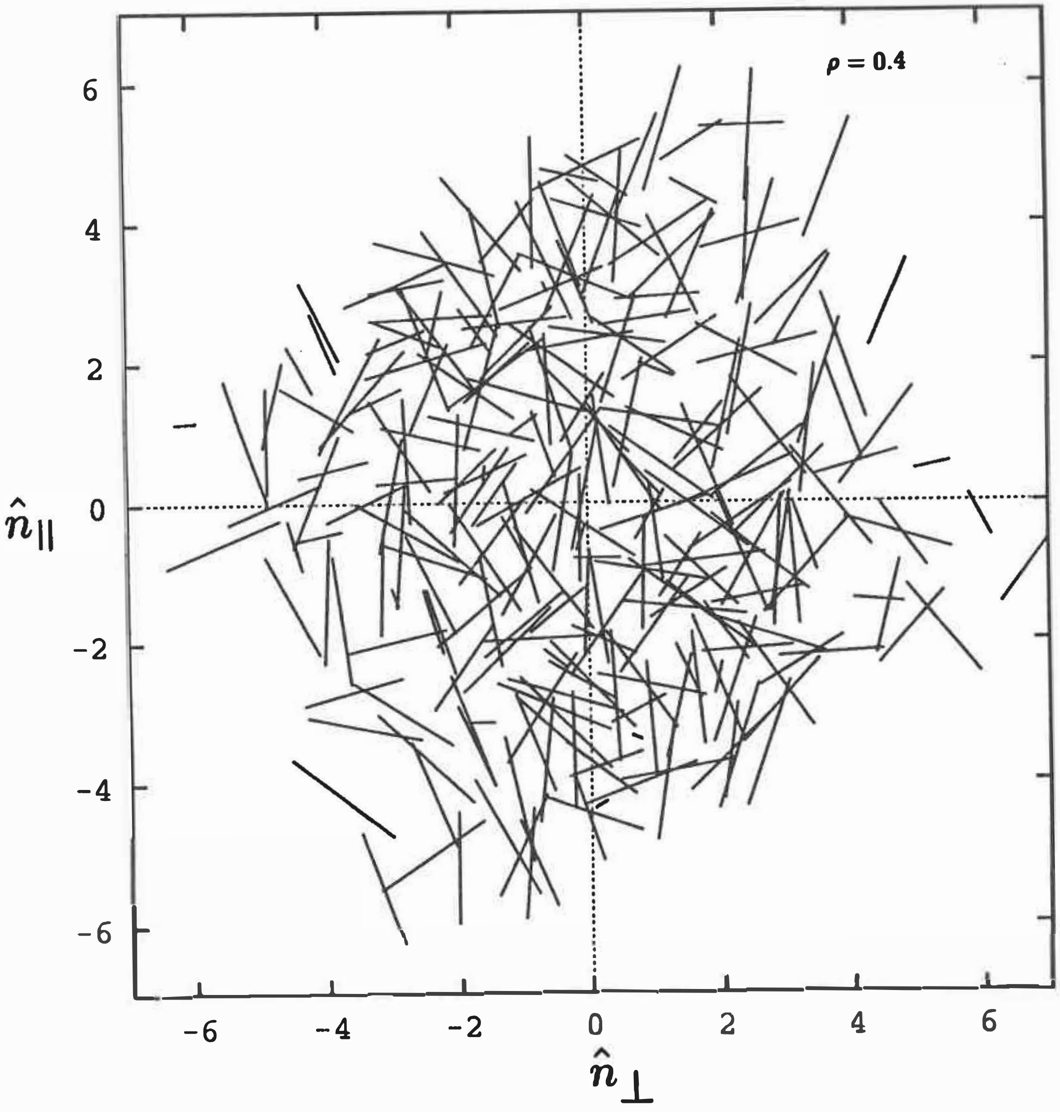


Figure 7b

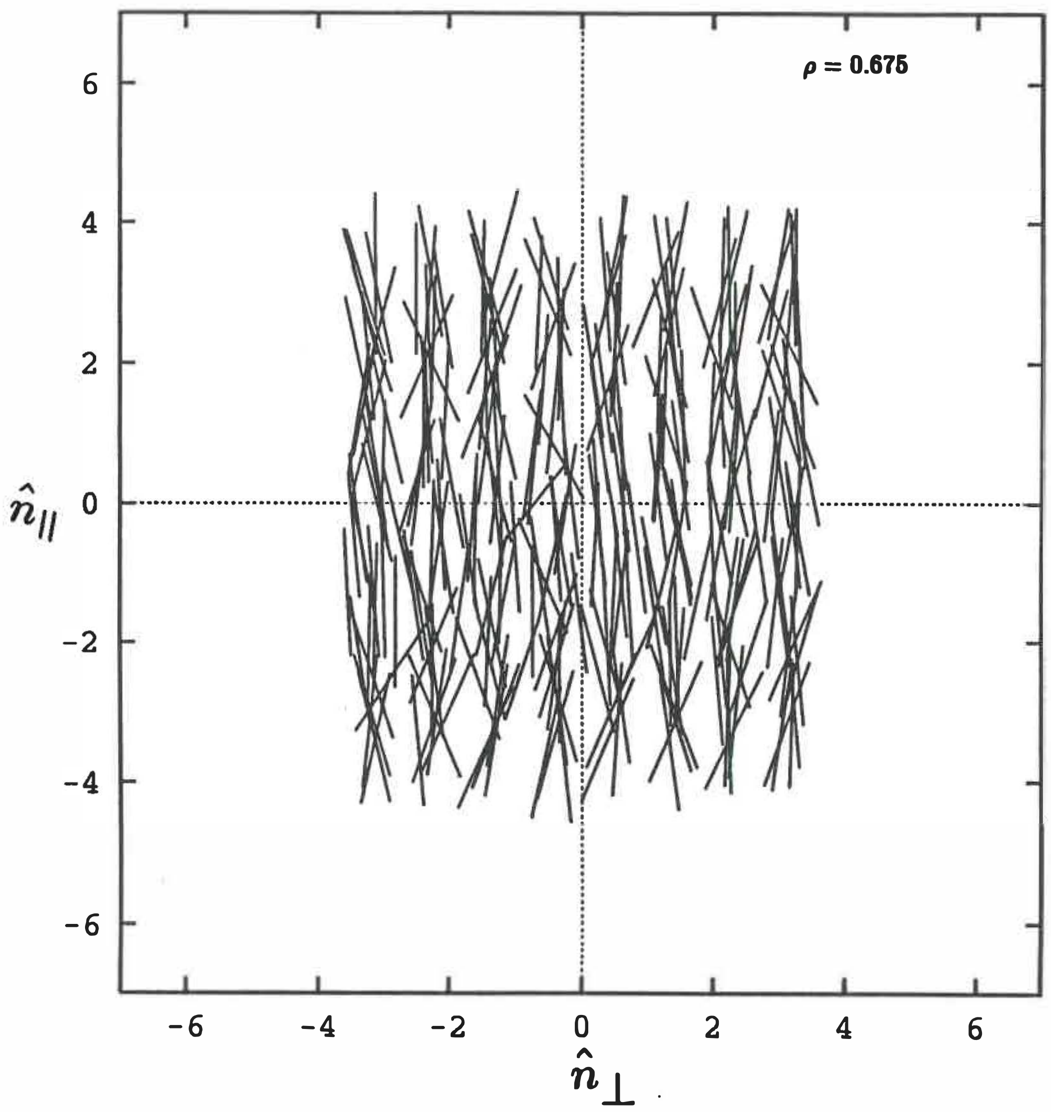


Figure 8a

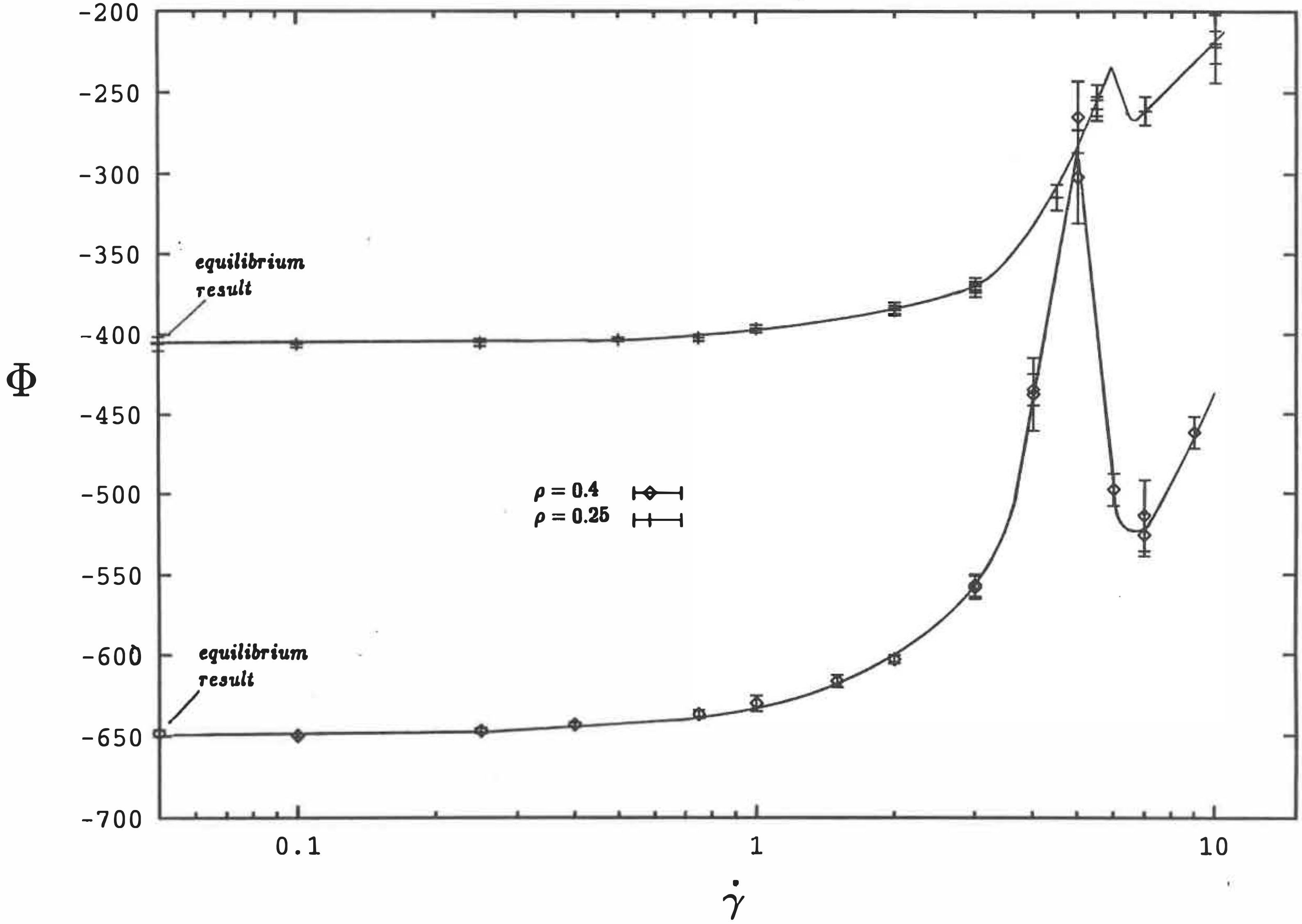

-200
-250
-300
-350
-400
-450
-500
-550
-600
-650
-700
Φ
0.1
1
10
$\dot{\gamma}$
equilibrium result
equilibrium result
$\rho = 0.4$
$\rho = 0.25$

Figure 8b

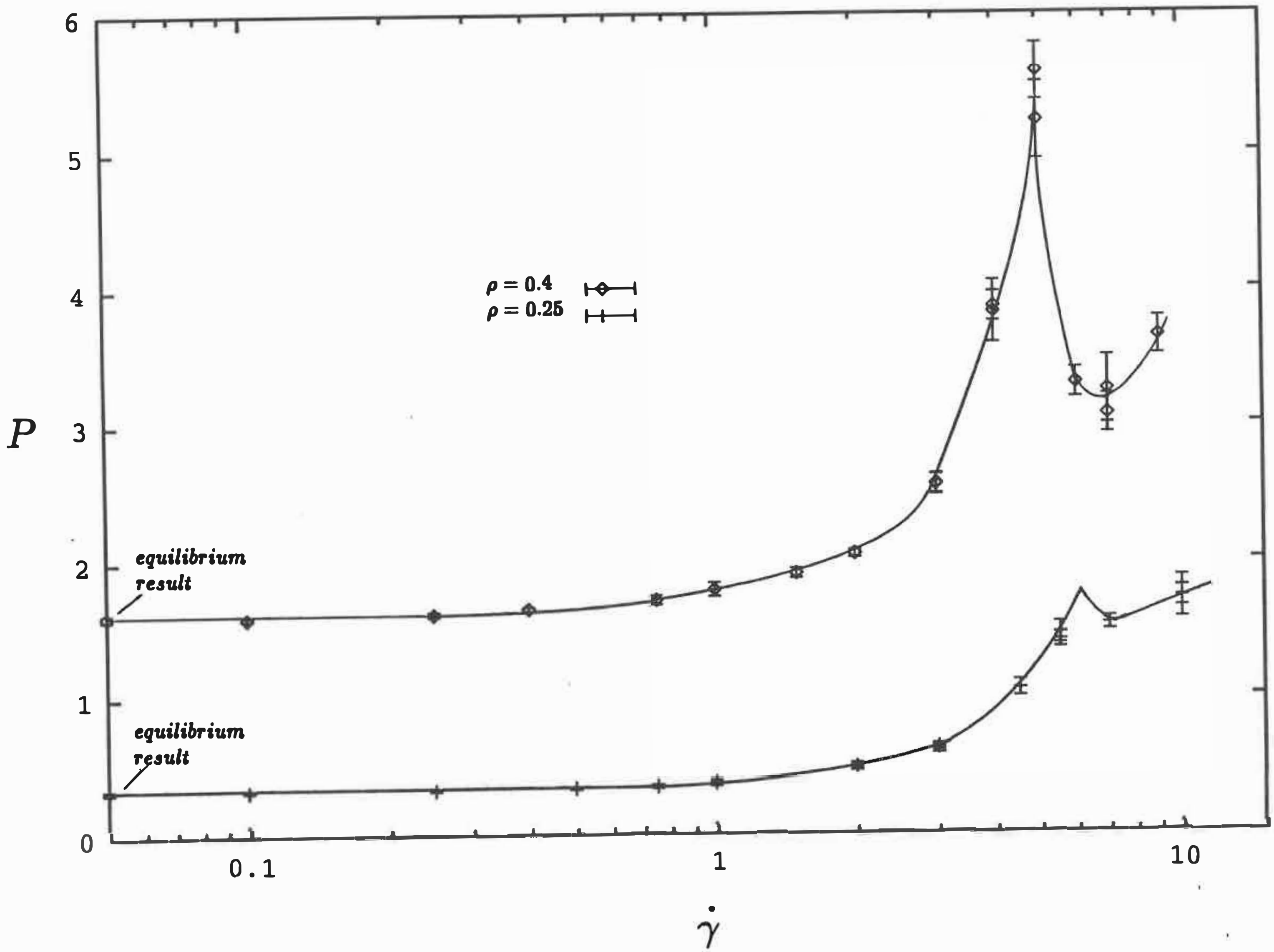

6
5
4
3
2
1
0
P
0.1
1
10
$\dot{\gamma}$
$\rho = 0.4$
$\rho = 0.25$
equilibrium result
equilibrium result

Figure 9

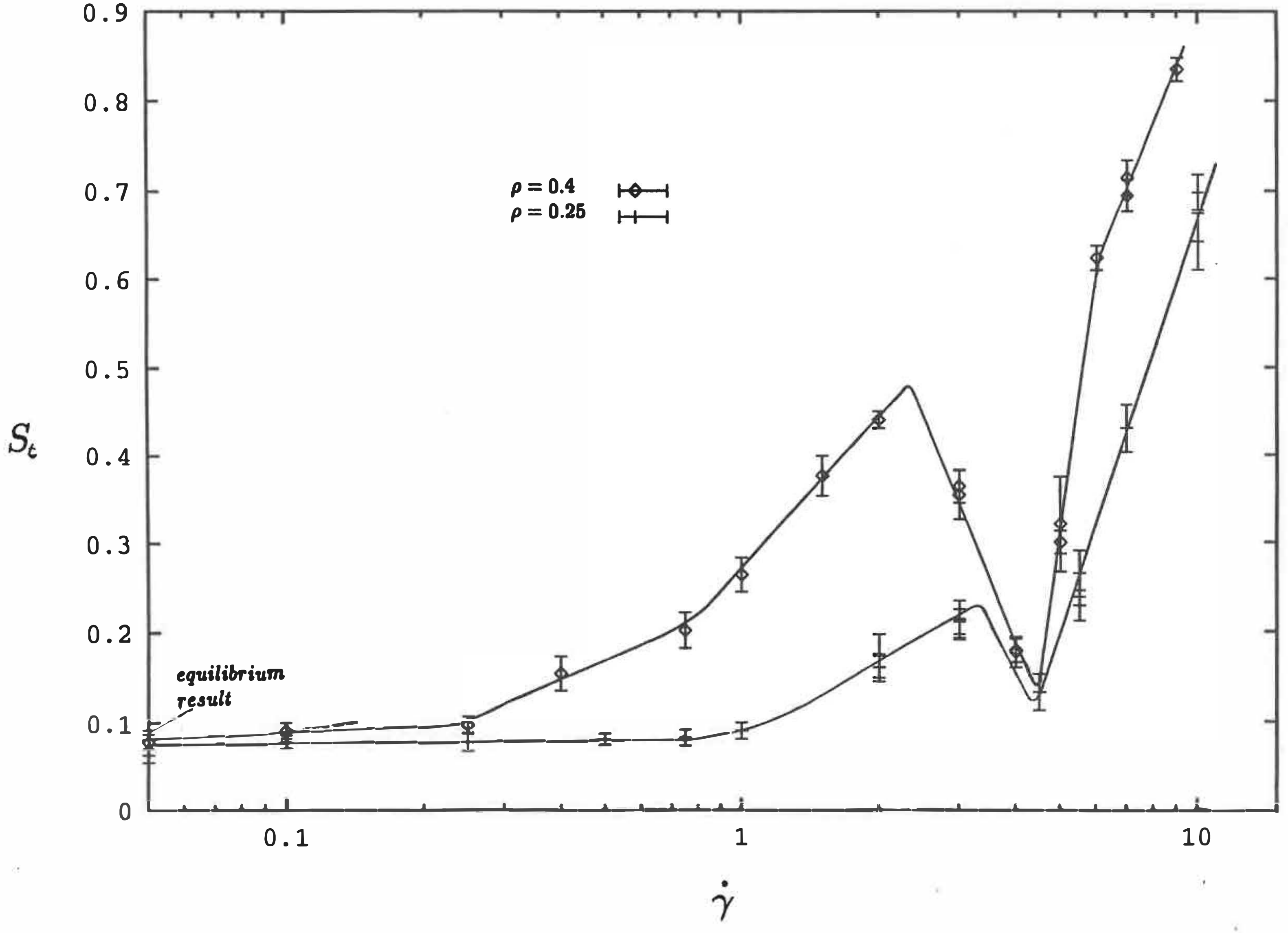
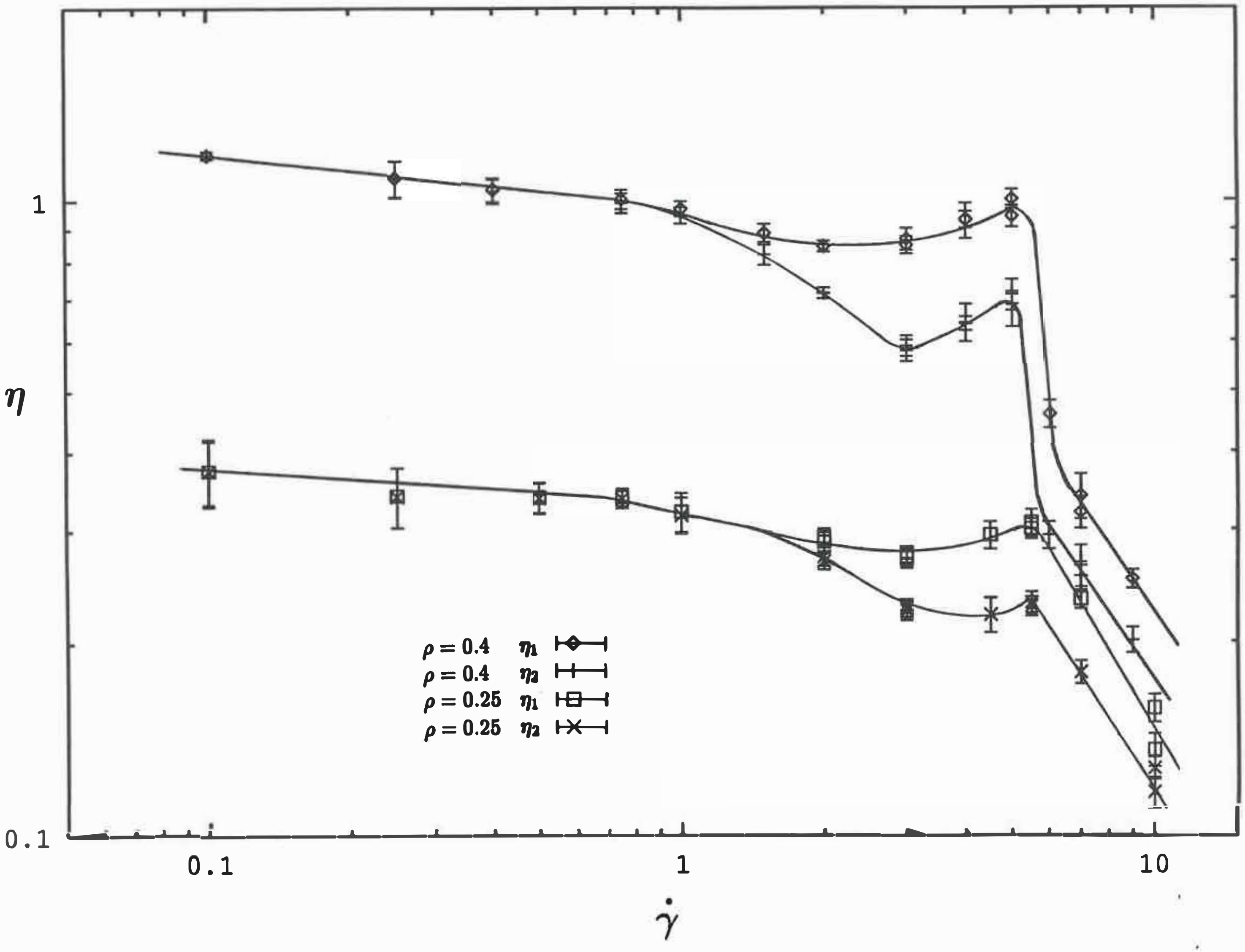

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
1
10
S_t
$\dot{\gamma}$
$\rho = 0.4$
$\rho = 0.25$
equilibrium
result

Figure 10a



1
0.1
η
0.1
1
10
$\dot{\gamma}$
$\rho = 0.4$ η_1 ◇
$\rho = 0.4$ η_2 +
$\rho = 0.25$ η_1 □
$\rho = 0.25$ η_2 ×

Figure 10b

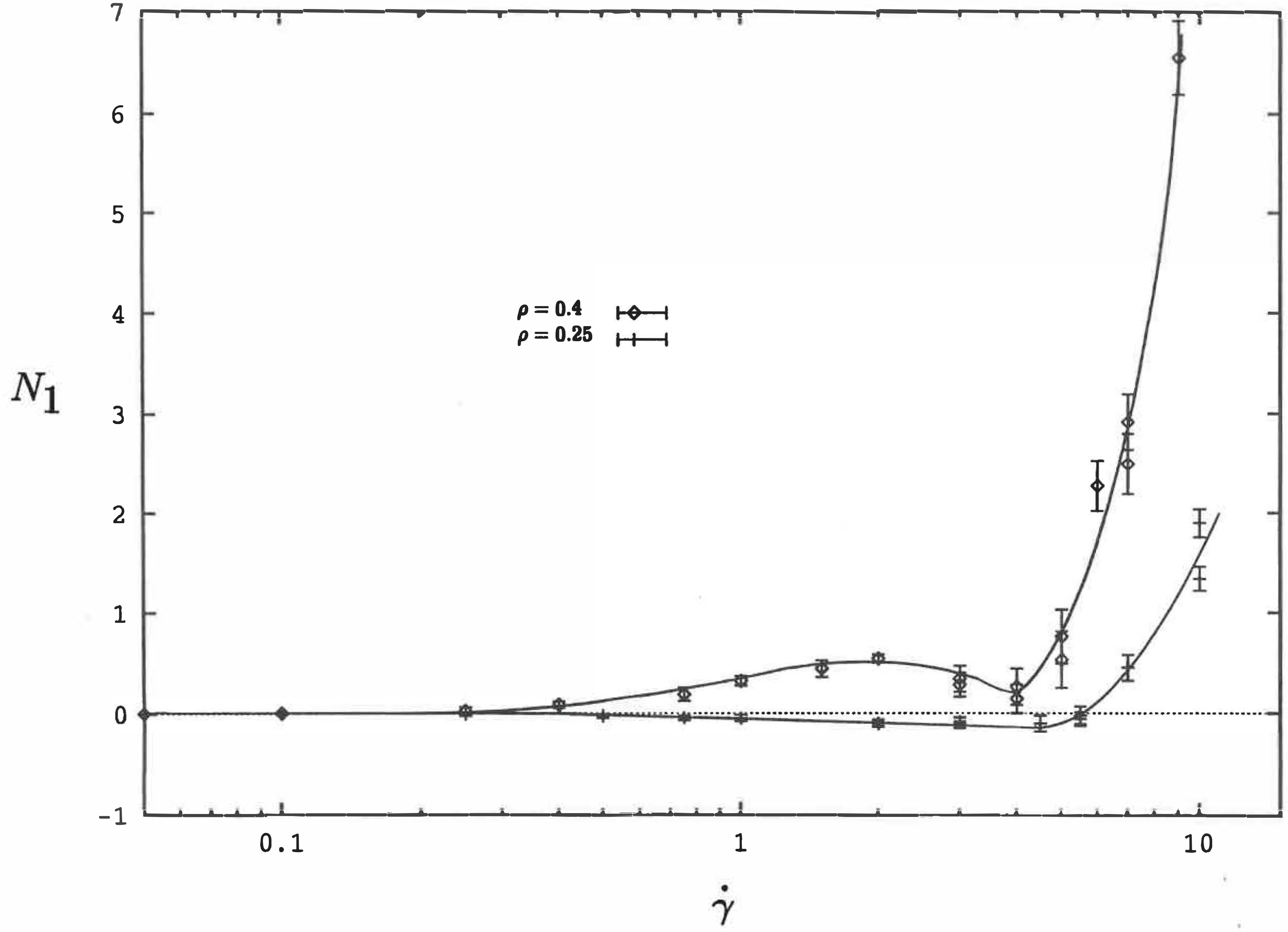


Figure 10c

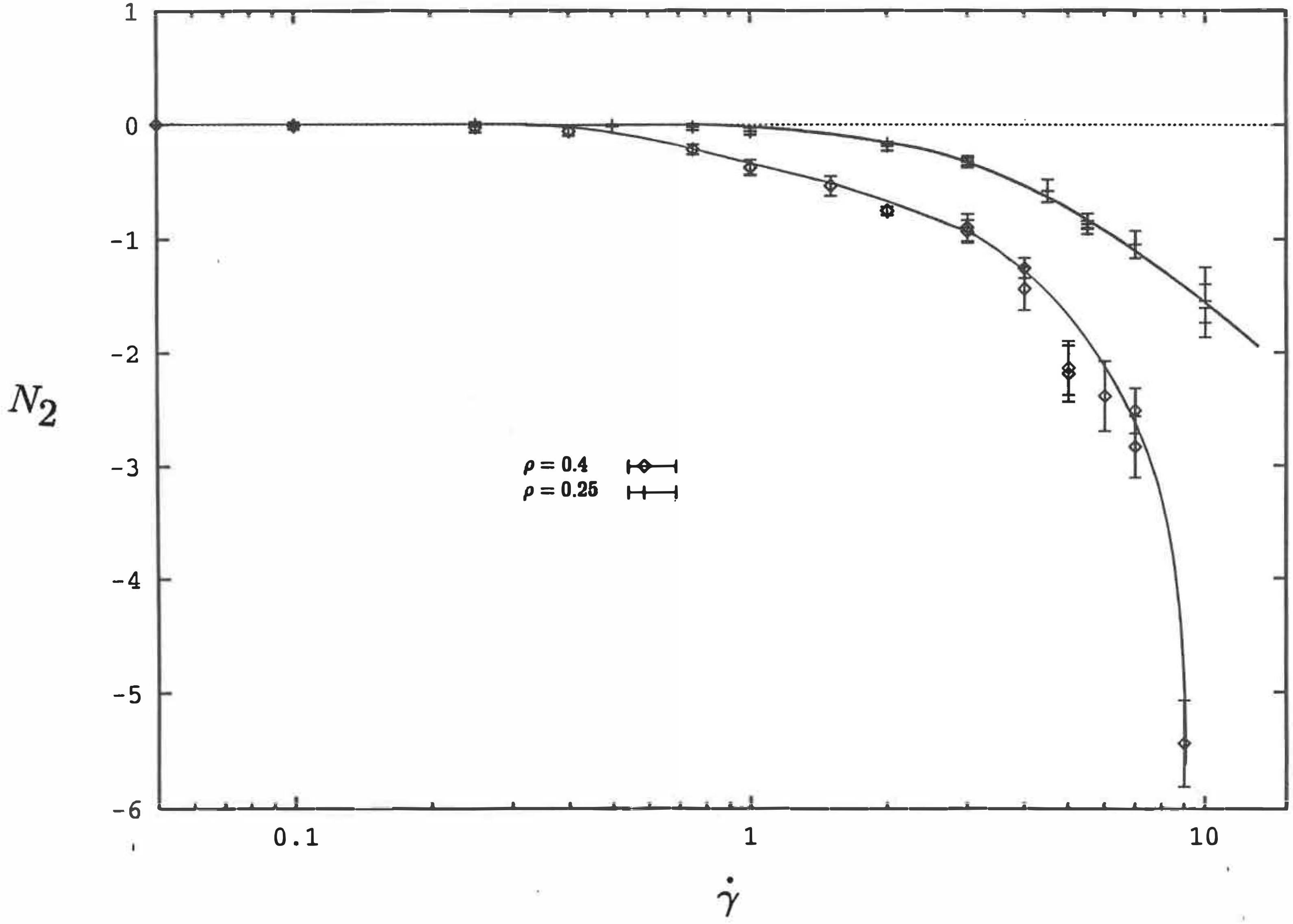
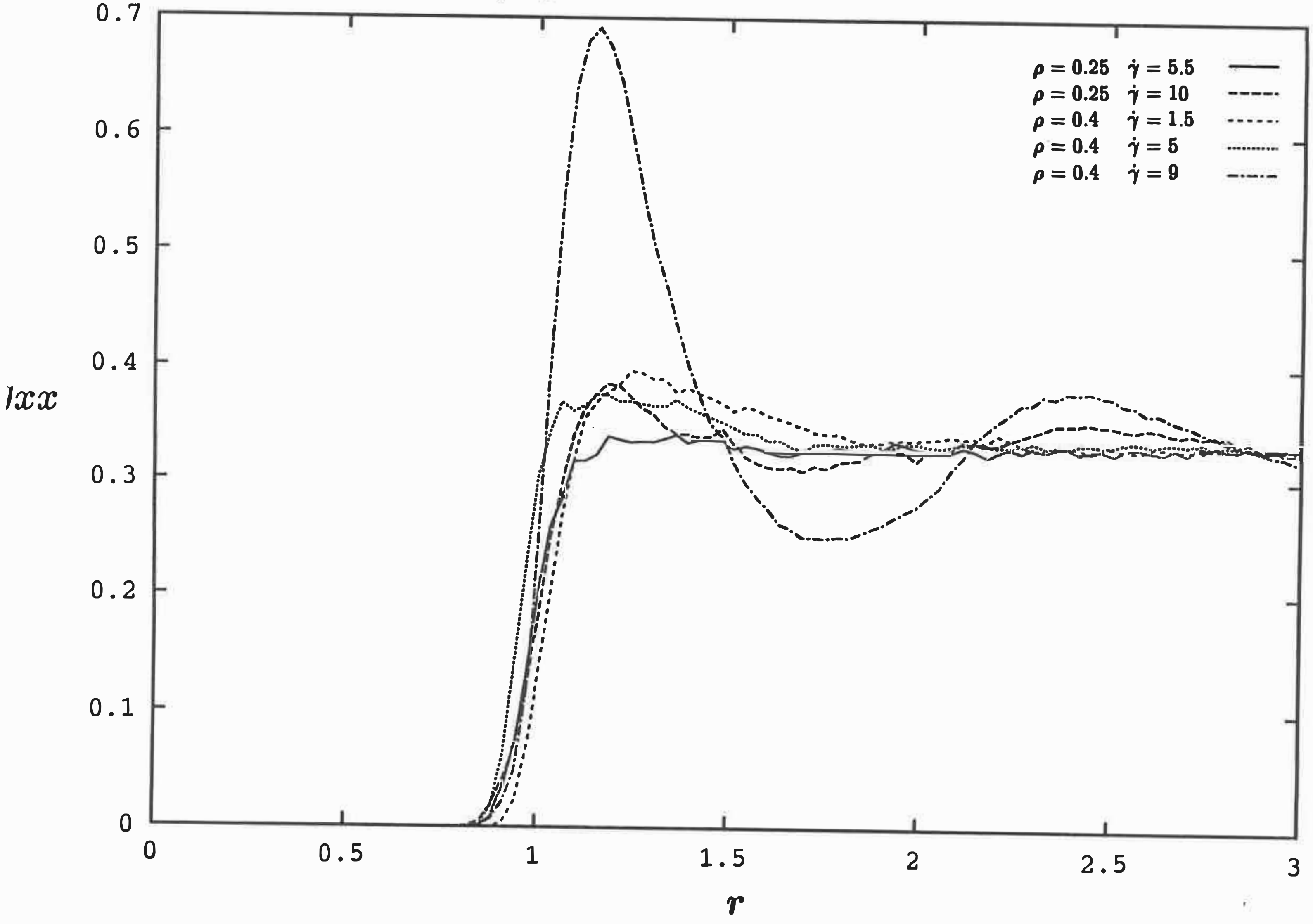
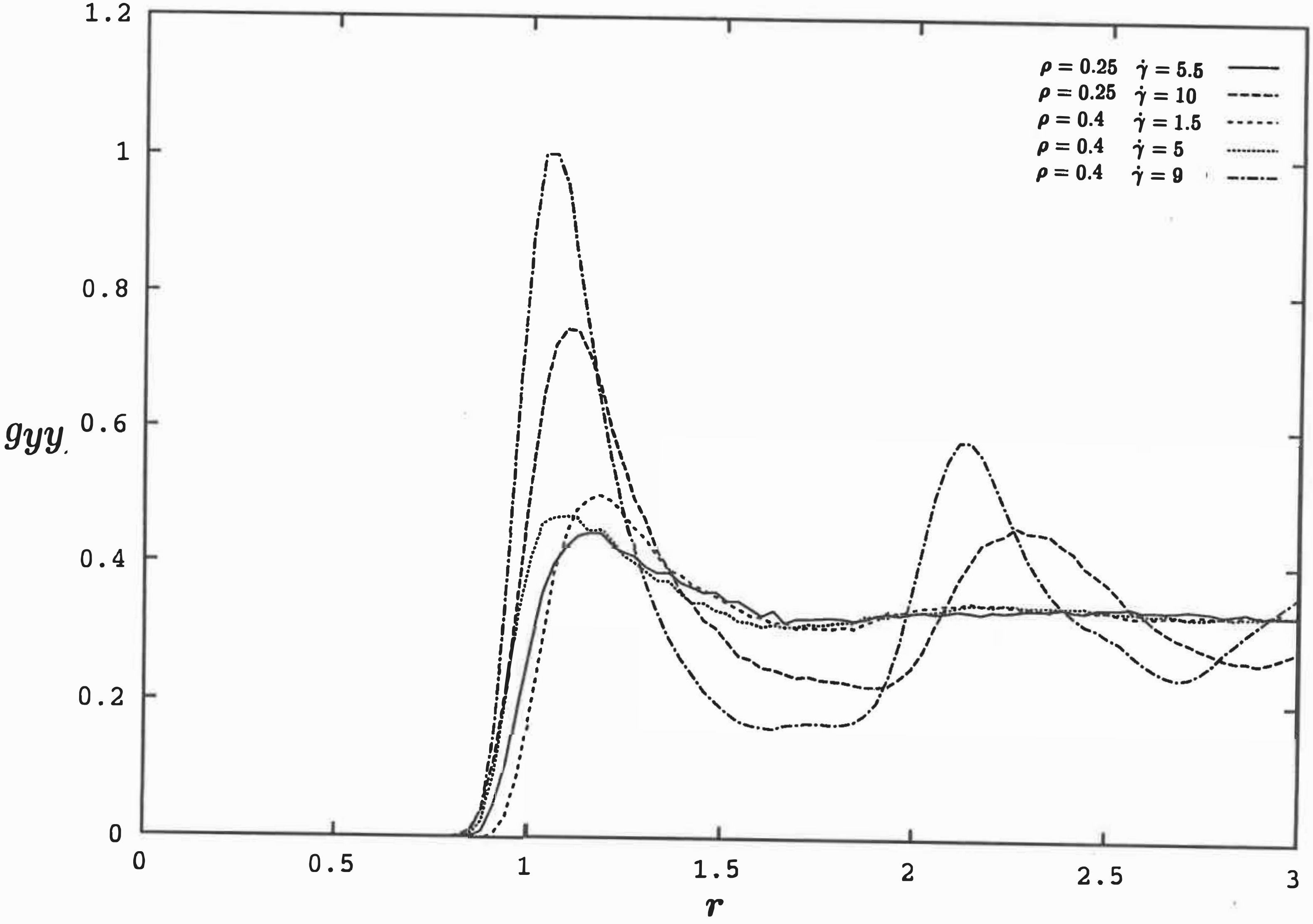


Figure 11a



0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
/xx
0
0.5
1
1.5
2
2.5
3
r
$\rho = 0.25$ $\dot{\gamma} = 5.5$
$\rho = 0.25$ $\dot{\gamma} = 10$
$\rho = 0.4$ $\dot{\gamma} = 1.5$
$\rho = 0.4$ $\dot{\gamma} = 5$
$\rho = 0.4$ $\dot{\gamma} = 9$

Figure 11b



$\rho = 0.25$ $\dot{\gamma} = 5.5$
$\rho = 0.25$ $\dot{\gamma} = 10$
$\rho = 0.4$ $\dot{\gamma} = 1.5$
$\rho = 0.4$ $\dot{\gamma} = 5$
$\rho = 0.4$ $\dot{\gamma} = 9$
gyy
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
3
r

Figure 11c

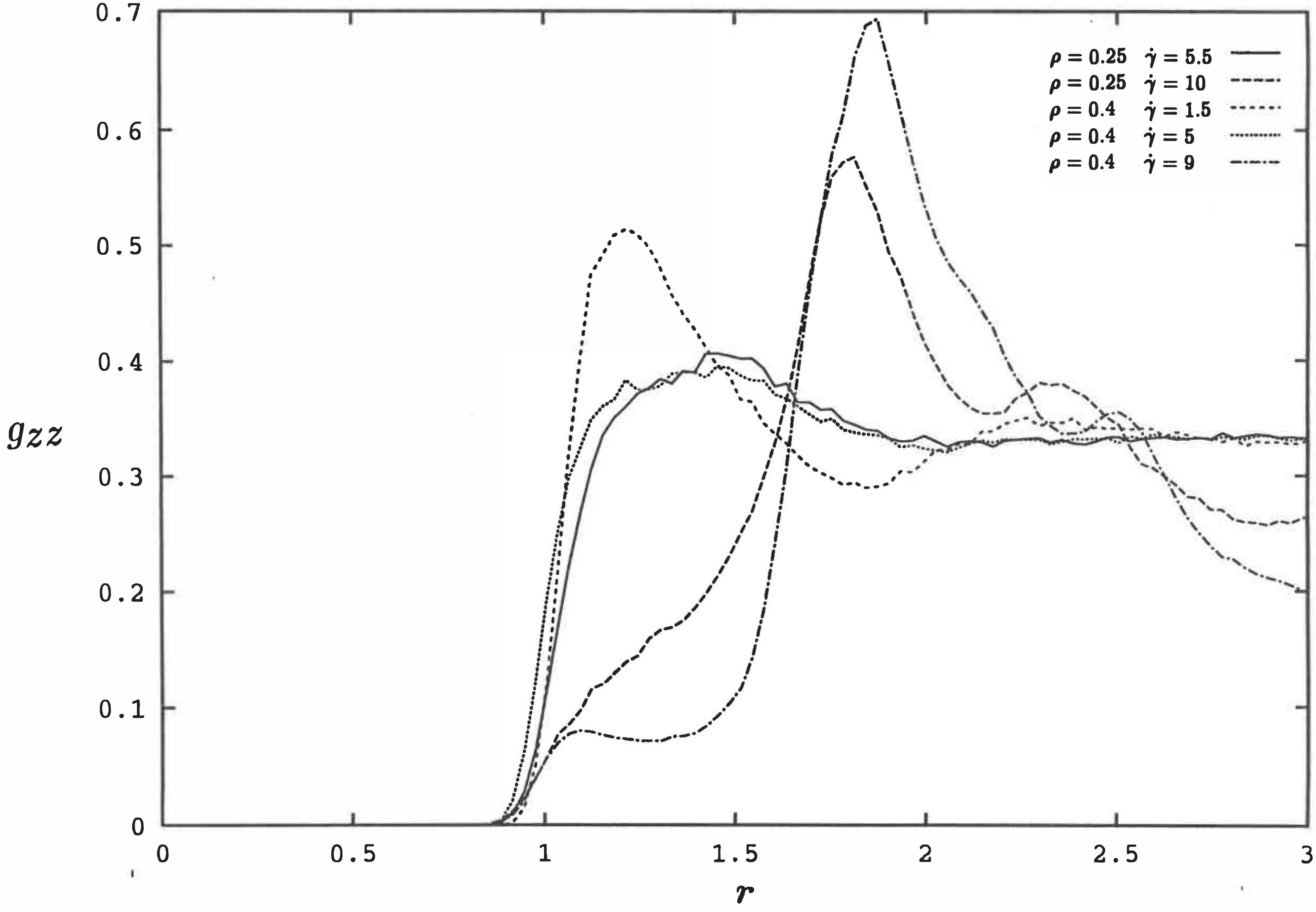
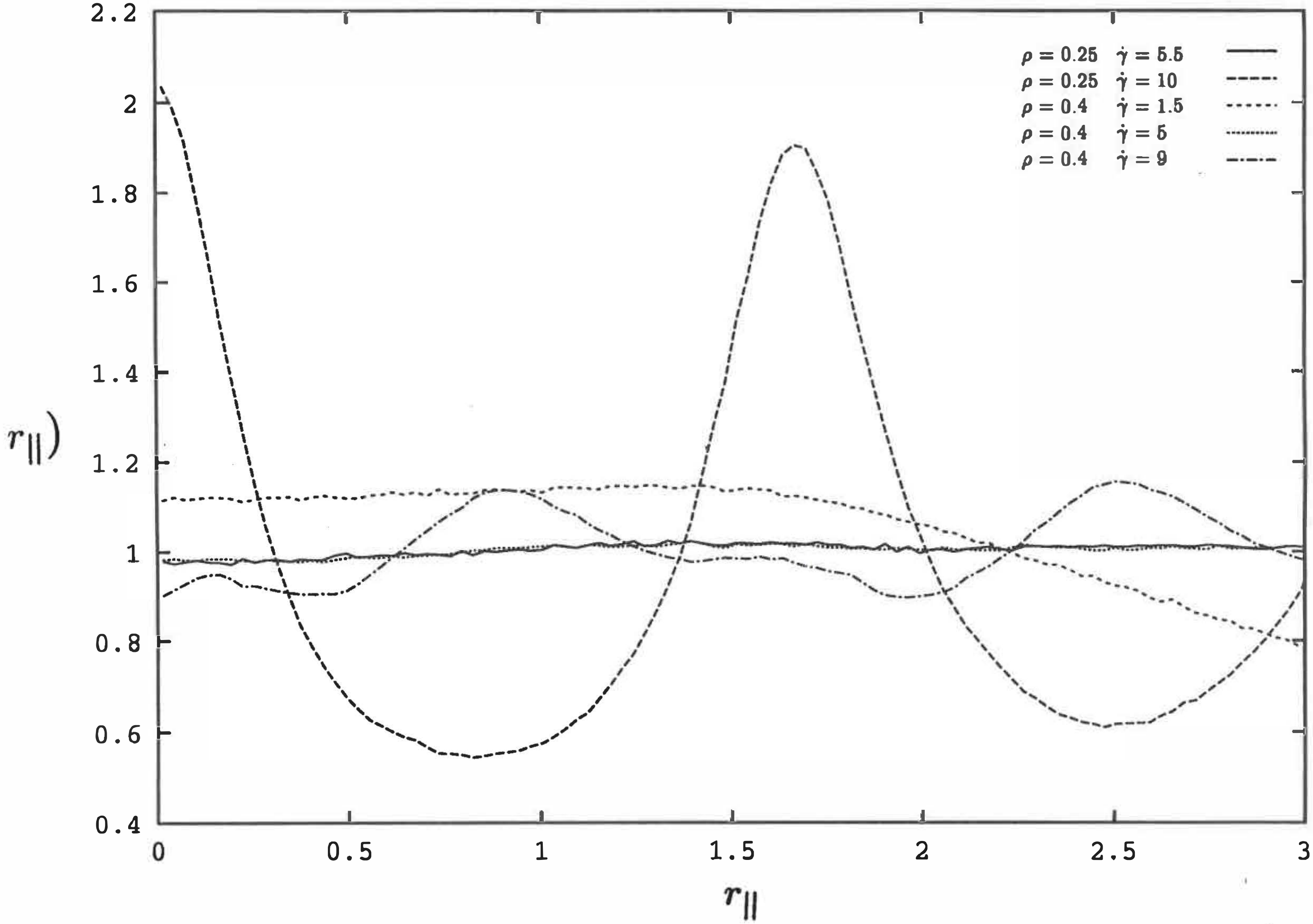
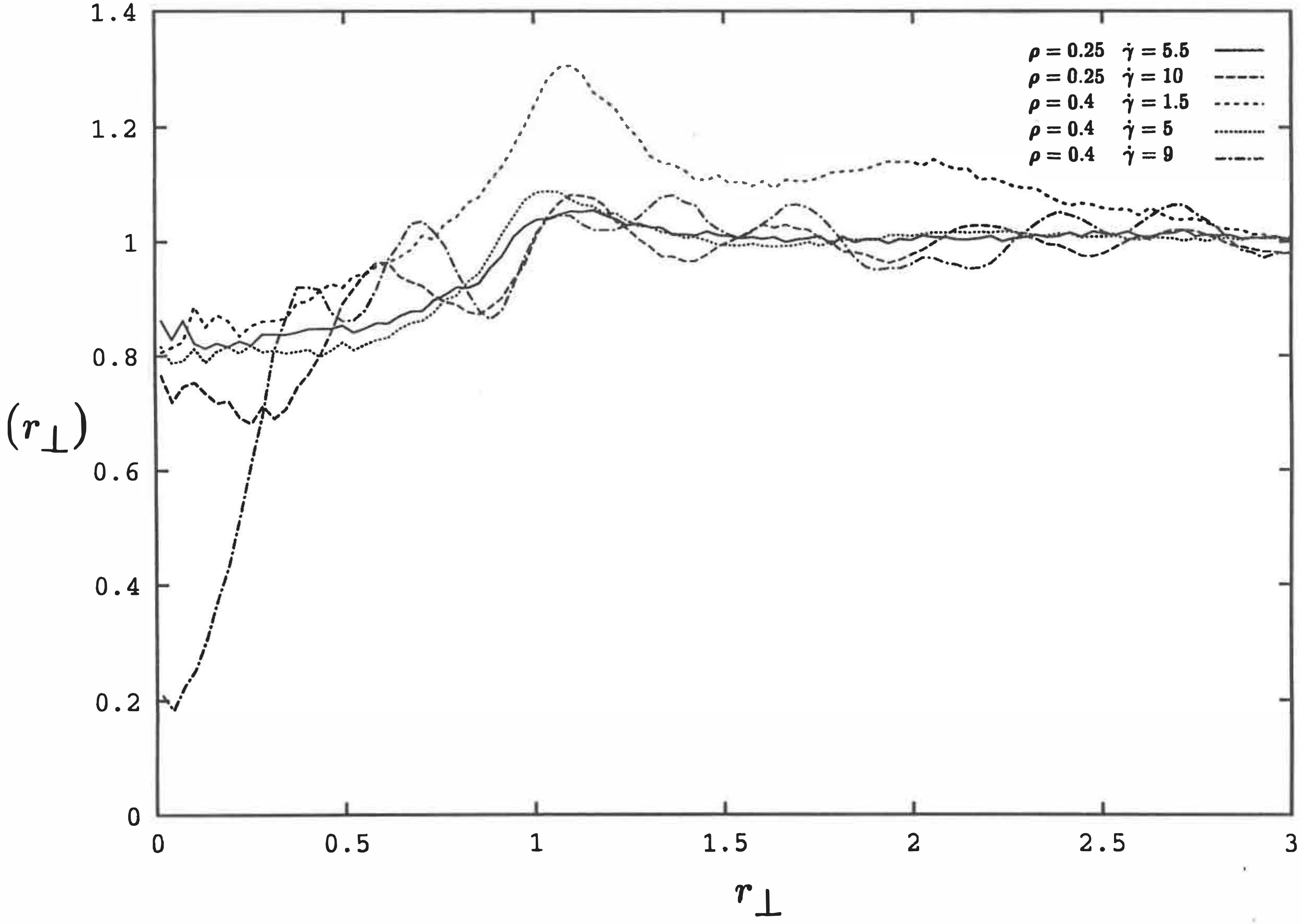

$\rho = 0.25$ $\dot{\gamma} = 5.5$
$\rho = 0.25$ $\dot{\gamma} = 10$
$\rho = 0.4$ $\dot{\gamma} = 1.5$
$\rho = 0.4$ $\dot{\gamma} = 5$
$\rho = 0.4$ $\dot{\gamma} = 9$
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
gzz
0
0.5
1
1.5
2
2.5
3
r

Figure 11d



$\rho = 0.25$ $\dot\gamma = 5.5$
$\rho = 0.25$ $\dot\gamma = 10$
$\rho = 0.4$ $\dot\gamma = 1.5$
$\rho = 0.4$ $\dot\gamma = 5$
$\rho = 0.4$ $\dot\gamma = 9$
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
$r_{||})$
0
0.5
1
1.5
2
2.5
3
$r_{||}$

Figure 11e



$\rho = 0.25$ $\dot{\gamma} = 5.5$
$\rho = 0.25$ $\dot{\gamma} = 10$
$\rho = 0.4$ $\dot{\gamma} = 1.5$
$\rho = 0.4$ $\dot{\gamma} = 5$
$\rho = 0.4$ $\dot{\gamma} = 9$
1.4
1.2
1
0.8
0.6
0.4
0.2
0
$(r_\perp)$
0
0.5
1
1.5
2
2.5
3
$r_\perp$

Figure 12a

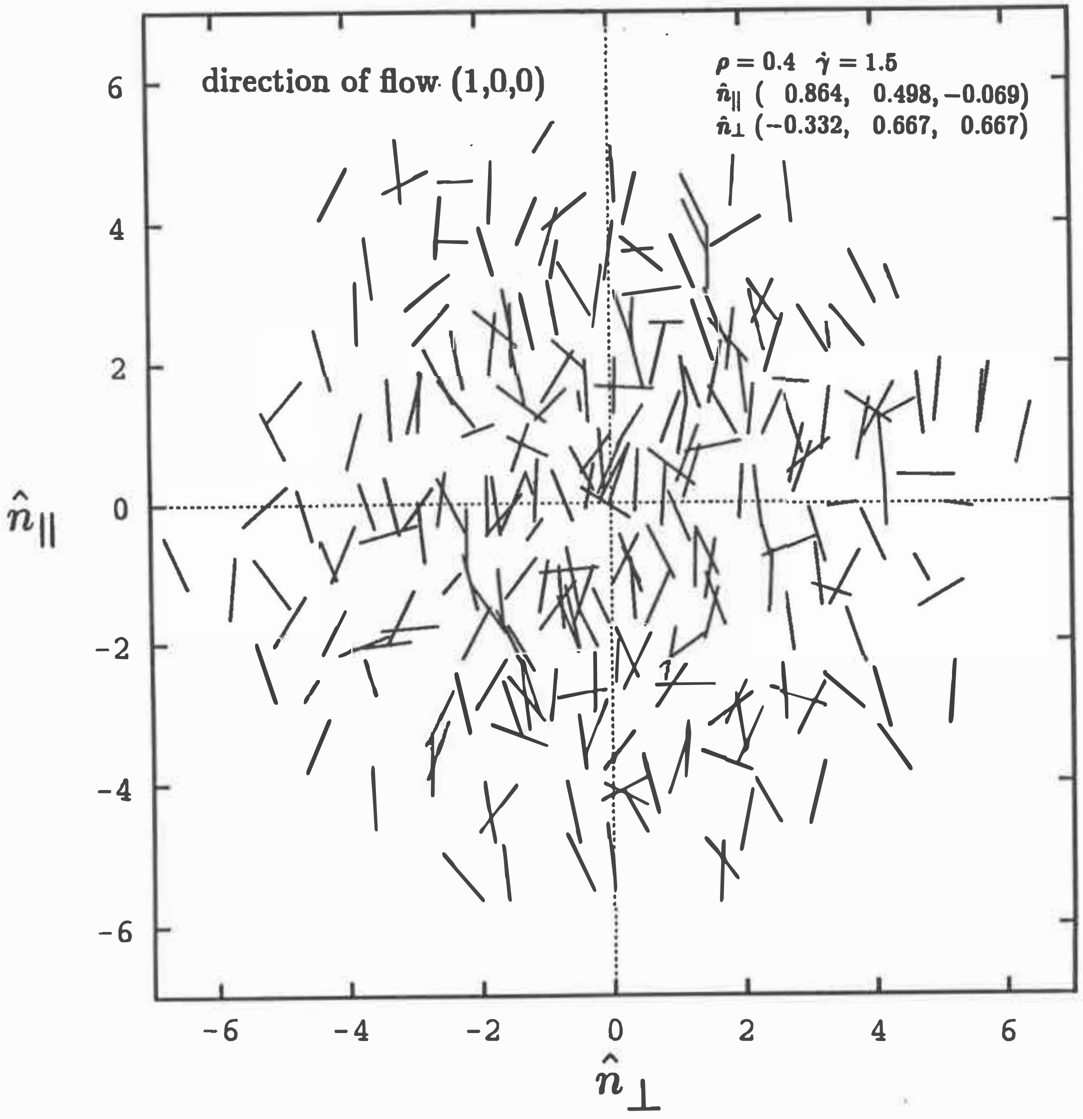

direction of flow (1,0,0)
$\rho = 0.4 \quad \dot{\gamma} = 1.5$
$\hat{n}_{\|}$ (0.864, 0.498, −0.069)
$\hat{n}_{\perp}$ (−0.332, 0.667, 0.667)
$\hat{n}_{\|}$
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6
$\hat{n}_{\perp}$

Figure 12b.

