

Titre: Utilisation d'un système d'information géographique dans la
Title: modélisation d'un bassin versant

Auteur: Stéphanie Thériault
Author:

Date: 1997

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Thériault, S. (1997). Utilisation d'un système d'information géographique dans la
Citation: modélisation d'un bassin versant [Master's thesis, École Polytechnique de
Montréal]. PolyPublie. <https://publications.polymtl.ca/9048/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/9048/>
PolyPublie URL:

**Directeurs de
recherche:** Claude Marche, & Pierre Tremblay
Advisors:

Programme: Unspecified
Program:



UNIVERSITÉ DE MONTRÉAL

UTILISATION D'UN SYSTÈME D'INFORMATION
GÉOGRAPHIQUE DANS LA MODÉLISATION
D'UN BASSIN VERSANT

STÉPHANIE THÉRIAULT
DÉPARTEMENT DE GÉNIE CIVIL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE CIVIL)
JANVIER 1997



© Stéphanie Thériault, 1