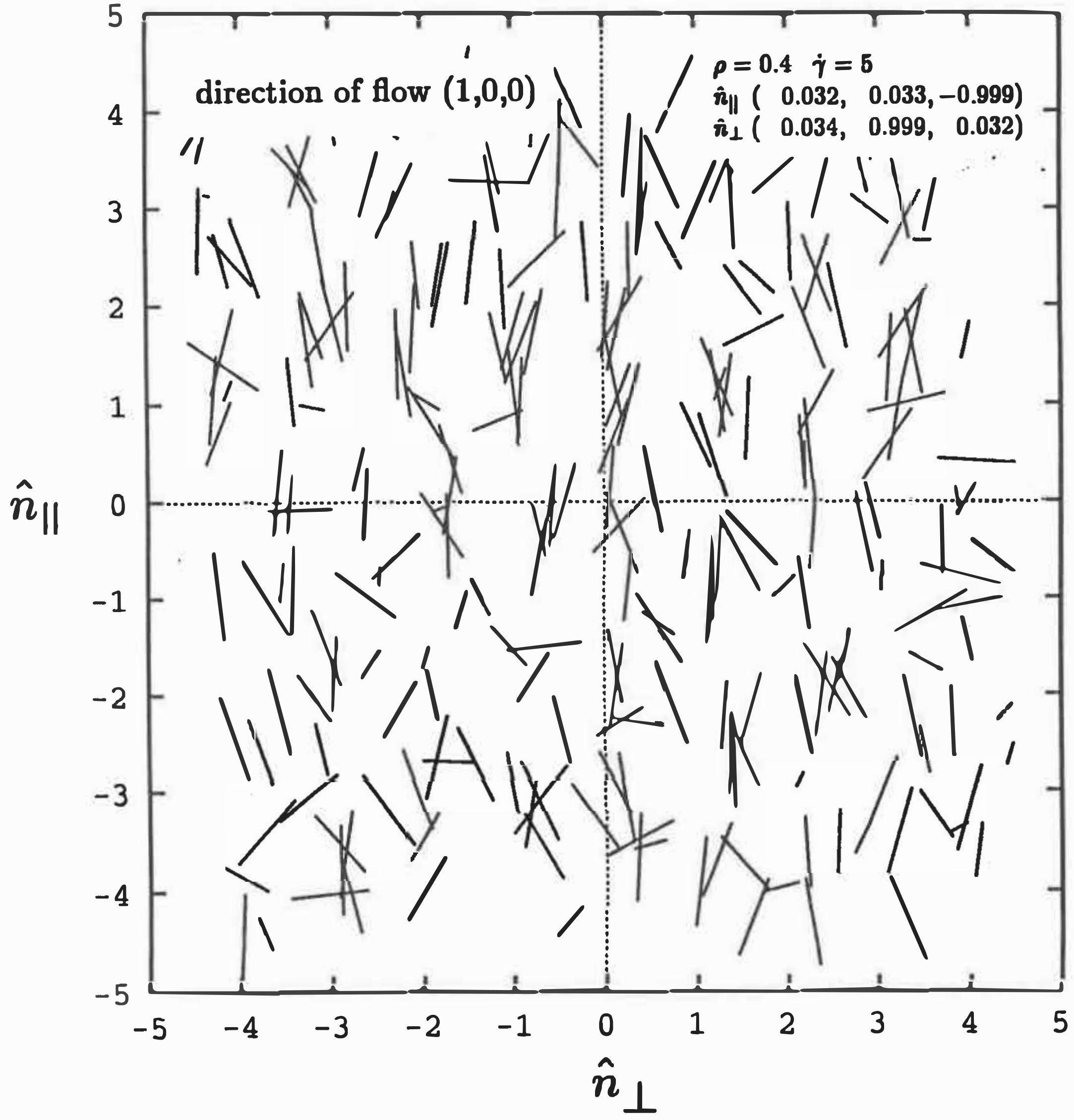

direction of flow (1,0,0)
$\rho = 0.4$ $\dot{\gamma} = 5$
$\hat{n}_{\|}$ (0.032, 0.033, −0.999)
$\hat{n}_{\perp}$ (0.034, 0.999, 0.032)
$\hat{n}_{\|}$
$\hat{n}_{\perp}$
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5

Figure 12c

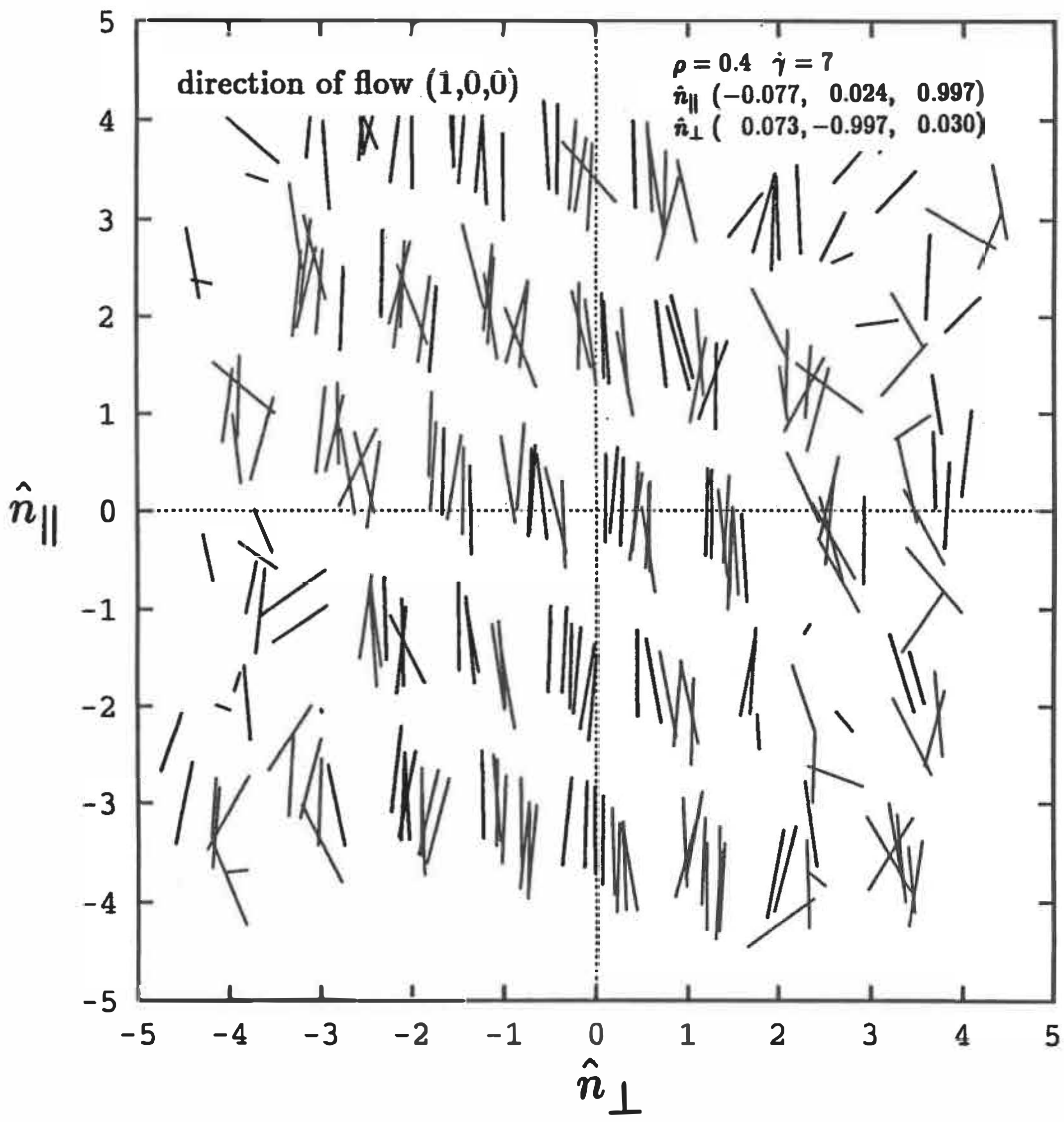

direction of flow (1,0,0)
$\rho = 0.4 \quad \dot{\gamma} = 7$
$\hat{n}_{\parallel}$ (−0.077, 0.024, 0.997)
$\hat{n}_{\perp}$ (0.073, −0.997, 0.030)
5
4
3
2
1
0
-1
-2
-3
-4
-5
$\hat{n}_{\parallel}$
-5
-4
-3
-2
-1
0
1
2
3
4
5
$\hat{n}_{\perp}$

Figure 12d

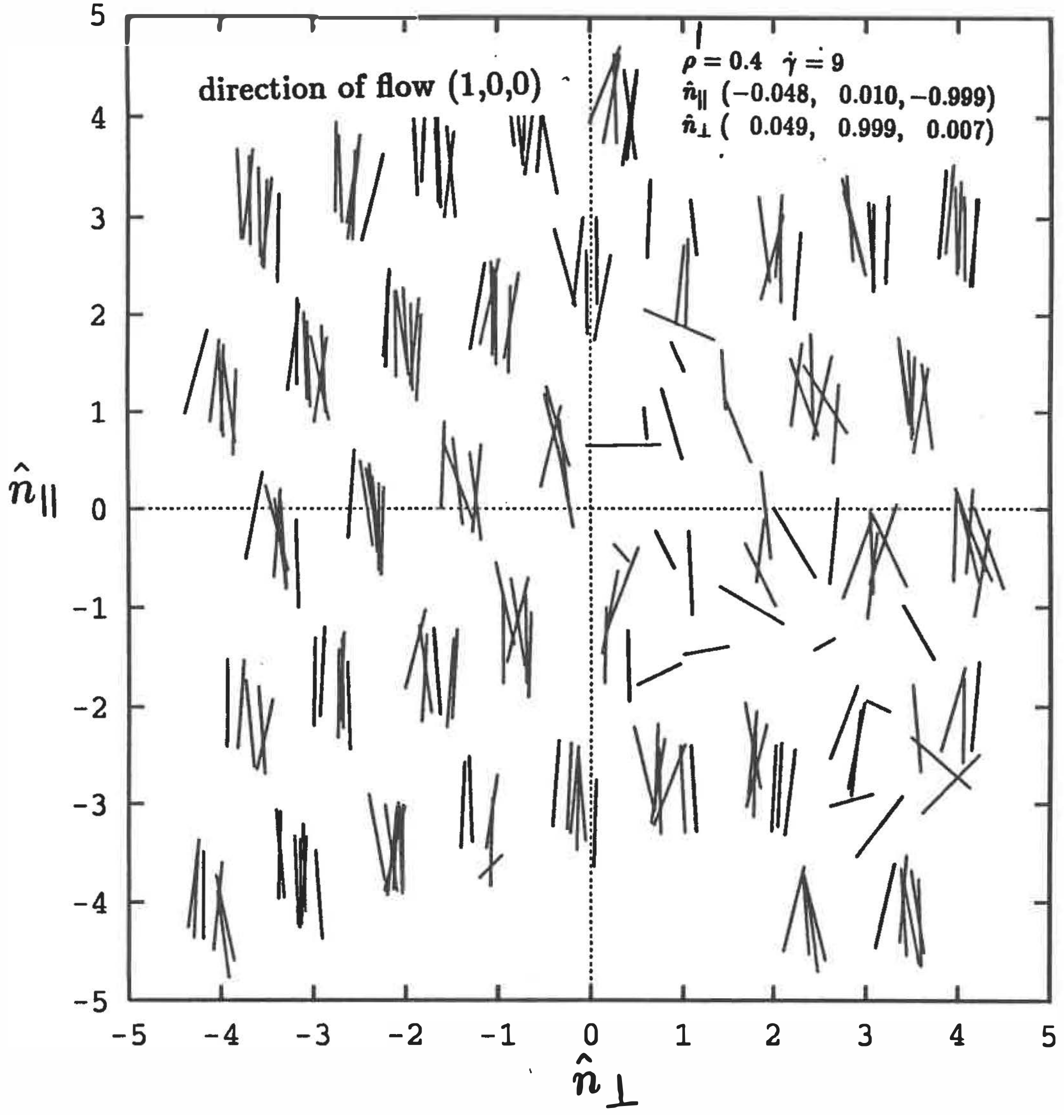

direction of flow (1,0,0)
$\rho = 0.4$ $\dot{\gamma} = 9$
$\hat{n}_{\parallel}$ (−0.048, 0.010, −0.999)
$\hat{n}_{\perp}$ (0.049, 0.999, 0.007)
$\hat{n}_{\parallel}$
$\hat{n}_{\perp}$
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5

Figure 12e

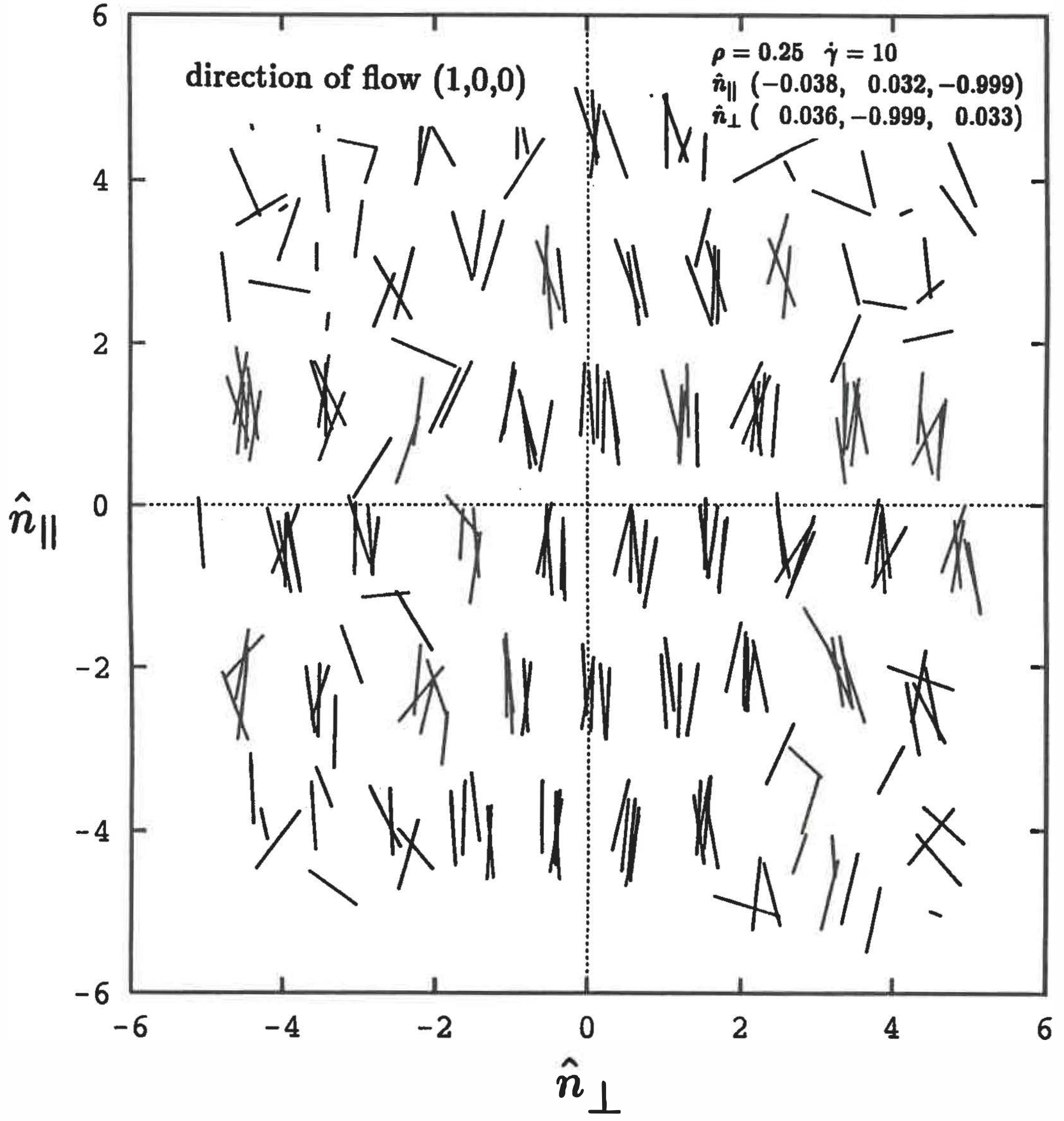

direction of flow (1,0,0)
$\rho = 0.25$ $\dot{\gamma} = 10$
$\hat{n}_{\|}$ (−0.038,   0.032, −0.999)
$\hat{n}_{\perp}$ (0.036, −0.999,   0.033)
$\hat{n}_{\|}$
$\hat{n}_{\perp}$
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

Figure 12f

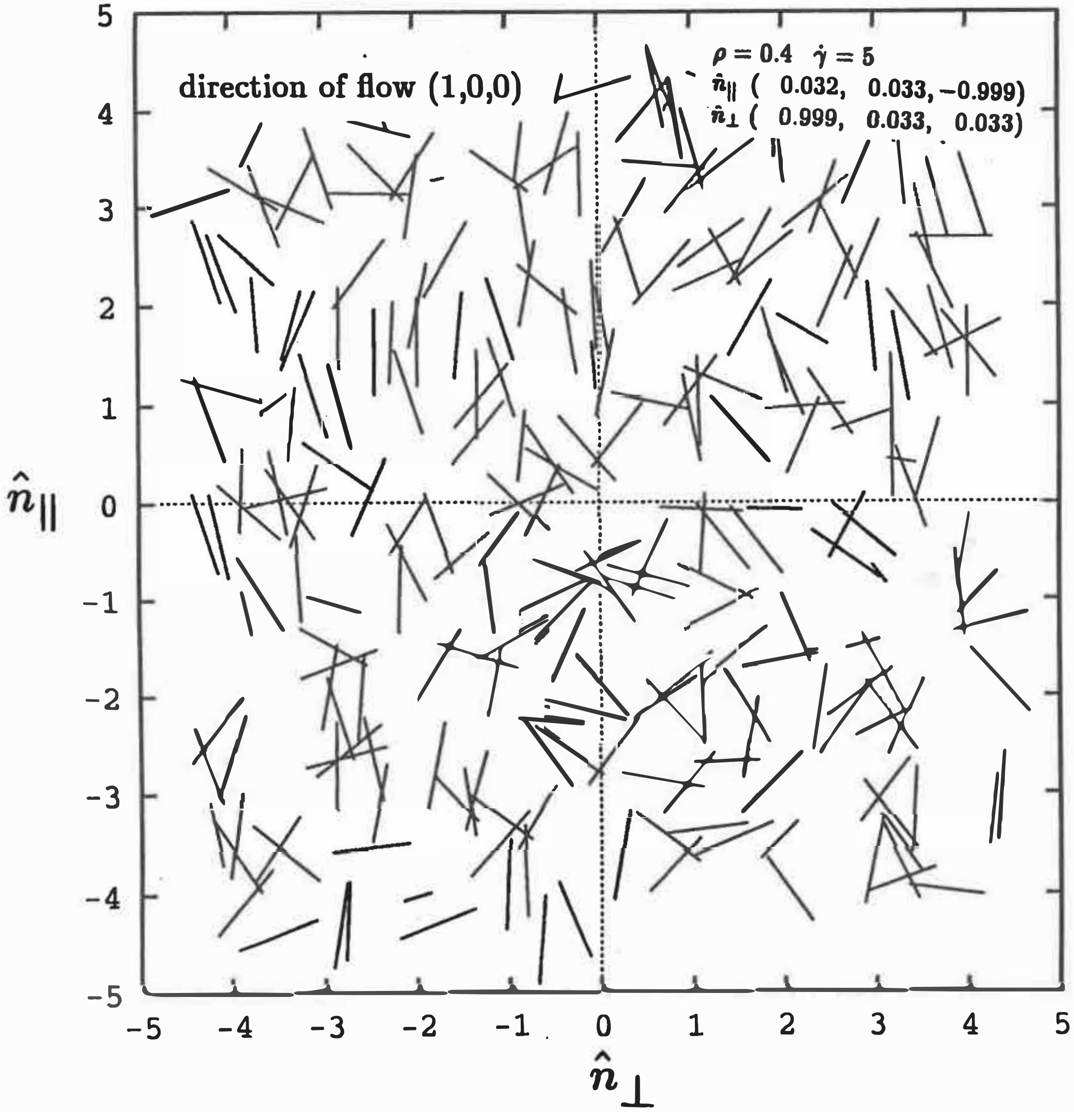

direction of flow (1,0,0)
$\rho = 0.4$ $\dot{\gamma} = 5$
$\hat{n}_{\parallel}$ (0.032, 0.033, −0.999)
$\hat{n}_{\perp}$ (0.999, 0.033, 0.033)
$\hat{n}_{\parallel}$
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5
$\hat{n}_{\perp}$

Figure 12g

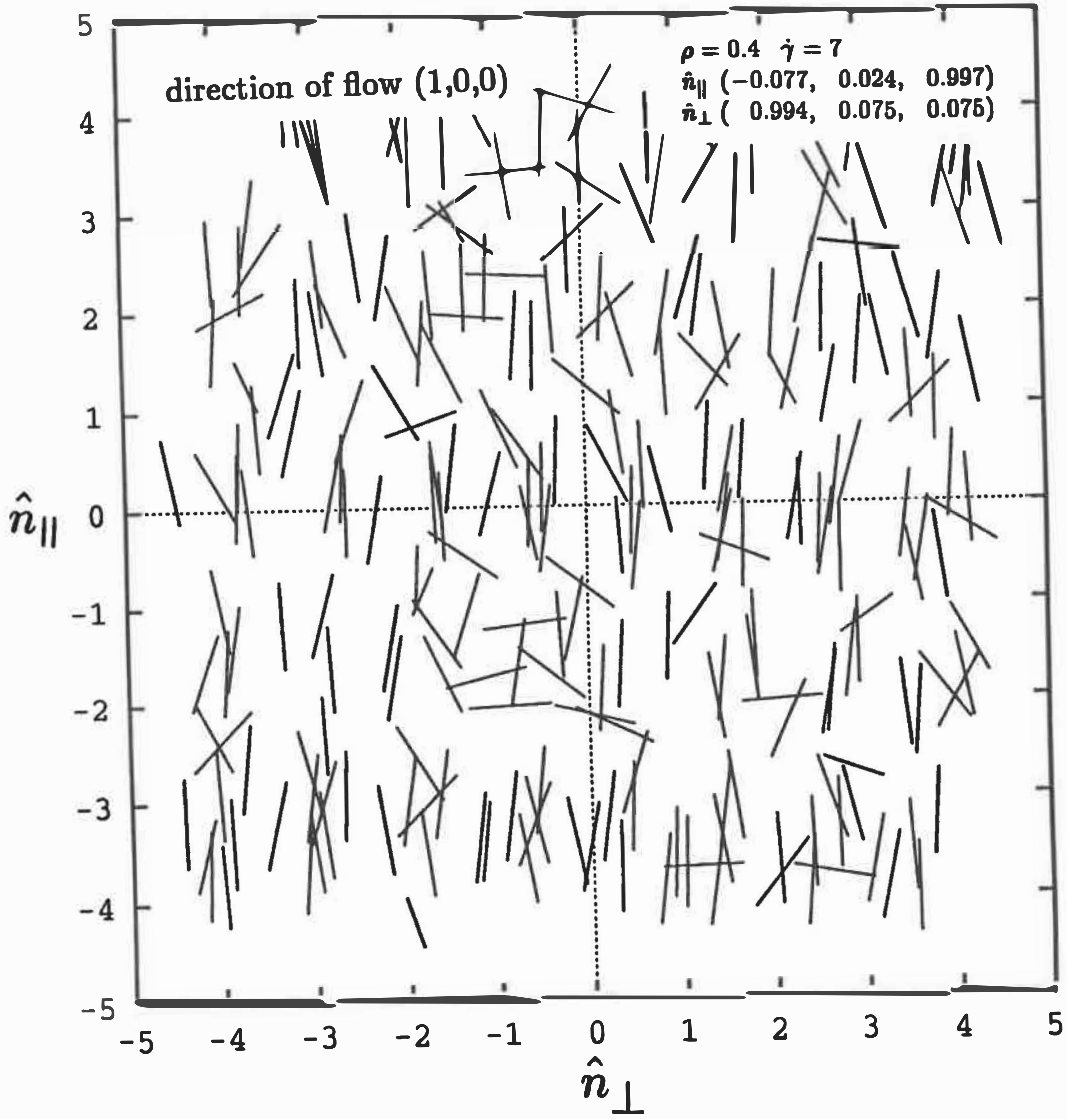

direction of flow (1,0,0)
$\rho = 0.4 \quad \dot{\gamma} = 7$
$\hat{n}_{\parallel}$ (−0.077, 0.024, 0.997)
$\hat{n}_{\perp}$ (0.994, 0.075, 0.075)
$\hat{n}_{\parallel}$
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5
$\hat{n}_{\perp}$

Figure 12h

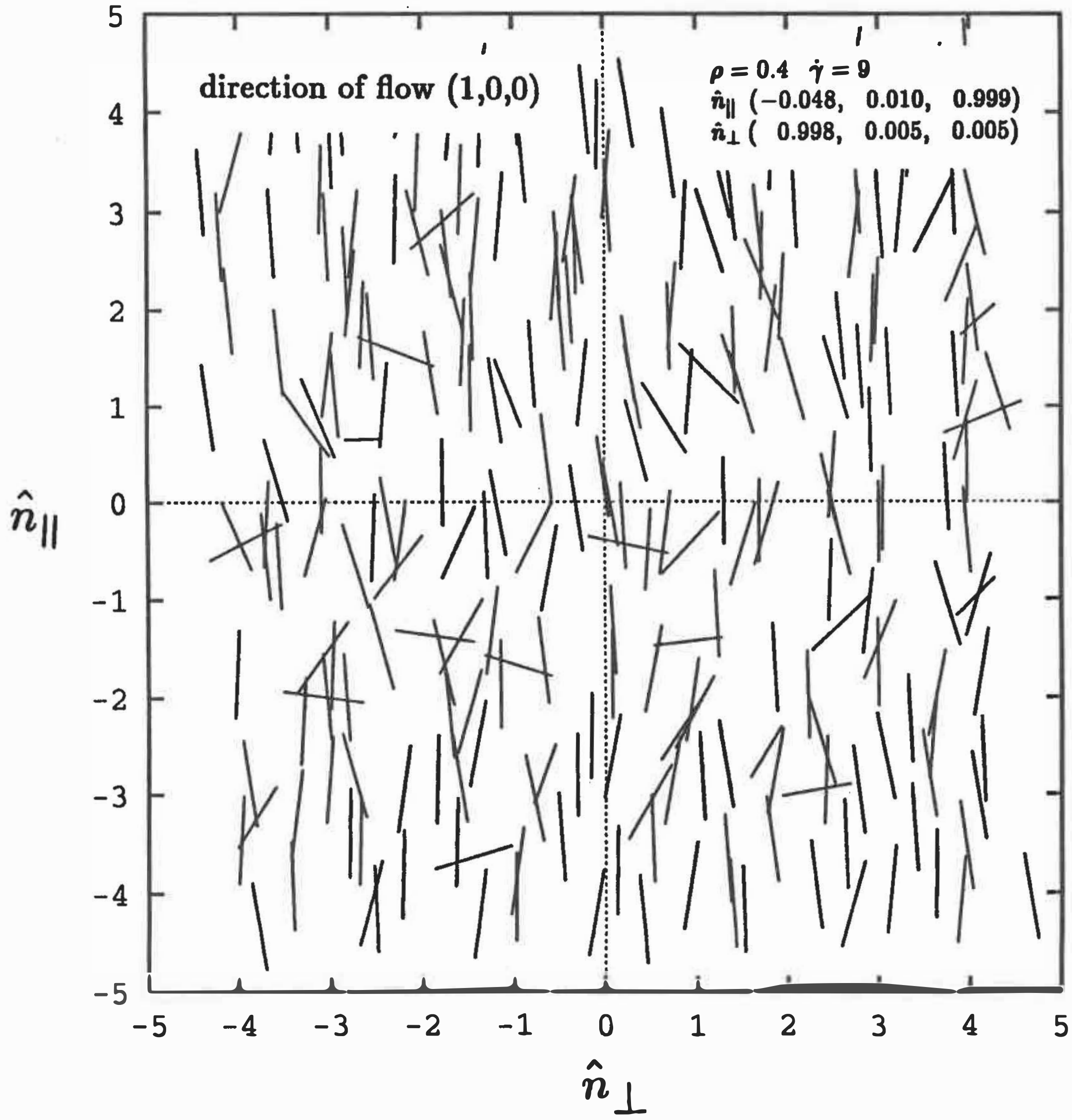

direction of flow (1,0,0)
$\rho = 0.4 \quad \dot{\gamma} = 9$
$\hat{n}_{\parallel}$ (−0.048, 0.010, 0.999)
$\hat{n}_{\perp}$ (0.998, 0.005, 0.005)
$\hat{n}_{\parallel}$
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5
$\hat{n}_{\perp}$

Figure 12i

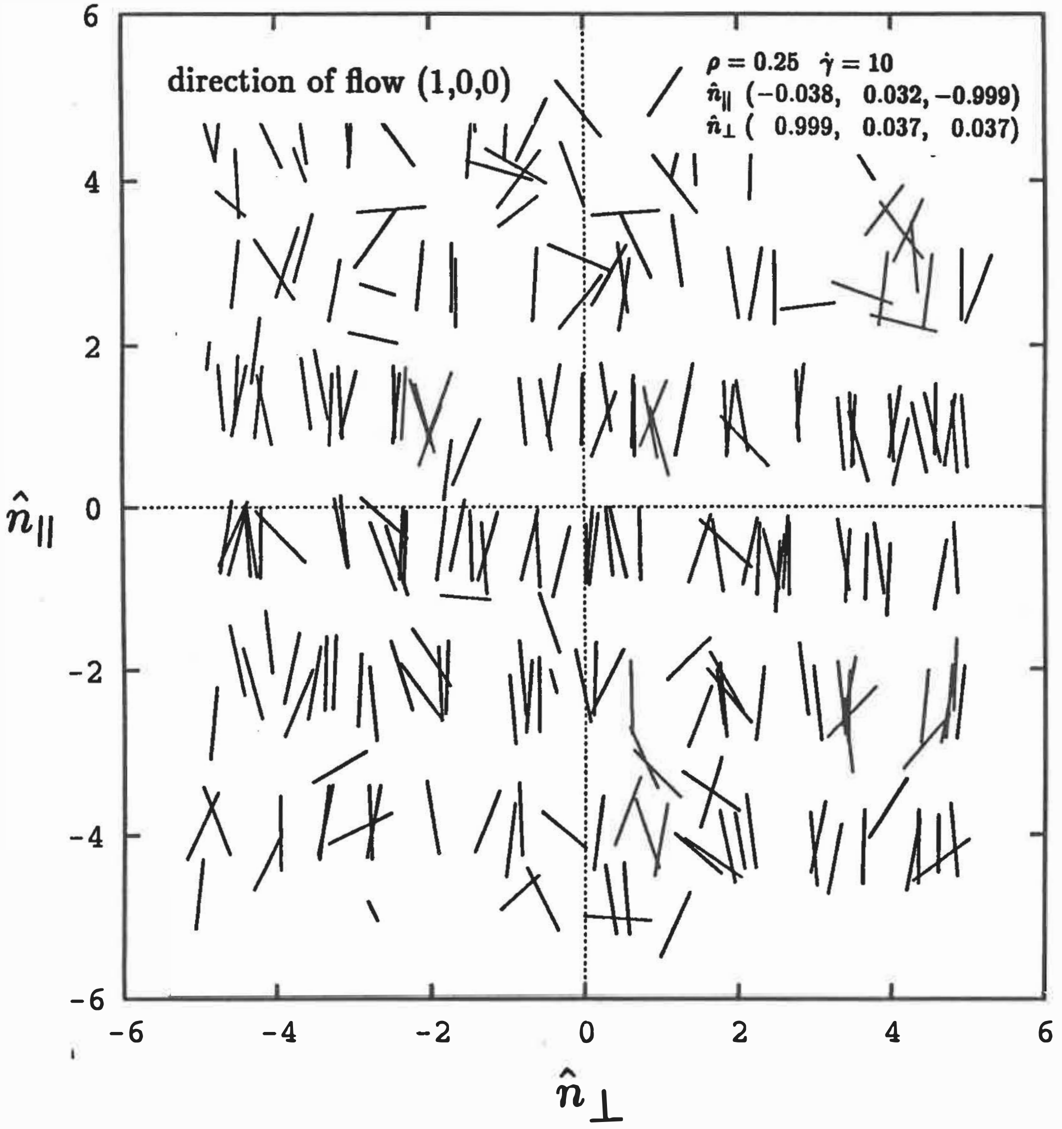

direction of flow (1,0,0)
$\rho = 0.25 \quad \dot{\gamma} = 10$
$\hat{n}_{\parallel}$ (−0.038, 0.032, −0.999)
$\hat{n}_{\perp}$ (0.999, 0.037, 0.037)
6
4
2
0
-2
-4
-6
$\hat{n}_{\parallel}$
-6
-4
-2
0
2
4
6
$\hat{n}_{\perp}$

Figure 12j

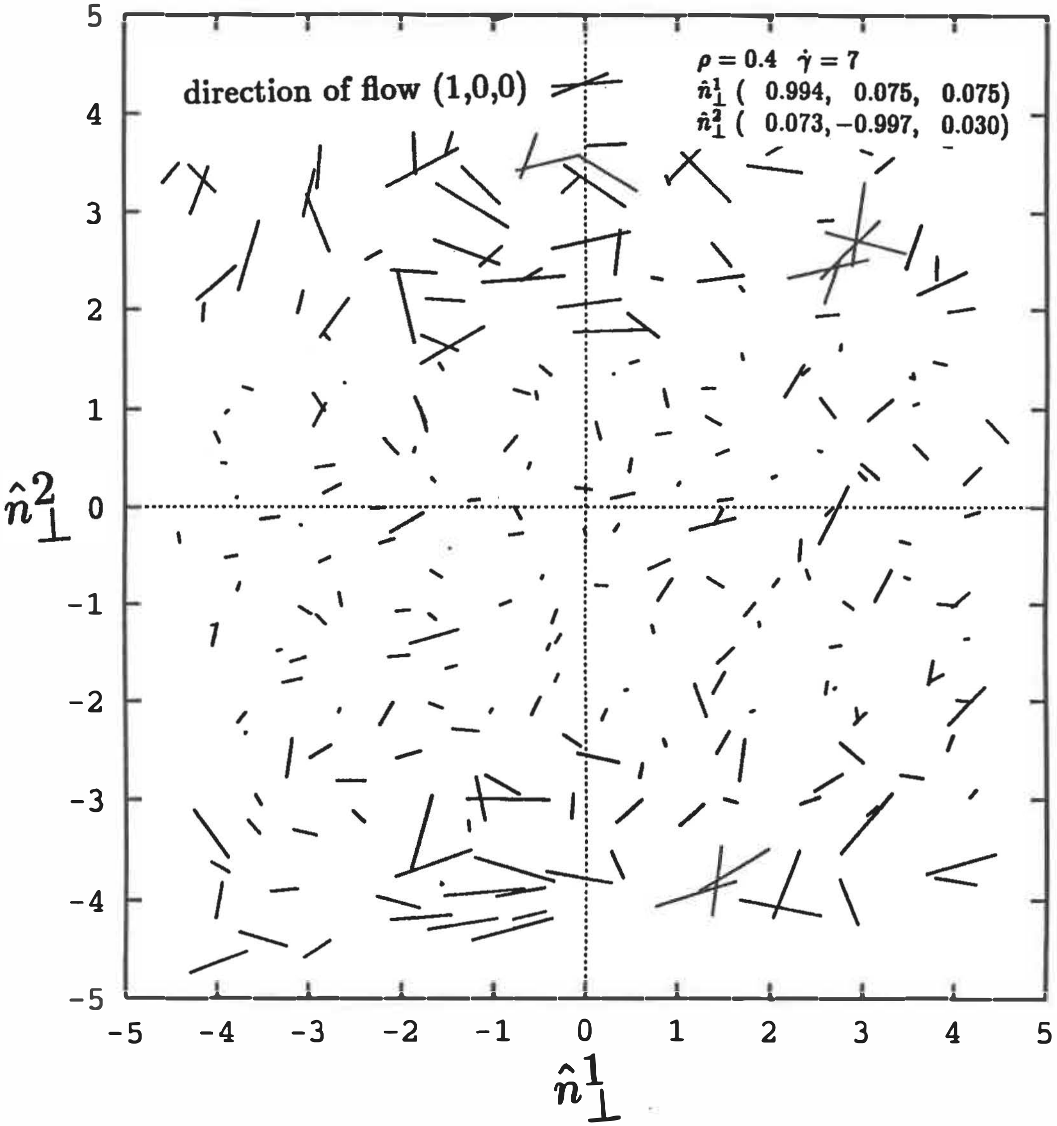

direction of flow (1,0,0)
$\rho = 0.4 \quad \dot{\gamma} = 7$
$\hat{n}^1_\perp$ (0.994, 0.075, 0.075)
$\hat{n}^2_\perp$ (0.073, −0.997, 0.030)
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5
$\hat{n}^2_\perp$
$\hat{n}^1_\perp$

Figure 13

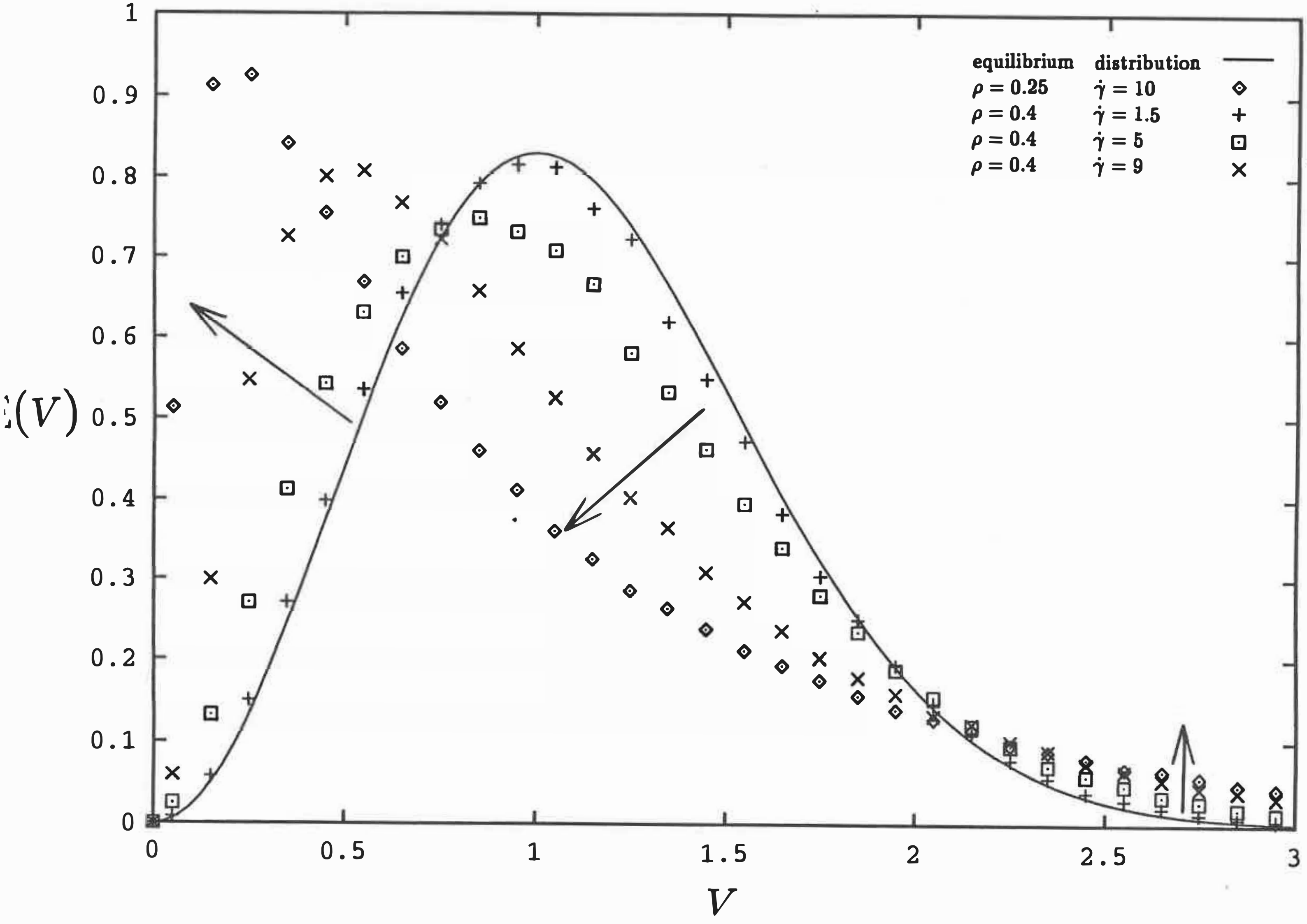

equilibrium distribution
ρ = 0.25 γ̇ = 10
ρ = 0.4 γ̇ = 1.5
ρ = 0.4 γ̇ = 5
ρ = 0.4 γ̇ = 9
(V)
V
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.5
1
1.5
2
2.5
3

Figure 14a

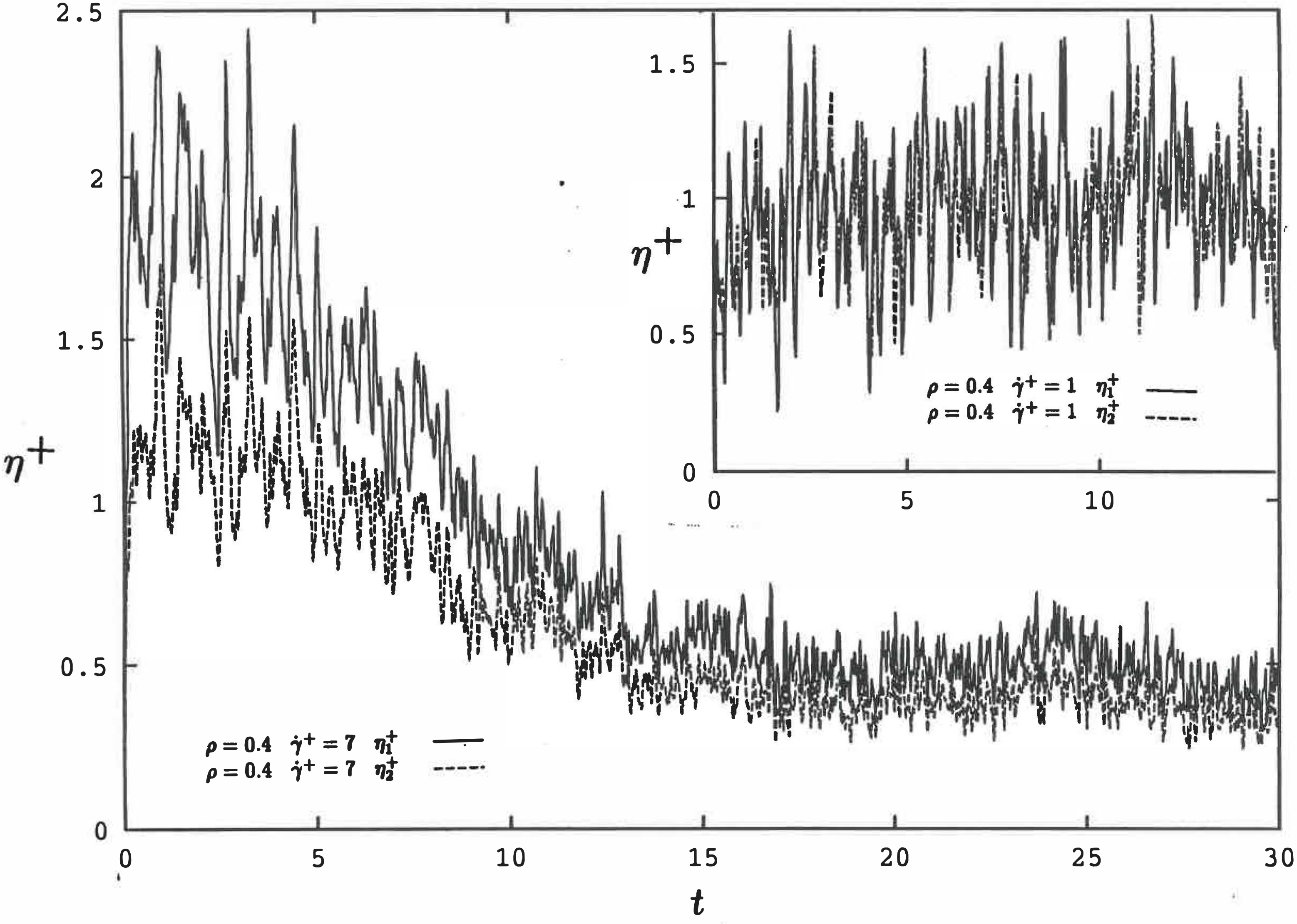

2.5
2
1.5
1
0.5
0
η^+
0
5
10
15
20
25
30
t
$\rho = 0.4$ $\dot{\gamma}^+ = 7$ η_1^+
$\rho = 0.4$ $\dot{\gamma}^+ = 7$ η_2^+
1.5
1
0.5
0
η^+
0
5
10
$\rho = 0.4$ $\dot{\gamma}^+ = 1$ η_1^+
$\rho = 0.4$ $\dot{\gamma}^+ = 1$ η_2^+

Figure 14b

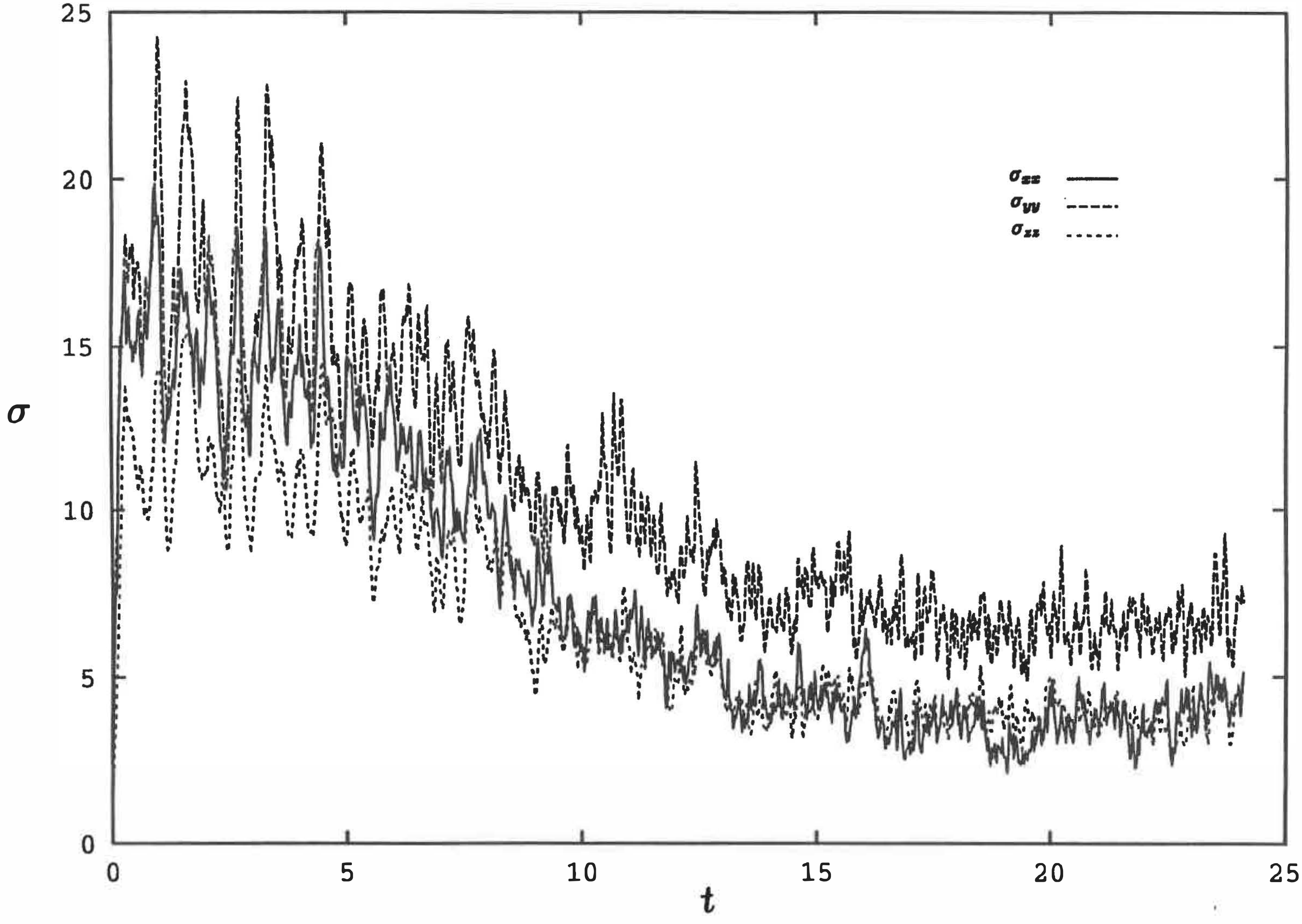
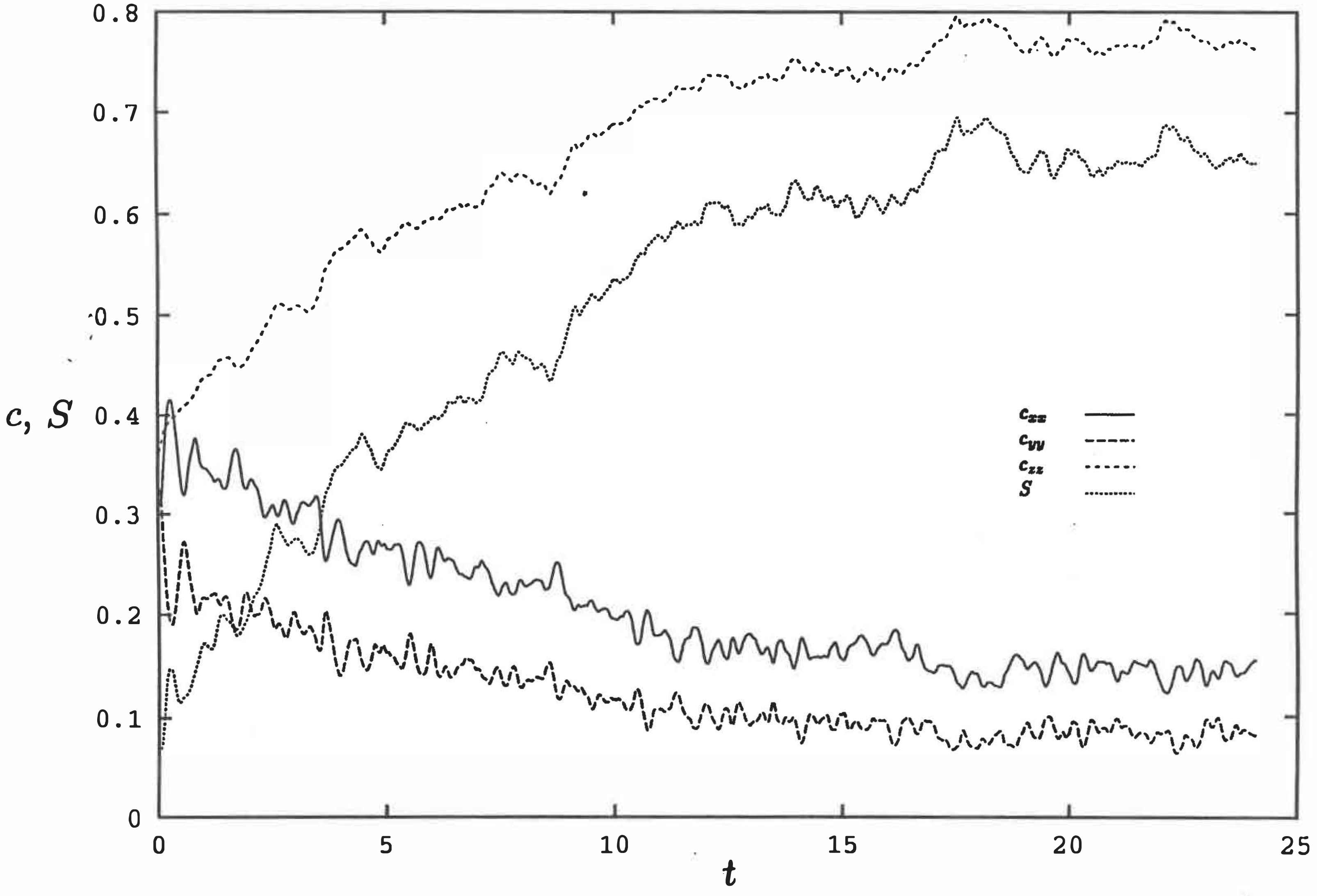


Figure 14c



4 FLOW OF ATOMIC AND FENE LIQUIDS

Direct Numerical Studies of Viscous Flow in Two-Dimensional Geometries

Bogdan Z. Dlugogorski*, Miroslav Grmela and Pierre J. Carreau

Département de génie chimique
Centre de recherche appliquée sur les polymères (CRASP)
École Polytechnique
Case postale 6079, succursale A
Montréal, Québec H3C 3A7, CANADA.

Present address:

Conseil national de recherches
Institut de recherche en construction
Laboratoire nationale de l'incendie
Ottawa, Ontario K1A 0R6, CANADA.

Synopsis

Direct simulations of macromolecular fluids are carried out for flows between parallel plates and in expanding and contracting channels. The macromolecules are modeled as FENE dumbbells with soft disks or Lennard-Jones dumbbell-dumbbell interactions. The results are presented in terms of profiles and contour plots of velocity, pressure, temperature, density, and flow fields. In addition the data for potential energy, shear stress, and the normal components of the stress tensor are collected. In general, an excellent quantitative agreement is found between the simulated profiles and the well known flow structures, such as flow separation and formation of viscous eddies proving that micro-hydrodynamics is a viable tool in linking macroscopic phenomena with the underlying physical mechanisms. The simulations are performed in the Newtonian regime, for medium-size systems comprising up to 3888 dumbbells. This number is sufficiently large to control boundary and particle number effects. The flow is induced by the gravity. The traditional stochastic (thermal) and periodic boundary conditions are employed, but also diffusive boundary conditions, that could include a stagnant fluid layer and repulsive potential walls, are developed. The scaling problems, which are related to the application of a large external force in microscopic system (of the size of the order $100\AA$), result in extreme pressure and temperature gradients. In addition, the viscosity and thermal conductivity coefficients obtained from velocity and temperature profiles of the channel flow are presented. These results are confirmed independently from modeling of Couette flow by the SLLOD equations of motion and from the Evans algorithm for thermal conductivity.

1 Introduction

Classical modeling of fluids proceeds according to the following scheme: (i) physical phenomena whose behavior are determined by interactions of atoms and molecules (always discrete at the microscopic level) → (ii) modeling in the form of continuous partial differential equations → (iii) discretization of the partial differential equations → (iv) solutions on digital computers (Toffoli, 1984). By modeling the flow of liquids directly, that is molecule by molecule, the scheme becomes simplified. The partial differential equations are replaced by a system of ordinary differential equations. Direct modeling can also be seen as a physically motivated discretization. In classical models the discretization plays the role of adjusting the models to the solution tool. In direct models the discretization takes up the role of modeling itself. It has been shown (Hannon, Lie and Clementi, 1988), that the direct approach gives flow predictions that are in qualitative agreement with results of classical hydrodynamics. In this paper, we explore direct modeling of flows of polymeric fluids in two dimensions. Two types of flows are considered: flow between parallel plates as well as in expanding and contracting channels. The macromolecules composing the fluids are modeled as dumbbells. The intermolecular potential (*i.e.* the potential between beads of different dumbbells) is assumed to be either soft disks or the shifted Lennard-Jones. The intramolecular potential (*i.e.* the potential between two beads of the dumbbell) is assumed to be the FENE (finitely extensible nonlinear elastic) potential. In order to be able to compare simple and macromolecular fluids, we also carry out simulations for simple fluids, *i.e.* for fluids composed of beads interacting through the soft disks or the shifted Lennard-Jones potentials.

The simulations are carried out in two dimensions in order to lessen the demand on our computer facilities. At the macroscopic level, in the Navier-Stokes theory, the equations of continuity and motion can be simplified from three into two dimensions if there is a sufficient degree of symmetry in the system and the effects of some of the boundary conditions may be neglected. In other words, the physical behavior of a fluid predicted from 2-D equations is the same as the behavior predicted for the 2-D part of a 3-D system. At the microscopic level, however, this analogy breaks down. The physics represented by the potentials in 2- or 3-D may not be consistent. This is evident in the calculation of transport coefficients which are different for two and three-dimensional fluids even at the same state point. Incidentally, this is reminiscent of many phenomena in physics, compare for example the formula for an electric field around a charged conducting sphere with the expression for an electric field in the vicinity of a charged cylindrical conductor. Thus the rheological and flow data generated for 2-D and 3-D geometries are in general different, at the macroscopic level. The restriction to two-dimensional geometries prevents us from comparing quantitatively our results with results of observations and results from classical models. The quantitative comparison shows a good agreement.

Being unable to compare the NEMD-computed viscosity and thermal conductivity data with the experimental result for simple (usually argon) fluids, we have followed an alternative route. We have calculated coefficients of shear viscosity and thermal conductivity in two dimensions using the SLLOD/LE/GAUSS (Evans, 1986) and Evans (Evans, 1982) algorithms; LE stands for Lees-Edwards shearing blocks boundary conditions (Lees and Edwards, 1972) and GAUSS denotes a method for thermostatting by a means of Lagrangian multipliers (Evans, Hoover, Failor, Moran and Ladd, 1983). These coefficients have been found to correspond very well with the viscosities and thermal conductivities extracted from the velocity and temperature profiles in flow between parallel plates.

The outline of the paper is as follows. The molecular model is introduced in Sec. 2. It consists of the boundary conditions and the equations of motion that incorporate the inter and intraparticle potentials. The passage between the microscopic and the macroscopic levels of description for model fluids and the technical details of simulations are elaborated in Sec. 2.2. Discussion of the channel flow in Sec. 3.1 illustrates the technique, and it is designed to emphasize that direct simulations represent an alternative to the continuum approach. The entry effects in a four-to-one contraction and steady recirculating eddies that develop in flow over a back-step are modeled together in Sec. 3.2 to take advantage of the periodic boundary conditions perpendicular to the direction of flow.

2 Details of the model

2.1 Equations of motion and potentials

Initially, the primitive geometries are filled with disks or dumbbells that emulate either the liquid argon or a model polymeric fluid in two dimensions. There are 4608 disks (2304 dumbbells) between parallel plates and 7776 disks (3888 dumbbells) in contracting and expanding sections of the channel (Tab. 1). A dumbbell is constructed from two disks by introducing stronger bonded interactions between the disks. The state point selected, namely $\frac{k_BT}{\epsilon} = 0.722$ and $\rho\sigma^2 = 0.832$ is the same as the one chosen by Hannon *et al.* (1988). Here, k_B denotes the Boltzmann constant, $\epsilon = 119.8\,k_B$ and $\sigma = 3.405 \times 10^{-10}\,m$ are the Lennard-Jones parameters that correspond to the depth of the potential well and to the collision diameter, T and ρ have their standard meaning of temperature and bead number density; the latter denotes number of single disks in a unit cube with the sides equal to σ. The disks interact either through the soft disks (SD), shifted Lennard-Jones (SLJ) or finitely extensible nonlinear elastic (FENE) (Rudisill and Cummings, 1991) potentials:

$$\phi_{ij}^{SD}(r_{ij}) = \begin{cases} \epsilon(\frac{\sigma}{r_{ij}})^{12} & \frac{r_{ij}}{\sigma} \leq 1.5 \\ 0 & \frac{r_{ij}}{\sigma} > 1.5, \end{cases} \tag{1}$$

$$\phi_{ij}^{SLJ}(r_{ij}) = \begin{cases} 4\epsilon[(\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^{6} - \frac{1}{4}] & \frac{r_{ij}}{\sigma} \leq 2^{\frac{1}{6}} \\ 0 & \frac{r_{ij}}{\sigma} > 2^{\frac{1}{6}}, \end{cases} \tag{2}$$

$$\phi_{ij}^{FENE}(r_{ij}) = \begin{cases} -\frac{H(r_{max}-r_{eq})^2}{2} \, ln\left[1-\left(\frac{r_{ij}-r_{eq}}{r_{max}-r_{eq}}\right)^2\right] & r_{min} < r_{ij} < r_{max} \\ \infty & r_{ij} \leq r_{min} \; or \; r_{ij} \geq r_{max}, \end{cases} \tag{3}$$

where,

$$r_{ij} = \mid \mathbf{r}_i - \mathbf{r}_j \mid .$$

The ratios $\frac{\sigma}{r_{ij}}$ of 1.5 and $2^{\frac{1}{6}}$ define the so-called cut-off radii beyond which there is no inter-molecular interaction; $\mathbf{r}_i$ denotes coordinates of an i^{th} particle. Note that there is no 4 before ϵ in SD potential. In addition, H denotes the spring constant, whereas r_{min} and r_{max} are the minimum and maximum extension of the FENE bond set to 0.9σ and 1.9σ, respectively, r_{eq}=1.4.

Four first order differential equations per molecule are solved either by the Verlet leap-frog method (Allen and Tildesley, 1987), or by the velocity form of the Verlet algorithm (Heermann, 1986),

$$\begin{aligned} \dot{\mathbf{r}}_i &= \mathbf{v_i} \qquad (4) \\ \dot{\mathbf{v}}_i &= \frac{\mathbf{F_i}}{m_i} + \mathbf{g} \qquad (5) \end{aligned}$$

where, $i = 1, \ldots, N$ (N is the number of beads), $\mathbf{g} = (g, 0)$, $\mathbf{F}_i$'s are forces derived from potentials according to:

$$\mathbf{F}_i = \sum_{i=1, i \neq j}^{N} \mathbf{F}_{ij} = - \sum_{i=1, i \neq j}^{N} \frac{1}{m_i} \frac{\partial}{\partial \mathbf{r}_i} \phi_{ij}(r_{ij}). \tag{6}$$

Note that $\phi_i j$ may assume different forms depending if the interaction is of inter (Eqs. 1&2) or intramolecular (Eq. 3) type.

The Verlet algorithms are easier to code than the Gear predictor-corrector method and, even at large time steps, they allows no significant energy drifts. Although, the Gear algorithm produces trajectories closer to the exact solution (Berendsen and van Gunsteren, 1986), the averaged properties obtained along leap-frog computed trajectories are known to be correct. In preliminary runs, for a single harmonic oscillator, we observed that 4 or 5 value Gear methods are more exact and show less energy drift if a single second order rather than two first order equations are solved. Thus for the Gear algorithm, Eqs. 4&5 should be combined as one second order equation, from the numerical point of view. We note in passing, that Eq. 5, solved by the Verlet leap-frog method have no implied time arrow, but the entropy dissipation is achieved through the numerical dispersion (round-off errors) and averaging. In addition, Eq. 5 together with Eqs. 1-3 and 6 exhibits the Galilean invariance, similarly as shown by the macroscopic Navier-Stokes equations.

An empirical, gravity-like force **g** is applied uniformly to all particles in the system, to induce flow; as a matter of fact flows described in Sec. 3.1 and 3.2 should have an adjective phrase "gravity driven" before their names. The minimum value of g is selected in such a manner that the resulting flow patterns (signal) are statistically recognizable against the background fluctuations (noise). In this sense, g decreases with the system size since the magnitude of fluctuations is inversely proportional to the square root of the particle number. The channel flow is investigated for nondimensional g = 0.002, 0.01 0.02 and the flow in a four-to-one contraction is modeled for g = 0.02, 0.05, 0.1. We note that g = 0.02 is of the order of $7.33 \times 10^{13} \frac{m}{s^2}$. It is remarkable that in spite of the extreme gradients, which are necessary to scale the macroscopic geometries to molecular proportions, in terms of the nondimensional quantities, the microscopic and macroscopic fluid structures are surprisingly similar.

The selection of nondimensional time step of between 0.001 and 0.005, used in this work, was based on the considerations relating to the magnitude of the energy drift and to the amplitude of the fluctuations in the total energy. In addition, for FENE dumbbells the time step was further limited to avoid the breaking up of dumbbells that may occur as the outcome of numerical calculations. Specifically, for small values of the spring constant ($H < 10$), the FENE potential curve is essentially flat close to r_{eq} but it rises rapidly close to r_{min} and r_{max}. This asymptotic increase of potential with respect to the dumbbell's extension may be undetected by a numerical algorithm if a time step is too long resulting in floating-point exception errors. On the other hand, for stiffer dumbbells ($H > 1000$), the FENE potential increases gradually and the longer time steps are allowed. Furthermore, larger gravitational forces, say for $g > 0.05$, in conjunction with more flexible bonded interactions tend to facilitate the dumbbells' break-up. If the breaking of dumbbells occurs seldom, it is possible to "widen" the arms of the potential curve for one or two time steps to allow the dumbbell's extension to fall back within the permitted region. In principle, the modeling can be done for very flexible FENE springs and the larger gravities provided that the selected time step is sufficiently short, at the price of longer simulations runs that are necessary for ensuring the statistical validity of the results. This is why we were able to conduct simulations for $H = 37.5$ with $g = 0.05$, in case of the flow between parallel plates, and only for $H = 3750$, for modeling the flow in a contracting/expanding channel.

The SD potential has been thoroughly investigated in the context of the rheological NEMD simulation of two-dimensional fluids, see for example (Evans, 1980; Evans, 1982). The SLJ has a much desired property of no discontinuity in potential and forces at the cut-off radius, and is of significant interest in equilibrium molecular dynamics. The viscosity curves for both potentials will be presented in Sec. 3.1. Small cut-off radii used in this study imply a reduced number of interactions, and more importantly, enable a computationally efficient implementation of cell linked-lists; see App. G in (Bird, 1976) or App. H in (Allen and Tildesley, 1987) for the code. A relatively large number of molecules that we use is mainly possible because of this technique.

It is worth to emphasize that the simulations are unusually straightforward. What is to be done is to solve a system of first order equations, however large, together with the chosen boundary conditions. All thermodynamic and transport properties of model liquids are adjusted solely through potentials (including potentials expressing interactions of molecules with boundaries). Both SD and SLJ signify simple liquids, but in general, they may be replaced by more complex expressions to simulate directly flow of viscoelastic fluids.

In the subsequent sections of this paper, the standard nondimensional quantities (see App. B in (Allen and Tildesley, 1987)), based on ϵ and σ, are used but the asterisks are omitted.

2.2 Boundary conditions and computational details

In this paper we explore four types of boundary conditions. The first type of the boundary conditions are the so-called stochastic or thermal (TB) walls. These boundary conditions have been implemented for both geometries to model the effects of the physical walls. The periodic boundary conditions are implemented perpendicular to flow to allow for the reintroduction of molecules that leave the domain of computation. This is necessary for inducing the flow. Incidentally, if the gravitational force is replaced with mass sinks (vacuum) and with sources of molecules (piston-like), as suggested in the recent literature (Sun and Ebner, 1992), then the density becomes nonuniform along a channel.

The next two types of boundary conditions are put into effect only for modeling of flows between parallel plates. They include either a layer of stagnant fluid (SL) at each plate or a stagnant layer bounded by a repulsive potential wall (PW). Each stagnant layer occupies $\frac{1}{21}$ of the total channel volume. The particles entering the layer have their velocities reset according to the Maxwell-Boltzmann distribution but are free to leave the layer with no velocity adjustment. At every times step, the total translational momentum of all particles in each stagnant layer is ensured to be zero. The upper and the lower stagnant layers are joined to create a periodic-like boundary condition also in the y direction. Particles may diffuse from the bottom stagnant layer to the top stagnant layer and their velocities are not adjusted in any manner during this process. It is clear that for this type of boundary conditions, the domain of calculations is confined to the surface of a torus.

Finally, we introduce yet another variation of the boundary conditions where no diffusion is allowed between the two stagnant layers. Rather two repulsive potential walls (PW) are positioned at a distance of 0.55 above the upper stagnant layer and 0.55 below the lower stagnant layer. The interactions between the potential walls and the approaching particles take place according to the SD potential with the cut-off radius of 2.5.

As shown in Tab. 1, the entire domain of computation is divided into 441 (21×21) boxes for flow between parallel plates and 735 boxes for flow in expanding/contracting channel; there are 15×21 boxes in each expanded subsection, and 15×7 in the contracted subsection.

The instantaneous thermodynamic (*e.g.* potential energy) and rheological (stress tensor) quantities are computed directly in every box, that is with no recourse to distribution functions. The average quantities are then found by dividing each simulation (200,000 or 300,000 time steps) into 10 intervals (bins) consisting of either 20,000 or 30,000 time steps, depending on the run, and taking 10 subaverages for every property in each box. These subaverages are subsequently used to compute the overall average and the standard deviation. In case of flow between parallel plates, in order to reduce the statistical uncertainty, the averages are calculated for horizontal layers, each including 21 boxes, and then presented in form of profiles (Fig. 1). The data for flow in expanding/contracting channel are presented in terms of isolines obtained by intrapolating the box averages. Since a number of molecules that could be found in any given box is small the fluctuations are quite significant, especially for $g = 0.02$. This is why the isolines for $g = 0.02$ are not as smooth as for the other values of the gravity.

It should be stressed that every simulation consists of three steps. During the equilibration, the formation of the velocity and temperature profiles is monitored and the execution is terminated once the steady state is reached. In the second stage, the average laboratory velocities are calculated in every box to be used afterwards in the final run in computation of the kinetic temperature (T), and the kinetic part of the stress tensor ($\underline{\underline{\sigma}}$). Thus in this paper, both T and $\underline{\underline{\sigma}}$ are based on the peculiar rather than laboratory velocities. In nondimensional quantities, for two-dimensional systems of particles, for a k^{th} box,

$$
\begin{aligned}
T^k &= \frac{1}{2\,N^k} < \sum_{i=1}^{N^k} (v_{i\alpha} - u_\alpha(\mathbf{r}))\,(v_{i\alpha} - u_\alpha(\mathbf{r})) >_t, && (7)\\
\sigma^k_{\alpha\beta} &= -\frac{1}{A^k} < \sum_{i=1}^{N^k} (v_{i\alpha} - u_\alpha(\mathbf{r}))\,(v_{i\beta} - u_\beta(\mathbf{r})) + \sum_{i=1}^{N^k} \sum_{i \neq j} r_{ij\alpha} F_{ij\beta} >_t . && (8)
\end{aligned}
$$

In the first expression the summation convention is assumed. The laboratory velocity field is denoted by $\mathbf{u}(\mathbf{r})$, $_t$ stands for the averaging procedure described above. The mass of a single bead is unity, A^k is an area of a single box, and N^k is the number of beads in a k^{th} box. The second summation in the potential contribution to the stress tensor (the last term on the right hand side in Eq. 8) is carried over all interactions within a box and over interactions between beads inside the box and beads in surrounding boxes; this is handled in a CPU-economic manner by a means of the linked-cell approach. If the interaction takes place between two beads that belong to separate boxes than the potential contribution to the stress tensor is apportioned equally to each box. For the SLLOD modeling, Eqs. 7&8 are simplified for two reasons. Firstly, the SLLOD method implies homogeneous shear and the domain of simulation (primitive cell) is not subdivided into boxes. Secondly, the SLLOD equations of motion are written in terms of peculiar rather than laboratory momenta and thus only two runs (equilibration and production) are required since $\mathbf{u}(\mathbf{r})$ in Eqs. 7&8 vanishes.

3 Results and discussion

3.1 Flow between parallel plates

It has been demonstrated that viscosities and thermal conductivities in three dimensions obtained from direct simulations of the flow between parallel plates, correspond perfectly to the available experimental data (Hannon, Lie and Clementi, 1986). On the other hand, the transport coefficients in two dimensions may not be calculated from the equilibrium molecular dynamics since the Green-Kubo equations predict the coefficients that diverge at low shear rates; flows in two dimensions at low shear rates are associated with inherent instabilities that are not accounted for in the Green-Kubo equations (Evans and Morriss, 1983). In this work we report the transport coefficients in the Newtonian limit that are calculated by both the direct simulations and the homogeneous NEMD algorithms.

The transport coefficients of a model fluid can be adjusted at the microscopic level by selecting inter and intramolecular potentials. Distinct potentials result in dissimilar viscosities and thermal conductivities, and eventually in different flow patterns. For this reason, in Figs. 1 a&b and 2 a&b one observes a variety of velocity and temperature profiles; due to the presence of the stagnant layers the profiles are in general less pronounced than for flows between thermal boundaries. It is not surprising that runs characterized by small viscosities (Tab. 2) have large centerline velocities and temperatures. For example the soft disk potential generates the least viscous fluids (Fig. 1 and Tab. 2). The profiles produced by the molecular FENE liquids depend on the value of the spring constant. For $H = 37.5$, the centerline velocity is slightly lower but at $H = 3750$ noticeably higher than for SD atomic fluids; see Fig. 1b for $g = 0.01$. Although, the temperature at the wall is maintained at 0.722 due to the viscous dissipation caused by the high gravity force, the temperature curves trace well defined fourth order polynomials, as expected from the standard hydrodynamics (Fig. 2 a&b). For all simulations of FENE fluids the average rotational velocities differ by a factor of 2 from the vibrational velocities.

For positions of −30 to 30 along the plate-plate distance, the effect of the boundary condition on the fluid density (Fig. 3) and pressure (Fig. 4) is small and thus one can fit the quadratic and fourth order polynomials to velocity and temperature profiles, respectively (Hannon *et al.*, 1986; Hannon *et al.*, 1988); the pressure is defined as minus half of the trace of the stress tensor. Using this approach, the transport coefficients are computed and summarized in Tab. 2. In addition, the viscosity can be calculated from the knowledge of the shear stress and velocity profiles (Tab. 2, col. 5), directly from the definition. For Newtonian fluids - $\sigma_{yx} = \mu \frac{dv_x}{dy}$. The shear stress profile in the channel (not shown) is linear with the exception of the boundary layer where the values of σ_{yx} are altered by the influence of the boundary conditions.

In order to verify the results of the direct modeling, we have carried out simulations with SLLOD and Evans algorithms, for data points that correspond to the density and the temperature at the center of the channel. The Newtonian viscosities have been obtained by extrapolating the viscosity curves to shear rates approaching zero (Fig. 5a). Similarly, by extrapolating the external field strength to zero (not shown) the thermal conductivity coefficients were computed. These results are assembled in Tab. 3. Obviously, the data in Tabs. 2&3 are very similar, a sign that the direct and homogeneous NEMD algorithms are consistent. This is analogous to real experiments, when one or several laboratory techniques may be used to investigate one phenomenon at the same time. As an additional benefit, the SLLOD algorithm gives the pressure (Fig. 5b) and the potential energy per particle (not shown), which correspond very well to the values generated by the direct approach (Fig. 4a and Tabs. 2&3).

Finally, as shown in Fig. 6, dumbbells that are closer to the centerline tend to be longer and those near the boundaries shorter, on the average. The large error bars associated with three data points in Fig. 6 indicate that the numerical algorithm experiences difficulties due to the small value of the spring constant.

3.2 Flow in contraction and in expansion

The entry flows and flows over a back-step, considered as benchmark problems in computational fluid mechanics, have sparked a significant experimental interest, especially within the context of polymeric fluids (Boger, 1987; White, Gotsis and Baird, 1987). The geometry of the channel and the type of the boundary conditions used in this study are shown in Fig. 7. Each channel wall is built from four stochastic (thermal) plates. Other kinds of the boundary conditions have not been modeled since the objective has been to observe the formation of flow structures rather than analyze interactions of particles with physical boundaries. The domain of calculations is divided into 735 boxes as pictured in Fig. 7 by horizontal and vertical solid lines. This number of boxes is a compromise between the magnitude of fluctuations in every box and our desire to obtain well defined contour plots for the properties of interests.

Our results are generated for both dumbbells and atomic fluids in a channel characterized by a four-to-one contraction. Direct modeling of flow of atomic liquids is performed for the shifted Lennard-Jones as well as the soft disks potentials. As before a model polymeric particle is constructed with the help of the FENE potential and the interaction between two model particles are accounted for by a sum of four bead-bead forces derived from the SD potential. The simulations are performed for low Reynolds numbers; for example, we estimate that the Reynolds numbers associated with the flow of SLJ atomic particles are approximately 2 for $g = 0.02$ and around 12 for $g = 0.1$, for runs 218 and 242, respectively.

From Figs. 8, a-f which present the comparison of the flow patterns, it follows that in the two corners of the upstream part of the channel, two recirculating vortices could be formed, especially for $g = 0.02$ and 0.05, but the large fluctuations in these areas make drawing definitive conclusions not possible. On the other hand, the entry length L_e (Fig. 9a) is estimated to be 0.6, for $g = 0.02$; L_e is the distance between the beginning of the contracted section and the location where the fully developed flow is first observed. This seems to be consistent with values observed in macroscopic experiments and calculated by the traditional algorithms of the computational fluid dynamics, (see for example - $L_e/R = 0.49+0.11N_{Re}$ - in (Boger, Hur and Binnington, 1986)). Unfortunately, at the higher Reynolds numbers the contracted section is too short to allow for the formation of the fully developed flow and to avoid the effects of the sudden expansion farther downstream. Thus for other runs the estimation of the entry length was not attempted. The characteristic recirculating eddies form inside the back-step and their size is related to the imposed gravitational force and the underlying microscopic potentials; the eddies for the atomic SD fluids are the largest and those for SLJ are the smallest. The larger fluctuations that appear to be present for FENE simulations (compare Fig. 8c&8d) are attributed to the smaller time step and hence shorter computational runs (Tab. 1).

For $g = 0.02$, the signal to noise ratio was insufficiently high to result in smooth isolines of the physical quantities (Figs. 8-12) . Upon increasing g to 0.05 and 0.1, the viscous dissipation becomes significant, as was also the case for the flow between parallel plates. The highest temperatures exist along the centerline of the expanded sections, perhaps due to the fact that the thermal boundaries (that act as energy sinks) are far away (Figs. 10 a-c). One notes that the isolines of the rotational (Fig. 9d) and vibrational (not shown) velocities follow the contours of the temperature, which was calculated from the translational kinetic energy. This observation indicates that the temperature of the dumbbell fluid could have been based on the internal motion of dumbbells. The temperature and pressure fields influence significantly the density of the fluids (compare Figs. 11 with 10 and 12).

The impact of the temperature on the density appears to be more pronounced; this may be readily noticed at the boundaries where the fluids are more dense (Figs.11 a-c). It is worthwhile to point out that at the entry to the contracted section the pressure builds up.

No alignment of dumbbells is observed in the bulk flow, but in the vicinity of the thermal plates dumbbells are aligned parallel to walls, on the average. Lastly, the potential contribution to the diagonal and off-diagonal components of the stress tensor are approximately five times as significant as the kinetic contribution.

4 Concluding remarks

Results of this paper indicate that direct simulations represent a viable alternative method of calculating flows of rheologically complex fluids. Its main advantage is that partial differential equations are completely avoided. The discrete equations that serve as an input for digital computers have a clear physical meaning, and the modeling can be made easily faithful to the complexity of intermolecular, intramolecular and molecules-boundary interactions. At this stage however some problems that remain unsolved in direct simulations prevent a widespread use of the method. Among these problems we mention in particular the problem of treating effectively the incompressibility and the problems related to the concepts of entropy and temperature. These problems represent an interesting challenge. Our strategy to solve these open problems is to combine appropriately (in the spirit of (Grmela, 1993)) the classical and the direct simulations methods.

The analysis of the generated data results in the following conclusions:

1. In the Newtonian limit, the NEMD simulations of flow and rheology of fluids composed of disks lead to similar coefficients of viscosity. Similarly, a close agreement is found between coefficients of thermal conductivity calculated from the temperature profiles and from the Evans algorithm. These results indicate that

the NEMD algorithms are consistent among themselves.

2. In case of the flow between parallel plates, stagnant layers of fluid at the plates affect the density and pressure profiles less that the thermal boundaries do.

3. Observations of the flow structures in 2-D contraction/expansion geometries confirm that the direct simulations and the classical hydrodynamics have common microscopic origins. The realistic flow predictions from the direct simulations have been obtained for large gravitational forces which are necessary in systems consisting of small (in comparison to Avogadro) number of molecules.

5 Acknowledgments

This research was supported by the *Natural Sciences and Engineering Research Council of Canada* and *Fonds pour la formation de chercheurs et l'aide à la recherche* of the Province of Québec.

References

Allen, M. P. and D. J. Tildesley, Computer Simulation of Liquids, Clarendon Press, Oxford (1987).

Berendsen, H. J. C. and W. F van Gunsteren, Practical algorithms for dynamic simulations, *in* G. Ciccotti and W. Hoover, eds, "Molecular-Dynamics Simulation of Statistical-Mechanical Systems," North-Holland, Oxford, pp. 43–65 (1986).

Bird, G. A., Molecular Gas Dynamics, Clarendon Press, Oxford (1976).

Boger, D. V., "Viscoelastic flows through contractions," Ann. Rev. Fluid Mech. **19**, 157–182 (1987).

Boger, D. V., D. U. Hur, and R. J. Binnington, "Further observations of elastic effects in tubular entry flows," J. Non-Newt. Fl. Mech. **20**, 31–49 (1986).

Evans, D. J., "Nonlinear viscous flow in two-dimensional systems," Phys. Rev. A **22**(1), 290–294 (1980).

Evans, D. J., "Homogeneous NEMD algorithm for thermal conductivity - application of non-canonical linear response theory," Phys. Lett. A **91**(9), 457–460 (1982).

Evans, D. J., Nonequilibrium molecular dynamics, *in* G. Ciccotti and W. Hoover, eds, "Molecular-Dynamics Simulation of Statistical-Mechanical Systems," North-Holland, Oxford, pp. 221–240 (1986).

Evans, D. J. and G. P. Morriss, "Nonequilibrium molecular-dynamics simulation of Couette flow in two-dimensional fluids," Phys. Rev. Lett. **51**(19), 1776–1779 (1983).

Evans, D. J., W. G. Hoover, B. C. Failor, B. Moran, and A. J. C. Ladd, "Nonequilibrium molecular dynamics *via* Gauss's principle of least constraint," Phys. Rev. A **28**, 1016–1021 (1983).

Grmela, M., "Coupling between microscopic and macroscopic dynamics in NEMD," Phys. Lett. A **174**, 59–65 (1993).

Hannon, L., G. C. Lie, and E. Clementi, "Molecular dynamics simulation of channel flow," Phys. Lett. A **119**(4), 174–177 (1986).

Hannon, L., G. C. Lie, and E. Clementi, "Micro-hydrodynamics," J. Stat. Phys. **51**(5/6), 965–979 (1988).

Heermann, D. W., Computer Simulation Methods in Theoretical Physics, Springer Verlag, Berlin (1986).

Lees, A. W. and S. F. Edwards, "The computer study of transport processes under extreme conditions," J. Phys. C: Solid State Phys. **5**, 1921–1929 (1972).

Rudisill, J. W. and P. T. Cummings, "The contribution of internal degrees of freedom to the non-Newtonian rheology of model polymer fluids," Rheol. Acta **30**, 33–43 (1991).

Sun, M. and C. Ebner, "Molecular-dynamics simulation of compressible fluid flow in two-dimensional channels," Phys. Rev. A **46**(8), 4813–4818 (1992).

Toffoli, T., "Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics," Physica D **10**, 117–127 (1984).

White, S. A., A. D. Gotsis, and D. G. Baird, "Review of the entry flow problem: experimental and numerical," J. Non-Newt. Fl. Mech. **24**, 121–160 (1987).

Table 1: Summary of the computational details. TB denotes thermal boundary at the channel's wall, SL is an abreviation for a stagnant layer and SL/PW signifies a stagnant layer in conjuction with a repulsive wall (the periodic boundary conditions are always used in direction perpendicular to flow). The intermolecular potentials are either soft disks (SD) or shifted Lennard-Jones (SLJ), and the intramolecular potential is the finitely extensible nonlinear elastic (FENE).

run no.	geometry	gravity	time step	potential	boundaries	molecules	boxes
217	plates	0.02	0.005	SD	TB	4608	441
211	plates	0.02	0.005	SLJ	TB	4608	441
322	plates	0.02	0.005	SLJ	SL	4608	441
221	plates	0.002	0.005	SLJ	TB	4608	441
g05*	plates	0.01	0.005	SD	SL/PW	4608	441
g08*	plates	0.02	0.005	SD	SL/PW	4608	441
l05*	plates	0.01	0.001	SD/FENE	SL/PW	2304	441
h03*	plates	0.01	0.0025	SD/FENE	SL/PW	2304	441
h07*	plates	0.02	0.0025	SD/FENE	SL/PW	2304	441
218	cont./exp.	0.02	0.005	SLJ	TB	7776	735
242	cont./exp.	0.1	0.005	SLJ	TB	7776	735
fe5*	cont./exp.	0.05	0.005	SD	TB	7776	735
co7*	cont./exp.	0.1	0.0025	SD	TB	7776	735
lo4*	cont./exp.	0.05	0.0025	SD/FENE	TB	3888	735
co3*	cont./exp.	0.1	0.0025	SD/FENE	TB	3888	735

* For these simulations the velocity form of the Verlet algorithm was used, the production runs lasted 200,000 steps. Otherwise, the leap frog algorithm was employed and the duration of the production runs was 300,000 steps.

Table 2: Transport coefficients and thermodynamic properties extracted from the simulations; numbers in subscript denote one standard deviation.

run no.	pressure	potential	viscosity (vel.)	viscosity (stress)	ther. conductivity
217	13.067_{13}	2.1477_{10}	1.832_{84}	1.834_{11}	11.34_{39}
211	11.047_{12}	1.0550_{83}	2.6787_{89}	2.680_{11}	13.25_{43}
322	9.2653_{54}	0.842_{13}	2.760_{17}	2.760_{20}	11.72_{22}
221	5.6787_{29}	0.44309_{39}	3.548_{50}	3.557_{79}	5.49_{13}

Table 3: Transport coefficients and thermodynamic quantities obtained from SLLOD/LE/GAUSS and Evans's algorithm for thermal conductivity; numbers in subscript denote one standard deviation.

potential	density	temperature	pressure	potential	viscosity	ther. conductivity
SD	0.798	3.473	13.0502_{49}	2.1466_{10}	1.83_{10}	11.36_{32}
SLJ	0.802	2.146	11.0278_{67}	1.0580_{10}	2.576_{55}	13.78_{39}
SLJ	0.807	1.626	9.2366_{82}	0.8460_{12}	2.66_{13}	12.16_{13}
SLJ	0.821	0.745	5.6963_{34}	0.4490_{38}	3.209_{88}	$6.1_{1.3}$

Figure Captions

- Figure 1. The translational (a,b), rotational and vibrational (c) velocity profiles for atomic (a,b) and molecular (b,c) fluids obtained by direct simulations of flow between two parallel plates as a function of the position inside the channel. In figures that follow TB, SL and PW denote thermal boundary, stagnant layer and potential walls. SD and SLJ are soft disks and shifted Lennard-Jones intermolecular potentials, whereas FENE signifies finitely extensible nonlinear elastic intramolecular potential. All data are in nondimensional quantities.

- Figure 2 a,b. Comparison between the temperature profiles generated by direct simulation of flow between parallel plates for different atomic (a,b) and dumbbell fluids (b).

- Figure 3 a,b. Profiles of the particle number density in the channel between parallel plates show the effect of the temperature and the type of boundary conditions.

- Figure 4 a,b. For the flow between parallel plates the pressure is constant except at the plates.

- Figure 5. Results generated by 2-D version of the SLLOD/LE/GAUSS algorithm in terms of (a) viscosity, and (b) pressure plotted against the shear rate.

- Figure 6. The average dumbbell extension is determined by the dumbbell's location in the flow field and by the rigidity of it's intramolecular bond.

- Figure 7. The domain of simulation and the boundary conditions for modeling of the contraction/expansion problem.

- Figure 8. Formation of the flow structures for the gravity induced flow of atomic fluids described by the shifted Lennard-Jones (a,b), and the soft disks (c,d) potentials. The molecular fluids characterized by the FENE intramolecular and SD intermolecular potentials are shown in (e,f).

- Figure 9. The translational (a,b,c) and the rotational (d) velocity field for atomic (a,b) and molecular (c,d) fluids.

- Figure 10 a-c. Isolines of the temperature, which is based on the translational kinetic energy, exhibit the consequences of the viscous heating at the channel's center, and cooling at the boundaries.

- Figure 11 a-c. The plots of particle number density show the influence of both the temperature and pressure.

- Figure 12 a-c. The effect of the boundary conditions is more pronounced for atomic fluids, especially for the lower values of the gravity.

Figure 1a

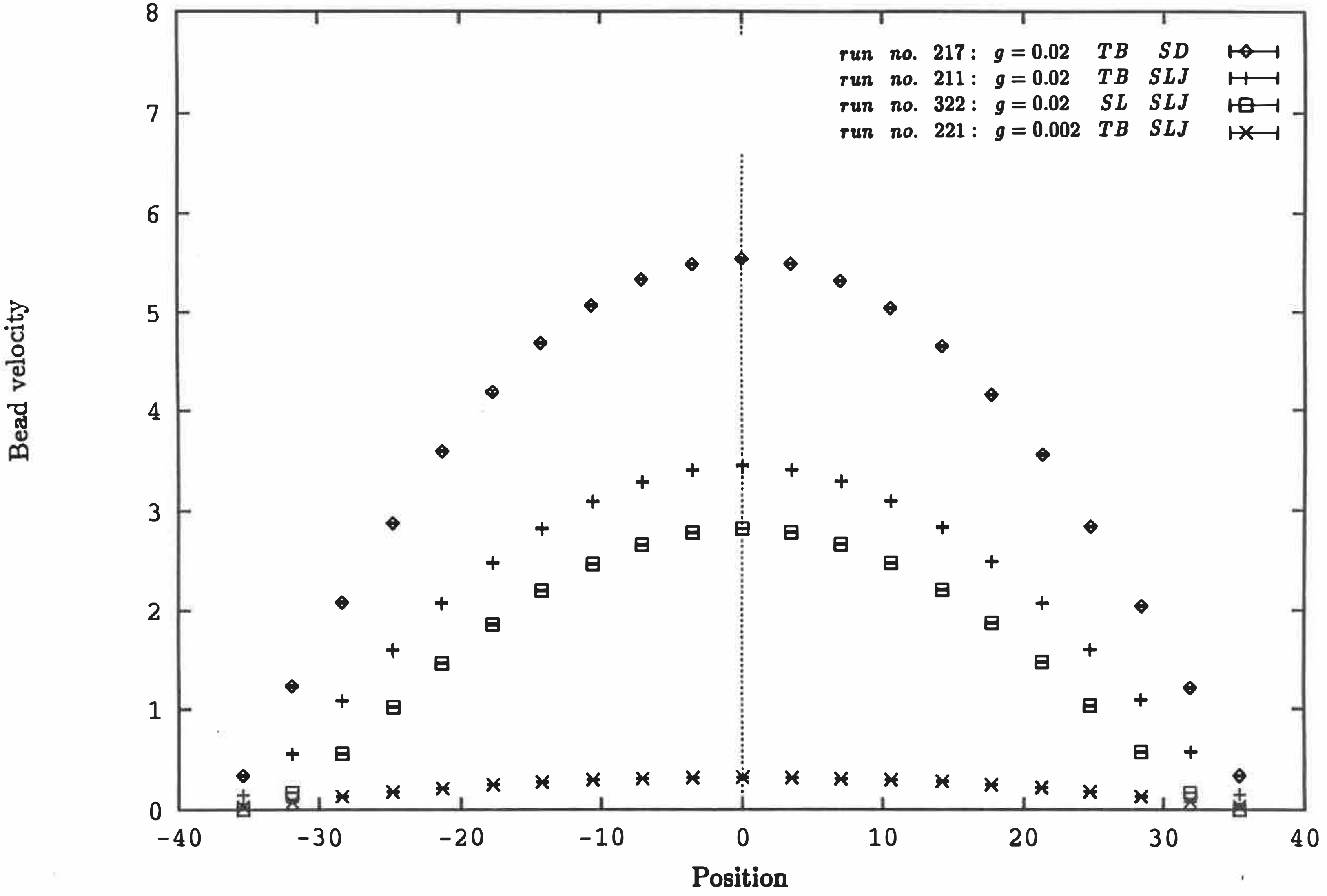

run no. 217 : g = 0.02 TB SD
run no. 211 : g = 0.02 TB SLJ
run no. 322 : g = 0.02 SL SLJ
run no. 221 : g = 0.002 TB SLJ
Bead velocity
0
1
2
3
4
5
6
7
8
-40
-30
-20
-10
0
10
20
30
40
Position

Figure 1b

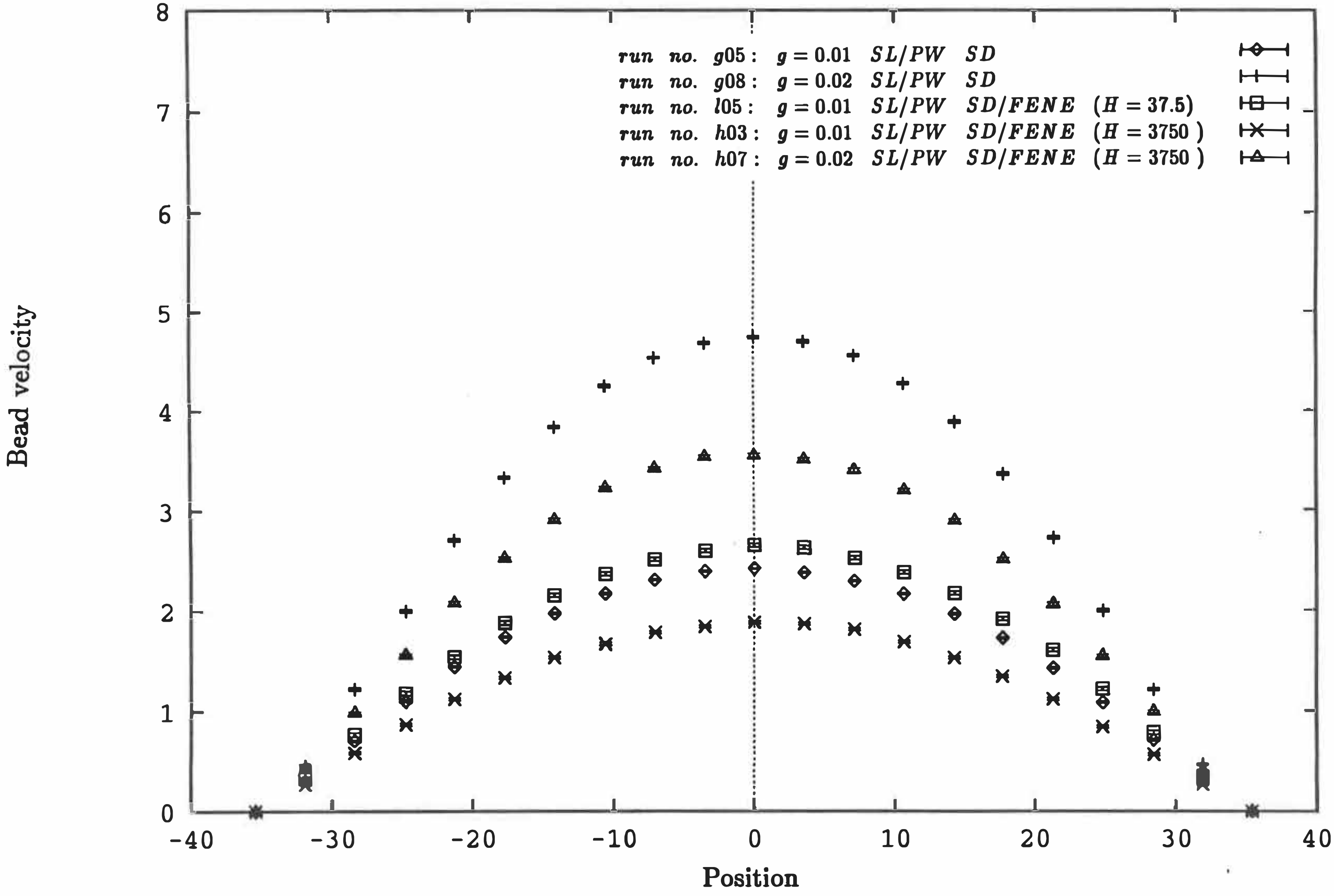

Bead velocity
Position
8
7
6
5
4
3
2
1
0
-40
-30
-20
-10
0
10
20
30
40
run no. g05 : g = 0.01 SL/PW SD
run no. g08 : g = 0.02 SL/PW SD
run no. l05 : g = 0.01 SL/PW SD/FENE (H = 37.5)
run no. h03 : g = 0.01 SL/PW SD/FENE (H = 3750)
run no. h07 : g = 0.02 SL/PW SD/FENE (H = 3750)

Figure 1c

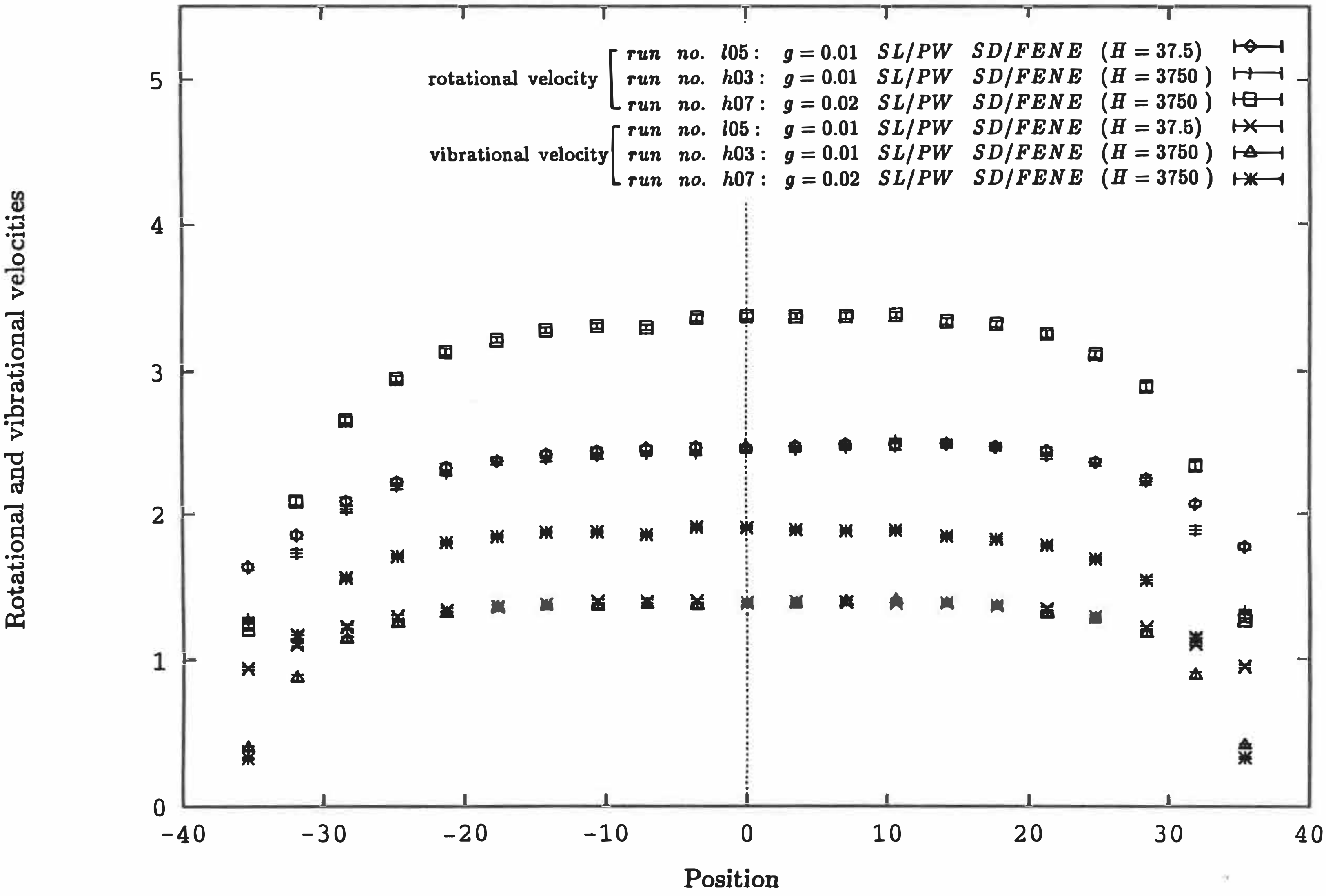

rotational velocity
run no. l05 : g = 0.01 SL/PW SD/FENE (H = 37.5)
run no. h03 : g = 0.01 SL/PW SD/FENE (H = 3750)
run no. h07 : g = 0.02 SL/PW SD/FENE (H = 3750)
vibrational velocity
run no. l05 : g = 0.01 SL/PW SD/FENE (H = 37.5)
run no. h03 : g = 0.01 SL/PW SD/FENE (H = 3750)
run no. h07 : g = 0.02 SL/PW SD/FENE (H = 3750)
Rotational and vibrational velocities
0
1
2
3
4
5
-40
-30
-20
-10
0
10
20
30
40
Position

Figure 2a

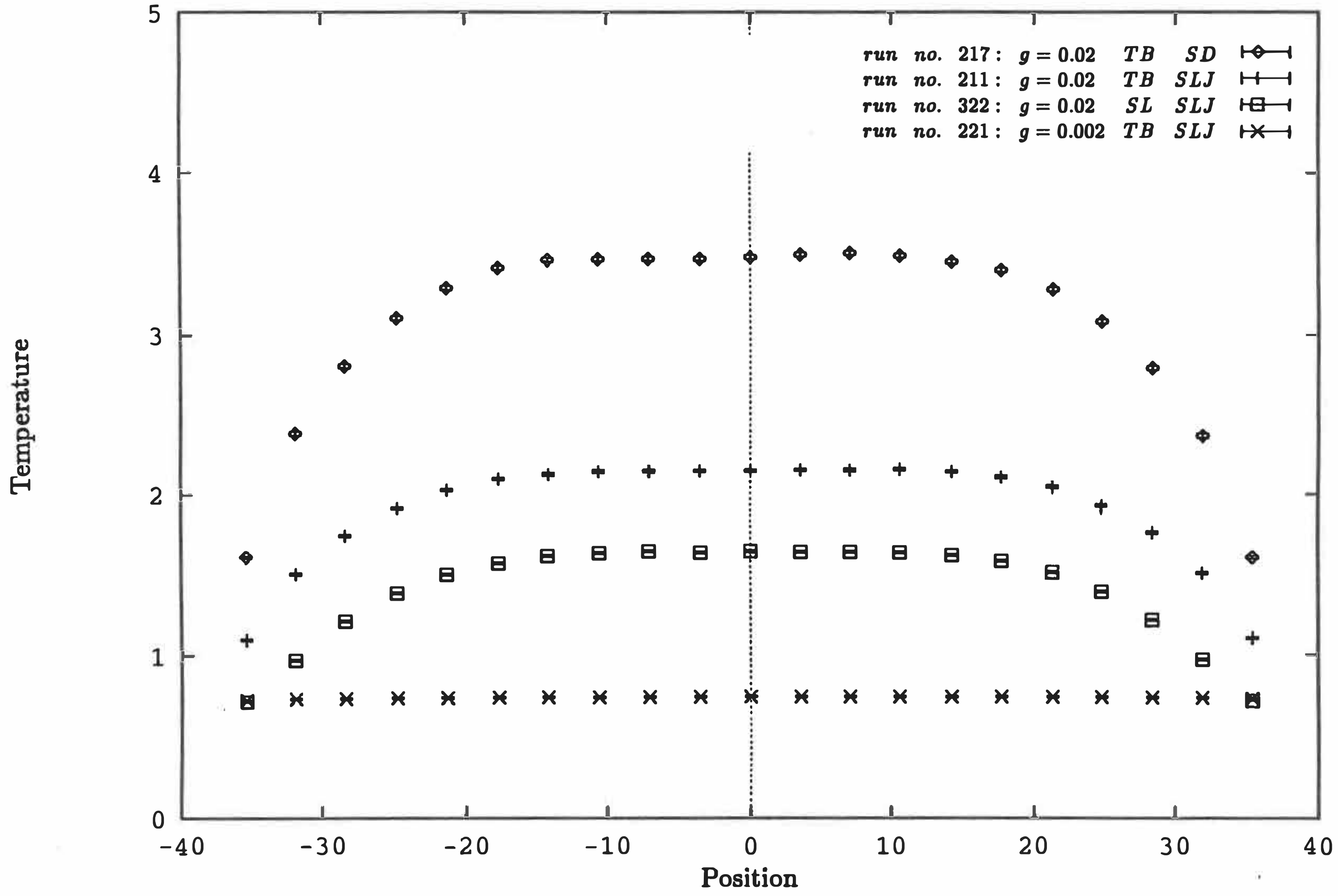
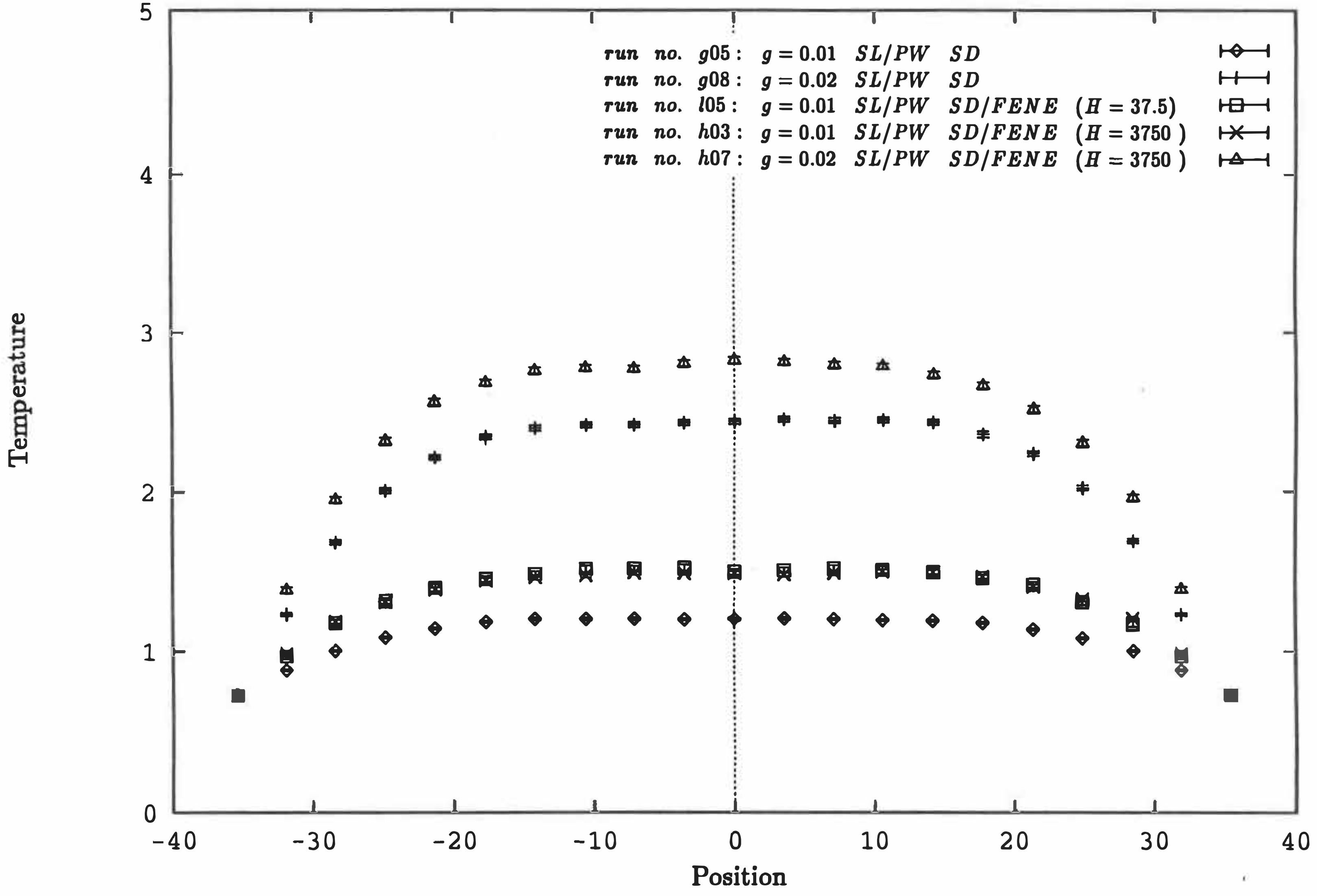
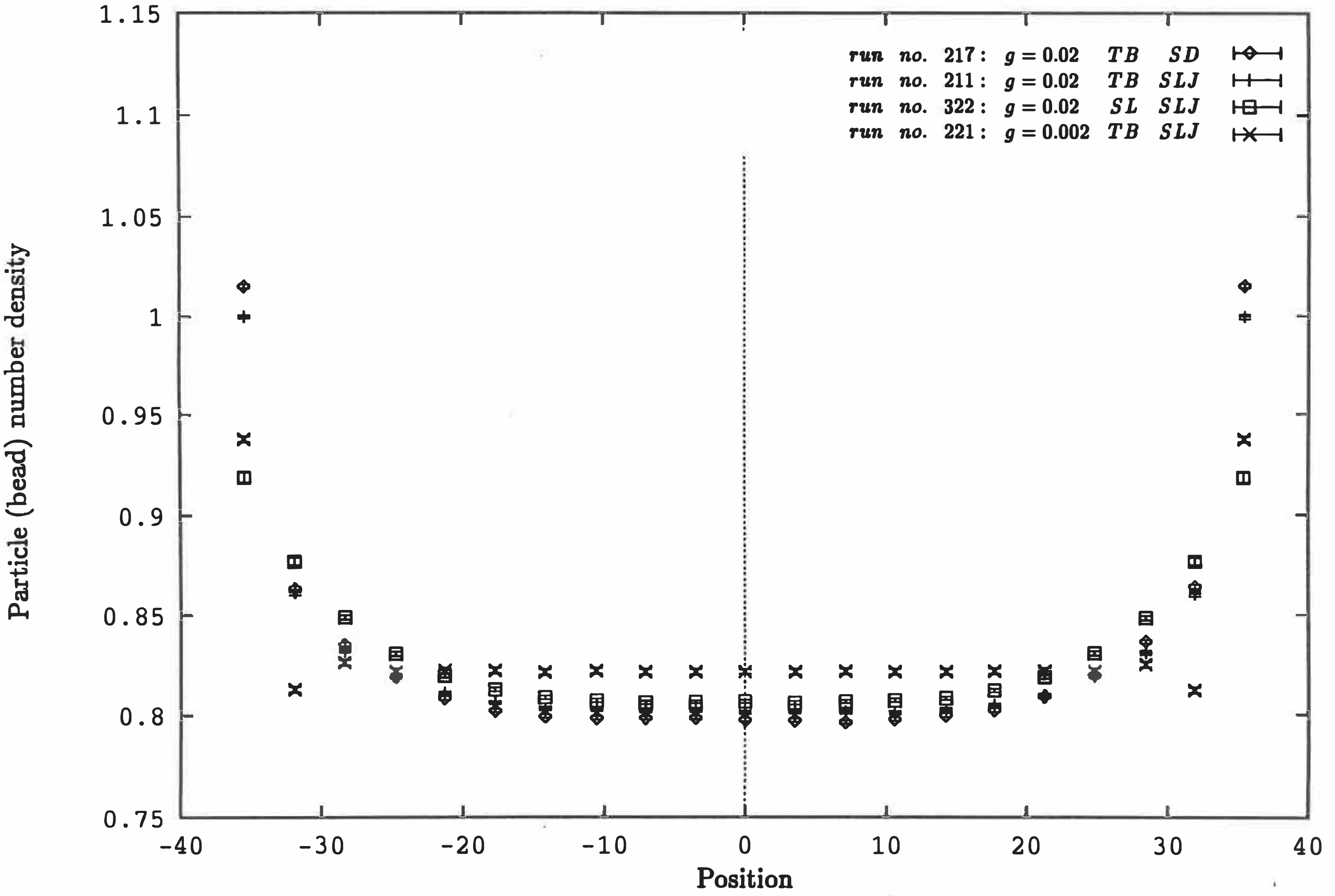

run no. 217 : g = 0.02 TB SD
run no. 211 : g = 0.02 TB SLJ
run no. 322 : g = 0.02 SL SLJ
run no. 221 : g = 0.002 TB SLJ
Temperature
Position
0
1
2
3
4
5
-40
-30
-20
-10
0
10
20
30
40

Figure 2b



run no. g05 : g = 0.01 SL/PW SD
run no. g08 : g = 0.02 SL/PW SD
run no. l05 : g = 0.01 SL/PW SD/FENE (H = 37.5)
run no. h03 : g = 0.01 SL/PW SD/FENE (H = 3750)
run no. h07 : g = 0.02 SL/PW SD/FENE (H = 3750)
Temperature
0
1
2
3
4
5
-40
-30
-20
-10
0
10
20
30
40
Position

Figure 3a



run no. 217 : $g = 0.02$ TB SD
run no. 211 : $g = 0.02$ TB SLJ
run no. 322 : $g = 0.02$ SL SLJ
run no. 221 : $g = 0.002$ TB SLJ
Particle (bead) number density
1.15
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
-40
-30
-20
-10
0
10
20
30
40
Position

Figure 3b

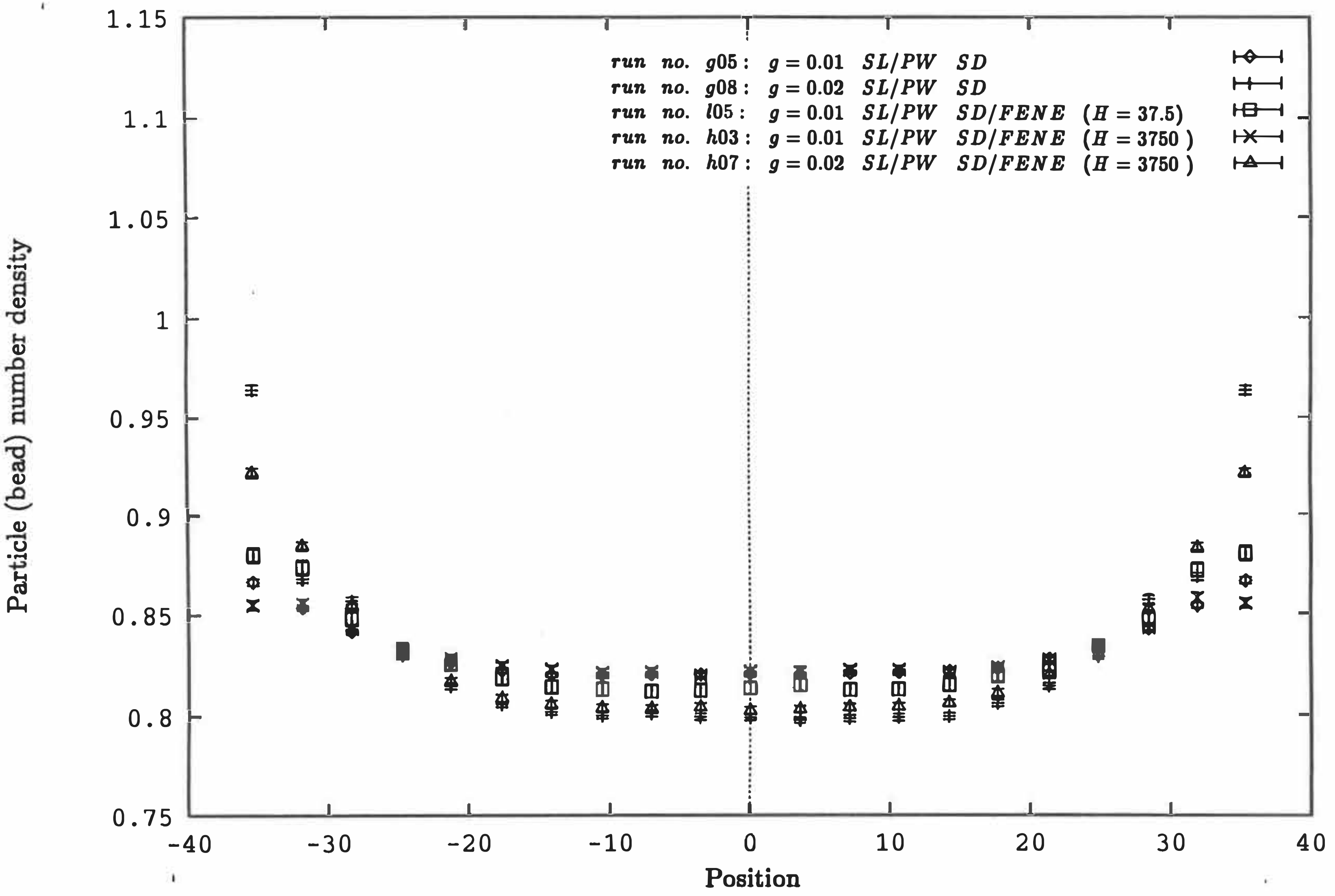

run no. g05 : g = 0.01 SL/PW SD
run no. g08 : g = 0.02 SL/PW SD
run no. l05 : g = 0.01 SL/PW SD/FENE (H = 37.5)
run no. h03 : g = 0.01 SL/PW SD/FENE (H = 3750)
run no. h07 : g = 0.02 SL/PW SD/FENE (H = 3750)
Particle (bead) number density
1.15
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
-40
-30
-20
-10
0
10
20
30
40
Position

Figure 4a

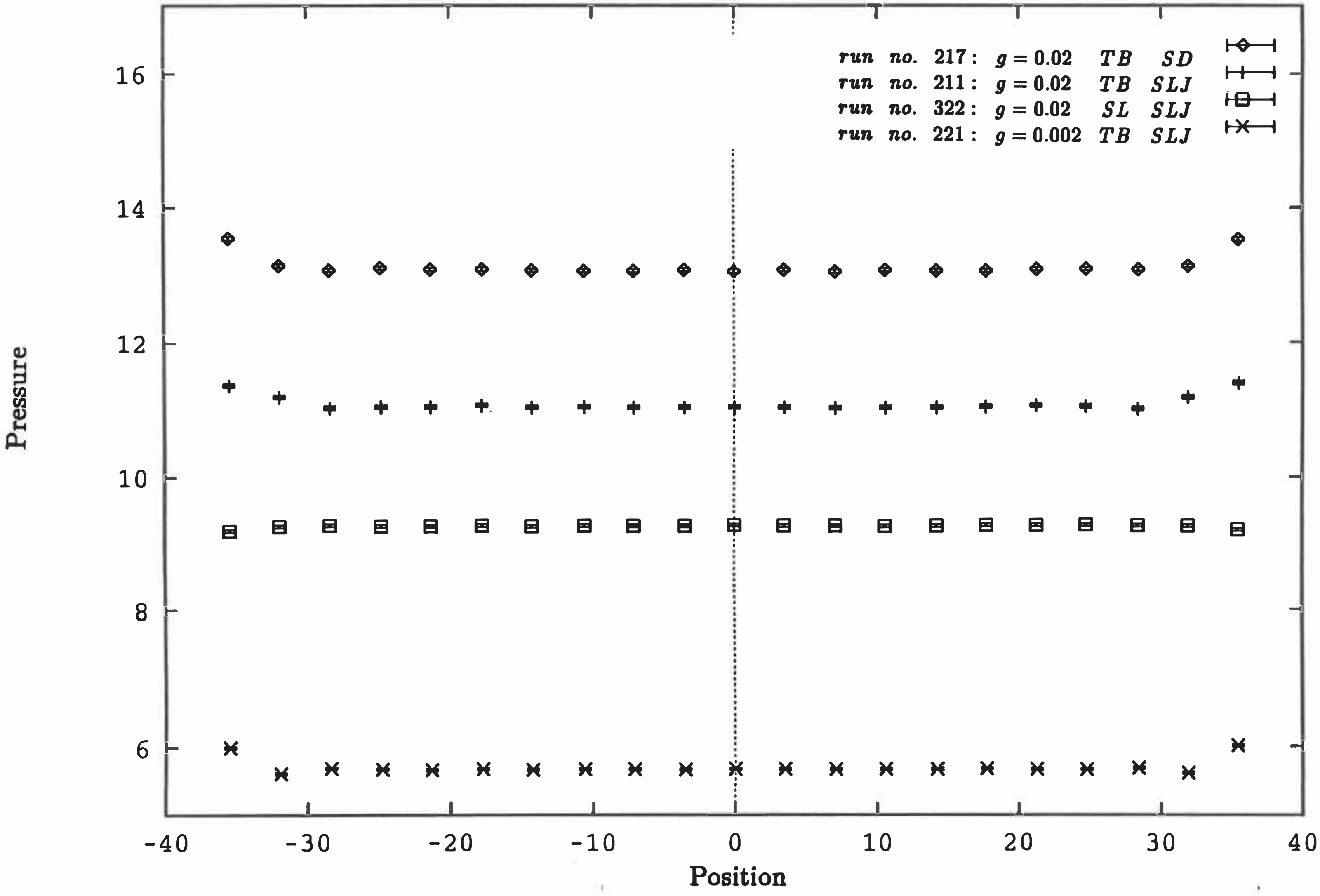

run no. 217 : g = 0.02 TB SD
run no. 211 : g = 0.02 TB SLJ
run no. 322 : g = 0.02 SL SLJ
run no. 221 : g = 0.002 TB SLJ
16
14
12
10
8
6
Pressure
-40
-30
-20
-10
0
10
20
30
40
Position

Figure 4b

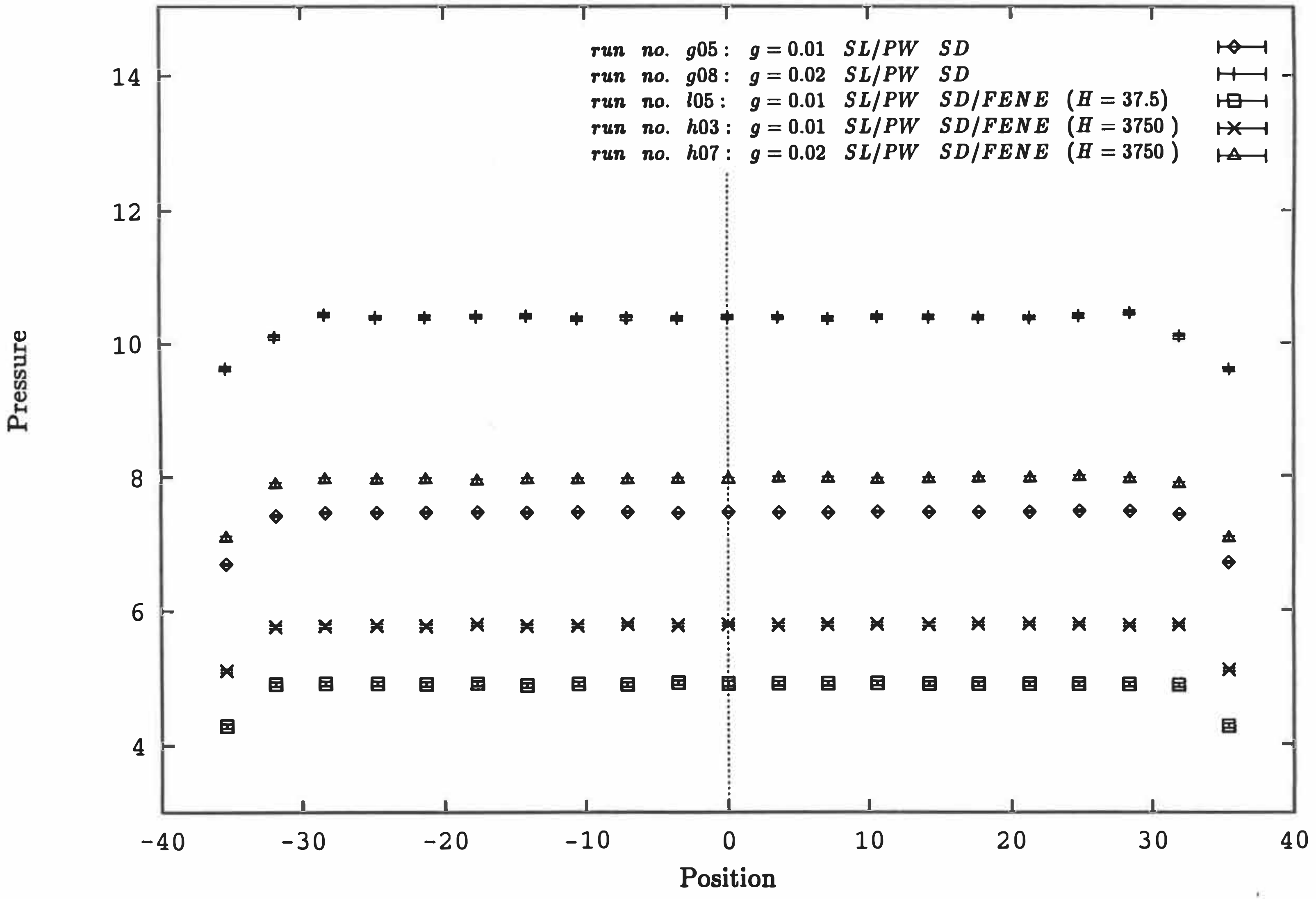

run no. g05 : g = 0.01 SL/PW SD
run no. g08 : g = 0.02 SL/PW SD
run no. l05 : g = 0.01 SL/PW SD/FENE (H = 37.5)
run no. h03 : g = 0.01 SL/PW SD/FENE (H = 3750)
run no. h07 : g = 0.02 SL/PW SD/FENE (H = 3750)
Pressure
14
12
10
8
6
4
-40
-30
-20
-10
0
10
20
30
40
Position

Figure 5a

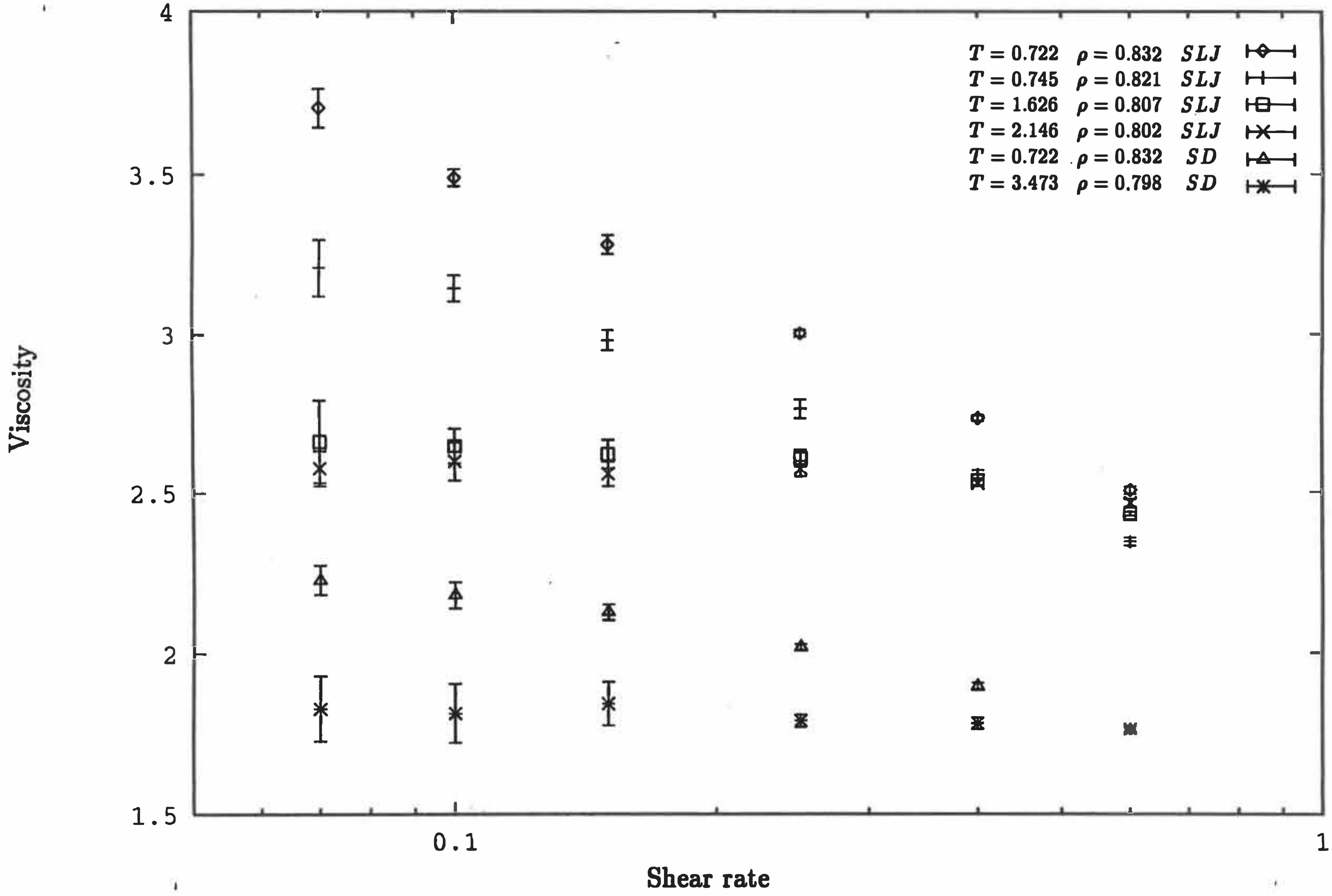

T = 0.722 ρ = 0.832 SLJ
T = 0.745 ρ = 0.821 SLJ
T = 1.626 ρ = 0.807 SLJ
T = 2.146 ρ = 0.802 SLJ
T = 0.722 ρ = 0.832 SD
T = 3.473 ρ = 0.798 SD
Viscosity
4
3.5
3
2.5
2
1.5
0.1
1
Shear rate

Figure 5b

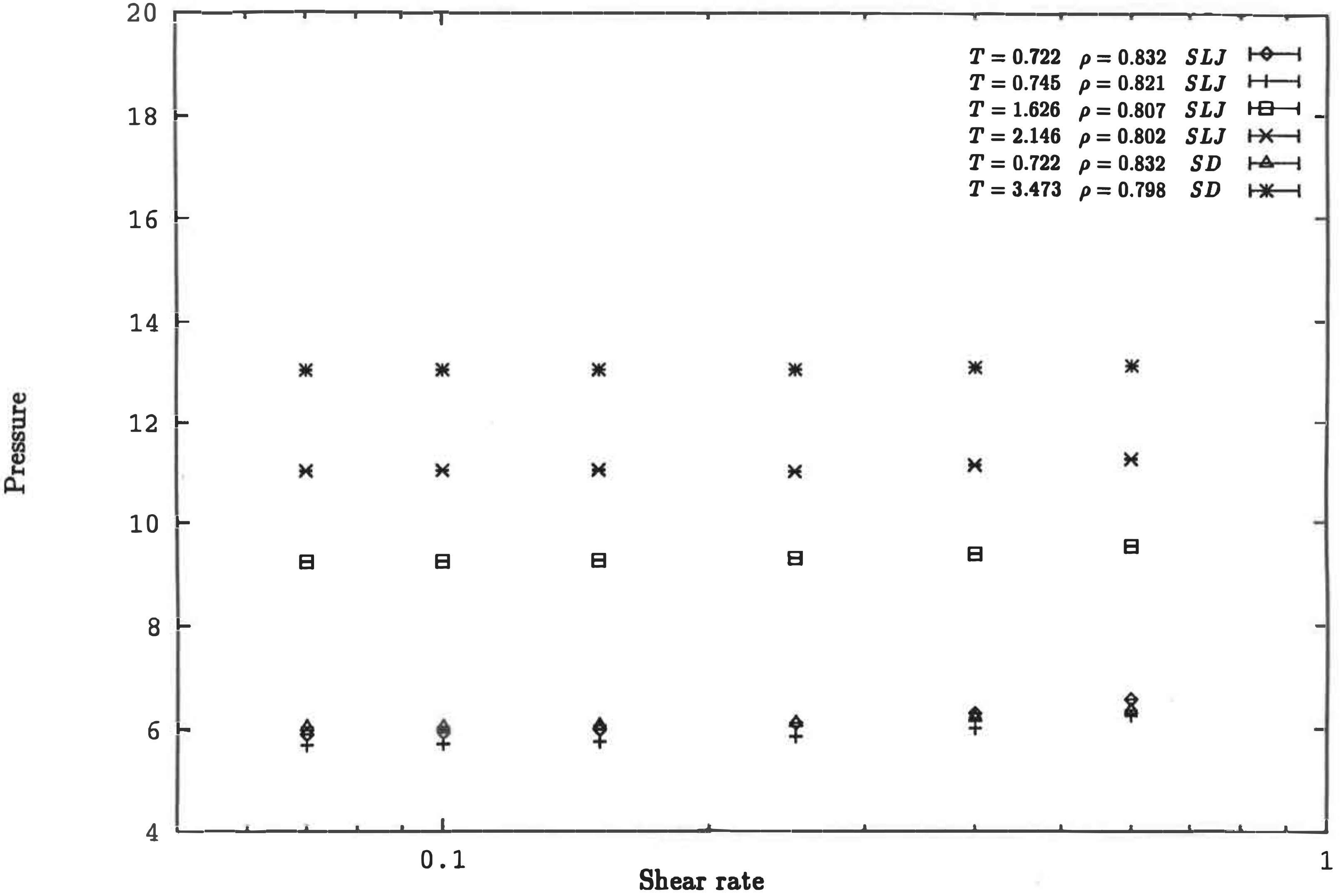

$T = 0.722$ $\rho = 0.832$ SLJ
$T = 0.745$ $\rho = 0.821$ SLJ
$T = 1.626$ $\rho = 0.807$ SLJ
$T = 2.146$ $\rho = 0.802$ SLJ
$T = 0.722$ $\rho = 0.832$ SD
$T = 3.473$ $\rho = 0.798$ SD
Pressure
20
18
16
14
12
10
8
6
4
0.1
1
Shear rate

Figure 6

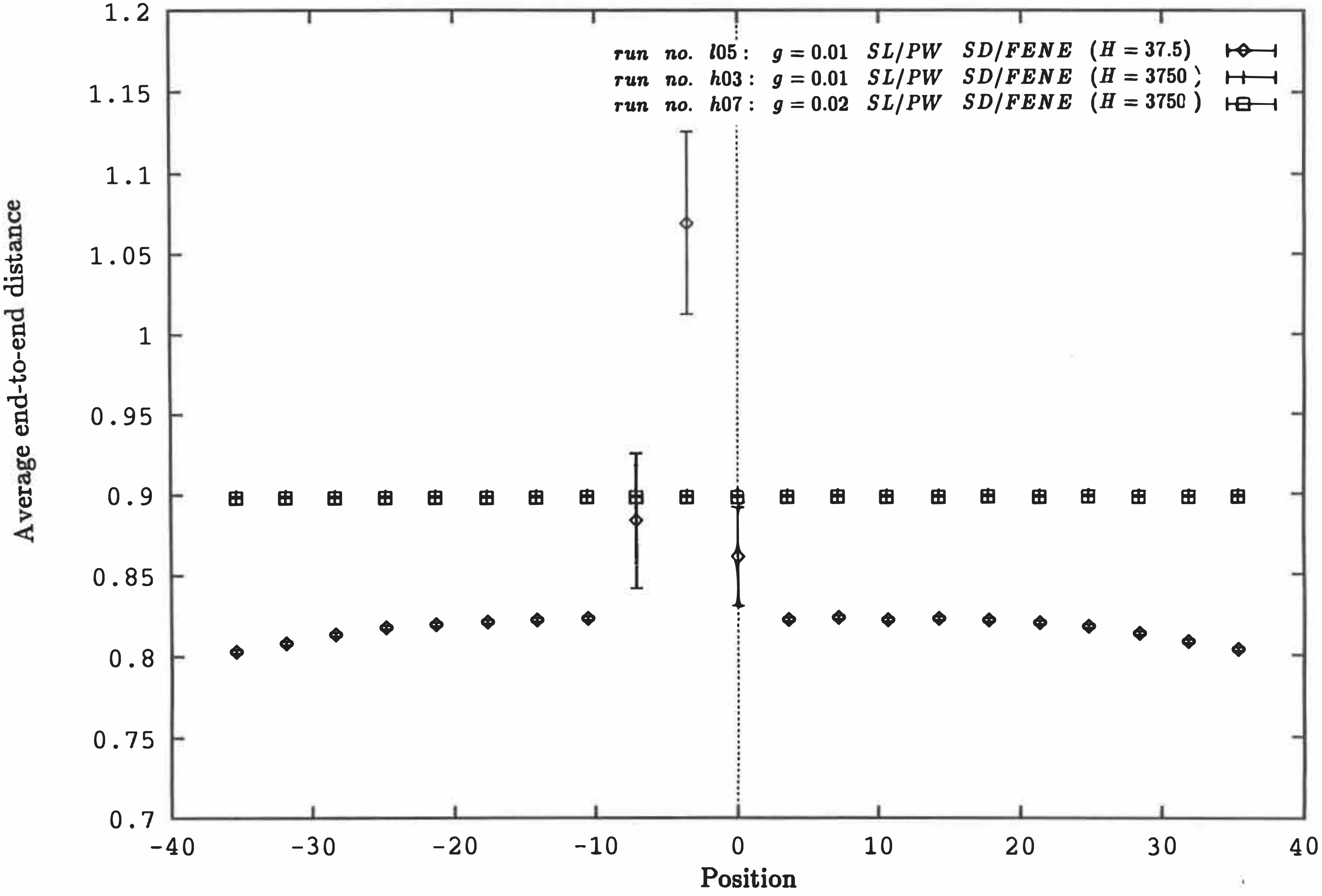

run no. l05 : g = 0.01 SL/PW SD/FENE (H = 37.5)
run no. h03 : g = 0.01 SL/PW SD/FENE (H = 3750)
run no. h07 : g = 0.02 SL/PW SD/FENE (H = 3750)
Average end-to-end distance
1.2
1.15
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
-40
-30
-20
-10
0
10
20
30
40
Position

Figure 7

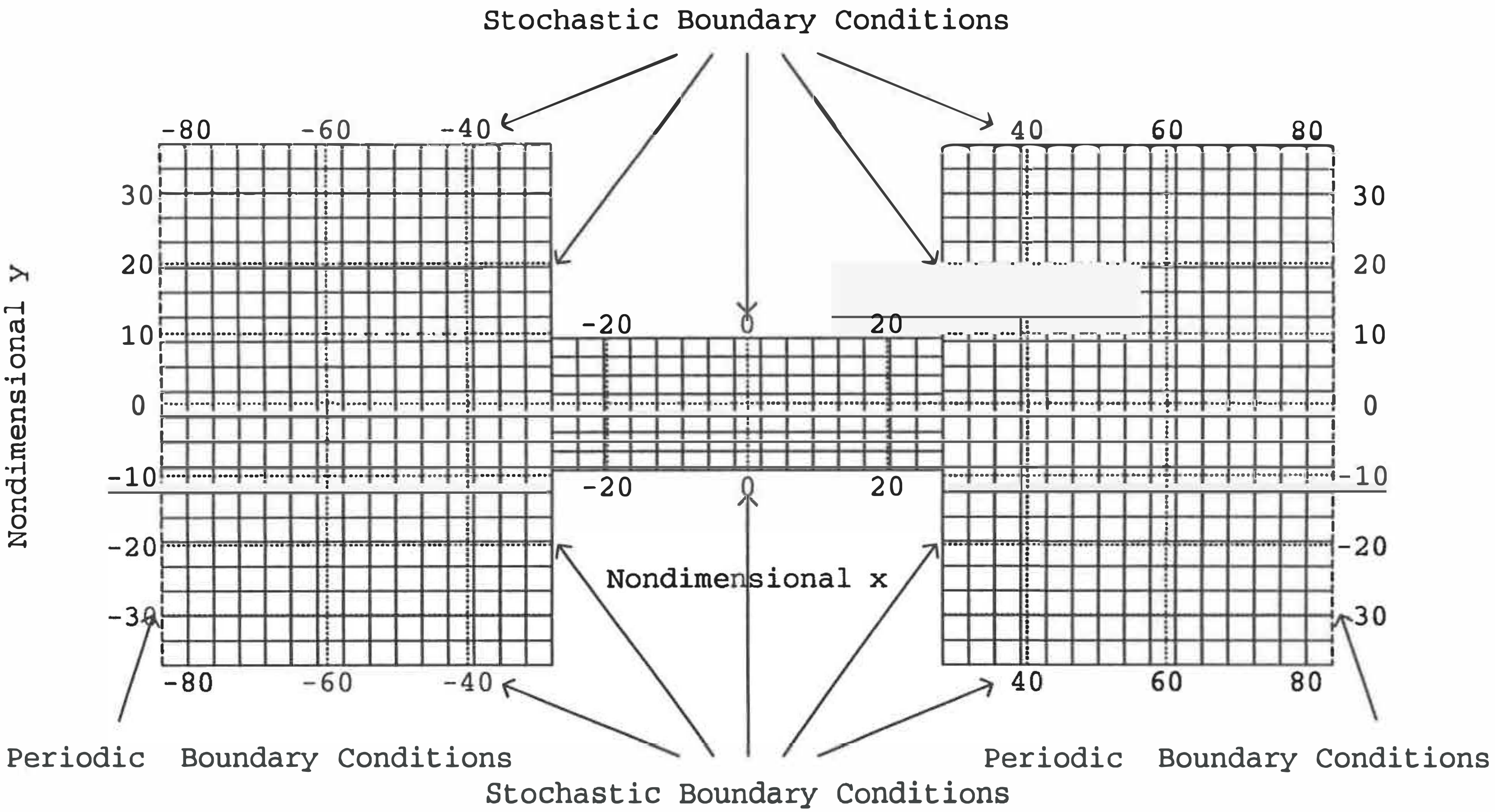

Stochastic Boundary Conditions
-80
-60
-40
40
60
80
30
20
10
0
-10
-20
-30
-20
0
20
-20
0
20
Nondimensional y
Nondimensional x
-80
-60
-40
40
60
80
Periodic Boundary Conditions
Periodic Boundary Conditions
Stochastic Boundary Conditions

Figure 8a

run no. 218: g=0.02 TW/PBC SLJ

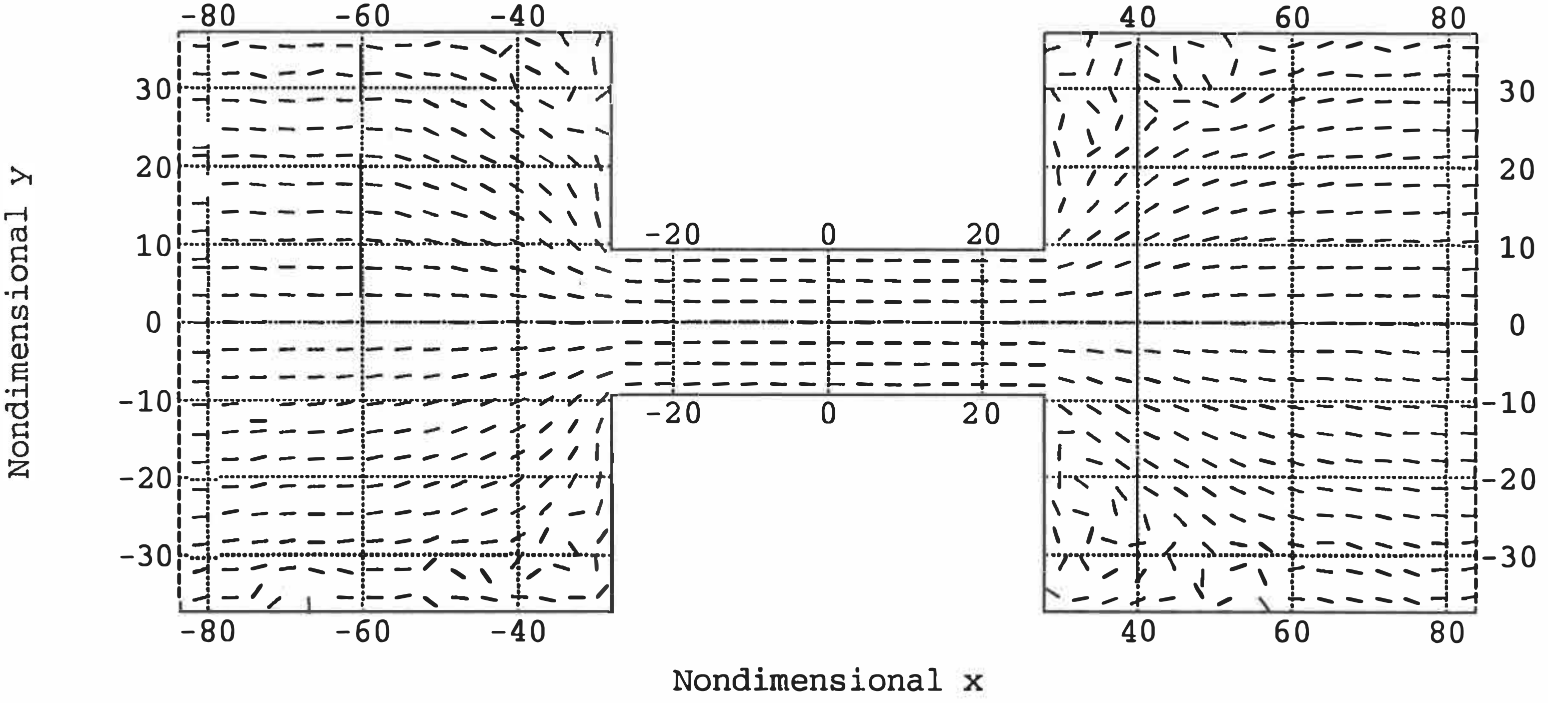


Figure 8b

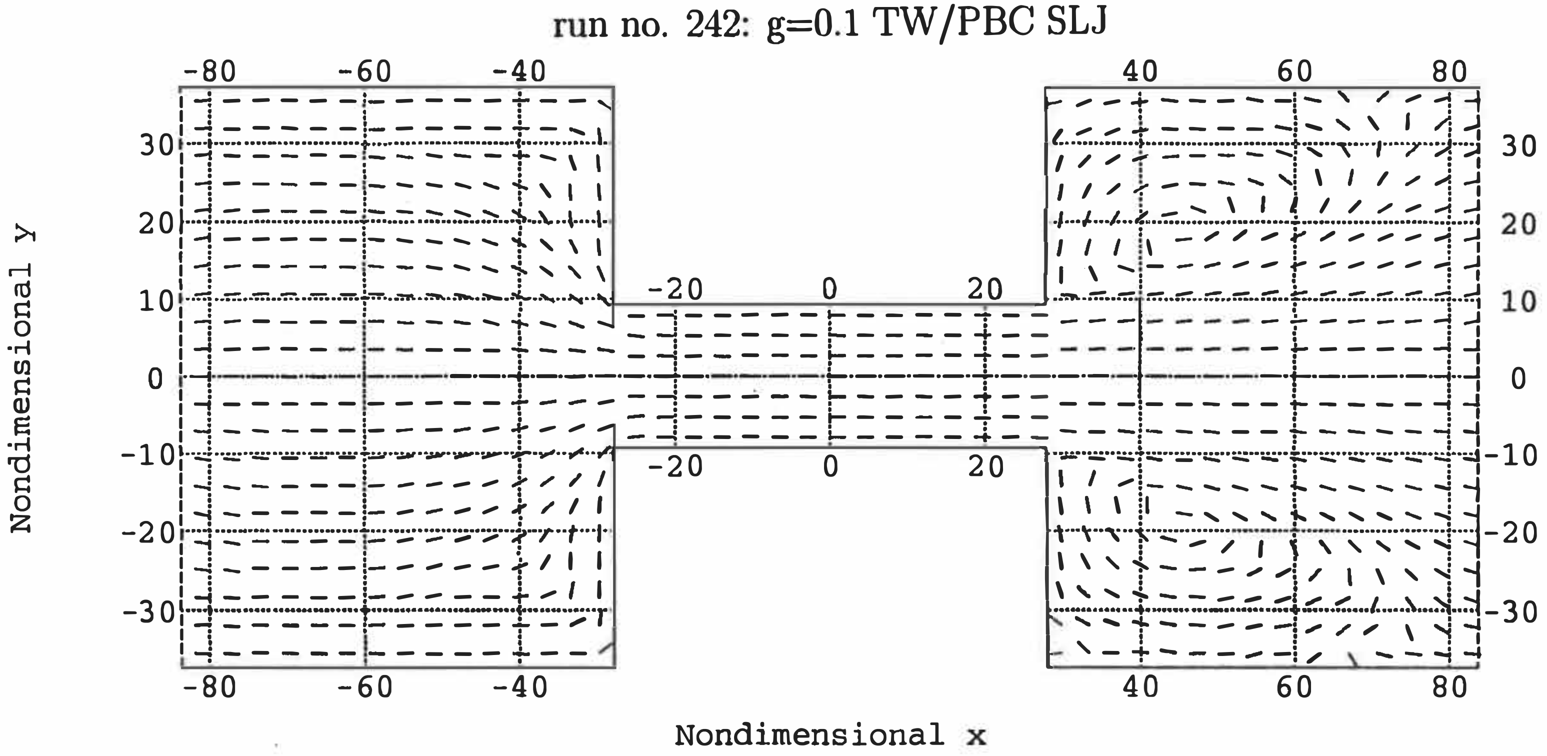

run no. 242: g=0.1 TW/PBC SLJ
-80
-60
-40
40
60
80
-20
0
20
30
20
10
0
-10
-20
-30
Nondimensional y
Nondimensional x

Figure 8c

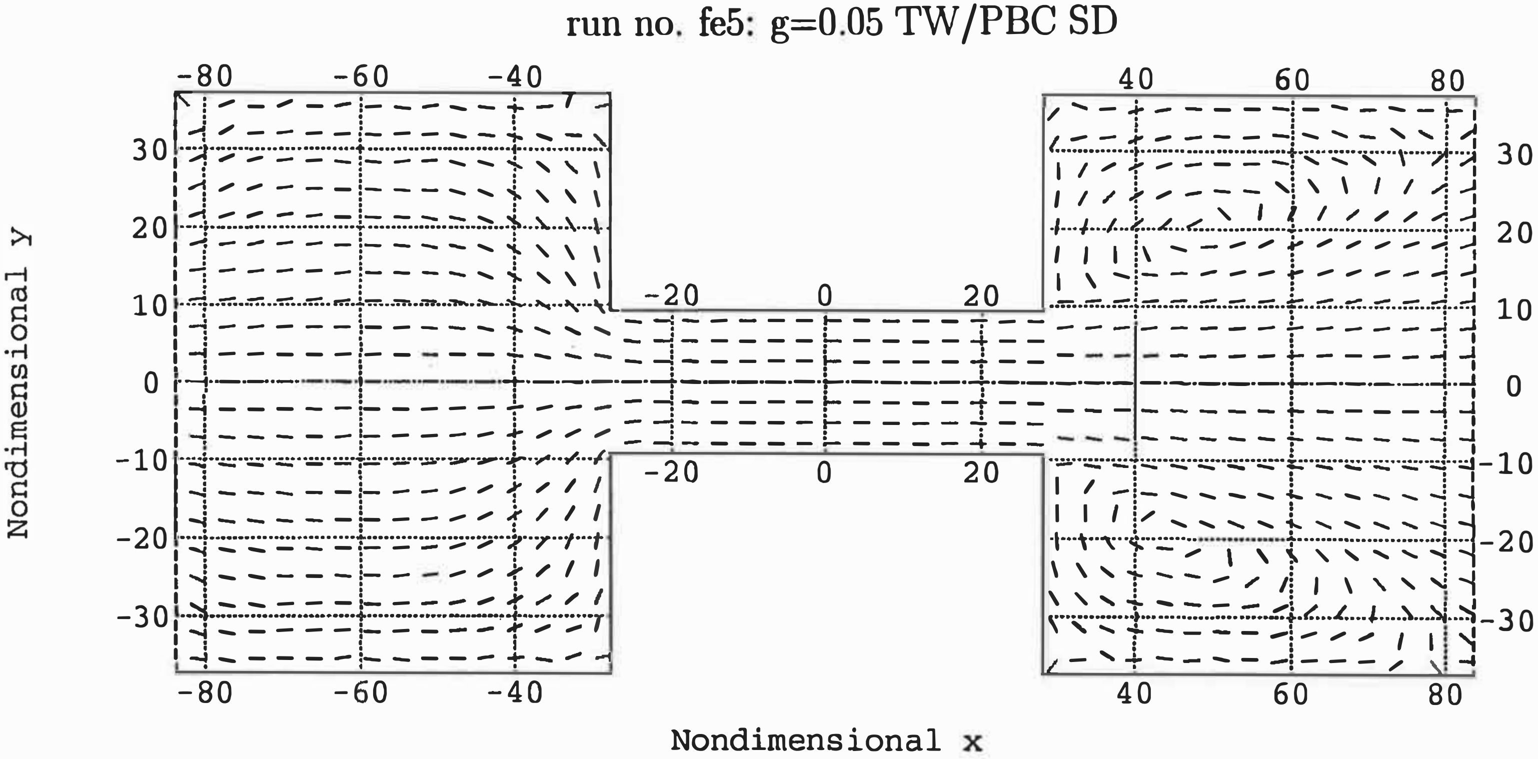

run no. fe5: g=0.05 TW/PBC SD
-80
-60
-40
40
60
80
-20
0
20
30
20
10
0
-10
-20
-30
Nondimensional y
Nondimensional x

Figure 8d

run no. co7: g=0.1 TW/PBC SD

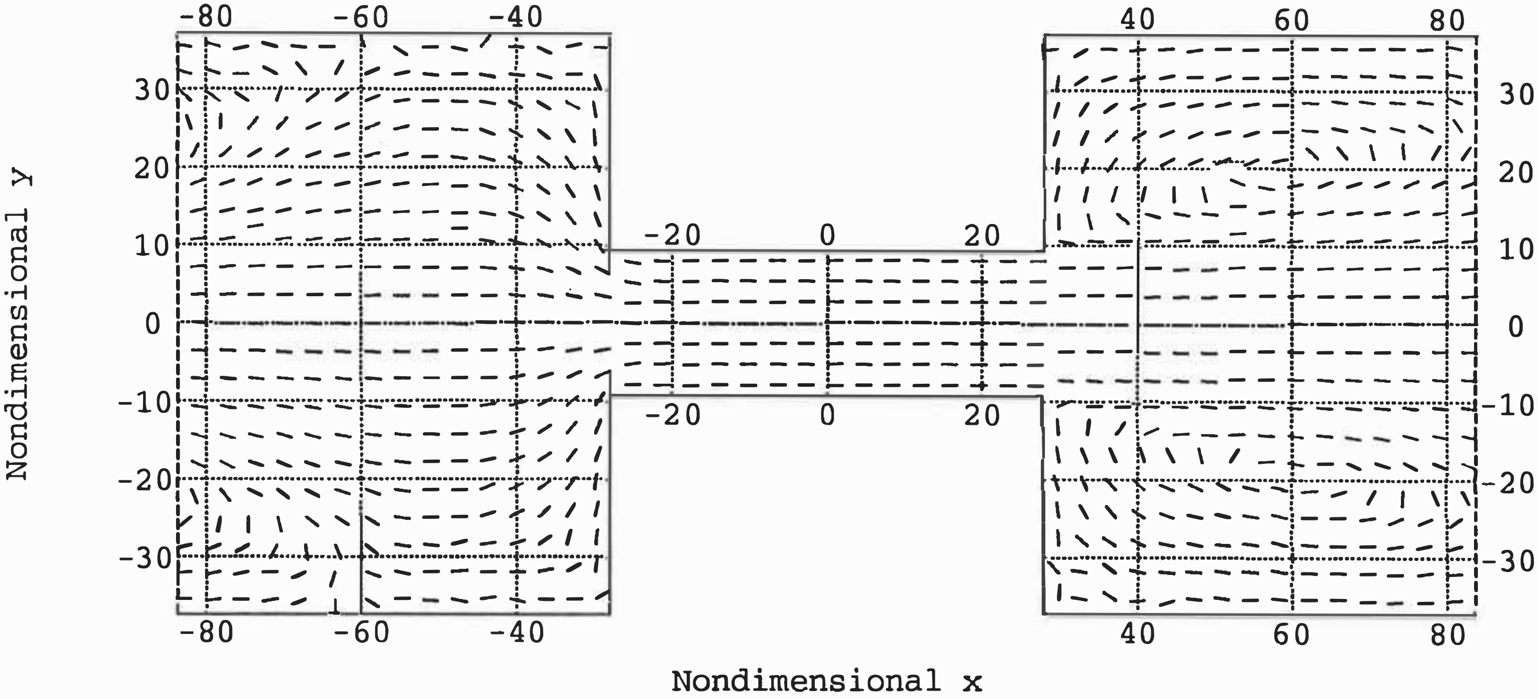


Figure 8e

run no. lo4: g=0.05 TW/PBC SD/FENE (H=3750)

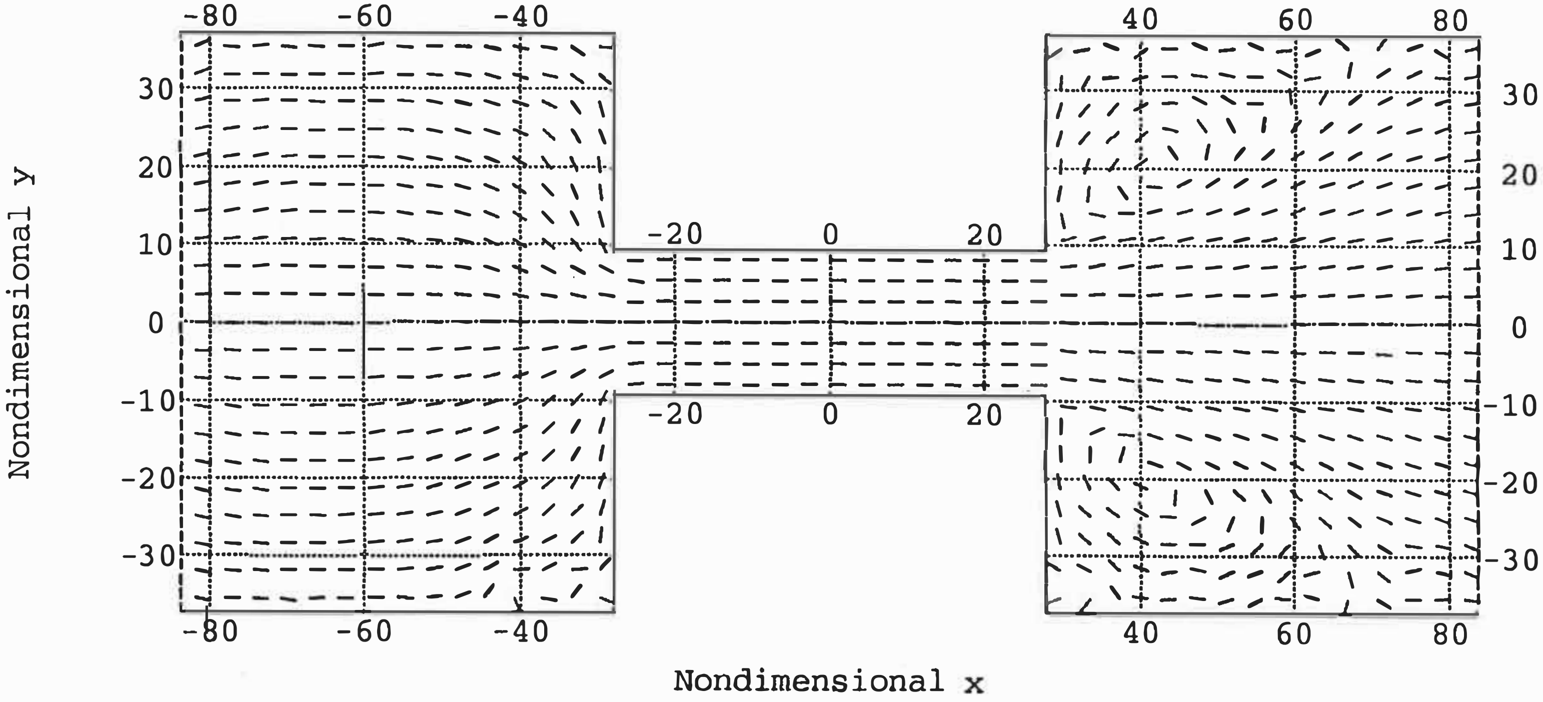


Figure 8f

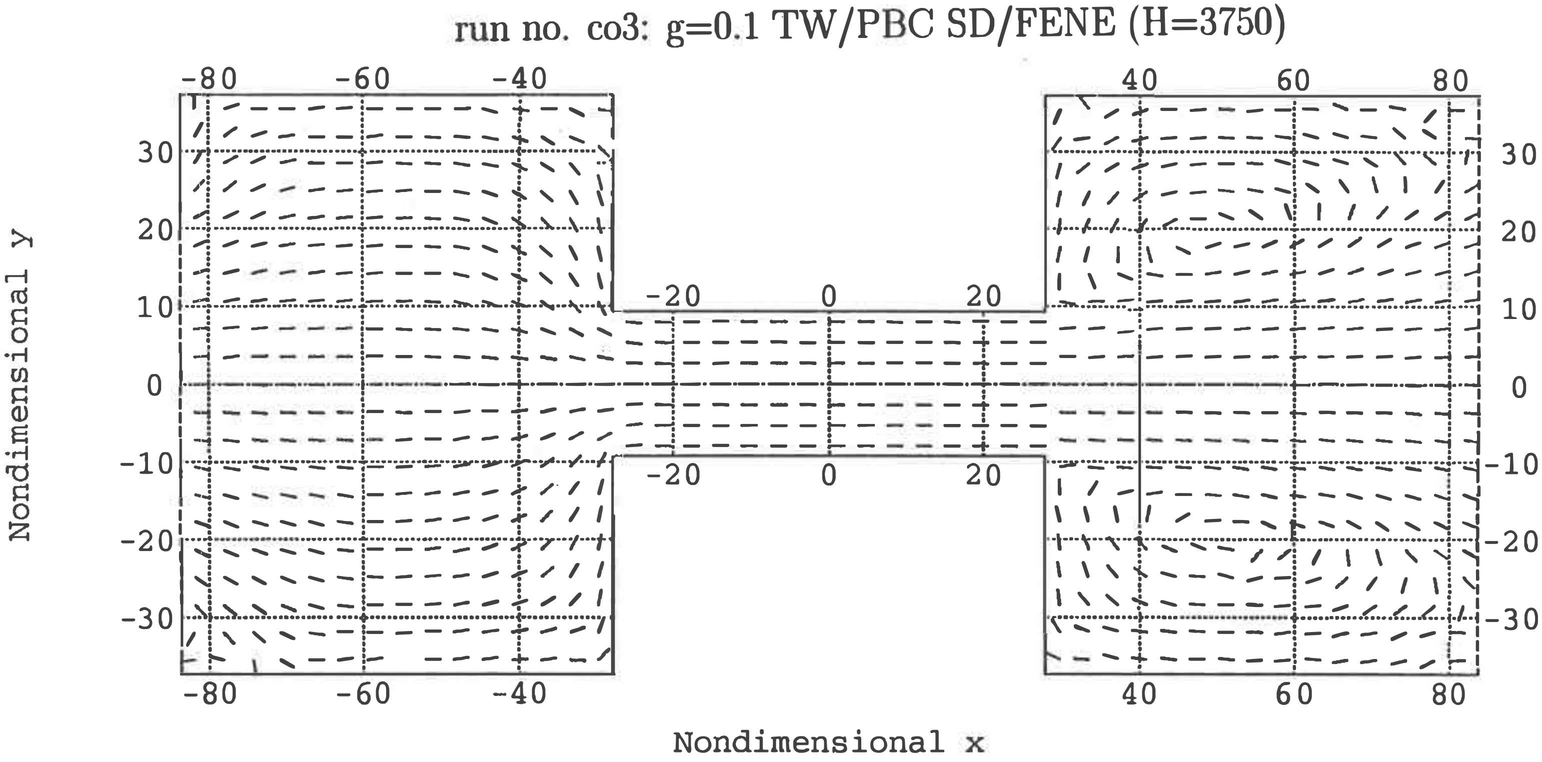

run no. co3: g=0.1 TW/PBC SD/FENE (H=3750)
-80
-60
-40
-20
0
20
40
60
80
30
20
10
0
-10
-20
-30
Nondimensional y
Nondimensional x

Figure 9a

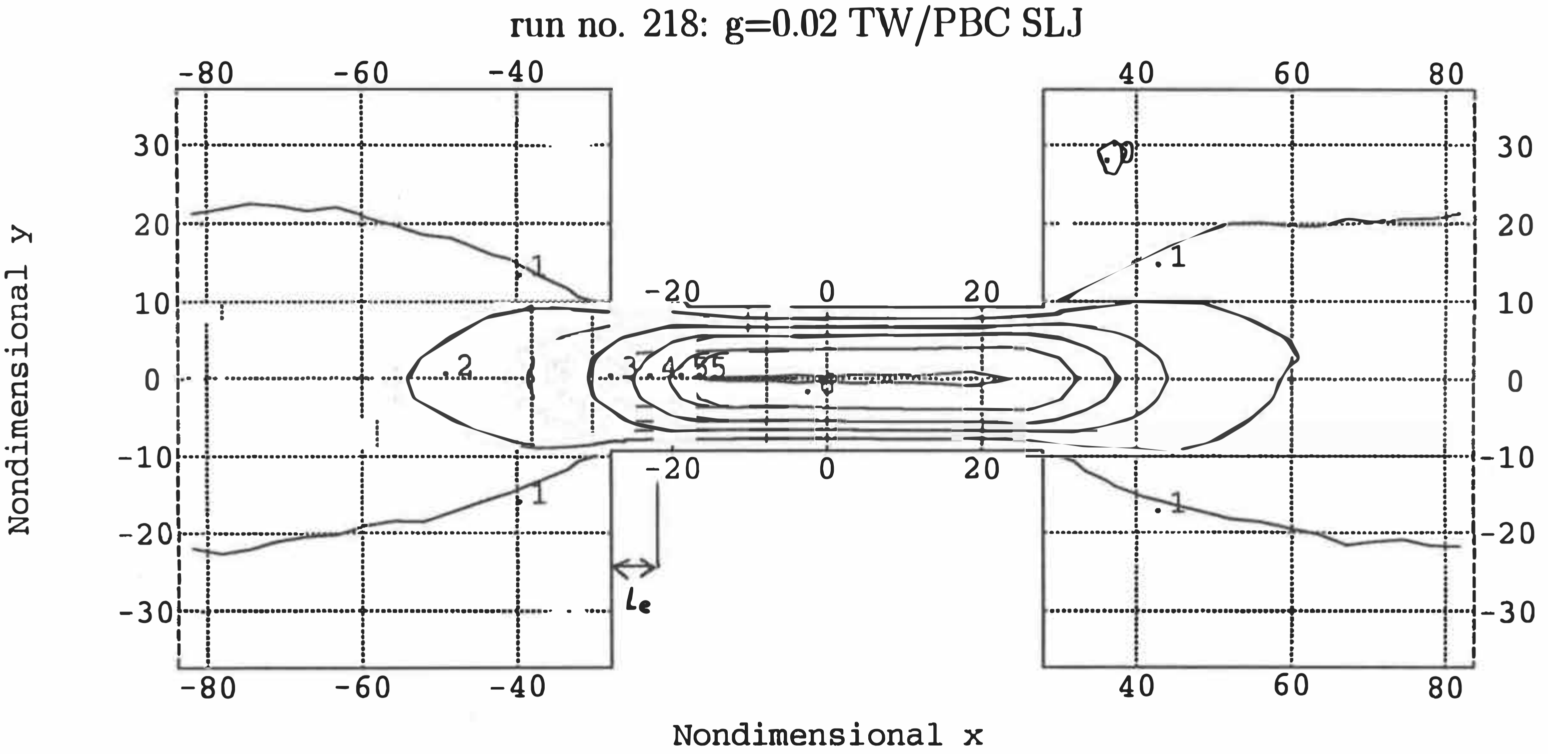

run no. 218: g=0.02 TW/PBC SLJ
-80
-60
-40
40
60
80
-20
0
20
30
20
10
0
-10
-20
-30
.1
.2
.3
.4
.5
.5
0
L_e
Nondimensional x
Nondimensional y

Figure 9b

run no. fe5: g=0.05 TW/PBC SD

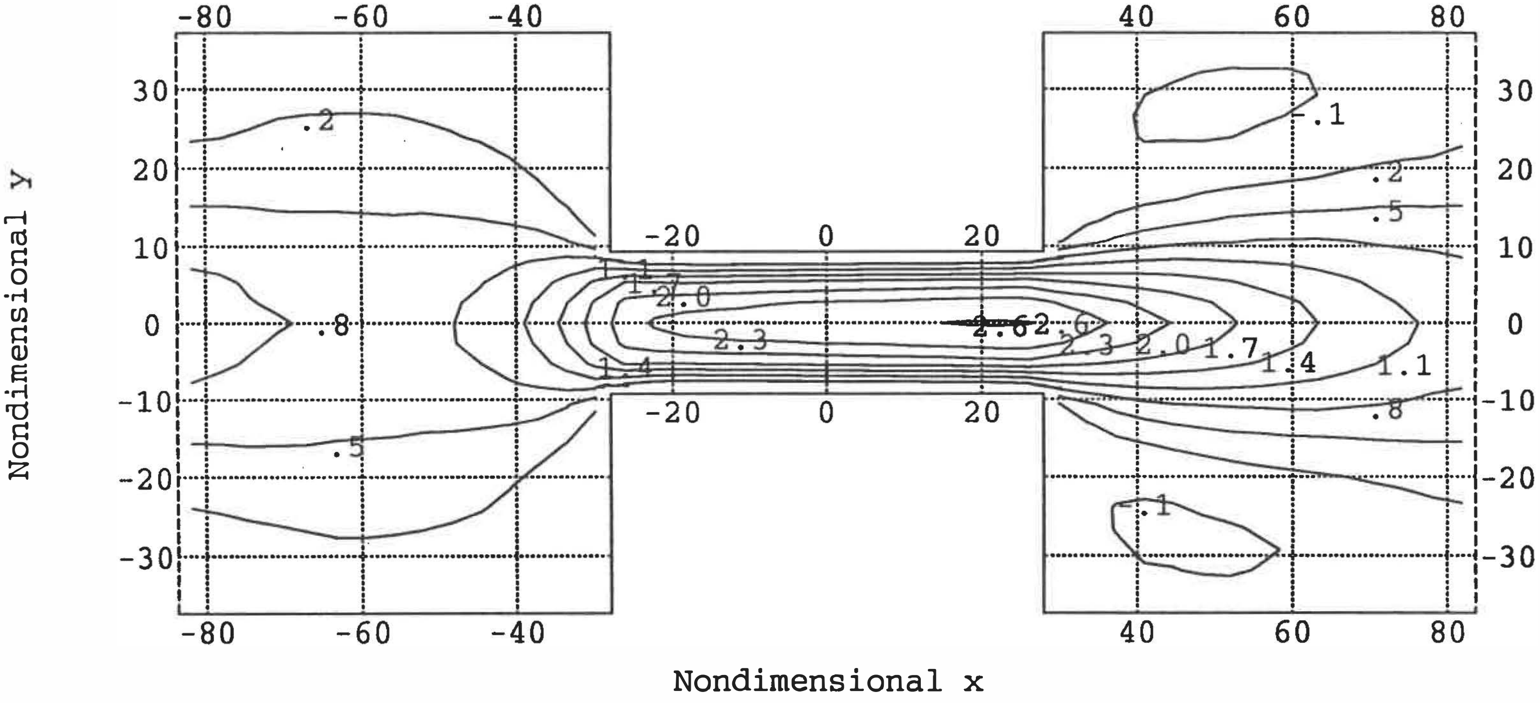


Figure 9c

run no. lo4: g=0.05 TW/PBC SD/FENE (H=3750)

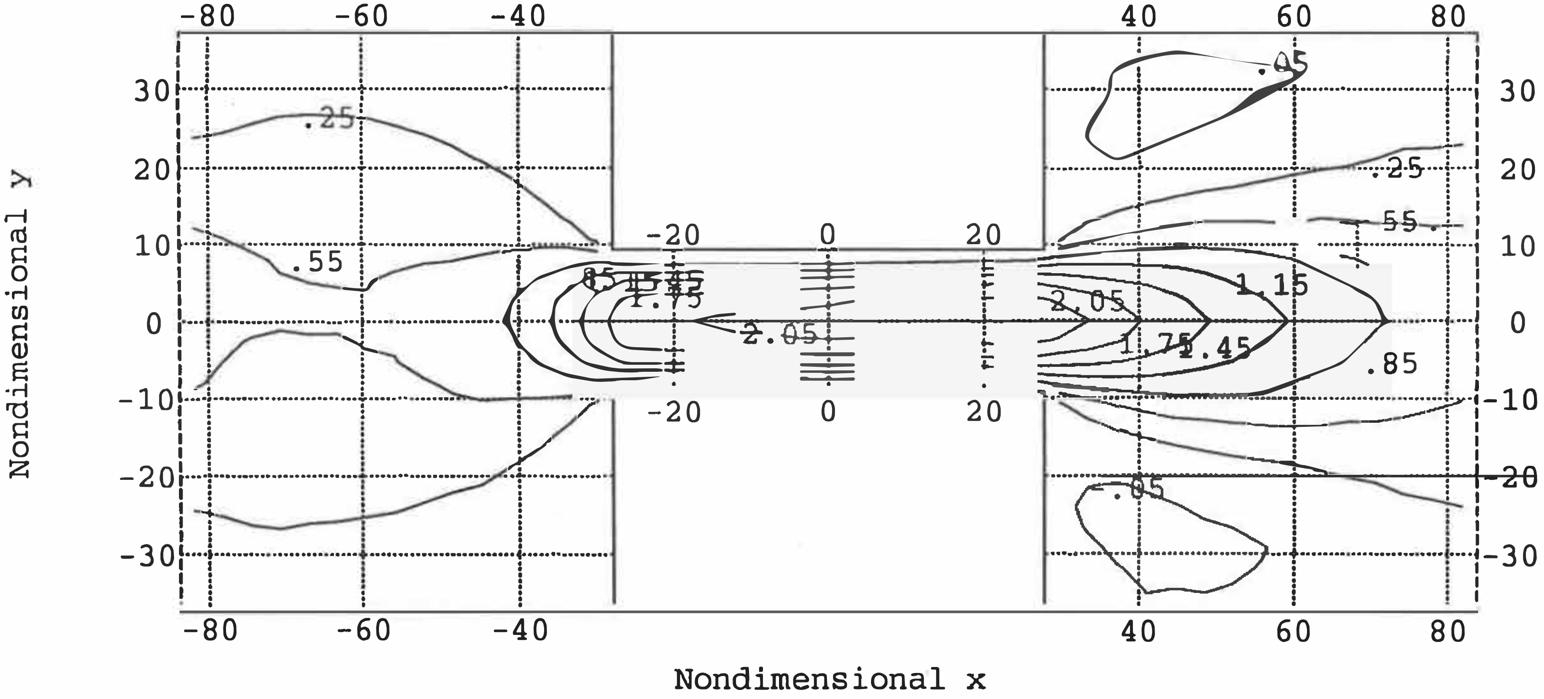


Figure 9d

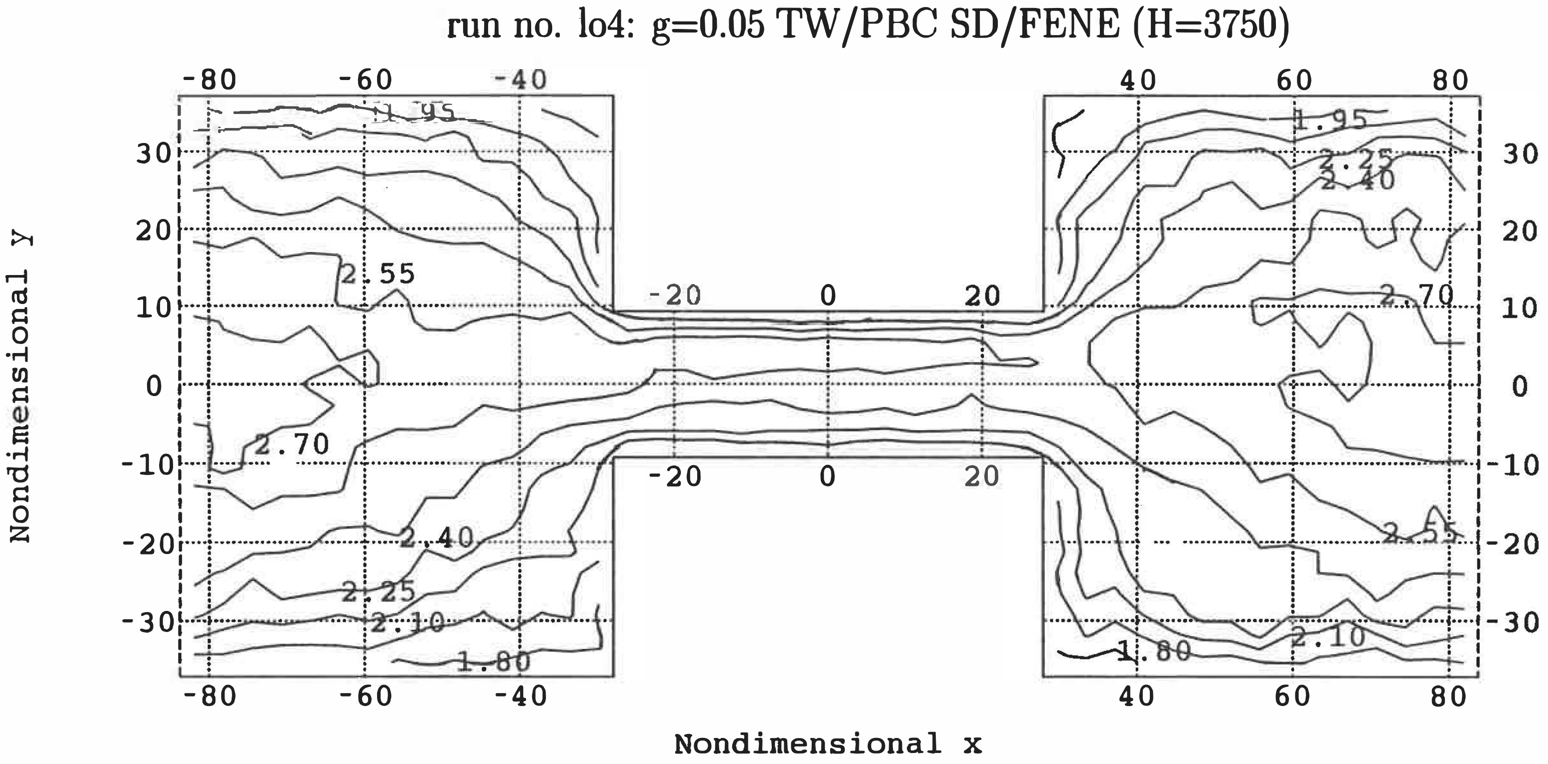

run no. lo4: g=0.05 TW/PBC SD/FENE (H=3750)
Nondimensional y
Nondimensional x
-80
-60
-40
40
60
80
-20
0
20
30
20
10
0
-10
-20
-30
1.95
2.25
2.40
2.55
2.70
2.10
1.80

Figure 10a

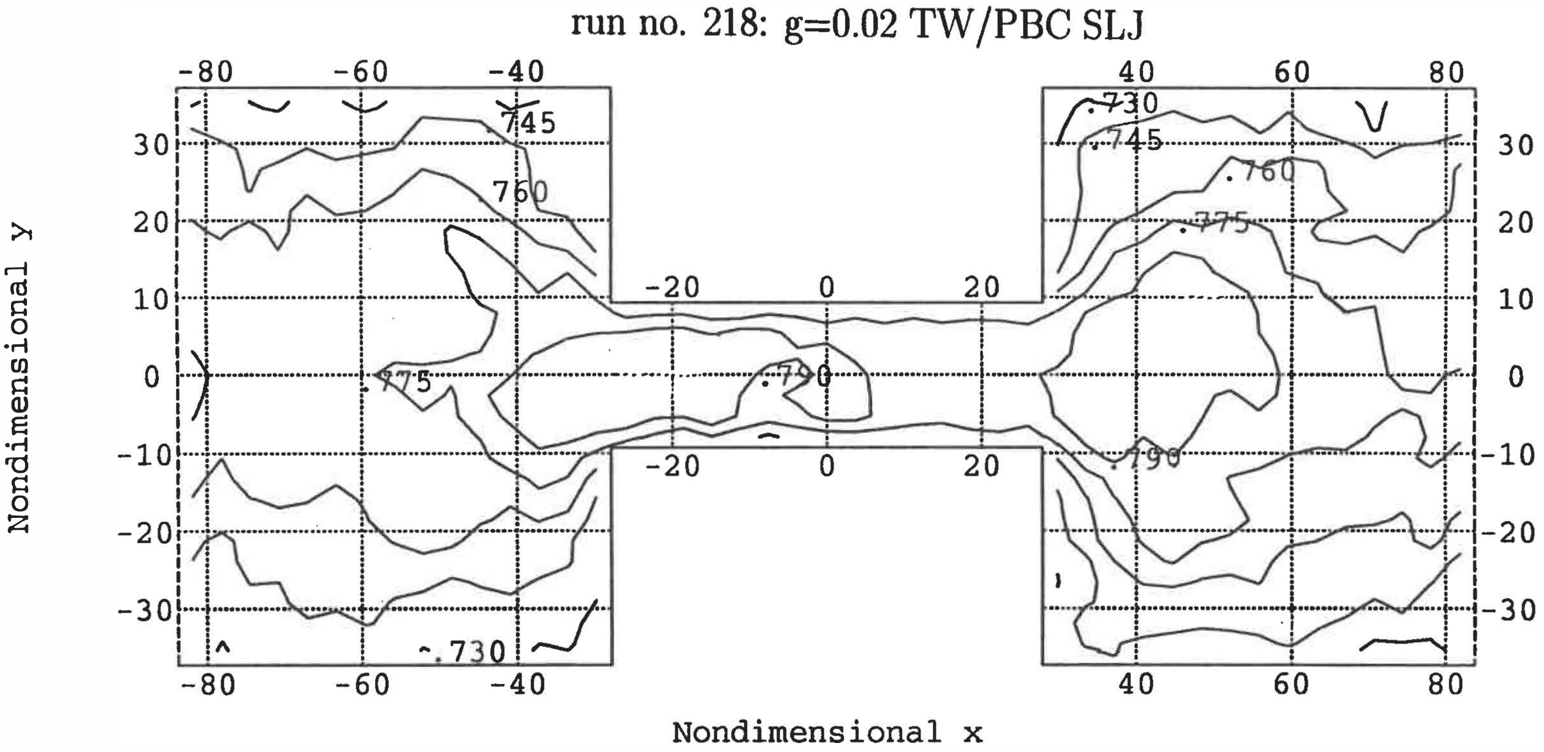

run no. 218: g=0.02 TW/PBC SLJ
-80
-60
-40
40
60
80
-20
0
20
30
20
10
0
-10
-20
-30
.730
.745
.760
.775
.790
Nondimensional y
Nondimensional x

Figure 10b

run no. fe5: g=0.05 TW/PBC SD

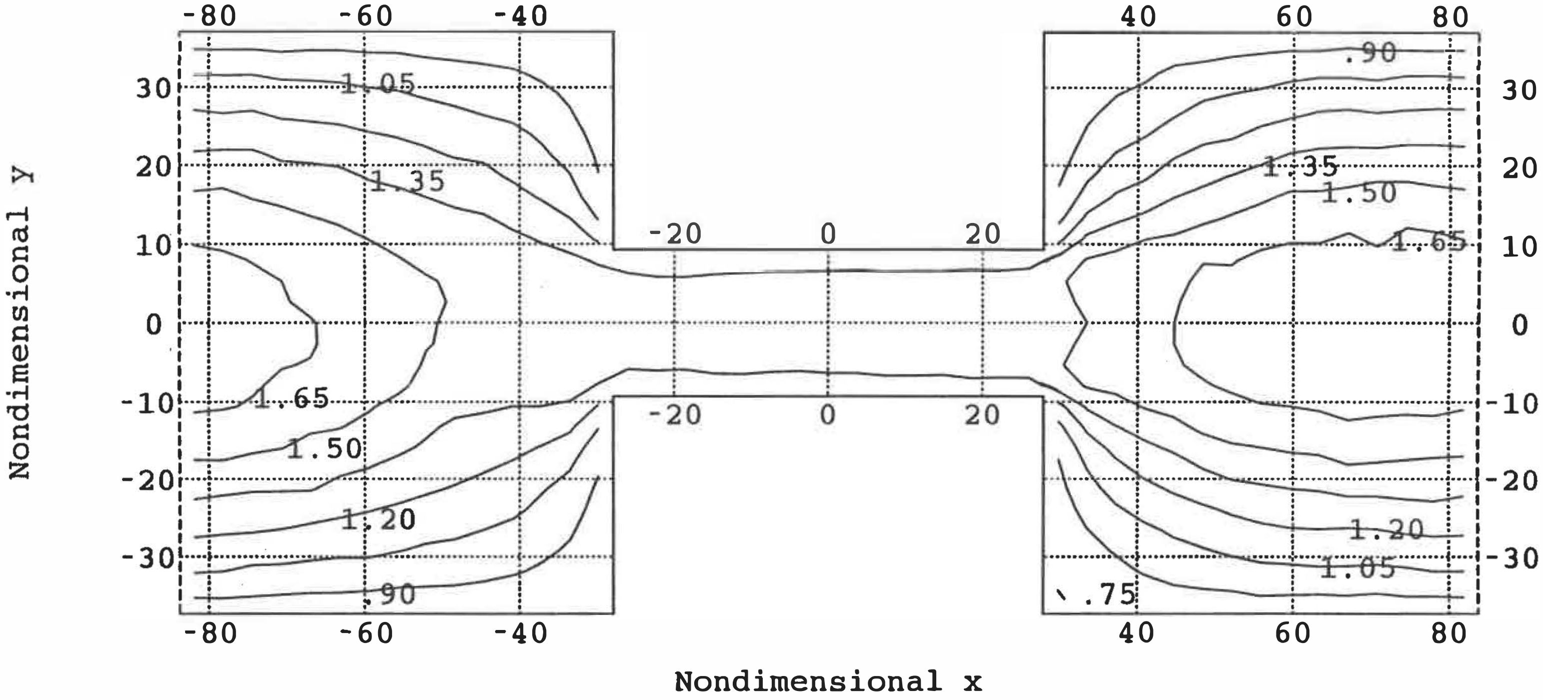


Figure 10c

run no. lo4: g=0.05 TW/PBC SD/FENE (H=3750)

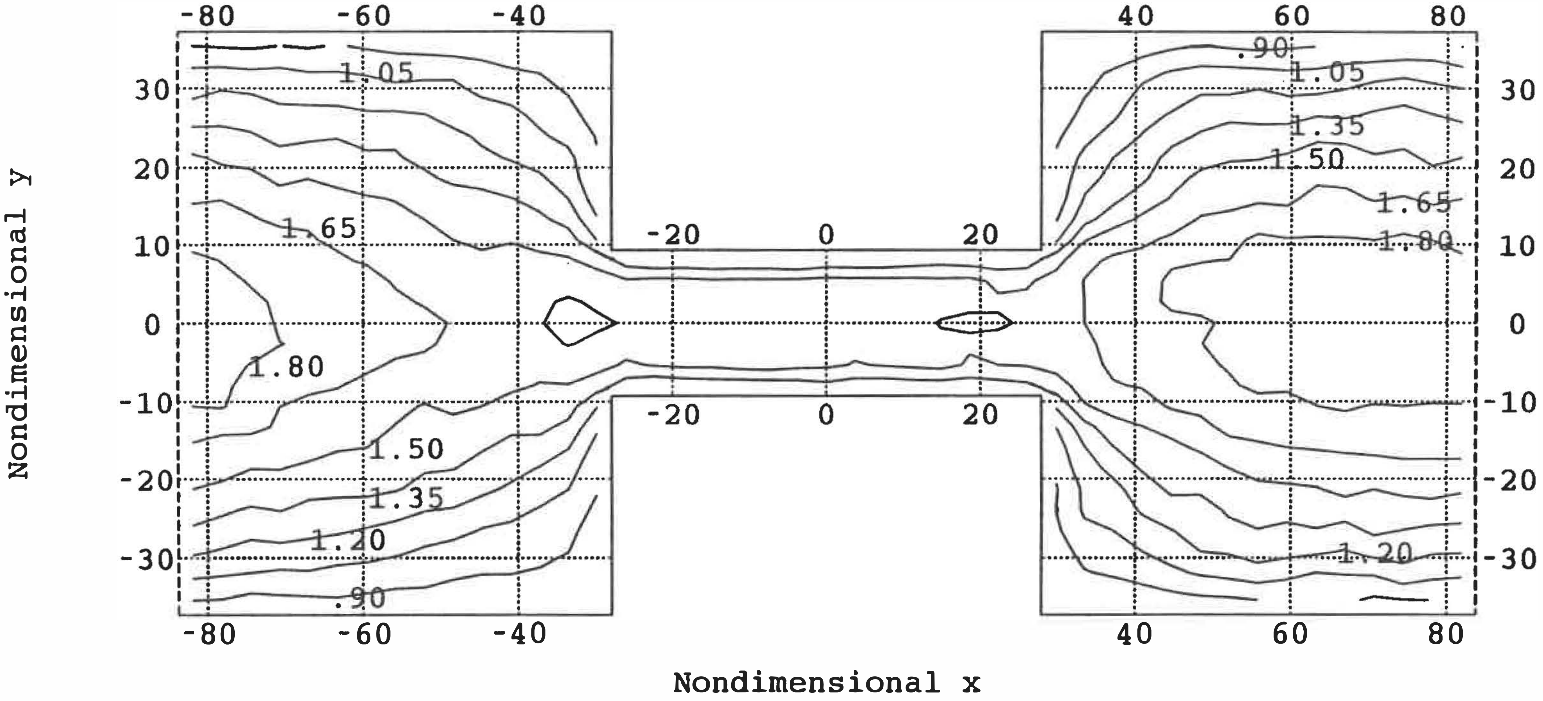
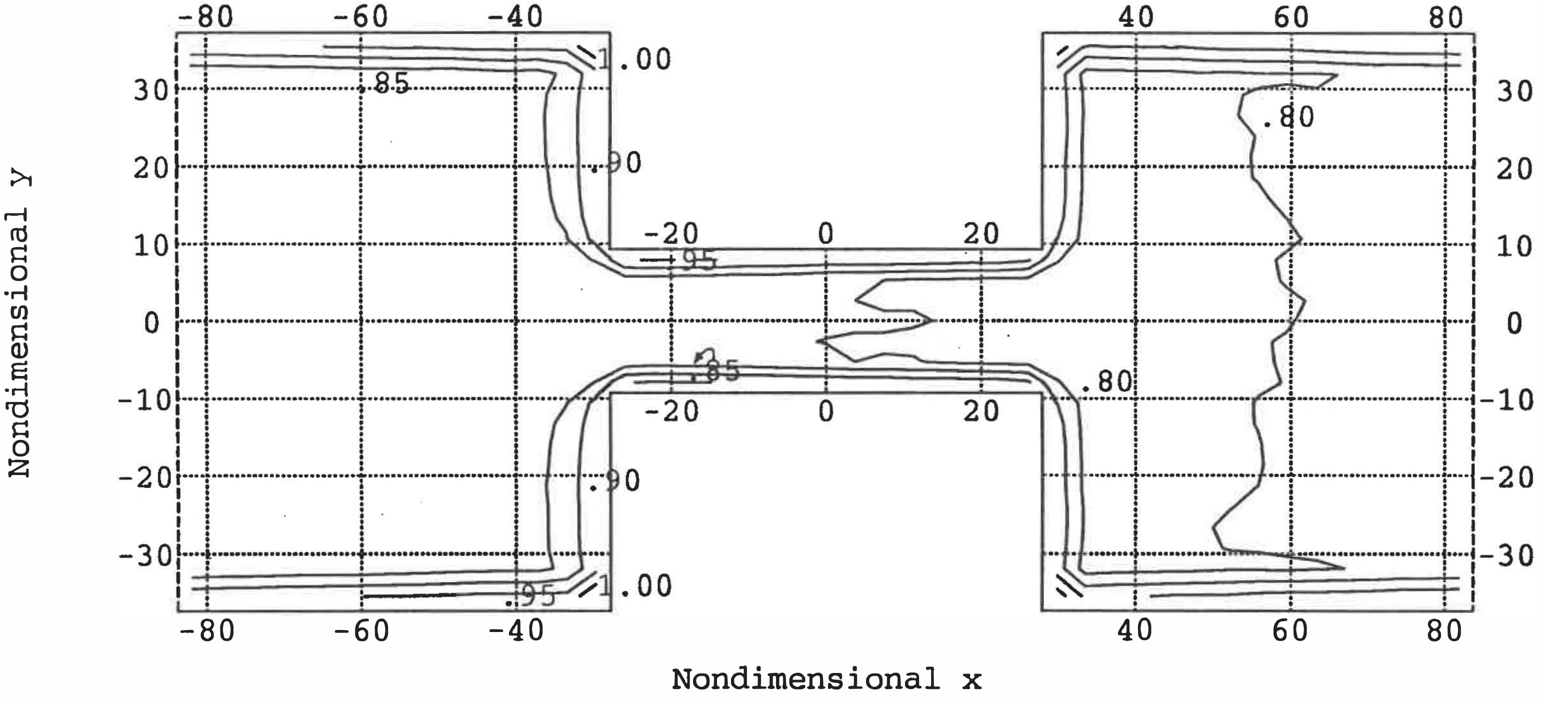


Figure 11a



run no. 218: g=0.02 TW/PBC SLJ

Figure 11b

run no. fe5: g=0.05 TW/PBC SD

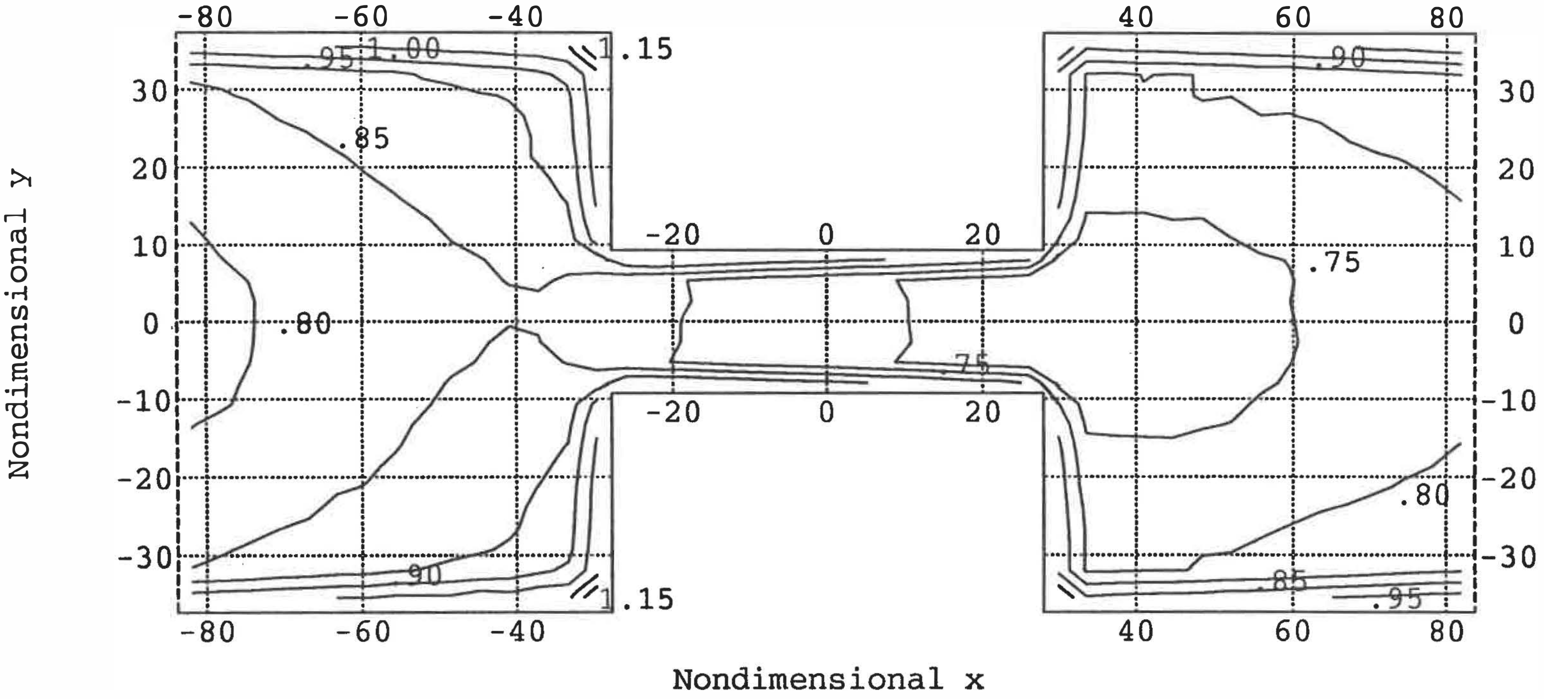


Figure 11c

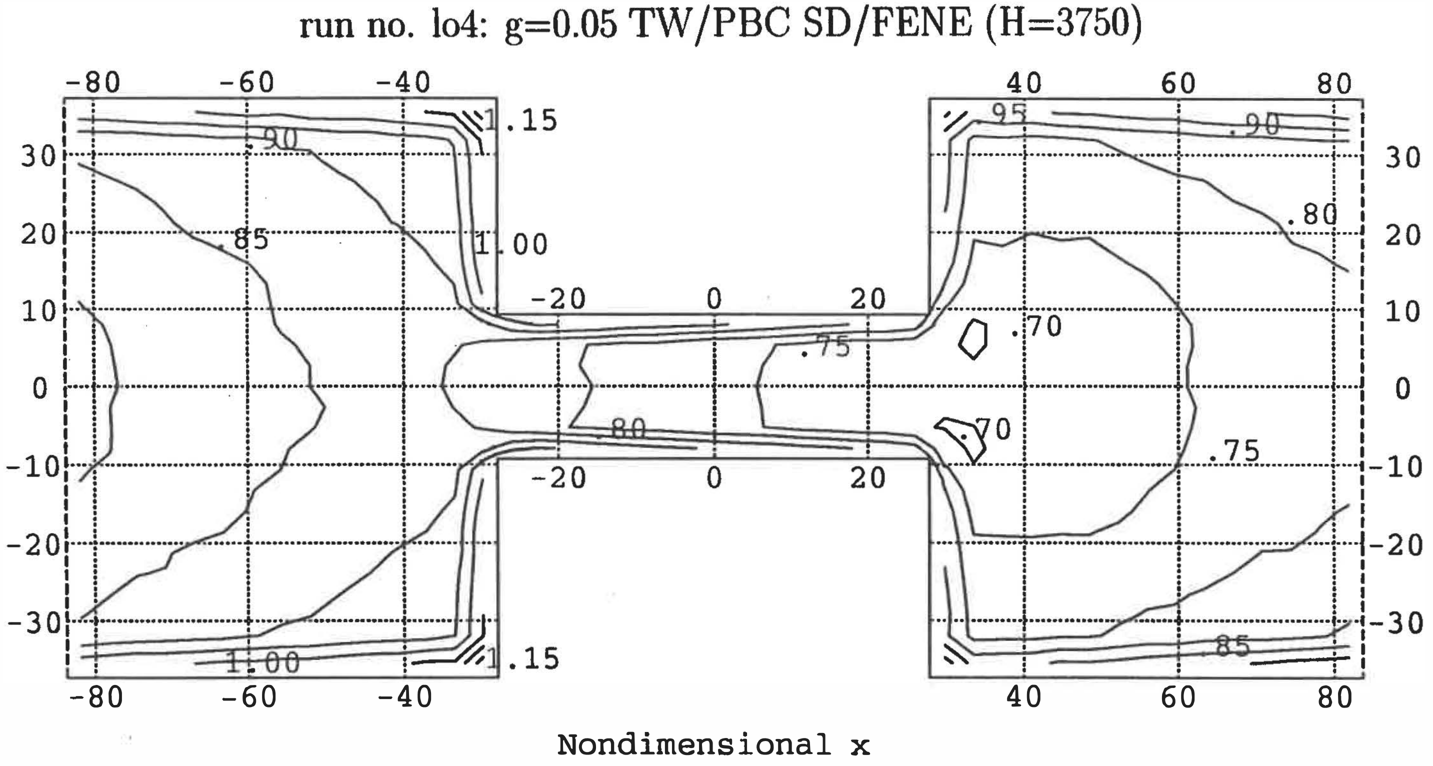

run no. lo4: g=0.05 TW/PBC SD/FENE (H=3750)
-80
-60
-40
40
60
80
-20
0
20
30
20
10
0
-10
-20
-30
.90
.85
1.15
1.00
.80
.75
.70
.95
.80
.75
.85
1.00
1.15
Nondimensional y
Nondimensional x

Figure 12a

run no. 218: g=0.02 TW/PBC SLJ

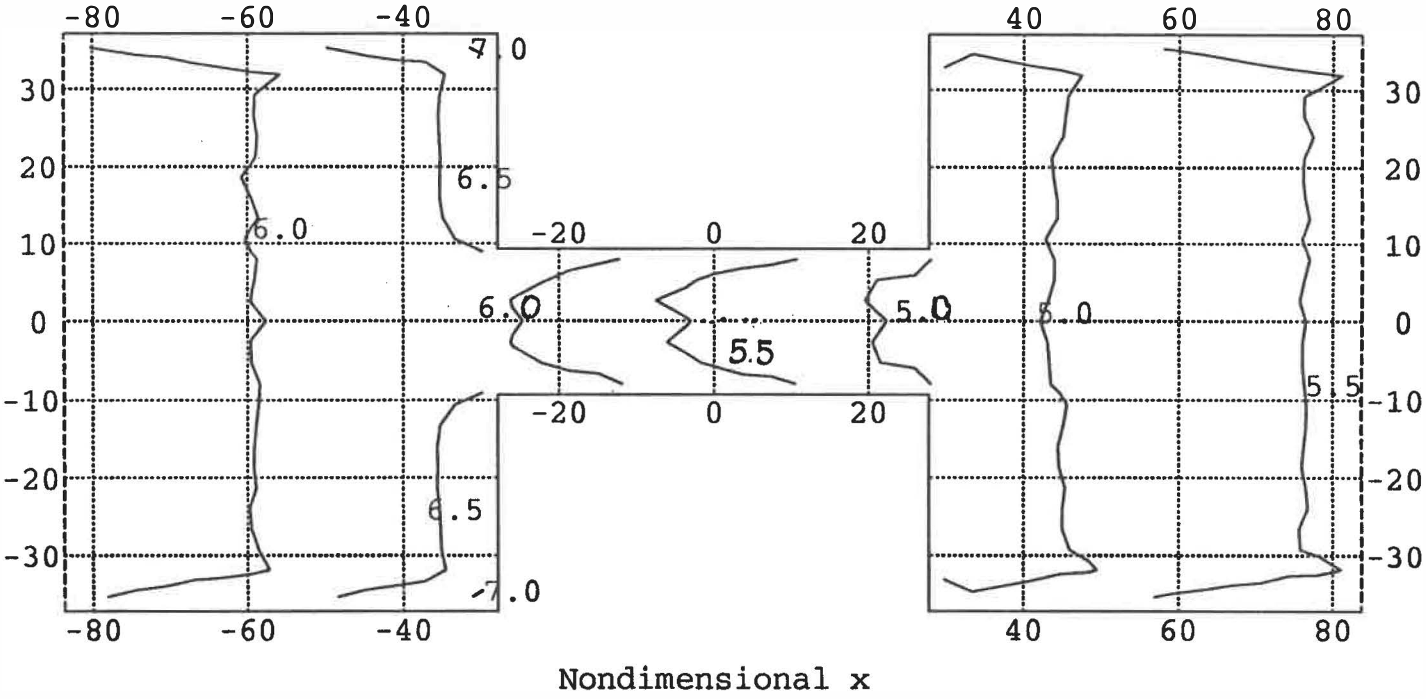


Figure 12b

run no. fe5: g=0.05 TW/PBC SD

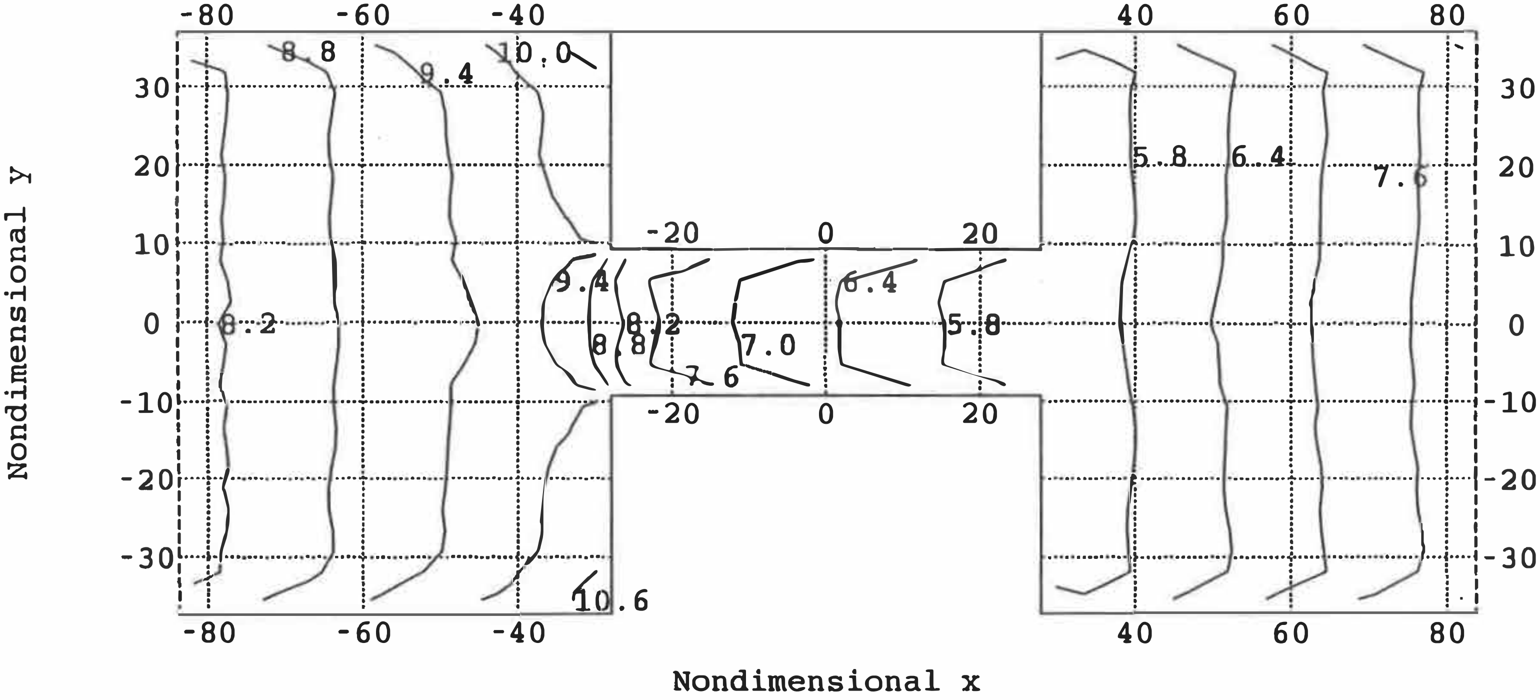
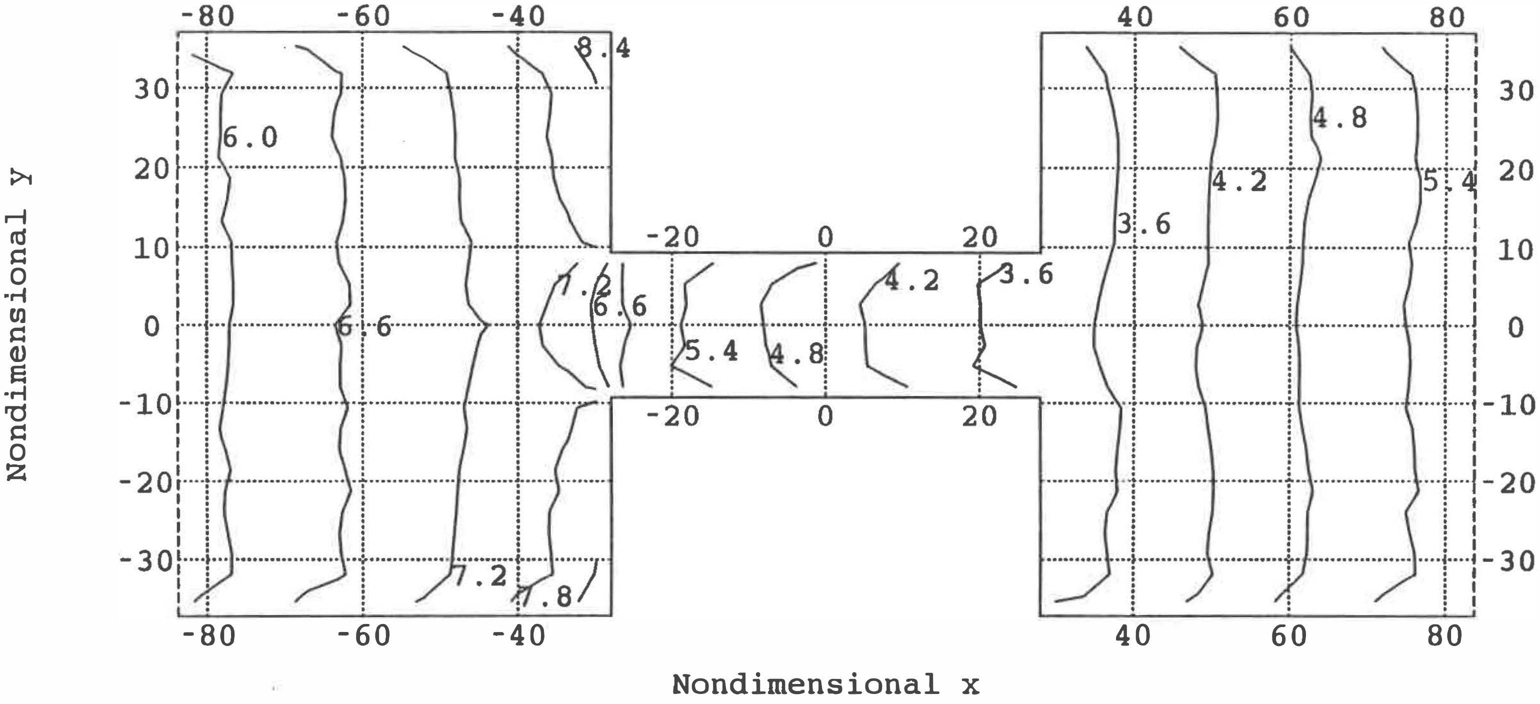


Figure 12c

run no. lo4: g=0.05 TW/PBC SD/FENE (H=3750)



5 CONCLUSIONS

5.1 Contribution to Knowledge

Our investigations of model microscopic fluids have resulted in a novel NEMD algorithm, led to new rheological, thermodynamical and microstructural data, and contributed to the understanding of the nature of the model fluids. The investigations have expanded the body of knowledge available to the humanity in general, and have produced interesting rheological results in particular. The most important achievements are described in the subsequent paragraphs.

1. Novel EMD Hamiltonian equations of motion are developed for rigid bodies and then tested for a particular implementation of the Gay-Berne (GB) potential. Following Grmela [1], these equations are extended to nonequilibrium systems by adding SLLOD-like terms to the equations for orientational unit vectors and angular momenta.

2. The viscometric functions are obtained for atomic and molecular fluids whose bonded (intramolecular) interaction are modeled by the finitely extensible nonlinear elastic (FENE) and the generalized Lennard-Jones (GLJ) potentials. In addition, the viscosity and the normal stress differences are computed for fluids composed of ellipsoids of revolution interacting according to the Gay-Berne potential. The FENE and GLJ fluids are studied with respect to the strength of the bonded interactions for several sizes of dumbbells. Vibrating dumbbells whose average end-to-end distance is 0.9 (in nondimensional units) are approximated by ellipsoids of revolution. The phenomena associated with viscoelastic fluids, such as shear thinning, Newtonian viscosity, non-zero normal stress difference, and volumetric dilatancy are observed for atomic and molecular fluids; however the curves of the viscometric functions are quite different than for vibrating dumbbells and ellipsoids of revolution. The stress tensor for fluids composed of rigid bodies is nonsymmetric and hence two shear viscosities are calculated. In the Newtonian limit both viscosity curves collapse into

one line. At the intermediate and higher shear rates, where the anisotropic effects are important, $\eta_1 = \frac{\sigma_{xy}}{\dot{\gamma}}$ is significantly above $\eta_2 = \frac{\sigma_{yx}}{\dot{\gamma}}$. Atomic liquids tend to thin at lower shear rates and are less elastic than molecular liquids.

3. The mesoscopic results are calculated directly in the simulations in terms of the contracted distribution functions in position and velocity spaces. The orientational and structural data are collected in the form of the conformation tensor and using several resolved pair distribution functions, one of them we denote as the second conformation tensor. Moreover, for rigid body fluids we compute the structure factor. The mesoscopic results not only confirm the structural rearrangements that take place at the microscopic level, but give additional information, such as the correlation between orientation of dumbbells and their average length. For all model liquids, the Maxwellian distribution of translational velocities is distorted at the higher shear rates. The nature of the very stiff dumbbell fluids is determined by the intramolecular interactions. At high shear rates, dumbbell fluids are liquid-like in the direction of flow and solid-like in the vorticity direction and (for the extremely high $\dot{\gamma}$) in the direction of the velocity gradient. The rigid body fluids are liquid-like in all directions.

4. At the molecular level, several new structures are identified for the first time, both for the vibrating dumbbells and for the rigid body fluids. When a fluid composed of rigid bodies is subjected to an imposed shear rate an oriented microstructure is formed in the direction of the fluid flow. At the intermediate shear rates, this microstructure is destroyed and replaced by an isotropic-like phase. At even higher shear rates, ellipsoids become oriented along the vorticity axis. Very small FENE dumbbell liquids tend to separate into horizontal layers (perpendicular to the velocity gradient, parallel to the direction of flow). Medium size FENE dumbbells build cross-spanning local microstructural arrangements which consists of two parallel strings that lie in the plane of constant velocity; each string contains only one bead of each dumbbell. On the other hand, two very large dumbbells may lock themselves into a cross-like arrangement that breaks at high $\dot{\gamma}$. For increasing shear

rates, the generalized Lennard-Jones dumbbells break if the intra-molecular bond is not stiff enough. Finally, the transient phenomena, such as the stress overshoot, are related to the formation of the directional order at the microstructural level.

5. Our thermodynamical results for Gay-Berne fluids remain in agreement with the published data for more elongated molecules [2, 3]. For the isotherm T=1, we observe the isotropic-nematic but no vapor-liquid phase transition.

6. It is shown that various NEMD algorithms yield consistent results, for fluids composed of disks. A stagnant layer of fluid adjacent to the stochastic wall tends to moderate the effect of the boundary condition on the bulk fluid. Flow structures obtained by direct simulations for flows in contracting and expanding channels are similar to those predicted by classical hydrodynamics and observed in experiments.

7. The computer codes developed in the course of this work are in the public domain and are freely available upon request.

5.2 Suggestions for Future Work

With the advent of fast computers, it has become feasible to investigate problems of engineering significance from the molecular point of view. The equilibrium molecular dynamics (EMD) simulations have come to be an accepted computational tool widely applied to problems of physical and medicinal chemistry (especially thermodynamics and drug design), materials science, physics and many other areas. This has led to the appearance of new fields of research, such as computational materials science, computational chemistry, computational solid and liquid state physics, computational astrophysics, *etc.* Recently, the molecular simulations have proliferated to chemical engineering to such an extent that during the annual AIChE meetings the sessions devoted to thermodynamics have been taken over by the Monte Carlo and EMD related calculations; papers on lattice gases have been also presented.

The work included in this thesis is one of the first application-driven attempts to use nonequilibrium molecular dynamics (NEMD) in polymer rheology. We have combined what is essentially methodology of theoretical and computational physics and chemistry with a well established procedures of chemical engineering with respect to formulating phenomenological models. We have not carried the process to the very end, that is to the modeling of a specific polymer processing operation. It will take some time until the direct methods gain enough maturity and robustness to be employed to such applications. Thus we make no specific recommendations for the immediate implementation of NEMD simulations within the context of the industrial polymer processing. However, in the subsequent paragraphs we give suggestions that could guide one on the way to the fulfillment of such a goal. We believe that in the course of this work a significant step forward has been made towards the ultimate industrial use of NEMD simulations.

It has taken several hundred years for classical (partial differential equations) methods to be developed to the point of yielding helpful predictions from the industrial perspective, by means of software packages. Yet, many problems simply cannot be solved by the continuum approach. Although useful these methods may not assist in answering some important questions, for they predict neither the morphology (e.g. fiber filled systems), nor mechanical/surface properties of the processed polymers. Large scale computational rheology or computational fluid dynamics calculations provide general information (pressure drop, flow structures) but fail to deliver detailed knowledge of a certain polymer property (*e.g.* surface quality). As the polymer processing industry grows mature, the problems to be tackled will shift from more general to more particular in nature. Applications seem to be endless for direct simulations, but one must be careful not to overestimate the current state-of-art of these techniques.

The specific suggestions for future research are enumerated below. The first spans directly from this work and is an attempt to improve direct simulation techniques to such an extend that they could be considered as an alternative to the classical methods. The second suggestion is a logical continuation of the first. Perhaps, it could be carried out, in cooperation with the industry, to result in the first commercial NEMD software package. In addition, it is feasible to investigate elongational viscosity by NEMD/SLLOD simulations, provided that the appropriate boundary conditions are found. Furthermore, EMD/NEMD algorithms may be applied to study advanced nanoscale ceramic/polymer composites, such as nanostructured electro-viscoelastic fluids; extension of Stockmayer fluids (see for example [4]) to nonequilibrium systems could be a convenient starting point. Nanostructured materials have become an exiting area of research in recent years mostly from the experimental point of view [5]. The computational techniques (such as NEMD) should provide an additional physical insight into the nature of these materials. Many other areas of polymer science could also be approached from the MD perspective, *e.g.* solid state polymer electrolytes.

1. *Unsolved problems of NEMD.*

- As observed experimentally, many polymeric fluids are incompressible but the incompressibility has not been incorporated into NEMD simulations. Under extreme gradients imposed on NEMD systems, this leads to an unacceptable density variation throughout a simulation domain.
- Up to know, all NEMD constant temperature calculations have been performed in reality for isokinetic systems, that is by controlling one or several kinetic (total, translational, rotational, vibrational) energies. In order to avoid computational artifacts and be able to relate NEMD results to laboratory experiments, the thermodynamic temperature should be used instead, requiring a method for calculation of the entropy for dense microscopic fluids. However, no such method exits. By the same token, thermodynamic rather than hydrostatic pressure ought to be used in NEMD simulations.
- The entropic dissipation in NEMD modeling comes about by the way of averaging and of round-off errors. A direct method of introducing the dissipation within NEMD algorithms would perhaps result in simulations requiring a smaller number of particles. To this end, we have attempted to implement several methods in the course of this work, but with no positive results to report.

2. *Novel approach to computational rheology and to computational fluid dynamics.*

Recently, Grmela [1] has formulated a system of coupled partial and ordinary differential equations that describe the behavior of suspensions by means of macroscopic (density and velocity fields), mesoscopic (conformation tensor and angular momentum fields) and microscopic (translational and rotational degrees of motion of single particles) variables. Although interesting on its own theoretical merit, due to the introduction of the angular momentum field within the context of hydrodynamics of suspensions, the system gives explicit coupling among variables at different levels of the physical description. If implemented numerically (provided that appropriate boundary conditions and potentials could be specified), the system would give predictions either at all levels of description, or separately at one particular level. In the latter case, equations formulated at other levels could be neglected. Judging by the complexity of the problem which in essence combines NEMD (or Brownian dynamics) and traditional simulations in one scheme, the equations would have to be coded on vector and/or parallel machines. 3-D visualizations routines could be incorporated to display simultaneously both macroscopic and microstructural information.

References

[1] M. Grmela. Mesoscopic hydrodynamics of suspensions. *Proc. Royal Soc. Lond. A (submitted)*, 1993.

[2] E de Miguel, L. F. Rull, M. K. Chalam, and K. E. Gubbins. Liquid-vapour coexistence of the Gay-Berne fluid by Gibbs-ensemble simulation. *Mol. Phys.*, 71(6):1223–1231, 1990.

[3] E de Miguel, L. F. Rull, M. K. Chalam, and K. E. Gubbins. Liquid crystal phase diagram of the Gay-Berne fluid. *Mol. Phys.*, 74(2):405–424, 1991.

[4] E. L. Pollock and B. J. Alder. Static dielectric properties of Stockmayers fluids. *Physica A*, 102:1–21, 1980.

[5] K. E. Drexler. *Nanosystems: Molecular Machinery, Manufacturing and Computations.* Wiley Intersience, New York, 1992.

APPENDIX: COMPUTER CODE

APPENDIX: COMPUTER CODE

As an example, we include computer code used for simulations of the behavior of a system of 256 rigid ellipsoids of revolution. Similar programs were developed for atomic and vibrating dumbbell fluids in case of rheological computations, and Lennard-Jones-like disk fluids for direct simulation of flow. The code is written in *C* and it compiles with any ANSI compliant compiler. Although the program was executed almost exclusively on RISC 320 with AIX 3.1.5 operational system, it was tested on SPARC station/SunOS (with GNU gcc compiler), 386/AT&T UNIX 3.2, 386/ESIX 4.0.3, and 486/386BSD 0.1.

Our approach was to make the program modular, easy to debug and test. The function prototypes which correspond to names of subroutines are listed in the include file rb.h. In this appendix, the subroutines are presented in the order in which they appear it the Makefile; rb.h is first. The global variables are declared in rb.h within the structure CurrentTrajectory. Since rb.h is included in all files, the structure CurrentTrajectory is common to all subroutines.

The flow of information in the program is controlled by *driver()* who initially sets up the faced-centered cubic lattice and assigns translational velocities to all particles by calling *fcc()* and *maxwell()* (in file max.c) respectively. If the current execution is a continuation of an old run then a call to *restore()* is issued. During the equilibration, *driver()* may issue calls to *scale_maxwell()* and *scale_rot()* if necessary.

The Gear order five predictor-corrector method *gear5()* is used to integrate the equations of motion. Runge-Kutta *rk4()* and the *leap-frog()* subroutines were also written to assist in evaluation of the Gear algorithm. Both *gear5()* and *rk4()* call other routines for evaluation of right hand sides of the equations of motion, for calculations of forces and for applications of the boundary conditions: *e.g gear5()* → *rh_sides()* → *gay_berne()* → (*virtual_pos()*, *ders()*). Subroutine *leap_frog()* by-passes *rh_sides()* calling directly *gay_berne()*.

Macroscopic and mesoscopic quantities are calculated either at every time step (*extra()*) or every 25 time steps (*radial()*). The averages are taken on-the-fly (*av_stress(), av_results(), av_velocity()*) during the execution, contributing to the overhead CPU time.

If necessary, particle trajectories could be also stored (*trstore()*) and then post-processed to extract the properties of interest or to prepare snapshots of the configurations.

In addition, subroutine *urng()* contains code for the universal random number generator (see header in urng.c). Subroutine *order()* monitors melting of the FCC lattice, calculates the order parameter (using the director found in *extra()*) and verifies the existence of the Maxwellian distribution function. Finally, the file utility.c contains an assembly of short codes used mostly for the dynamic allocation and deallocation of memory.

```
#Makefile
FILES =  driver.c urng.c fcc.c order.c max.c store.c forces.c ders.c\
         extra.c leap_frog.c gear5.c average.c utility.c rk4.c radial.c

OBJECTS =  driver.o urng.o fcc.o order.o max.o store.o forces.o ders.o\
           extra.o leap_frog.o gear5.o average.o utility.o rk4.o radial.o

HD = rb.h

LIBS = -lm

CC = cc
#CC = gcc
CFLAGS = -O

make: $(OBJECTS)
        $(CC) $(CFLAGS) $(OBJECTS) $(LIBS) -o rb25g10c

driver.o     : driver.c $(HD)
fcc.o        : fcc.c $(HD)
order.o      : order.c $(HD)
max.o        : max.c $(HD)
store.o      : store.c $(HD)
forces.o     : forces.c $(HD)
extra.o      : extra.c $(HD)
gear5.o      : gear5.c $(HD)
rk4.o        : rk4.c $(HD)
leap_frog.o  : leap_frog.c $(HD)
average.o    : average.c $(HD)
utility.o    : utility.c
urng.o       : urng.c
ders.o       : ders.c $(HD)
radial.o     : radial.c $(HD)
```

```
/*** rb.h ***/
/* This is an include file that contains all definitions
   and parameter for simulation of rheology of rigid bodies
   using the gear5 algorithm. */

/* Functions prototypes. */
/* In file utility.c     */
double **dmatrix(int nrl, int nrh, int ncl, int nch);
float **matrix(int nrl, int nrh, int ncl, int nch);
double *dvector(int nl, int nh);
float *vector(int nl, int nh);
int *ivector(int nl, int nh);
void free_dmatrix(double **m, int nrl, int nrh, int ncl, int nch);
void free_matrix(float **m, int nrl, int nrh, int ncl, int nch);
void free_dvector(double *m, int nl, int nh);
void free_vector(float *m, int nl, int nh);
void free_ivector(int *m, int nl, int nh);
void mderror(char *p);
void doublechange(double *tbc);
void intchange(int *tbc);
void floatchange(float *tbc);
/* In file fcc.c         */
void fcc(void);
/* In file urng.c        */
int urng(void);
double gauss(void);
/* In file order.c       */
void order(void);
/* In file max.c         */
void maxwell(void);
void check_maxwell(void);
void scale_maxwell(void);
void scale_rot(void);
/* In file store.c       */
void store(void);
void restore(void);
void trstore(void);
/* In file forces.c      */
void virtual_pos(void);
void gay_berne(void);
/* In file gear5.c       */
void gear5(double step);
void rh_sides(void);
/* In file rk4.c         */
void rk4(double step);
/* In file ders.c        */
void ders(double pi[], double pj[], double r[], double derv[]);
double pot(double pi[], double pj[], double r[]);
/* In file extra.c       */
void extra(void);
/* In file average.c     */
void av_stress(void);
void av_results(void);
void av_velocity(void);
/* In file radial.c      */
void radial(void);

struct CurrentTrajectory {
  int    istep,       /* current time step                 */
```

```
         fstep,       /* first time step                     */
         lstep,       /* last time step                      */
         shear_step,  /* when to start shearing              */
         isokin_step, /* when to start isokin calculations */
         ncut;        /* number of potential interactions   */
  double **rv,  /* mass-center virtual position - in primitive cube */
         **rp,**grp[4],**Rrp, /* mass-center physical positions     */
         **P,**gP[4],**RP,   /* momenta of mass-center              */
         **p,**gp[4],**Rp,   /* unit vector along long axis         */
         **m,**gm[4],**Rm,   /* angular momenta                     */
         **pot_forces,  /* forces -dp(V)/dp(xi), etc.  */
         **rot_forces,  /* forces -pi x dp(V)/dp(pi)   */
         **stresspot,   /* pot part of stress tensor   */
         **stresstkin,  /* trans part of stress tensor */
         **stresstot,   /* total stress tensor         */
         **conf,        /* conformation tensor         */
         ttemp,         /* translational temperature   */
         rtemp,         /* rotational temperature      */
         tekin,         /* translational kin energy    */
         ktx,kty,ktz,   /* components of tekin         */
         rekin,         /* rotational kinetic energy   */
         krx,kry,krz,   /* components of rekin         */
         tvel,          /* translational velocity      */
         rvel,          /* rotational velocity         */
         vor,   /* Verlet order parameter              */
         dor,   /* direction order parameter           */
         dorx,dory,dorz, /* components of dor          */
         tvelor,/* transl velocity order parameter     */
         ortx,orty,ortz, /* components of tvelor       */
         rvelor,/* rotat velocity order parameter      */
         orrx,orry,orrz, /* components of rvelor       */
         momx,momy,momz,mz, /* total trans and rot mom */
         epot,  /* potential energy                    */
         etot,  /* total energy                        */
         press, /* total energy                        */
         strain,        /* total strain                */
         strain_rate,   /* strain rate                 */
         dummy1,dummy2; /* to hold space in store file */
};

/* Particles */
#define NDIM    3   /* Number of dimensions */
#define NPART   256 /* Number of particles (2*NCALPAR^2*NLAYERS).  */
#define NLAYERS 8   /* Number of layers in the (2*NCALPAR).        */
#define NCALPAR 4   /* To calculate number of particles in a layer.*/

/* Geometry */
#define LX  10.079369 /* 0f Direction of flow              */
#define LY  10.079369 /* 1f Direction of velocity gradient */
#define LZ  10.079369 /* 2f LZ must be equal to LX         */
/* To avoid placing particles outside of the simulation box we should have
   LY>=NLAYERS/(2xNCALPAR)*LZ */

/* Diagonal elements of I and I^-1 tensors */
#define I1   0.461
#define I2   0.461
#define I3   0.2
#define II1  2.169197397
#define II2  2.169197397
#define II3  5.0
```

```
/* Initialization */
#define RESTART      1           /* 0 for a new run. */
#define NSEQUIL      0           /* Number of steps for equilibration.       */
#define NRESVEL      50          /* How often to rescale velocities >0.      */
#define NSMIC        0           /* Number of steps for microcanonical ens.  */
#define NSKIN        0           /* Number of steps for isokinetic ens.      */
#define NSSHEAR      20000       /* Number of steps for SLLOD.               */

/* Temperature and friends */
#define TREF           1.000   /* Reference temperature.           */
#define RCUTOFF        1000.0  /* Square of the cut-off radius.    */
#define STRAIN_RATE    10.0    /* Strain rate.                     */
#define TSTEP          0.00125 /* Time step.                       */

/* Start, restart and trajectory */
#define STORENAME      "rb25g10c.store"
#define RESTORENAME    "rb25g10b.store"
#define STORETRAJ      "rb25g10c.traj"
#define STORESTRESS    "rb25g10c.stress"
#define STORERESULT    "rb25g10c.result"
#define STOREVELOCITY  "rb25g10c.velocity"
#define RADIAL         "rb25g10c.radial"
#define VELDIST        "rb25g10c.veldist"
#define NSTORE       100      /* How often to store results for restarting. */
#define NTRAJSTORE   1000000  /* How often to store trajectory.              */
#define BINSIZE      2000     /* Number of steps in every bin.               */
```

```
/* driver.c */

/* Performs administrative functions calling subroutines to
   do particular tasks.
*/

#include <stdio.h>
#include <math.h>
#include <time.h>
#include "rb.h"

struct CurrentTrajectory theTrajectory;

void main(void)
{
  int i,j,k;
  unsigned long t1,t2,t3;
  struct tm *local;
  time_t t1t,t2t,t3t;
  double **rv,**rp,**P,**p,**m,r2;

  t1=clock(); t1t=time(NULL); local=localtime(&t1t);
  fprintf(stdout,"Time before initialization: %s",asctime(local));

  for(i=0;i<20000;i++) urng(); /* Warm-up the urng. */

  /* Memory for dynamic variables */
  for(i=0;i<=3;i++) {
    theTrajectory.grp[i]=dmatrix(1,NDIM,1,NPART);
    theTrajectory.gP[i]=dmatrix(1,NDIM,1,NPART);
    theTrajectory.gp[i]=dmatrix(1,NDIM,1,NPART);
    theTrajectory.gm[i]=dmatrix(1,NDIM,1,NPART);
  }
  theTrajectory.rv=dmatrix(1,NDIM,1,NPART); rv=theTrajectory.rv;
  theTrajectory.rp=dmatrix(1,NDIM,1,NPART); rp=theTrajectory.rp;
  theTrajectory.P=dmatrix(1,NDIM,1,NPART);  P=theTrajectory.P;
  theTrajectory.p=dmatrix(1,NDIM,1,NPART);  p=theTrajectory.p;
  theTrajectory.m=dmatrix(1,NDIM,1,NPART);  m=theTrajectory.m;
  theTrajectory.Rrp=dmatrix(1,NDIM,1,NPART);
  theTrajectory.RP=dmatrix(1,NDIM,1,NPART);
  theTrajectory.Rp=dmatrix(1,NDIM,1,NPART);
  theTrajectory.Rm=dmatrix(1,NDIM,1,NPART);
  theTrajectory.stresspot=dmatrix(1,NDIM,1,NDIM);
  theTrajectory.stresstot=dmatrix(1,NDIM,1,NDIM);
  theTrajectory.stresstkin=dmatrix(1,NDIM,1,NDIM);
  theTrajectory.conf=dmatrix(1,NDIM,1,NDIM);
  theTrajectory.pot_forces=dmatrix(1,NDIM,1,NPART);
  theTrajectory.rot_forces=dmatrix(1,NDIM,1,NPART);
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      theTrajectory.pot_forces[i][j]=0.0; theTrajectory.rot_forces[i][j]=0.0;
      theTrajectory.Rrp[i][j]=0.0; theTrajectory.RP[i][j]=0.0;
      theTrajectory.Rp[i][j]=0.0; theTrajectory.Rm[i][j]=0.0;
  } }

  if(!RESTART) {
    theTrajectory.fstep=1; theTrajectory.strain=0.0;
    for(i=1;i<=NDIM;i++) {           /* Zero everything - just in case */
      for(j=1;j<=NPART;j++) {
        rv[i][j]=0.0; rp[i][j]=0.0; P[i][j]=0.0; p[i][j]=0.0; m[i][j]=0.0;
  } }
  for(i=0;i<=3;i++) {                              /* Zero these array   */
    for(j=1;j<=NDIM;j++) {
      for(k=1;k<=NPART;k++) {
        theTrajectory.grp[i][j][k]=0.0;
        theTrajectory.gP[i][j][k]=0.0;
        theTrajectory.gp[i][j][k]=0.0;
        theTrajectory.gm[i][j][k]=0.0;
  } } }
  fcc(); maxwell();
}
else {
  restore(); theTrajectory.fstep=theTrajectory.istep+1;
}
theTrajectory.lstep=theTrajectory.fstep+NSEQUIL+NSMIC+NSKIN+NSSHEAR-1;
theTrajectory.shear_step=theTrajectory.fstep+NSEQUIL+NSMIC+NSKIN;
theTrajectory.isokin_step=theTrajectory.fstep+NSEQUIL+NSMIC;

/* Start the calculation loop */
for(theTrajectory.istep=theTrajectory.fstep;
    theTrajectory.istep<=theTrajectory.lstep;theTrajectory.istep++) {

  /* Scale temp during equilibration */
  if(theTrajectory.istep<=(theTrajectory.fstep+NSEQUIL)) {
    if(theTrajectory.istep%NRESVEL==0) { scale_maxwell(); scale_rot();};
    if((theTrajectory.istep==(theTrajectory.fstep+NSEQUIL))&&
       (NSEQUIL==0||NSMIC==0)&&(theTrajectory.istep!=1))
         { scale_maxwell(); scale_rot(); };
  }

  /*leap(TSTEP); rk4(TSTEP); */
  gear5(TSTEP); extra(); order(); av_stress(); av_results(); av_velocity();
  if(theTrajectory.istep%25==0) radial();             /* Radial dist. func. */
  if(theTrajectory.istep%NSTORE==0) store();          /* Restart            */
  if(theTrajectory.istep%NTRAJSTORE==0) trstore();    /* Trajectory         */

  if(theTrajectory.istep>=theTrajectory.isokin_step) {
    if((theTrajectory.ttemp>1.001)||(theTrajectory.ttemp<0.999)) {
      scale_maxwell();
      fprintf(stderr, "Trans energy flow rescale %7d\n",theTrajectory.istep);
    }
    if((theTrajectory.rtemp>1.001)||(theTrajectory.rtemp<0.999)) {
      scale_rot();
      fprintf(stderr, "Rot energy flow rescale %7d\n",theTrajectory.istep);
    }
  }

  if(theTrajectory.istep%10==0) {
    fprintf(stdout, "%5d%11.3e%11.3e%11.3e%11.3e%11.3e%11.3e%11.3e%11.3e%11.3e
      theTrajectory.istep,theTrajectory.ttemp,theTrajectory.ktx,
      theTrajectory.kty,theTrajectory.ktz,theTrajectory.rtemp,
      theTrajectory.krx,theTrajectory.kry,theTrajectory.krz,
      theTrajectory.epot,theTrajectory.etot);
  }
  /* check_maxwell();*/
}

/* Release memory for dynamic variables */
free_dmatrix(theTrajectory.rv,1,NDIM,1,NPART);
free_dmatrix(theTrajectory.rp,1,NDIM,1,NPART);
```

```
  free_dmatrix(theTrajectory.P,1,NDIM,1,NPART);
  free_dmatrix(theTrajectory.p,1,NDIM,1,NPART);
  free_dmatrix(theTrajectory.m,1,NDIM,1,NPART);
  free_dmatrix(theTrajectory.pot_forces,1,NDIM,1,NPART);
  free_dmatrix(theTrajectory.rot_forces,1,NDIM,1,NPART);
  free_dmatrix(theTrajectory.stresspot,1,NDIM,1,NDIM);
  free_dmatrix(theTrajectory.stresstot,1,NDIM,1,NDIM);
  free_dmatrix(theTrajectory.stresstkin,1,NDIM,1,NDIM);
  free_dmatrix(theTrajectory.conf,1,NDIM,1,NDIM);
  free_dmatrix(theTrajectory.Rrp,1,NDIM,1,NPART);
  free_dmatrix(theTrajectory.RP,1,NDIM,1,NPART);
  free_dmatrix(theTrajectory.Rp,1,NDIM,1,NPART);
  free_dmatrix(theTrajectory.Rm,1,NDIM,1,NPART);
  for(i=0;i<=3;i++) {
    free_dmatrix(theTrajectory.grp[i],1,NDIM,1,NPART);
    free_dmatrix(theTrajectory.gP[i],1,NDIM,1,NPART);
    free_dmatrix(theTrajectory.gp[i],1,NDIM,1,NPART);
    free_dmatrix(theTrajectory.gm[i],1,NDIM,1,NPART);
  }

  t3=clock(); t3t=time(NULL); local=localtime(&t3t);
  fprintf(stdout,"Clock cycles = %u, Total time: %s", t3-t2,asctime(local));
}
```

```
/* fcc.c */

/* This routine sets up the face-centered cubic lattice (FCC) for 2kn^2 atoms
   in a parallelepiped (see description of k and n in rb.h). A simpler
   program to set up the lattice in a cubic box for 256 atoms may be found in
   Dieter W. Heermann, Computer Simulation Methods in Theoretical Physics,
   Springer-Verlag, 1986 p122.
*/

#include <stdio.h>
#include <math.h>
#include "rb.h"

extern struct CurrentTrajectory theTrajectory;

void fcc(void)
{
  int i,j,k;
  double r,dist,dist4,dist2,*Posx,*Posy,*Posz,lxh,lyh,lzh;

  fprintf(stdout, "Setting the initial positions and directions of molecules ...
  dist=LZ/NCALPAR;
  dist4=dist/4.0;
  dist2=dist/2.0;

  /* Get some useful pointers. */
  Posx=theTrajectory.rp[1];
  Posy=theTrajectory.rp[2];
  Posz=theTrajectory.rp[3];
  *Posx++;*Posy++;*Posz++;

  /* Place particles on the FCC lattice - loop over no. of layers / 2 */
  for(k=0;k<NLAYERS/2;k++) { /* note y is vertical */
    /* Get odd planes x-z */
    for(j=0;j<NCALPAR;j++) {
      for(i=0;i<2*NCALPAR;i++) {
        *Posz++=dist4+i*dist2;
        *Posx++=dist4+j*dist+i%2*dist2;
        *Posy++=dist4+k*dist;
    } }
    /* Get even planes x-z */
    for(j=0;j<NCALPAR;j++) {
      for(i=0;i<2*NCALPAR;i++) {
        *Posz++=dist4+i*dist2;
        *Posx++=dist4+dist2+j*dist-i%2*dist2;
        *Posy++=dist4+dist2+k*dist;
  } } }
  /* Shift centre of box to the origin */
  Posx=theTrajectory.rp[1];
  Posy=theTrajectory.rp[2];
  Posz=theTrajectory.rp[3];
  lxh=0.5*LX;lyh=0.5*LY;lzh=0.5*LZ;
  for(i=1;i<=NPART;i++) {
    Posx[i]-=lxh;
    Posy[i]-=lyh;
    Posz[i]-=lzh;
  }
  /* Set orientations - in z (neutral) direction */
  Posx=theTrajectory.p[1];
  Posy=theTrajectory.p[2];
  Posz=theTrajectory.p[3];
  for(i=1;i<=NPART;i++) {
    Posx[i]=0.0; Posy[i]=0.0; Posz[i]=1.0;
  }
  theTrajectory.dorx=0.0; theTrajectory.dory=0.0; theTrajectory.dorz=1.0;
  /* Set orientations randomly - not recommended for higher densities
  for(i=1;i<=NPART;i++) {
    Posx[i]=urng(); Posy[i]=urng(); Posz[i]=urng();
    r=sqrt(Posx[i]*Posx[i]+Posy[i]*Posy[i]+Posz[i]*Posz[i]);
    Posx[i]/=r; Posy[i]/=r; Posz[i]/=r;
  }*/
}
```

```
/* order.c */

/* This routine calculates the Verlet and direction order parameters,
   and in addition the kinetic part of the H-funtion for velocities (see Haile 1
   it is useful for equilibration.
*/

#include <stdio.h>
#include <math.h>
#include "rb.h"

extern struct CurrentTrajectory theTrajectory;

void order(void)
{
  int i,index1,index2,index3,px[601],py[601],pz[601];
  double hx,hy,hz,fx[601],fy[601],fz[601];
  double dist,*Posx,*Posy,*Posz,**P;
  double r;

  /* Verlet order parameter  */
  theTrajectory.vor=0.0; dist=4.0*M_PI*NCALPAR/LX;
  Posx=theTrajectory.rp[1]; Posy=theTrajectory.rp[2]; Posz=theTrajectory.rp[3];
  for(i=1;i<=NPART;i++)
    theTrajectory.vor+=cos((Posx[i]+Posy[i]+Posz[i])*dist);
  theTrajectory.vor/=NPART;

  /* Direction order parameter */
  Posx=theTrajectory.p[1]; Posy=theTrajectory.p[2]; Posz=theTrajectory.p[3];
  theTrajectory.dor=0.0;
  for(i=1;i<=NPART;i++) {
    r=Posx[i]*theTrajectory.dorx+Posy[i]*theTrajectory.dory
     +Posz[i]*theTrajectory.dorz;
    theTrajectory.dor+=r*r;
  }
  theTrajectory.dor=1.5*theTrajectory.dor/NPART-0.5;

  /* Verify that r is constant */
  for(i=1;i<=NPART;i++) {
    r=Posx[i]*Posx[i]+Posy[i]*Posy[i]+Posz[i]*Posz[i];
    if(r>1.0005||r<.9995) {
      r=sqrt(r); Posx[i]/=r; Posy[i]/=r; Posz[i]/=r;
        fprintf(stderr,"p %4d in step %5d %13.6e%13.6e%13.6e%13.6e\n",
                         i,theTrajectory.istep,Posx[i],Posy[i],Posz[i],r);
    }
  }

  /* Translational velocity distribution */
  P=theTrajectory.P;
  for(i=0;i<=600;i++) { px[i]=0; py[i]=0; pz[i]=0; }
  for(i=1;i<=NPART;i++) {
    index1=(int)((0.5*P[1][i]+15.015)*20.0);   /* 0.5 is for mass */
    if(index1>-1&&index1<601) { px[index1]+=1; }
     else { fprintf(stderr,"order: tvelx out of bounds%5d%13.3e\n",i,P[1][i]); }
    index2=(int)((0.5*P[2][i]+15.015)*20.0);
    if(index2>-1&&index2<601) { py[index2]+=1; }
     else { fprintf(stderr,"order: tvely out of bounds%5d%13.3e\n",i,P[2][i]); }
    index3=(int)((0.5*P[3][i]+15.015)*20.0);
    if(index3>-1&&index3<601) { pz[index3]+=1; }
```

```
      else { fprintf(stderr,"order: tvelz out of bounds%5d%13.3e\n",i,P[3][i]); }
  }
  hx=1.0/(NPART*0.05);
  for(i=0;i<=600;i++)  {
    fx[i]=(double)px[i]*hx; fy[i]=(double)py[i]*hx;
    fz[i]=(double)pz[i]*hx;
  }
  hx=0.0; hy=0.0; hz=0.0;
  for(i=0;i<=600;i++)  {
    if(px[i]!=0) { hx+=fx[i]*log(fx[i]); }
    if(py[i]!=0) { hy+=fy[i]*log(fy[i]); }
    if(pz[i]!=0) { hz+=fz[i]*log(fz[i]); }
  }
  hx*=0.05; hy*=0.05; hz*=0.05;
  theTrajectory.ortx=hx; theTrajectory.orty=hy; theTrajectory.ortz=hz;
  theTrajectory.tvelor=(hx+hy+hz)/3.0;
  /* Rotational velocity distribution - reuse pointers */
  P=theTrajectory.m;
  for(i=0;i<=600;i++) { px[i]=0; py[i]=0; pz[i]=0; }
  for(i=1;i<=NPART;i++) {
    index1=(int)((II1*P[1][i]+15.015)*20.0);   /* 0.5 is for mass */
    if(index1>-1&&index1<601) { px[index1]+=1; }
     else { fprintf(stderr,"order: rvelx out of bounds%5d%13.3e\n",i,P[1][i]); }
    index2=(int)((II2*P[2][i]+15.015)*20.0);
    if(index2>-1&&index2<601) { py[index2]+=1; }
     else { fprintf(stderr,"order: rvely out of bounds%5d%13.3e\n",i,P[2][i]); }
    index3=(int)((II3*P[3][i]+15.015)*20.0);
    if(index3>-1&&index3<601) { pz[index3]+=1; }
     else { fprintf(stderr,"order: rvelz out of bounds%5d%13.3e\n",i,P[3][i]); }
  }
  hx=1.0/(NPART*0.05);
  for(i=0;i<=600;i++)  {
    fx[i]=(double)px[i]*hx; fy[i]=(double)py[i]*hx; fz[i]=(double)pz[i]*hx;
  }
  hx=0.0; hy=0.0; hz=0.0;
  for(i=0;i<=600;i++)  {
    if(px[i]!=0) { hx+=fx[i]*log(fx[i]); }
    if(py[i]!=0) { hy+=fy[i]*log(fy[i]); }
    if(pz[i]!=0) { hz+=fz[i]*log(fz[i]); }
  }
  hx*=0.05; hy*=0.05; hz*=0.05;
  theTrajectory.orrx=hx; theTrajectory.orry=hy; theTrajectory.orrz=hz;
  theTrajectory.rvelor=(hx+hy+hz)/3.0;
}
```

```
/* max.c */

/* Subroutine maxwell() initializes the translational momenta
   according to the Maxwell-Boltzmann distribution. Subroutines
   scale_maxwell() and scale_rot() rescale the translational and
   rotational velocities. Note: the mass of a single molecule
   is 2.
*/

#include <stdio.h>
#include <math.h>
#include "rb.h"

extern struct CurrentTrajectory theTrajectory;

void maxwell(void)
{
  int i,j;
  double **P,rtemp,sum;

  fprintf(stdout, "Setting the initial translational and angular momenta ...\n")

  P=theTrajectory.P;

  /* TRANSLATIONAL MOMENTA      */
  /** Adjust the second moment */
  rtemp=sqrt(2.0*TREF);
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) P[i][j]=rtemp*gauss();
  }
  /** Set the momenta to zero */
  sum=0.0; for(j=1;j<=NPART;j++) sum+=P[1][j];
  sum/=NPART; for(j=1;j<=NPART;j++) P[1][j]-=sum;
  sum=0.0; for(j=1;j<=NPART;j++) sum+=P[2][j];
  sum/=NPART; for(j=1;j<=NPART;j++) P[2][j]-=sum;
  sum=0.0; for(j=1;j<=NPART;j++) sum+=P[3][j];
  sum/=NPART; for(j=1;j<=NPART;j++) P[3][j]-=sum;
}

/* To check the velocity distribution - normally not used. */
void check_maxwell(void)
{
 int i,j;
 double **P,Px,Py,Pz,temp,vel,PiPi;

 P=theTrajectory.P;

 /* TRANSLATIONAL MOMENTA      */
 Px=0.0; for(j=1;j<=NPART;j++) Px+=P[1][j];
 Py=0.0; for(j=1;j<=NPART;j++) Py+=P[2][j];
 Pz=0.0; for(j=1;j<=NPART;j++) Pz+=P[3][j];
 temp=0.0; vel=0.0;
 for(j=1;j<=NPART;j++) {
   PiPi=P[1][j]*P[1][j]+P[2][j]*P[2][j]+P[3][j]*P[3][j];
   temp+=PiPi; vel+=sqrt(PiPi);
 }
 temp/=(6.0*NPART);       /* from kinetic energy; mass is 2 */
 vel/=NPART; vel=vel*vel;
 vel*=M_PI/16.0;          /* from average speed; mass is 2   */
 fprintf(stderr,"(Trans.) TREF=%11.3e T(<p^2>)=%11.3e T(<p>)=%11.3e\n",
              TREF,temp,vel);
}

void scale_maxwell(void)
{
  int i,j;
  double **P,alpha,sum;

  P=theTrajectory.P;

  /* TRANSLATIONAL MOMENTA     */
  /** Set the momenta to zero */
  sum=0.0; for(j=1;j<=NPART;j++) sum+=P[1][j];
  sum/=NPART; for(j=1;j<=NPART;j++) P[1][j]-=sum;
  sum=0.0; for(j=1;j<=NPART;j++) sum+=P[2][j];
  sum/=NPART; for(j=1;j<=NPART;j++) P[2][j]-=sum;
  sum=0.0; for(j=1;j<=NPART;j++) sum+=P[3][j];
  sum/=NPART; for(j=1;j<=NPART;j++) P[3][j]-=sum;
  alpha=0;
  for(j=1;j<=NPART;j++) {
    alpha+=P[1][j]*P[1][j]+P[2][j]*P[2][j]+P[3][j]*P[3][j];
  }
  alpha=sqrt(6.0*NPART*TREF/alpha);
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) P[i][j]*=alpha;
  }
  fprintf(stderr,"(Trans.) alpha=%11.3e\n",alpha);
}

void scale_rot(void)
{
  int i,j;
  double **m,alpha;

  m=theTrajectory.m;

  /* ROTATIONAL ENERGY */
  alpha=0;
  for(j=1;j<=NPART;j++) {
    alpha+=II1*m[1][j]*m[1][j]+II2*m[2][j]*m[2][j]+II3*m[3][j]*m[3][j];
  }
  alpha=sqrt(2.0*NPART*TREF/alpha);
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) m[i][j]*=alpha;
  }
  fprintf(stderr,"(Rot.) alpha=%11.3e\n",alpha);
}
```

```
/* store.c */

/* These routines store and restore the restart files.
   Subroutine trstore() stores the trajectory.
*/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include "rb.h"

extern struct CurrentTrajectory theTrajectory;

void store(void)
{
  int i,j,k;
  double strain,strain_rate,dummy1,dummy2,**rv,**rp,**P,**p,**m;
  double **grp[4],**gP[4],**gp[4],**gm[4];
  FILE *fp;

  if((fp=fopen(STORENAME,"wb"))==NULL) {
    fprintf(stderr,"Cannot open file\n");
    exit(1);
  }
  strain=theTrajectory.strain; strain_rate=theTrajectory.strain_rate;
  dummy1=theTrajectory.dummy1; dummy2=theTrajectory.dummy2;
  rv=theTrajectory.rv; rp=theTrajectory.rp; P=theTrajectory.P;
  p=theTrajectory.p; m=theTrajectory.m;
  for(k=0;k<4;k++) {
    grp[k]=theTrajectory.grp[k]; gP[k]=theTrajectory.gP[k];
    gp[k]=theTrajectory.gp[k]; gm[k]=theTrajectory.gm[k];
  }
  if(fwrite(&theTrajectory.istep,sizeof(int),1,fp)!=1) {
    fprintf(stderr,"W error\n");
  }
  if(fwrite(&strain,sizeof(double),1,fp)!=1) {
    fprintf(stderr,"W error\n");
  }
  if(fwrite(&strain_rate,sizeof(double),1,fp)!=1) {
    fprintf(stderr,"W error\n");
  }
  if(fwrite(&dummy1,sizeof(double),1,fp)!=1) {
    fprintf(stderr,"W error\n");
  }
  if(fwrite(&dummy2,sizeof(double),1,fp)!=1) {
    fprintf(stderr,"W error\n");
  }
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      if(fwrite(&rv[i][j],sizeof(double),1,fp)!=1) fprintf(stderr,"W error\n");
      if(fwrite(&rp[i][j],sizeof(double),1,fp)!=1) fprintf(stderr,"W error\n");
      if(fwrite(&P[i][j],sizeof(double),1,fp)!=1) fprintf(stderr,"W error\n");
      if(fwrite(&p[i][j],sizeof(double),1,fp)!=1) fprintf(stderr,"W error\n");
      if(fwrite(&m[i][j],sizeof(double),1,fp)!=1) fprintf(stderr,"W error\n");
      for(k=0;k<4;k++) {
        if(fwrite(&grp[k][i][j],sizeof(double),1,fp)!=1)
          fprintf(stderr,"W error\n");
        if(fwrite(&gP[k][i][j],sizeof(double),1,fp)!=1)
          fprintf(stderr,"W error\n");
        if(fwrite(&gp[k][i][j],sizeof(double),1,fp)!=1)
          fprintf(stderr,"W error\n");
        if(fwrite(&gm[k][i][j],sizeof(double),1,fp)!=1)
          fprintf(stderr,"W error\n");
      }
  } }
  fclose(fp);
}


void restore(void)
{
  int i,j,k;
  double strain,strain_rate,dummy1,dummy2,**rv,**rp,**P,**p,**m;
  double **grp[4],**gP[4],**gp[4],**gm[4];
  FILE *fp;

  rv=theTrajectory.rv; rp=theTrajectory.rp; P=theTrajectory.P;
  p=theTrajectory.p; m=theTrajectory.m;
  for(k=0;k<4;k++) {
    grp[k]=theTrajectory.grp[k]; gP[k]=theTrajectory.gP[k];
    gp[k]=theTrajectory.gp[k]; gm[k]=theTrajectory.gm[k];
  }
  if((fp=fopen(RESTORENAME,"rb"))==NULL) {
    fprintf(stderr,"Cannot open file\n"); exit(1);
  }
  if(fread(&theTrajectory.istep,sizeof(int),1,fp)!=1) {
    fprintf(stderr,"R error (nstep)\n"); exit(1);
  }
  if(fread(&strain,sizeof(double),1,fp)!=1) {
    fprintf(stderr,"R error (strain)\n"); exit(1);
  }
  if(fread(&strain_rate,sizeof(double),1,fp)!=1) {
    fprintf(stderr,"R error (strain_rate)\n"); exit(1);
  }
  if(fread(&dummy1,sizeof(double),1,fp)!=1) {
    fprintf(stderr,"R error (dummy1)\n"); exit(1);
  }
  if(fread(&dummy2,sizeof(double),1,fp)!=1) {
    fprintf(stderr,"R error (dummy2)\n"); exit(1);
  }
  theTrajectory.strain=strain; theTrajectory.strain_rate=strain_rate;
  theTrajectory.dummy1=dummy1; theTrajectory.dummy2=dummy2;
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      if(fread(&rv[i][j],sizeof(double),1,fp)!=1) {
        fprintf(stderr,"R error (rv)\n"); exit(1);
      }
      if(fread(&rp[i][j],sizeof(double),1,fp)!=1) {
        fprintf(stderr,"R error (rp)\n"); exit(1);
      }
      if(fread(&P[i][j],sizeof(double),1,fp)!=1) {
        fprintf(stderr,"R error (P)\n"); exit(1);
      }
      if(fread(&p[i][j],sizeof(double),1,fp)!=1) {
        fprintf(stderr,"R error (p)\n"); exit(1);
      }
      if(fread(&m[i][j],sizeof(double),1,fp)!=1) {
        fprintf(stderr,"R error (m)\n"); exit(1);
      }
      for(k=0;k<4;k++) {
        if(fread(&grp[k][i][j],sizeof(double),1,fp)!=1) {
          fprintf(stderr,"R error (grp)\n"); exit(1);
        }
        if(fread(&gP[k][i][j],sizeof(double),1,fp)!=1) {
          fprintf(stderr,"R error (gP)\n"); exit(1);
        }
        if(fread(&gp[k][i][j],sizeof(double),1,fp)!=1) {
          fprintf(stderr,"R error (gp)\n"); exit(1);
        }
        if(fread(&gm[k][i][j],sizeof(double),1,fp)!=1) {
          fprintf(stderr,"R error (gm)\n"); exit(1);
        }
      }
   } }
  fclose(fp);
}

void trstore(void)
{
  static int first=1,whence;
  int i,j,terminate=-1;
  float **rv,**rp,**P,**p,**m,strain;
  FILE *fp;

  rp=matrix(1,NDIM,1,NPART); rv=matrix(1,NDIM,1,NPART);
  P=matrix(1,NDIM,1,NPART); p=matrix(1,NDIM,1,NPART); m=matrix(1,NDIM,1,NPART);

  for(i=1;i<=NDIM;i++) {              /* Store floats */
    for(j=1;j<=NPART;j++) {
      rv[i][j]=(float)theTrajectory.rv[i][j];
      rp[i][j]=(float)theTrajectory.rp[i][j];
      P[i][j]=(float)theTrajectory.P[i][j];
      p[i][j]=(float)theTrajectory.p[i][j];
      m[i][j]=(float)theTrajectory.m[i][j];
  } }
  strain=(float)theTrajectory.strain;

  if(first) {                          /* Open the file  */
    first=0;
    if((fp=fopen(STORETRAJ,"wb"))==NULL) {
      fprintf(stderr,"Cannot open file (first)\n"); exit(1);
  } }
  else {
    if((fp=fopen(STORETRAJ,"r+b"))==NULL) {
      fprintf(stderr,"Cannot open file (append)\n"); exit(1);
    }
    if(fseek(fp,whence,0)!=0) {
      fprintf(stderr,"Cannot backup (trstore)\n"); exit(1);
    }
  }

  if(fwrite(&theTrajectory.istep,sizeof(int),1,fp)!=1) {
    fprintf(stderr,"Write error (istep)\n"); exit(1);
  }
  if(fwrite(&strain,sizeof(float),1,fp)!=1) {
    fprintf(stderr,"Write error (strain)\n"); exit(1);
  }
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      if(fwrite(&rv[i][j],sizeof(float),1,fp)!=1) {
        fprintf(stderr,"Write error (rv)\n"); exit(1);
      }
      if(fwrite(&rp[i][j],sizeof(float),1,fp)!=1) {
        fprintf(stderr,"Write error (rp)\n"); exit(1);
      }
      if(fwrite(&P[i][j],sizeof(float),1,fp)!=1) {
        fprintf(stderr,"Write error (P)\n"); exit(1);
      }
      if(fwrite(&p[i][j],sizeof(float),1,fp)!=1) {
        fprintf(stderr,"Write error (p)\n"); exit(1);
      }
      if(fwrite(&m[i][j],sizeof(float),1,fp)!=1) {
        fprintf(stderr,"Write error (m)\n"); exit(1);
      }
  } }
  whence=ftell(fp);
  if(fwrite(&terminate,sizeof(int),1,fp)!=1) {
    fprintf(stderr,"Write error (m)\n"); exit(1);
  }
  fclose(fp);

  free_matrix(rv,1,NDIM,1,NPART); free_matrix(rp,1,NDIM,1,NPART);
  free_matrix(P,1,NDIM,1,NPART); free_matrix(p,1,NDIM,1,NPART);
  free_matrix(m,1,NDIM,1,NPART);
}
```

```
/* forces.c */

/* Routine gay_berne() evaluates forces and handles the minimum image.
   Routine virtual_pos() calculates virtual from physical positions.
*/

#include <stdio.h>
#include <math.h>
#include "rb.h"

extern struct CurrentTrajectory theTrajectory;

void virtual_pos(void)
{
  int i,j;
  double sideh1,sideh2,sideh3,bring,**Pos,**Vir_pos;

  sideh1=0.5*LX; sideh2=0.5*LY; sideh3=0.5*LZ;

  Pos=theTrajectory.rp; Vir_pos=theTrajectory.rv;
  for(i=1;i<=NDIM;i++){
    for(j=1;j<=NPART;j++) Vir_pos[i][j]=Pos[i][j];
  }
  for(i=1;i<=NPART;i++){
    if(Pos[3][i]>  sideh3 ) {
      bring=floor(Pos[3][i]/LZ+0.5);
      Vir_pos[3][i]=Pos[3][i]-(bring*LZ);
    }
    if(Pos[3][i]<(-sideh3)) {
      bring=ceil(Pos[3][i]/LZ-0.5);
      Vir_pos[3][i]=Pos[3][i]-(bring*LZ);
    }
    if(Pos[1][i]>  sideh1 ) {    /* We'll recal. if nec. for istep>shear_step */
      bring=floor(Pos[1][i]/LX+0.5);
      Vir_pos[1][i]=Pos[1][i]-(bring*LX);
    }
    if(Pos[1][i]<(-sideh1)) {
      bring=ceil(Pos[1][i]/LX-0.5);
      Vir_pos[1][i]=Pos[1][i]-(bring*LX);
    }
    if(Pos[2][i]>  sideh2 ) {
      bring=floor(Pos[2][i]/LY+0.5);
      Vir_pos[2][i]=Pos[2][i]-(bring*LY);
      if(theTrajectory.istep>=theTrajectory.shear_step) {
        Vir_pos[1][i]=Pos[1][i]-theTrajectory.strain*bring;
        if(Vir_pos[1][i]>  sideh1 ) {
          bring=floor(Vir_pos[1][i]/LX+0.5);
          Vir_pos[1][i]=Vir_pos[1][i]-(bring*LX);
        }
        if(Vir_pos[1][i]<(-sideh1)) {
          bring=ceil(Vir_pos[1][i]/LX-0.5);
          Vir_pos[1][i]=Vir_pos[1][i]-(bring*LX);
    } } }
    if(Pos[2][i]<(-sideh2)) {
      bring=ceil(Pos[2][i]/LY-0.5);
      Vir_pos[2][i]=Pos[2][i]-(bring*LY);
      if(theTrajectory.istep>=theTrajectory.shear_step) {
        Vir_pos[1][i]=Pos[1][i]-theTrajectory.strain*bring;
        if(Vir_pos[1][i]>  sideh1 ) {
          bring=floor(Vir_pos[1][i]/LX+0.5);
          Vir_pos[1][i]=Vir_pos[1][i]-(bring*LX);
        }
        if(Vir_pos[1][i]<(-sideh1)) {
          bring=ceil(Vir_pos[1][i]/LX-0.5);
          Vir_pos[1][i]=Vir_pos[1][i]-(bring*LX);
    } } }
    if(Vir_pos[1][i]>sideh1||Vir_pos[1][i]<-sideh1){
      fprintf(stderr,"*** ERROR %d %.5e\n",theTrajectory.istep,Vir_pos[1][i]);
      exit(1);
    }
    if(Vir_pos[2][i]>sideh2||Vir_pos[2][i]<-sideh2){
      fprintf(stderr,"*** ERROR %d %.5e\n",theTrajectory.istep,Vir_pos[2][i]);
      exit(1);
    }
    if(Vir_pos[3][i]>sideh3||Vir_pos[3][i]<-sideh3){
      fprintf(stderr,"*** ERROR %d %.5e\n",theTrajectory.istep,Vir_pos[3][i]);
      exit(1);
} } }

void gay_berne(void)
{
  int i,j,ncut;
  double rd2,epot,**pforces,**rforces,**Stresspot,**p,pi[3],pj[3],r[3],derv[10];
  double posix,posiy,posiz,xij,yij,zij,kx,ky,kz,sideh1,sideh2,sideh3;
  double a,b,vol,xj,bring,**Pos,**Vir_pos;

  vol=1.0/(LX*LY*LZ);epot=0.0;ncut=0;sideh1=0.5*LX;sideh2=0.5*LY;sideh3=0.5*LZ;
  Stresspot=theTrajectory.stresspot;
  Pos=theTrajectory.rp; Vir_pos=theTrajectory.rv;
  pforces=theTrajectory.pot_forces; rforces=theTrajectory.rot_forces;
  p=theTrajectory.p;

  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) { pforces[i][j]=0.0; rforces[i][j]=0.0; }
  }
  for(i=1;i<=NDIM;i++) { for(j=1;j<=NDIM;j++) Stresspot[i][j]=0.0; }

  virtual_pos();

  for(i=1;i<NPART;i++){                  /* Outer force loop */
    posix=Vir_pos[1][i]; posiy=Vir_pos[2][i]; posiz=Vir_pos[3][i];
    for(j=i+1;j<=NPART;j++) {            /* Inner force loop */
      xij=posix-Vir_pos[1][j]; yij=posiy-Vir_pos[2][j]; zij=posiz-Vir_pos[3][j];
      if(zij<(-sideh3))  zij+=LZ;
      if(zij>  sideh3 )  zij-=LZ;
      if(xij<(-sideh1))  xij+=LX;   /* If istep>shear_step recalculate xij */
      if(xij>  sideh1 )  xij-=LX;
      if(theTrajectory.istep<theTrajectory.shear_step) {  /* Minimum image */
        if(yij<(-sideh2))  yij+=LY;
        if(yij>  sideh2 )  yij-=LY;
      }
      else {                             /* Lees-Edwards BC */
        if(yij>  sideh2 ) {              /* ABOVE */
          yij-=LY; xj=Vir_pos[1][j]; xj+=theTrajectory.strain;
          bring=floor(xj/LX+0.5); xj-=(bring*LX); xij=posix-xj;
          if(xij>  sideh1 )  xij-=LX;  if(xij<(-sideh1))  xij+=LX;
        }
        if(yij<(-sideh2)) {              /* BELOW */
          yij+=LY; xj=Vir_pos[1][j]; xj-=theTrajectory.strain;
          bring=ceil(xj/LX-0.5); xj-=(bring*LX); xij=posix-xj;
          if(xij>  sideh1 )  xij-=LX;  if(xij<(-sideh1))  xij+=LX;
      } }
      rd2=xij*xij+yij*yij+zij*zij;
      if(rd2<RCUTOFF) {
        ncut+=1;
        pi[0]=p[1][i]; pi[1]=p[2][i]; pi[2]=p[3][i];
        pj[0]=p[1][j]; pj[1]=p[2][j]; pj[2]=p[3][j];
        r[0]=xij; r[1]=yij; r[2]=zij;

        ders(pi, pj, r, derv); epot+=derv[0];
        pforces[1][i]-=derv[1]; pforces[1][j]+=derv[1]; /* sign !!! */
        pforces[2][i]-=derv[2]; pforces[2][j]+=derv[2]; /* includes minus */
        pforces[3][i]-=derv[3]; pforces[3][j]+=derv[3];
        rforces[1][i]+=pi[2]*derv[5]-pi[1]*derv[6];
        rforces[2][i]+=pi[0]*derv[6]-pi[2]*derv[4];
        rforces[3][i]+=pi[1]*derv[4]-pi[0]*derv[5];
        rforces[1][j]+=pj[2]*derv[8]-pj[1]*derv[9];
        rforces[2][j]+=pj[0]*derv[9]-pj[2]*derv[7];
        rforces[3][j]+=pj[1]*derv[7]-pj[0]*derv[8];

        Stresspot[1][1]-=derv[1]*xij; Stresspot[2][1]-=derv[1]*yij;
        Stresspot[3][1]-=derv[1]*zij; Stresspot[2][2]-=derv[2]*yij;
        Stresspot[3][2]-=derv[2]*zij; Stresspot[3][3]-=derv[3]*zij;
        Stresspot[1][2]-=derv[2]*xij; Stresspot[1][3]-=derv[3]*xij;
        Stresspot[2][3]-=derv[3]*yij;
    } }
  }                                        /* Inner loop and outer loop end. */
  for(i=1;i<=NDIM;i++) { for(j=1;j<=NDIM;j++) Stresspot[i][j]*=vol; }
  theTrajectory.epot=epot; theTrajectory.ncut=ncut;
}
```

```
/* extra.c */

/* This subroutine calculates mesoscopic and macroscopic
   instantaneous variables.
*/

#include <stdio.h>
#include <math.h>
#include "rb.h"

extern struct CurrentTrajectory theTrajectory;

void extra(void)
{
  int i,j;
  double krx,kry,krz,PiPi,temp,vel;
  double **p,**m,**P,**Stresstkin,**Stresstot,vol=LX*LY*LZ;
  double **conf,a1,a2,a3,q,r,theta,x1,x2,x3,lambda;

  /* Get kinetic contribution to the stress tensor */
  P=theTrajectory.P; Stresstkin=theTrajectory.stresstkin;
  Stresstot=theTrajectory.stresstot; m=theTrajectory.m;
  p=theTrajectory.p;
  theTrajectory.mz=0.0; theTrajectory.momx=0.0;
  theTrajectory.momy=0.0; theTrajectory.momz=0.0;
  for(i=1;i<=NPART;i++) {
    theTrajectory.mz+=p[1][i]*m[1][i]+p[2][i]*m[2][i]+p[3][i]*m[3][i];
    theTrajectory.momx+=P[1][i];
    theTrajectory.momy+=P[2][i];
    theTrajectory.momz+=P[3][i];
  }
  for(i=1;i<=NDIM;i++) { for(j=1;j<=NDIM;j++) Stresstkin[i][j]=0.0; }
  for(i=1;i<=NPART;i++) {
    Stresstkin[1][1]+=P[1][i]*P[1][i]; Stresstkin[1][2]+=P[1][i]*P[2][i];
    Stresstkin[1][3]+=P[1][i]*P[3][i]; Stresstkin[2][2]+=P[2][i]*P[2][i];
    Stresstkin[2][3]+=P[2][i]*P[3][i]; Stresstkin[3][3]+=P[3][i]*P[3][i];
    Stresstkin[2][1]+=P[2][i]*P[1][i]; Stresstkin[3][1]+=P[3][i]*P[1][i];
    Stresstkin[3][2]+=P[3][i]*P[2][i];
  }
  theTrajectory.ktx=Stresstkin[1][1]/4.0; theTrajectory.kty=Stresstkin[2][2]/4.0
  theTrajectory.ktz=Stresstkin[3][3]/4.0;
  for(i=1;i<=3;i++) {
    for(j=1;j<=3;j++) {
      Stresstkin[i][j]/=(2.0*vol);  /* mass is 2 */
      Stresstot[i][j]=Stresstkin[i][j]+theTrajectory.stresspot[i][j];
  } }

  temp=0.0; vel=0.0; krx=0.0; kry=0.0; krz=0.0;
  for(j=1;j<=NPART;j++) {
    PiPi=P[1][j]*P[1][j]+P[2][j]*P[2][j]+P[3][j]*P[3][j];
    krx+=II1*m[1][j]*m[1][j]; kry+=II2*m[2][j]*m[2][j];
    krz+=II3*m[3][j]*m[3][j]; temp+=PiPi; vel+=sqrt(PiPi);
  }
  theTrajectory.krx=0.5*krx;
  theTrajectory.kry=0.5*kry; theTrajectory.krz=0.5*krz;

  theTrajectory.ttemp=temp/(6.0*NPART);     /* from kinetic energy; mass is 2 */
  theTrajectory.rtemp=(krx+kry+krz)*0.5/NPART;
  theTrajectory.press=(Stresstot[1][1]+Stresstot[2][2]+Stresstot[3][3])/3.0;
  theTrajectory.tekin=temp*0.25;            /* mass is 2 */
  temp=0.0;
  for(j=1;j<=NPART;j++) {
    temp+=II1*m[1][j]*m[1][j]+II2*m[2][j]*m[2][j]+II3*m[3][j]*m[3][j];
  }
  theTrajectory.rekin=0.5*temp;
  theTrajectory.etot=theTrajectory.tekin+theTrajectory.rekin+theTrajectory.epot;
  theTrajectory.tvel=vel/(2.0*NPART);       /* from average speed; mass is 2  */

  /* Conformation tensor */
  conf=theTrajectory.conf;
  conf[1][1]=0.0; conf[1][2]=0.0; conf[1][3]=0.0; conf[2][2]=0.0;
  conf[2][3]=0.0; conf[3][3]=0.0;
  for(j=1;j<=NPART;j++) {
    conf[1][1]+=p[1][j]*p[1][j]; conf[1][2]+=p[1][j]*p[2][j];
    conf[1][3]+=p[1][j]*p[3][j]; conf[2][2]+=p[2][j]*p[2][j];
    conf[2][3]+=p[2][j]*p[3][j]; conf[3][3]+=p[3][j]*p[3][j];
  }
  conf[1][1]/=NPART; conf[1][2]/=NPART; conf[1][3]/=NPART; conf[2][2]/=NPART;
  conf[2][3]/=NPART; conf[3][3]/=NPART;
  if(theTrajectory.istep>50) {  /* skip first 50 steps - see the end of fcc.c */
    a1=-conf[1][1]-conf[2][2]-conf[3][3];
    a2=conf[1][1]*conf[2][2]+conf[1][1]*conf[3][3]+conf[2][2]*conf[3][3]-
       conf[1][2]*conf[1][2]-conf[1][3]*conf[1][3]-conf[2][3]*conf[2][3];
    a3=conf[1][2]*conf[1][2]*conf[3][3]+conf[1][3]*conf[1][3]*conf[2][2]+
       conf[2][3]*conf[2][3]*conf[1][1]-conf[1][1]*conf[2][2]*conf[3][3]-
       2.0*conf[1][2]*conf[1][3]*conf[2][3];
    q=(a1*a1-3*a2)/9.0;r=(2.0*a1*a1*a1-9.0*a1*a2+27.0*a3)/54.0;lambda=q*q*q-r*r;
    if(lambda<0) { fprintf(stderr,"extra: eigenvalue problem\n"); exit(1);}
    theta=acos(r/sqrt(q*q*q)); x1=-2.0*sqrt(q)*cos(theta/3.0)-a1/3.0;
    x2=-2.0*sqrt(q)*cos((theta+2.0*M_PI)/3.0)-a1/3.0;
    x3=-2.0*sqrt(q)*cos((theta+4.0*M_PI)/3.0)-a1/3.0;
    lambda=abs(x1);
    if(x2>lambda) lambda=x2; if(x3>lambda) lambda=x3;
    x1=conf[2][3]*conf[1][2]-conf[1][3]*(conf[2][2]-lambda);
    x3=x1/(conf[1][3]*conf[2][3]-conf[1][2]*(conf[3][3]-lambda));
    x1/=(conf[1][3]*conf[1][2]-conf[2][3]*(conf[1][1]-lambda));
    r=sqrt(1.0+x1*x1+x3*x3); theTrajectory.dorx=x1/r;
    theTrajectory.dory=1/r; theTrajectory.dorz=x3/r;
  }
}
```

```
/* gear5.c */

/* This routine performs the Gear five value integration,
   for the first order ODE. The algorithm uses the Nordsieck representation.
   Subroutine rh_sides() calculates right hand sides of the
   equations of motion.
*/

#include <stdio.h>
#include <math.h>
#include "rb.h"
extern struct CurrentTrajectory theTrajectory;


#define GEAR0  0.3486111111111111111
#define GEAR1  1.0
#define GEAR2  0.9166666666666666667
#define GEAR3  0.3333333333333333333
#define GEAR4  0.0416666666666666667

void gear5(double tstep)
{
  int i,j,k;
  double c1,c2,c3,c4,crv1,crv3,crv4,crv5;
  double **Pos1,**Pos2,**Pos3,**Pos4,**Pos5,**Vel1,**Vel2,**Vel3,**Vel4,**Vel5;
  double **gp1,**gp2,**gp3,**gp4,**gp5,**gm1,**gm2,**gm3,**gm4,**gm5;
  double **PosR, **VelR, **Poscorr, **Velcorr;

  if(theTrajectory.istep>=theTrajectory.shear_step) {
    /* Gentle ramp for preshearing
    theTrajectory.strain_rate=4.5+2.5*(theTrajectory.istep
                             -theTrajectory.fstep)/15000.0; */
    theTrajectory.strain_rate=STRAIN_RATE;
    theTrajectory.strain+=tstep*LY*theTrajectory.strain_rate;
  }
  else { theTrajectory.strain_rate=0.0; }

  Pos1=theTrajectory.rp; Pos2=theTrajectory.grp[0]; Pos3=theTrajectory.grp[1];
  Pos4=theTrajectory.grp[2]; Pos5=theTrajectory.grp[3];
  Vel1=theTrajectory.P; Vel2=theTrajectory.gP[0]; Vel3=theTrajectory.gP[1];
  Vel4=theTrajectory.gP[2]; Vel5=theTrajectory.gP[3];
  gp1=theTrajectory.p; gp2=theTrajectory.gp[0]; gp3=theTrajectory.gp[1];
  gp4=theTrajectory.gp[2]; gp5=theTrajectory.gp[3];
  gm1=theTrajectory.m; gm2=theTrajectory.gm[0]; gm3=theTrajectory.gm[1];
  gm4=theTrajectory.gm[2]; gm5=theTrajectory.gm[3];

  /* Predictor */
  c1=tstep; c2=c1*tstep/2.0; c3=c2*tstep/3.0; c4=c3*tstep/4.0;
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      Pos1[i][j]+=c1*Pos2[i][j]+c2*Pos3[i][j]+c3*Pos4[i][j]+c4*Pos5[i][j];
      Pos2[i][j]+=c1*Pos3[i][j]+c2*Pos4[i][j]+c3*Pos5[i][j];
      Pos3[i][j]+=c1*Pos4[i][j]+c2*Pos5[i][j]; Pos4[i][j]+=c1*Pos5[i][j];
      Vel1[i][j]+=c1*Vel2[i][j]+c2*Vel3[i][j]+c3*Vel4[i][j]+c4*Vel5[i][j];
      Vel2[i][j]+=c1*Vel3[i][j]+c2*Vel4[i][j]+c3*Vel5[i][j];
      Vel3[i][j]+=c1*Vel4[i][j]+c2*Vel5[i][j]; Vel4[i][j]+=c1*Vel5[i][j];

      gp1[i][j]+=c1*gp2[i][j]+c2*gp3[i][j]+c3*gp4[i][j]+c4*gp5[i][j];
      gp2[i][j]+=c1*gp3[i][j]+c2*gp4[i][j]+c3*gp5[i][j];
      gp3[i][j]+=c1*gp4[i][j]+c2*gp5[i][j]; gp4[i][j]+=c1*gp5[i][j];
      gm1[i][j]+=c1*gm2[i][j]+c2*gm3[i][j]+c3*gm4[i][j]+c4*gm5[i][j];
      gm2[i][j]+=c1*gm3[i][j]+c2*gm4[i][j]+c3*gm5[i][j];
      gm3[i][j]+=c1*gm4[i][j]+c2*gm5[i][j]; gm4[i][j]+=c1*gm5[i][j];
  } }

  /* Evaluate right hand sides (forces). */
  rh_sides();

  /* Corrector */
  crv1=GEAR0*c1; crv3=GEAR2*c1/c2; crv4=GEAR3*c1/c3; crv5=GEAR4*c1/c4;
  Poscorr=dmatrix(1,NDIM,1,NPART); Velcorr=dmatrix(1,NDIM,1,NPART);

  PosR=theTrajectory.Rrp;
  VelR=theTrajectory.RP;
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      Poscorr[i][j]=PosR[i][j]-Pos2[i][j]; Velcorr[i][j]=VelR[i][j]-Vel2[i][j];
      Pos1[i][j]+=crv1*Poscorr[i][j]; Pos2[i][j]=PosR[i][j];
      Pos3[i][j]+=crv3*Poscorr[i][j]; Pos4[i][j]+=crv4*Poscorr[i][j];
      Pos5[i][j]+=crv5*Poscorr[i][j]; Vel1[i][j]+=crv1*Velcorr[i][j];
      Vel2[i][j]=VelR[i][j]; Vel3[i][j]+=crv3*Velcorr[i][j];
      Vel4[i][j]+=crv4*Velcorr[i][j]; Vel5[i][j]+=crv5*Velcorr[i][j];
  } }
  PosR=theTrajectory.Rp;
  VelR=theTrajectory.Rm;
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      Poscorr[i][j]=PosR[i][j]-gp2[i][j]; Velcorr[i][j]=VelR[i][j]-gm2[i][j];
      gp1[i][j]+=crv1*Poscorr[i][j]; gp2[i][j]=PosR[i][j];
      gp3[i][j]+=crv3*Poscorr[i][j]; gp4[i][j]+=crv4*Poscorr[i][j];
      gp5[i][j]+=crv5*Poscorr[i][j]; gm1[i][j]+=crv1*Velcorr[i][j];
      gm2[i][j]=VelR[i][j]; gm3[i][j]+=crv3*Velcorr[i][j];
      gm4[i][j]+=crv4*Velcorr[i][j]; gm5[i][j]+=crv5*Velcorr[i][j];
  } }

  /* Release memory. */
  free_dmatrix(Poscorr,1,NDIM,1,NPART);
  free_dmatrix(Velcorr,1,NDIM,1,NPART);
}

void rh_sides(void) {
  int i,j;
  double **rp,**P,**p,**m,**Rrp,**RP,**Rp,**Rm;
  double num,numst,den,alpha,lambda,t1,t2,t3;

  rp=theTrajectory.rp; P=theTrajectory.P;
  p=theTrajectory.p; m=theTrajectory.m;
  Rrp=theTrajectory.Rrp; RP=theTrajectory.RP;
  Rp=theTrajectory.Rp; Rm=theTrajectory.Rm;

  /* Get Gay-Berne forces */
  gay_berne();

  /* Compute the right hand sides */
  if(theTrajectory.istep<theTrajectory.fstep+NSEQUIL+NSMIC) {
    /* Translational equations */
    for(i=1;i<=NDIM;i++){
      for(j=1;j<=NPART;j++){
        Rrp[i][j]=P[i][j]*0.5; /* 0.5 for mass */
      RP[i][j]=theTrajectory.pot_forces[i][j];
  } }
  /* Rotational equations */
  for(j=1;j<=NPART;j++){
    t1=m[1][j]*II1; t2=m[2][j]*II2; t3=m[3][j]*II3;
    Rp[1][j]=t2*p[3][j]-t3*p[2][j];
    Rp[2][j]=t3*p[1][j]-t1*p[3][j];
    Rp[3][j]=t1*p[2][j]-t2*p[1][j];
    lambda=1.0/(p[1][j]*p[1][j]+p[2][j]*p[2][j]+p[3][j]*p[3][j]);
    lambda*=(p[1][j]*Rp[1][j]+p[2][j]*Rp[2][j]+p[3][j]*Rp[3][j]);
    Rp[1][j]-=lambda*p[1][j];
    Rp[2][j]-=lambda*p[2][j];
    Rp[3][j]-=lambda*p[3][j];
    Rm[1][j]=t2*m[3][j]-t3*m[2][j]+theTrajectory.rot_forces[1][j];
    Rm[2][j]=t3*m[1][j]-t1*m[3][j]+theTrajectory.rot_forces[2][j];
    Rm[3][j]=t1*m[2][j]-t2*m[1][j]+theTrajectory.rot_forces[3][j];
} }

if(theTrajectory.istep>=theTrajectory.fstep+NSEQUIL+NSMIC) {
  /* get alpha */
  num=0.0;den=0.0;numst=0.0;
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      den+=P[i][j]*P[i][j];
      num+=P[i][j]*theTrajectory.pot_forces[i][j];
  } }
  for(j=1;j<=NPART;j++) numst+=P[1][j]*P[2][j];
  alpha=(num-theTrajectory.strain_rate*numst)/den;

  /* Translational equations */
  for(j=1;j<=NPART;j++){
    Rrp[1][j]=P[1][j]*0.5+theTrajectory.strain_rate*rp[2][j];
    Rrp[2][j]=P[2][j]*0.5; Rrp[3][j]=P[3][j]*0.5;
    RP[1][j]=theTrajectory.pot_forces[1][j]-theTrajectory.strain_rate
               *P[2][j]-alpha*P[1][j];
    RP[2][j]=theTrajectory.pot_forces[2][j]-alpha*P[2][j];
    RP[3][j]=theTrajectory.pot_forces[3][j]-alpha*P[3][j];
  }
  /* Rotational equations */
  den=0.0;num=0.0;
  for(j=1;j<=NPART;j++){
    t1=m[1][j]*II1; t2=m[2][j]*II2; t3=m[3][j]*II3;
    Rp[1][j]=t2*p[3][j]-t3*p[2][j]+theTrajectory.strain_rate*p[2][j];
    Rp[2][j]=t3*p[1][j]-t1*p[3][j]-theTrajectory.strain_rate*p[1][j];
    Rp[3][j]=t1*p[2][j]-t2*p[1][j];
    lambda=1.0/(p[1][j]*p[1][j]+p[2][j]*p[2][j]+p[3][j]*p[3][j]);
    lambda*=(p[1][j]*Rp[1][j]+p[2][j]*Rp[2][j]+p[3][j]*Rp[3][j]);
    Rp[1][j]-=lambda*p[1][j];
    Rp[2][j]-=lambda*p[2][j];
    Rp[3][j]-=lambda*p[3][j];
    Rm[1][j]=t2*m[3][j]-t3*m[2][j]+theTrajectory.rot_forces[1][j]
            +theTrajectory.strain_rate*m[2][j];
    Rm[2][j]=t3*m[1][j]-t1*m[3][j]+theTrajectory.rot_forces[2][j]
            -theTrajectory.strain_rate*m[1][j];
    Rm[3][j]=t1*m[2][j]-t2*m[1][j]+theTrajectory.rot_forces[3][j];
    den+=t1*m[1][j]+t2*m[2][j]+t3*m[3][j];
    num+=t1*Rm[1][j]+t2*Rm[2][j]+t3*Rm[3][j];
  }
  lambda=num/den;
  for(j=1;j<=NPART;j++){
      Rm[1][j]-=lambda*m[1][j];
      Rm[2][j]-=lambda*m[2][j];
      Rm[3][j]-=lambda*m[3][j];
} } }
```

```
/* rk4.c */

/* This routine performs the Runge-Kutta integration. Not
   used in production runs - only to compare the energy
   and temperature flows in the gear5 routine.
*/

#include <stdio.h>
#include <math.h>
#include "rb.h"
extern struct CurrentTrajectory theTrajectory;

void rk4(double dt) {
  int i,j,k,km1;
  double k1,**krp[4],**kP[4],**kp[4],**km[4],**irp,**iP,**ip,**im;

  if(theTrajectory.istep>=theTrajectory.shear_step) {
    theTrajectory.strain_rate=STRAIN_RATE;
  }
  else { theTrajectory.strain_rate=0.0; }

  for(i=0;i<=3;i++) {
    krp[i]=dmatrix(1,NDIM,1,NPART); kP[i]=dmatrix(1,NDIM,1,NPART);
    kp[i]=dmatrix(1,NDIM,1,NPART); km[i]=dmatrix(1,NDIM,1,NPART);
  }
  irp=dmatrix(1,NDIM,1,NPART); iP=dmatrix(1,NDIM,1,NPART);
  ip=dmatrix(1,NDIM,1,NPART); im=dmatrix(1,NDIM,1,NPART);

  /* Save initial positions and velocities. */
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      irp[i][j]=theTrajectory.rp[i][j]; iP[i][j]=theTrajectory.P[i][j];
      ip[i][j]=theTrajectory.p[i][j]; im[i][j]=theTrajectory.m[i][j];
  } }
  rh_sides();
  /* Save k[1]. */
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      krp[0][i][j]=dt*theTrajectory.Rrp[i][j];
      kP[0][i][j]=dt*theTrajectory.RP[i][j];
      kp[0][i][j]=dt*theTrajectory.Rp[i][j];
      km[0][i][j]=dt*theTrajectory.Rm[i][j];
  } }

  /* Get other ks. */
  for(k=1;k<=3;k++) {
    /* Set pos and vel first. */
    km1=k-1; k1=((int)(k+3)/3)/2.0;
    for(i=1;i<=NDIM;i++) {
      for(j=1;j<=NPART;j++) {
        theTrajectory.rp[i][j]=irp[i][j]+krp[km1][i][j]*k1;
        theTrajectory.P[i][j]=iP[i][j]+kP[km1][i][j]*k1;
        theTrajectory.p[i][j]=ip[i][j]+kp[km1][i][j]*k1;
        theTrajectory.m[i][j]=im[i][j]+km[km1][i][j]*k1;
    } }
    /* Call forces - we need to adjust the strain. */
    theTrajectory.strain+=0.5*(k%2)*dt*LY*theTrajectory.strain_rate;
    rh_sides();
    /* Save kpos's and kvol's. */
    for(i=1;i<=NDIM;i++) {
      for(j=1;j<=NPART;j++) {
        krp[k][i][j]=dt*theTrajectory.Rrp[i][j];
        kP[k][i][j]=dt*theTrajectory.RP[i][j];
        kp[k][i][j]=dt*theTrajectory.Rp[i][j];
        km[k][i][j]=dt*theTrajectory.Rm[i][j];
  } } }
  /* Get positions and velocities. */
  for(i=1;i<=NDIM;i++) {
    for(j=1;j<=NPART;j++) {
      theTrajectory.rp[i][j]=irp[i][j]+(krp[0][i][j]+
      2.0*krp[1][i][j]+2.0*krp[2][i][j]+krp[3][i][j])/6.0;
      theTrajectory.P[i][j]=iP[i][j]+(kP[0][i][j]+
      2.0*kP[1][i][j]+2.0*kP[2][i][j]+kP[3][i][j])/6.0;
      theTrajectory.p[i][j]=ip[i][j]+(kp[0][i][j]+
      2.0*kp[1][i][j]+2.0*kp[2][i][j]+kp[3][i][j])/6.0;
      theTrajectory.m[i][j]=im[i][j]+(km[0][i][j]+
      2.0*km[1][i][j]+2.0*km[2][i][j]+km[3][i][j])/6.0;
  } }

  for(i=0;i<=3;i++) {
    free_dmatrix(krp[i],1,NDIM,1,NPART); free_dmatrix(kP[i],1,NDIM,1,NPART);
    free_dmatrix(kp[i],1,NDIM,1,NPART); free_dmatrix(km[i],1,NDIM,1,NPART);
  }
  free_dmatrix(irp,1,NDIM,1,NPART); free_dmatrix(iP,1,NDIM,1,NPART);
  free_dmatrix(ip,1,NDIM,1,NPART); free_dmatrix(im,1,NDIM,1,NPART);
}
```

```
/* leap_frog.c */

/* This routine performs the leap-frog integration. The routine is not
   complete since it does not incorporate rotational equations of motion.
   Used to test energy and temperature flows, not used in the
   production runs. The mass of a single particle is 2.
*/

#include <stdio.h>
#include <math.h>
#include "rb.h"

extern struct CurrentTrajectory theTrajectory;

void leap(double tstep)
{
  int i,j,nwhile;
  static double alpha=0.0;
  double **Pos,**Vel,**Forces, **half_vel,oalpha,alpha_coeff,
         htstep,epsilon,num,den,numst,hgts,hgts2;

  Pos=theTrajectory.rp; Vel=theTrajectory.P;

  if(theTrajectory.istep>=theTrajectory.shear_step) {
    theTrajectory.strain_rate=STRAIN_RATE;
    theTrajectory.strain+=tstep*LY*theTrajectory.strain_rate;
  }
  else { theTrajectory.strain_rate=0.0; }

  gay_berne();

  Forces=theTrajectory.pot_forces; htstep=0.5*tstep;
  if(theTrajectory.istep<theTrajectory.fstep+NSEQUIL+NSMIC) {
  /* to check the velocity form
    for(i=1;i<=NDIM;i++) {
      for(j=1;j<=NPART;j++) {
        Pos[i][j]+=0.5*(Vel[i][j]*tstep+Forces[i][j]*htstep*tstep);
        Vel[i][j]+=Forces[i][j]*htstep;
      }
    }
    gay_berne();
    for(i=1;i<=NDIM;i++) {
      for(j=1;j<=NPART;j++) Vel[i][j]+=Forces[i][j]*htstep;
    } */
    for(i=1;i<=NDIM;i++) {
      for(j=1;j<=NPART;j++) {
        Vel[i][j]+=Forces[i][j]*tstep;   /* Vel denotes mass-center momenta */
        Pos[i][j]+=Vel[i][j]*htstep;     /* 0.5 for mass                    */
    } }
  }

  if(theTrajectory.istep>=theTrajectory.fstep+NSEQUIL+NSMIC) {
    half_vel=dmatrix(1,NDIM,1,NPART);
    epsilon=100.0; nwhile=0;
    while(epsilon>1.0e-5) {
      alpha_coeff=1.0/(1.0+alpha*htstep);
      for(i=1;i<=NPART;i++) {
        half_vel[2][i]=alpha_coeff*(Vel[2][i]+htstep*Forces[2][i]);
        half_vel[3][i]=alpha_coeff*(Vel[3][i]+htstep*Forces[3][i]);
        half_vel[1][i]=alpha_coeff*(Vel[1][i]+htstep*(Forces[1][i]-
                                    theTrajectory.strain_rate*half_vel[2     );
      }
      num=0.0;den=0.0;numst=0.0;
      for(i=1;i<=NDIM;i++) {
        for(j=1;j<=NPART;j++) {
          den+=half_vel[i][j]*half_vel[i][j]; num+=half_vel[i][j]*Forces[i][j];
      } }
      for(i=1;i<=NPART;i++) numst+=half_vel[1][i]*half_vel[2][i];
      oalpha=alpha; alpha=(num-theTrajectory.strain_rate*numst)/den;
      epsilon=fabs(1-oalpha/alpha);
      if(nwhile>100) {
         fprintf(stderr,"*** Number of iterations for alpha > 100\n");
         fprintf(stderr,"*** See subroutine leap\n");
         fprintf(stderr,"n=%d o=%.5e al=%.5e ep=%.9e\n",
                     theTrajectory.istep,oalpha,alpha,epsilon);
         break;
      }
      nwhile+=1;
    }
    for(i=1;i<=NPART;i++) {
        Vel[2][i]+=(Forces[2][i]-alpha*half_vel[2][i])*tstep;
        Vel[3][i]+=(Forces[3][i]-alpha*half_vel[3][i])*tstep;
        Vel[1][i]+=(Forces[1][i]-alpha*half_vel[1][i]-
                                theTrajectory.strain_rate*half_vel[2][i])*tstep;
    }
    hgts=theTrajectory.strain_rate*htstep;
    hgts2=theTrajectory.strain_rate*tstep;
    for(i=1;i<=NPART;i++) {
      /*Pos[1][i]+=Vel[1][i]*htstep+hgts*Pos[2][i]; Pos[2][i]+=Vel[2][i]*htstep;
      Pos[3][i]+=Vel[3][i]*htstep; Pos[1][i]+=hgts*Pos[2][i];*/
      Pos[1][i]+=Vel[1][i]*htstep+hgts2*Pos[2][i]; Pos[2][i]+=Vel[2][i]*htstep;
      Pos[3][i]+=Vel[3][i]*htstep;
    }
    free_dmatrix(half_vel,1,NDIM,1,NPART);
  }
}
```

```
/* average.c */

/* This routine stores average stresses and kinetic energies and other
   useful quantities. The bin averages are taken over each bin and then the
   global average and the st. dev. are found.
*/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include "rb.h"
extern struct CurrentTrajectory theTrajectory;

void av_stress(void)
{
  static int first=1,scounter=-1,*nstepa,counter;
  int i,j,k,kp9,kp18,kp27;
  double tp[9],binav[46],binstdev[46];
  static double **data,avdata[46];
  FILE *fp;

  if(first) {
    counter=NSEQUIL+NSMIC+NSKIN+NSSHEAR;
    if(counter<BINSIZE) {
      fprintf(stderr,"av_stress: BINSIZE must be greater then NSEQUIL+NSMIC+NSKI
      exit(1);
    }
    first=0; counter=counter/BINSIZE-1;
    data=dmatrix(0,counter,0,45); nstepa=ivector(0,counter);
  }
  if((theTrajectory.istep-theTrajectory.fstep)%BINSIZE==0) {
    for(i=0;i<=45;i++) avdata[i]=0.0; scounter+=1;
  }
  k=0;
  for(i=1;i<=3;i++) {
    for(j=1;j<=3;j++) {
      tp[k]=theTrajectory.stresspot[i][j]+theTrajectory.stresstkin[i][j];
      k+=1;
  } }
  k=0;
  for(i=1;i<=3;i++) {
      for(j=1;j<=3;j++) {
        kp9=k+9; kp18=k+18; kp27=k+27;
        avdata[k]+=theTrajectory.stresspot[i][j];
        avdata[kp9]+=kp9;
        avdata[kp18]+=theTrajectory.stresstkin[i][j];
        avdata[kp27]+=tp[k];
        k+=1;
  } }
  avdata[36]+=0.33333333333*(tp[0]+tp[4]+tp[8]);
  avdata[37]+=tp[4]-tp[0]; avdata[38]+=tp[8]-tp[4];
  k=0;
  for(i=1;i<=3;i++) {
      for(j=i;j<=3;j++) { avdata[k+39]+=theTrajectory.conf[i][j]; k++; }
  }
  avdata[45]+=theTrajectory.conf[1][1]+theTrajectory.conf[2][2]
             +theTrajectory.conf[3][3];

  if((theTrajectory.istep-theTrajectory.fstep+1)%BINSIZE==0) {
```

```
nstepa[scounter]=theTrajectory.istep;
for(i=0;i<=45;i++) data[scounter][i]=avdata[i]/(double)BINSIZE;
/* get bin averages */
for(i=0;i<=45;i++) binav[i]=0.0;
for(i=0;i<=scounter;i++) { for(j=0;j<=45;j++) binav[j]+=data[i][j]; }
for(i=0;i<=45;i++) binav[i]/=(scounter+1);
for(i=0;i<=45;i++) binstdev[i]=0.0;
for(i=0;i<=scounter;i++) {
  for(j=0;j<=45;j++) {
    binstdev[j]+=(data[i][j]-binav[j])*(data[i][j]-binav[j]);
} }
if(scounter>0) { for(i=0;i<=45;i++) binstdev[i]/=scounter; }
for(i=0;i<=45;i++) binstdev[i]=sqrt(binstdev[i]);
if((fp=fopen(STORESTRESS,"w"))==NULL)
  { fprintf(stderr,"Cannot open file (first)\n"); exit(1); }
fprintf(fp,"  nstep    Ppot11     Ppot12     Ppot13     Ppot21     Ppot22
for(i=0;i<=scounter; i++)
  fprintf(fp,"%7d %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %1
            nstepa[i],data[i][0],data[i][1],data[i][2],data[i][3],
            data[i][4],data[i][5],data[i][6],data[i][7],data[i][8]);
/* fprintf(fp,"\n  nstep    Ppoti11    Ppoti12    Ppoti13    Ppoti21    Ppot
for(i=0;i<=scounter; i++)
  fprintf(fp,"%7d %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %1
           nstepa[i],data[i][9],data[i][10],data[i][11],data[i][12],
           data[i][13],data[i][14],data[i][15],data[i][16],data[i][17]); */
fprintf(fp,"\n  nstep     Ptkin11    Ptkin12    Ptkin13    Ptkin21    Ptkin2
for(i=0;i<=scounter; i++)
  fprintf(fp,"%7d %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %1
            nstepa[i],data[i][18],data[i][19],data[i][20],data[i][21],
            data[i][22],data[i][23],data[i][24],data[i][25],data[i][26]);
fprintf(fp,"\n  nstep     Ptot11     Ptot12     Ptot13     Ptot21     Ptot22
for(i=0;i<=scounter; i++)
  fprintf(fp,"%7d %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %1
            nstepa[i],data[i][27],data[i][28],data[i][29],data[i][30],
            data[i][31],data[i][32],data[i][33],data[i][34],data[i][35]);
fprintf(fp,"\n  nstep     cote11     cote12     cote13     cote22     cote23
for(i=0;i<=scounter; i++)
  fprintf(fp,"%7d %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e\n",
            nstepa[i],data[i][39],data[i][40],data[i][41],data[i][42],
            data[i][43],data[i][44],data[i][45]);
fprintf(fp,"\n  nstep      Press         N1         N2\n");
for(i=0;i<=scounter; i++)
  fprintf(fp,"%7d %10.3e %10.3e %10.3e\n",
              nstepa[i],data[i][36],data[i][37],data[i][38]);
fprintf(fp,"\n");
for(i=0;i<=8;i++)  {
  j=i+18; k=i+27;
  fprintf(fp,"%3d %12.5e (+/-)%9.2e %3d %12.5e (+/-)%9.2e %3d %12.5e (+/-)%9
              binav[i],binstdev[i],j,binav[j],binstdev[j],k,binav[k],binstde
}
fprintf(fp,"\n");
/* for(i=9;i<=17;i++)
  fprintf(fp,"%3d %12.5e (+/-)%12.5e\n",i,binav[i],binstdev[i]);
fprintf(fp,"\n");
for(i=18;i<=26;i++)
  fprintf(fp,"%3d %12.5e (+/-)%12.5e\n",i,binav[i],binstdev[i]);
fprintf(fp,"\n");
for(i=27;i<=35;i++)
  fprintf(fp,"%3d %12.5e (+/-)%12.5e\n",i,binav[i],binstdev[i]);
fprintf(fp,"\n"); */
```

```
    for(i=36;i<=38;i++)
      fprintf(fp,"%3d %12.5e (+/-)%12.5e\n",i,binav[i],binstdev[i]);
    fprintf(fp,"\n");
    for(i=39;i<=45;i++)
      fprintf(fp,"%3d %12.5e (+/-)%12.5e\n",i,binav[i],binstdev[i]);
    fprintf(fp,"\x0C\n");
    fclose(fp);
    if(theTrajectory.istep==theTrajectory.lstep) {
      free_ivector(nstepa,0,counter); free_dmatrix(data,0,counter,0,45);
    }
  }
}

void av_results(void) {
  static char c[6][80];
  static int first=1,scounter=-1,*nstepa,counter,intstore[33];
  int j,i;
  static float floatstore[15];
  float store[17];
  static double **data,avdata[17];
  double binav[17],binstdev[17];
  FILE *fp;

  store[0]=(float)theTrajectory.strain;
  store[1]=(float)theTrajectory.rtemp;
  store[2]=(float)theTrajectory.rekin;
  store[3]=(float)theTrajectory.ttemp;
  store[4]=(float)theTrajectory.tvel;
  store[5]=(float)theTrajectory.tekin;
  store[6]=(float)theTrajectory.epot;
  store[7]=(float)theTrajectory.etot;
  store[8]=(float)theTrajectory.mz;
  store[9]=(float)theTrajectory.dor;
  store[10]=(float)theTrajectory.vor;
  store[11]=(float)theTrajectory.tvelor;
  store[12]=(float)theTrajectory.rvelor;
  store[13]=(float)theTrajectory.ncut;

  if(first) {
    first=0;
    intstore[0]=NDIM; intstore[1]=NPART; intstore[2]=NLAYERS;
    intstore[3]=NCALPAR; intstore[4]=RESTART; intstore[5]=NSEQUIL;
    intstore[6]=NRESVEL; intstore[7]=NSMIC; intstore[8]=NSKIN;
    intstore[9]=NSSHEAR; intstore[10]=NSTORE; intstore[11]=NTRAJSTORE;
    intstore[12]=BINSIZE; intstore[13]=-1; intstore[14]=-1;
    intstore[15]=-1; intstore[16]=-1; intstore[17]=-1;
    intstore[18]=-1; intstore[19]=-1; intstore[20]=-1;
    intstore[21]=-1; intstore[22]=-1; intstore[23]=-1;
    intstore[24]=-1;
#ifdef double
    intstore[25]=1;
#else
    intstore[25]=0;
#endif
    intstore[26]=-1; intstore[27]=-1; intstore[28]=-1;
    intstore[29]=-1; intstore[30]=-1; intstore[31]=-1;
    intstore[32]=-1;

    floatstore[0]=LX; floatstore[1]=LY; floatstore[2]=LZ;
    floatstore[3]=TREF; floatstore[4]=RCUTOFF; floatstore[5]=TSTEP;
    floatstore[6]=theTrajectory.strain_rate; floatstore[7]=-1.0;
    floatstore[8]=-1.0;
    strcpy(c[0],STORENAME); strcpy(c[1],RESTORENAME);
    strcpy(c[2],STORERESULT); strcpy(c[3],STORETRAJ);
     strcpy(c[4],STORESTRESS); strcpy(c[5],STOREVELOCITY);

    counter=NSEQUIL+NSMIC+NSKIN+NSSHEAR;
    if(counter<BINSIZE) {
      fprintf(stderr,"av_result: BINSIZE must be greater then NSEQUIL+NSMIC+NSKI
    }
    counter=counter/BINSIZE-1;

    data=dmatrix(0,counter,0,12); nstepa=ivector(0,counter);
  }
  if((theTrajectory.istep-theTrajectory.fstep)%BINSIZE==0)
    { for(i=0;i<=13;i++) avdata[i]=0.0; scounter+=1; }
  for(i=0;i<=13;i++) avdata[i]+=store[i];
  if((theTrajectory.istep-theTrajectory.fstep+1)%BINSIZE==0) {
    nstepa[scounter]=theTrajectory.istep;
    for(i=0;i<=13;i++) data[scounter][i]=avdata[i]/(double)BINSIZE;
    for(i=0;i<=13;i++) binav[i]=0.0;
    for(i=0;i<=scounter;i++) { for(j=0;j<=13;j++) binav[j]+=data[i][j]; }
    for(i=0;i<=13;i++) binav[i]/=(scounter+1);
    for(i=0;i<=13;i++) binstdev[i]=0.0;
    for(i=0;i<=scounter;i++) {
      for(j=0;j<=13;j++) {
        binstdev[j]+=(data[i][j]-binav[j])*(data[i][j]-binav[j]);
    } }
    if(scounter>0) { for(i=0;i<=13;i++) binstdev[i]/=scounter; }
    for(i=0;i<=13;i++) binstdev[i]=sqrt(binstdev[i]);
    if((fp=fopen(STORERESULT,"w"))==NULL)
      { fprintf(stderr,"Cannot open file (storedata)\n"); exit(1); }
    for(i=0;i<=32;i++) fprintf(fp,"%d ",intstore[i]);
    fprintf(fp,"\n");
    for(i=0;i<=8;i++) fprintf(fp,"%2.6lf ",floatstore[i]);
    fprintf(fp,"\n");
    for(i=0;i<=5;i++) {fprintf(fp,c[i]);fprintf(fp,"  ");}
    fprintf(fp,"\n");
    fprintf(fp,"  istep    ncut         strain          rtemp          rekin
    for(i=0;i<=scounter;i++)
      fprintf(fp,"%7d %5.1f %14.7e %14.7e %14.7e %14.7e %14.7e %14.7e\n",
                   nstepa[i],data[i][13],data[i][0],data[i][1],
                   data[i][2],data[i][3],data[i][4],data[i][5]);

    fprintf(fp,"\n  istep         epot         etot           mz          dor
    for(i=0;i<=scounter;i++)
      fprintf(fp,"%7d %12.5e %12.5e %12.5e %12.5e %12.5e %12.5e %12.5e\n",
                   nstepa[i],data[i][6],data[i][7],data[i][8],
                   data[i][9],data[i][10],data[i][11],data[i][12]);
    fprintf(fp,"\n");
    fprintf(fp,"ncut %12.5e (+/-)%12.5e\n",binav[13],binstdev[13]);
    for(i=0;i<=12;i++)
      fprintf(fp,"%4d %12.5e (+/-)%12.5e\n",i,binav[i],binstdev[i]);
    fprintf(fp,"\x0C\n");
    fclose(fp);
    if(theTrajectory.istep==theTrajectory.lstep)
      { free_ivector(nstepa,0,counter); free_dmatrix(data,0,counter,0,13); }
  }
}
```

```
void av_velocity(void)
{
  static int first=1,scounter=-1,*nstepa,counter;
  int i,j,k;
  static double **data,avdata[18];
  double binav[18],binstdev[18];
  FILE *fp;

  if(first) {
    counter=NSEQUIL+NSMIC+NSKIN+NSSHEAR;
    if(counter<BINSIZE) {
      fprintf(stderr,"velocitystore: BINSIZE must be greater then NSEQUIL+NSMIC+
    }
    first=0; counter=counter/BINSIZE-1;
    data=dmatrix(0,counter,0,17); nstepa=ivector(0,counter);
  }
  if((theTrajectory.istep-theTrajectory.fstep)%BINSIZE==0) {
    for(i=0;i<=17;i++) avdata[i]=0.0; scounter+=1;
  }
  avdata[0]+=theTrajectory.ktx; avdata[1]+=theTrajectory.kty;
  avdata[2]+=theTrajectory.ktz; avdata[6]+=theTrajectory.krx;
  avdata[7]+=theTrajectory.kry; avdata[8]+=theTrajectory.krz;
  avdata[3]+=theTrajectory.ortx; avdata[4]+=theTrajectory.orty;
  avdata[5]+=theTrajectory.ortz; avdata[9]+=theTrajectory.orrx;
  avdata[10]+=theTrajectory.orry; avdata[11]+=theTrajectory.orrz;
  avdata[12]+=theTrajectory.dorx; avdata[13]+=theTrajectory.dory;
  avdata[14]+=theTrajectory.dorz; avdata[15]+=theTrajectory.momx;
  avdata[16]+=theTrajectory.momy; avdata[17]+=theTrajectory.momz;
  if((theTrajectory.istep-theTrajectory.fstep+1)%BINSIZE==0) {
    nstepa[scounter]=theTrajectory.istep;
    for(i=0;i<=17;i++) data[scounter][i]=avdata[i]/(double)BINSIZE;
    for(i=0;i<=17;i++) binav[i]=0.0;
    for(i=0;i<=scounter;i++) { for(j=0;j<=17;j++) binav[j]+=data[i][j]; }
    for(i=0;i<=17;i++) binav[i]/=(scounter+1);
    for(i=0;i<=17;i++) binstdev[i]=0.0;
    for(i=0;i<=scounter;i++) {
      for(j=0;j<=17;j++) {
        binstdev[j]+=(data[i][j]-binav[j])*(data[i][j]-binav[j]);
    } }
    if(scounter>0) { for(i=0;i<=17;i++) binstdev[i]/=scounter; }
    for(i=0;i<=17;i++) binstdev[i]=sqrt(binstdev[i]);
    if((fp=fopen(STOREVELOCITY,"w"))==NULL) {
      fprintf(stderr,"Cannot open file (velocitystore)\n"); exit(1);
    }
    fprintf(fp,"  istep       Ktx        Kty        Ktz        ortx       orty
    for(i=0;i<=scounter; i++) {
      fprintf(fp,"%7d %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %1
                     nstepa[i],data[i][0],data[i][1],data[i][2],data[i][3],
                     data[i][4],data[i][5],data[i][12],data[i][13],data[i][14]);
    }
    fprintf(fp,"\n");
    fprintf(fp,"  istep       Krx        Kry        Krz       orrx        orry
    for(i=0;i<=scounter; i++) {
      fprintf(fp,"%7d %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %10.3e %1
                    nstepa[i],data[i][6],data[i][7],data[i][8],data[i][9],
                   data[i][10],data[i][11],data[i][15],data[i][16],data[i][17]);
    }
    fprintf(fp,"\n");
    for(i=0;i<=5;i++) {
      k=i+6;
    fprintf(fp,"%3d %12.5e (+/-)%12.5e   %3d %12.5e (+/-)%12.5e\n",i,    av[i],
  }
  fprintf(fp,"\n");
  for(i=12;i<=17;i++)
    fprintf(fp,"%3d %12.5e (+/-)%12.5e\n",i,binav[i],binstdev[i]);
  fprintf(fp,"\x0C\n");
  fclose(fp);
  if(theTrajectory.istep==theTrajectory.lstep) {
    free_ivector(nstepa,0,counter);
    free_dmatrix(data,0,counter,0,45);
  }
 }
}
```

```
/* utility.c */

/* Contains some utility routines - all but last three taken from
   "Numerical Recipes in C". The last three exchange order of bytes
   in binary numbers - there are useful when restarting a simulation
   on big-endian machine with a restart file created on a
   small-endian machine (e.g. 386/ESIX <---> SPARC/SunOS).
*/

#include <malloc.h>
#include <stdio.h>

double **dmatrix(int nrl, int nrh, int ncl, int nch)
{
  void mderror(char *p);
  int    i;
  double **m;

  /* Allocate pointers to raws */
  m=(double **)malloc((unsigned) (nrh-nrl+1)*sizeof(double*));
  if (!m) mderror("allocation failure 1 in dmatrix()");
  if (!m) printf("allocation failure 1 in dmatrix()\n");

  /* Allocate raws and set pointers to them */
  m -= nrl;
  for(i=nrl;i<=nrh;i++) {
    m[i]=(double *)malloc((unsigned) (nch-ncl+1)*sizeof(double));
    if(!m[i]) mderror("allocation failure 2 in dmatrix()");
    if(!m[i]) printf("allocation failure 2 in dmatrix()\n");
    m[i] -= ncl;
  }
  return m;
}

float **matrix(int nrl, int nrh, int ncl, int nch)
{
  void mderror(char *p);
  int    i;
  float **m;

  /* Allocate pointers to raws */
  m=(float **)malloc((unsigned) (nrh-nrl+1)*sizeof(float*));
  if (!m) mderror("allocation failure 1 in dmatrix()");

  /* Allocate raws and set pointers to them */
  m -= nrl;
  for(i=nrl;i<=nrh;i++) {
    m[i]=(float *)malloc((unsigned) (nch-ncl+1)*sizeof(float));
    if(!m[i]) mderror("allocation failure 2 in dmatrix()");
    m[i] -= ncl;
  }
  return m;
}

double *dvector(int nl, int nh)
{
  void mderror(char *p);
  double *m;
  /* Allocate a vector */
  m=(double *)malloc((unsigned) (nh-nl+1)*sizeof(double));
  if (!m) mderror("allocation failure 1 in dvector()");
  m -= nl;
  return m;
}

float *vector(int nl, int nh)
{
  void mderror(char *p);
  float *m;

  /* Allocate a vector */
  m=(float *)malloc((unsigned) (nh-nl+1)*sizeof(float));
  if (!m) mderror("allocation failure 1 in vector()");
  m -= nl;
  return m;
}

int *ivector(int nl, int nh)
{
  void mderror(char *p);
  int *m;

  /* Allocate a vector */
  m=(int *)malloc((unsigned) (nh-nl+1)*sizeof(int));
  if (!m) mderror("allocation failure 1 in vector()");
  m -= nl;
  return m;
}


void mderror(char *p)
{
  void exit();

  fprintf(stderr, "*** ERROR ***\n");
  fprintf(stderr, "%s\n", p);
  fprintf(stderr, "*** EXITING ***\n");
  exit(1);
}

void free_dmatrix(double **m, int nrl, int nrh, int ncl, int nch)
{
  int i;
  for(i=nrh;i>=nrl;i--) free((char*)(m[i]+ncl));
  free((char*)(m+nrl));
}

void free_matrix(float **m, int nrl, int nrh, int ncl, int nch)
{
  int i;
  for(i=nrh;i>=nrl;i--) free((char*)(m[i]+ncl));
  free((char*)(m+nrl));
}

void free_dvector(double *m, int nl, int nh)
{
  free((char*)(m+nl));
}
```

```
void free_vector(float *m, int nl, int nh)
{
  free((char*)(m+nl));
}

void free_ivector(int *m, int nl, int nh)
{
  free((char*)(m+nl));
}

void intchange(int *tbc)
{
  char *byte1,*byte2,*byte3,*byte4;
  unsigned char store;

  byte1=(char*)tbc;
  byte2=byte1+1;  byte3=byte2+1;  byte4=byte3+1;
  store=*byte4;  *byte4=*byte1;  *byte1=store;
  store=*byte2;  *byte2=*byte3;  *byte3=store;
}

void floatchange(float *tbc)
{
  char *byte1,*byte2,*byte3,*byte4;
  unsigned char store;

  byte1=(char*)tbc;
  byte2=byte1+1;  byte3=byte2+1;  byte4=byte3+1;
  store=*byte4;  *byte4=*byte1;  *byte1=store;
  store=*byte2;  *byte2=*byte3;  *byte3=store;
}

void doublechange(double *tbc)
{
  char *byte1,*byte2,*byte3,*byte4;
  char *byte5,*byte6,*byte7,*byte8;
  unsigned char store;

  byte1=(char*)tbc;
  byte2=byte1+1;  byte3=byte2+1;  byte4=byte3+1;
  byte5=byte4+1;  byte6=byte5+1;  byte7=byte6+1; byte8=byte7+1;
  store=*byte8;  *byte8=*byte1;  *byte1=store;
  store=*byte7;  *byte7=*byte2;  *byte2=store;
  store=*byte6;  *byte6=*byte3;  *byte3=store;
  store=*byte5;  *byte5=*byte4;  *byte4=store;
}
```

```
/* urng.c */

/* The universal random number generator is described by G. Marsaglia
   and A. Zaman in TOWARD A UNIVERSAL RANDOM NUMBER GENERATOR, Statistics
   & Probability Letters 9 (1990) 35-39.

   We have adapted the algorithm to perform calculation on long integers
   (32 bits). This makes the algorithm less portable but faster.
   The program returns numbers between 0 and 16 777 215 and it requires
   four initial guesses. g1, g2, g3 must be in the range 1 to 178, and
   not all 1, while g4 may be any integer from 0 to 168.

   The algorithm that uses floating point calculation was written by
   Jim Butler, and was based on a FORTRAN program posted by David
   LaSalle. Check SIMTEL20 (or serveur on MUSIC) in directories C and
   FORTRAN for listings of these programs.

   According to its authors, this generator "passes all the standard
   tests, and all the latest - more stringent - tests for randomness,
   has an incredibly long period 2^144 ...".

   The generator is 1.3 times slower than RANP.
*/

#define G1 12  /* 4 */
#define G2 34  /* 5 */
#define G3 56  /* 6 */
#define G4 78  /* 7 */

int urng(void)
{
  static unsigned int uni[97];            /* register for Fibonacci       */
  static unsigned int *pto97=uni-1;       /* initialized for not accessed */
  static unsigned int *pto33=uni-1;       /* initialized for not accessed */
  static int first=1;                     /* to initialize the generator  */
  static unsigned int c=362436,d=7654321,dm=16777213;
  int guess_g1=G1,guess_g2=G2,guess_g3=G3,guess_g4=G4;
  int i,j,m;

  if(first) {   /* initialize the generator */
    if (guess_g1<=0||guess_g1>=179) guess_g1=12;
    if (guess_g2<=0||guess_g2>=179) guess_g2=34;
    if (guess_g3<=0||guess_g3>=179) guess_g3=56;
    if (guess_g4<=-1||guess_g4>=168) guess_g4=78;
    if (guess_g1==1&&guess_g2==1&&guess_g3==1) guess_g1=12;
    pto97 = uni;
    for(i=0;i<97;i++) {
      for(j=0;j<24;j++) {
        m=(((guess_g1*guess_g2)%179)*guess_g3)%179;
        guess_g1=guess_g2;guess_g2=guess_g3;guess_g3=m;
        guess_g4=(53*guess_g4+1)%169;
        *pto97<<=1;                              /* shift up by one */
        if((m*guess_g4)&0x20) *pto97+=1;         /* get a sixth bit */
      }
      pto97++;
    }
    pto97=uni;
    pto97+=96;                           /* point to 97th element          */
    pto33=uni+32;                        /* point to 33rd element          */
    first=0;
  }

  /* This is the generator */
  if(pto97<uni) pto97=uni+96;           /* start calculation of random */
  if(pto33<uni) pto33=uni+96;           /* numbers                       */
  *pto97=(*pto97-*pto33)&0xffffff;      /* update Fibonacci mod(2^24)  */
  if(c>=d) c-=d;                        /* update arithmetic sequence  */
    else c+=(dm-d);
  pto97--;                              /* update pointers to Fibonacci*/
  pto33--;
  return((*(pto97+1)-c)&0xffffff);      /* return a random number        */
}

/* Generates random variate from the gaussian distribution
   with zero mean and unit standard deviation. Taken from
   Allen and Tildesley, microfiche F. 24.
*/

#define A1 3.949846138
#define A3 0.252408784
#define A5 0.076542912
#define A7 0.008355968
#define A9 0.029899776

double gauss(void)
{
  int i;
  double sum,r,r2;

  sum=0.0;
  for(i=0;i<12;i++) sum+=urng()/16777215.0;
  r=(sum-6.0)/4.0;
  r2=r*r;
  return( ((((A9*r2+A7)*r2+A5)*r2+A3)*r2+A1)*r );
}
```

```
/* ders.c */

/* This subroutine calculates derivatives of the Gay-Berne potential
   (phi) using the analytical expressions. As input, it
   accepts two unit vectors pi[3], and pj[3], as well as the
   center-to-center vector r[3]. As output, it gives derv[10] that
   contains: derv[0]=phi
             derv[1]=pd(phi)/pd(xi)=-pd(pot)/pd(xj)
             derv[2]=pd(phi)/pd(yi)=-pd(pot)/pd(yj)
             derv[3]=pd(phi)/pd(xi)=-pd(pot)/pd(xj)
             derv[4]=pd(phi)/pd(pxi)
             derv[5]=pd(phi)/pd(pyi)
             derv[6]=pd(phi)/pd(pzi)
             derv[7]=pd(phi)/pd(pxj)
             derv[8]=pd(phi)/pd(pyj)
             derv[9]=pd(phi)/pd(pzj),
   where pd denotes a partial derivative.
   Subroutine pot(double pi[], double pj[], double r[]), accepts the
   same input as ders() but returns only phi - used to verify the
   results from ders(), not used in production runs. It may be used to
   calculate the derivatives numerically. Note: the Gay-Berne
   parameters are hard-coded in the define statements.
*/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include "rb.h"

#define KAPPA  0.566160520607
#define K      0.225148226553
#define KAPPAH 0.283080260304
#define KAPPA2 0.320537735095
#define KAPPAN 0.824295010846
#define KH     0.112574113276
#define PI180  0.017453292520

void ders(double pi[], double pj[], double r[], double derv[])
{
  double ru[3],rr,rupimpj,rupippj,pipj,sigma,rrms,eps1,eps2;
  double rrms2,rrms6,pippj[3],pimpj[3];
  double sigma_1,sigma_2,sigma_1s,sigma_2s,eps2_1,eps2_2,eps2_1s,eps2_2s;
  double eps,potp,axi,ayi,azi,bxi,byi,bzi;
  double ke1e2,exi,eyi,ezi,sig,sigma_st,sxi,syi,szi;
  double rrms7,potd,eps1_1,e1e2t2,eps1K,sig1,sig2;

  /* calculate model potential */
  rr=r[0]*r[0]+r[1]*r[1]+r[2]*r[2]; rr=sqrt(rr);
  ru[0]=r[0]/rr; ru[1]=r[1]/rr; ru[2]=r[2]/rr;
  pimpj[0]=pi[0]-pj[0]; pimpj[1]=pi[1]-pj[1]; pimpj[2]=pi[2]-pj[2];
  pippj[0]=pi[0]+pj[0]; pippj[1]=pi[1]+pj[1]; pippj[2]=pi[2]+pj[2];
  rupimpj=ru[0]*pimpj[0]+ru[1]*pimpj[1]+ru[2]*pimpj[2];
  rupippj=ru[0]*pippj[0]+ru[1]*pippj[1]+ru[2]*pippj[2];
  pipj=pi[0]*pj[0]+pi[1]*pj[1]+pi[2]*pj[2];
  sigma_1=rupippj/(1.0+KAPPA*pipj);
  sigma_2=rupimpj/(1.0-KAPPA*pipj);
  sigma_1s=sigma_1*rupippj; sigma_2s=sigma_2*rupimpj;
  sigma_st=1.0/(1.0-KAPPAH*(sigma_1s+sigma_2s)); sigma=sqrt(sigma_st);
  eps2_1=rupippj/(1.0+K*pipj); eps2_2=rupimpj/(1.0-K*pipj);
  eps2_1s=eps2_1*rupippj; eps2_2s=eps2_2*rupimpj;
  eps2=eps2_1s+eps2_2s; eps2=1.0-KH*eps2;
  rrms=1.0/(rr-sigma+1.0);
  rrms2=rrms*rrms; rrms6=rrms2*rrms2*rrms2;
  eps1_1=1.0/(1.0-KAPPA2*pipj*pipj);
  eps1=KAPPAN*sqrt(eps1_1); /* normalize eps1 for side-by-side */
  /* get potential */
  potp=rrms6*rrms6-rrms6; eps=eps1*eps2*eps2; derv[0]=4.0*potp*eps;
  /* get derivatives over components of r */
  /** derivatives of epsilon **/
  axi=(pippj[0]-rupippj*ru[0])/rr; ayi=(pippj[1]-rupippj*ru[1])/rr;
  azi=(pippj[2]-rupippj*ru[2])/rr; bxi=(pimpj[0]-rupimpj*ru[0])/rr;
  byi=(pimpj[1]-rupimpj*ru[1])/rr; bzi=(pimpj[2]-rupimpj*ru[2])/rr;
  ke1e2=-2.0*eps1*eps2*K;
  exi=ke1e2*(eps2_1*axi+eps2_2*bxi);
  eyi=ke1e2*(eps2_1*ayi+eps2_2*byi);
  ezi=ke1e2*(eps2_1*azi+eps2_2*bzi);
  /** derivatives of sigma **/
  rrms7=rrms6*rrms;
  potd=6.0*(rrms7-2.0*rrms7*rrms6);
  sig1=sigma_st*sigma*KAPPAH;
  sxi=potd*(ru[0]-sig1*(sigma_1*axi+sigma_2*bxi));
  syi=potd*(ru[1]-sig1*(sigma_1*ayi+sigma_2*byi));
  szi=potd*(ru[2]-sig1*(sigma_1*azi+sigma_2*bzi));
  /**  derivatives over r **/
  derv[1]=4.0*(potp*exi+eps*sxi);
  derv[2]=4.0*(potp*eyi+eps*syi);
  derv[3]=4.0*(potp*ezi+eps*szi);
  /* get derivatives over unit vectors */
  /** derivatives of epsilon over pi */
  sig=KAPPA2*eps1*eps2*eps1_1*pipj; /* reuse variables */
  eps1K=eps1*K; ke1e2=K*eps1K*(eps2_2*eps2_2-eps2_1*eps2_1);
  sig2=4.0*eps2*potp; e1e2t2=2.0*sig2*eps1K*(eps2_1+eps2_2);
  ke1e2=sig2*(sig-ke1e2);
  derv[4]=ke1e2*pj[0]-e1e2t2*ru[0]; /* over pi */
  derv[5]=ke1e2*pj[1]-e1e2t2*ru[1];
  derv[6]=ke1e2*pj[2]-e1e2t2*ru[2];
  /** derivatives of epsilon over pj */
  e1e2t2=2.0*sig2*eps1K*(eps2_1-eps2_2);
  derv[7]=ke1e2*pi[0]-e1e2t2*ru[0]; /* over pj */
  derv[8]=ke1e2*pi[1]-e1e2t2*ru[1];
  derv[9]=ke1e2*pi[2]-e1e2t2*ru[2];
  /** derivatives of sigma over pi */
  sig1=-potd*sig1*2.0*eps;
  ke1e2=sig1*KAPPA*(sigma_2*sigma_2-sigma_1*sigma_1);
  e1e2t2=2.0*sig1*(sigma_1+sigma_2);
  derv[4]+=e1e2t2*ru[0]+ke1e2*pj[0];  /* over pi */
  derv[5]+=e1e2t2*ru[1]+ke1e2*pj[1];
  derv[6]+=e1e2t2*ru[2]+ke1e2*pj[2];
  /** derivatives of sigma over pj */
  e1e2t2=2.0*sig1*(sigma_1-sigma_2);
  derv[7]+=e1e2t2*ru[0]+ke1e2*pi[0];  /* over pj */
  derv[8]+=e1e2t2*ru[1]+ke1e2*pi[1];
  derv[9]+=e1e2t2*ru[2]+ke1e2*pi[2];
}

double pot(double pi[], double pj[], double r[])
{
  double ru[3],rr,pipjp,pipjm,rupi,rupj,pipj,sigma,rrms,eps1,eps2;
  /* calculate model potential */
  rr=r[0]*r[0]+r[1]*r[1]+r[2]*r[2]; rr=sqrt(rr);
  ru[0]=r[0]/rr; ru[1]=r[1]/rr; ru[2]=r[2]/rr;
  rupi=ru[0]*pi[0]+ru[1]*pi[1]+ru[2]*pi[2];
  rupj=ru[0]*pj[0]+ru[1]*pj[1]+ru[2]*pj[2];
  pipj=pi[0]*pj[0]+pi[1]*pj[1]+pi[2]*pj[2];
  pipjp=(rupi+rupj)*(rupi+rupj);
  pipjm=(rupi-rupj)*(rupi-rupj);
  sigma=pipjp/(1.0+KAPPA*pipj)+pipjm/(1.0-KAPPA*pipj);
  sigma=1.0/sqrt(1.0-KAPPAH*sigma);
  eps2=pipjp/(1.0+K*pipj)+pipjm/(1.0-K*pipj);
  eps2=1.0-KH*eps2; rrms=1.0/(rr-sigma+1.0);
  rrms=rrms*rrms*rrms; rrms=rrms*rrms;
  eps1=KAPPAN/sqrt(1.0-KAPPA2*pipj*pipj);
  return(4.0*eps1*eps2*eps2*(rrms*rrms-rrms));
}
```

```
/* radial.c */

/* Calculate radial distribution function. This routine has
   hard-coded parameters - may be used only for 256 molecules. */

#include <stdio.h>
#include <math.h>
#include "rb.h"
extern struct CurrentTrajectory theTrajectory;

void radial()
{
  static double **g;
  double sideh1,sideh2,sideh3,bring,xij,yij,zij,rd,rdpot;
  double posix,posiy,posiz,xj,rdsr,g1,g2,g3,g4,g5,g6,g7,g8,g9,g10,g11,g12;
  double lon,trans,r,deltar,rho,f,vel,faccm,avcm[101];
  static int first=1,maxcm[101],steps=0;
  int i,j,binno;
  FILE *fp,*fp1;

  steps++;
  if(first) {
    g=dmatrix(1,12,0,600);
    for(i=1;i<=12;i++) { for(j=0;j<=600;j++) g[i][j]=0.0; }
    for(i=0;i<=100;i++) { maxcm[i]=0;};
    first=0;
  }
  /* get xij, yij, zij */
  sideh1=0.5*LX; sideh2=0.5*LY; sideh3=0.5*LZ;
  for(i=1;i<NPART;i++){                      /* Outer force loop */
    posix=theTrajectory.rv[1][i]; posiy=theTrajectory.rv[2][i];
    posiz=theTrajectory.rv[3][i];
    for(j=i+1;j<=NPART;j++) {                   /* Inner force loop */
      xij=posix-theTrajectory.rv[1][j]; yij=posiy-theTrajectory.rv[2][j];
      zij=posiz-theTrajectory.rv[3][j];
      if(zij<(-sideh3))  zij+=LZ; if(zij>  sideh3 )  zij-=LZ;
      if(xij<(-sideh1))  xij+=LX; if(xij>  sideh1 )  xij-=LX;
      if(theTrajectory.istep<theTrajectory.shear_step) {
        if(yij<(-sideh2))  yij+=LY; if(yij>  sideh2 )  yij-=LY;
      }
      else {
        if(yij>  sideh2 ) {               /* ABOVE */
          yij-=LY; xj=theTrajectory.rv[1][j]; xj+=theTrajectory.strain;
          bring=floor(xj/LX+0.5); xj-=(bring*LX); xij=posix-xj;
          if(xij>  sideh1 )  xij-=LX; if(xij<(-sideh1))  xij+=LX;
        }
        if(yij<(-sideh2)) {               /* BELOW */
          yij+=LY; xj=theTrajectory.rv[1][j]; xj-=theTrajectory.strain;
          bring=ceil(xj/LX-0.5); xj-=(bring*LX); xij=posix-xj;
          if(xij>  sideh1 )  xij-=LX; if(xij<(-sideh1))  xij+=LX;
      } }
      rd=xij*xij+yij*yij+zij*zij; rdsr=sqrt(rd);
      binno=(int)(rdsr*33.333333333);
      if(binno<0||binno>599) {
         fprintf(stderr,"Wrong binno *** stop (rdsr=%13.3e%13.3e%13.3e%13.3e)\n"
                          rdsr,xij,yij,zij);
         exit(1);
      }
      g[1][binno]+=xij*xij/rd; g[2][binno]+=xij*yij/rd;
    g[3][binno]+=xij*zij/rd; g[4][binno]+=yij*yij/rd;
    g[5][binno]+=yij*zij/rd; g[6][binno]+=zij*zij/rd;
    /* get transverse and longitudinal */
    lon=theTrajectory.dorx*xij+theTrajectory.dory*yij+theTrajectory.dorz*zij;
    lon=lon*lon; trans=rd-lon;
    g[7][binno]+=lon/rd; g[8][binno]+=trans/rd;
    binno=(int)(sqrt(lon)*33.333333333);
    if(binno<0||binno>599) {
       fprintf(stderr,"%13.3e%13.3e%13.3e%13.3e\n",lon,xij,yij,zij);
       fprintf(stderr,"%13.3e%13.3e%13.3e%13.3e\n",
        lon,theTrajectory.dorx,theTrajectory.dory,theTrajectory.dorz);
       fprintf(stderr,"Wrong binno *** stop lon %7d\n",binno); exit(1);
    }
    g[9][binno]+=1.0;
    binno=(int)(sqrt(trans)*33.333333333);
    if(binno<0||binno>599) {
       fprintf(stderr,"Wrong binno *** stop trans\n"); exit(1);
    }
    g[10][binno]+=1.0;
  }
}
for(i=1;i<=NPART;i++) {
  vel=0.5*sqrt(theTrajectory.P[1][i]*theTrajectory.P[1][i]+
                   theTrajectory.P[2][i]*theTrajectory.P[2][i]+
                   theTrajectory.P[3][i]*theTrajectory.P[3][i]);
  binno=(int)(vel*10.0);
  if(binno>99) binno=99; maxcm[binno]++;
}

if(theTrajectory.istep==theTrajectory.lstep) {
  if((fp=fopen(RADIAL,"w"))==NULL) {
    fprintf(stderr,"Cannot open file\n");
    exit(1);
  }
  rho=NPART/(LX*LY*LZ);
  for(i=0;i<=599;i++) {
    r=((double)i+0.5)/33.3333333333;
    deltar=0.03;   /* change this */
    f=1.0/(2.0*M_PI*r*r*deltar*rho*steps*NPART);
    g1=g[1][i]*f; g2=g[2][i]*f; g3=g[3][i]*f; g4=g[4][i]*f;
    g5=g[5][i]*f; g6=g[6][i]*f; g7=g1+g4+g6;
    g8=g[7][i]*f; g9=g[8][i]*f; g10=g8+g9;
    f=1.0/(LX*LX*deltar*rho*steps*NPART); /* twice the volume */
    g11=g[9][i]*f;
    f=2.0/(LX*2.0*M_PI*r*deltar*rho*steps*NPART);
    g12=g[10][i]*f;
    fprintf(fp,"%12.3e%12.3e%12.3e%12.3e%12.3e%12.3e%12.3e%12.3e%12.3e%12.3e%1
     r,g1,g2,g3,g4,g5,g6,g7,g8,g9,g10,g11,g12);
  }
  free_dmatrix(g,1,12,0,600);
  close(fp);
  if((fp1=fopen(VELDIST,"wb"))==NULL) {
    fprintf(stderr,"Cannot open file\n");
    exit(1);
  }
  faccm=0.0;
  for(i=0;i<=99;i++) { faccm+=(double)maxcm[i]; }; faccm*=0.1;
  for(i=0;i<=99;i++) { avcm[i]=(double)maxcm[i]/faccm; }
  for(i=0;i<=99;i++) {
    fprintf(fp1,"%12e%12.3e\n", (i+0.5)*0.1, avcm[i]);
    }
    close(fp1);
} }
```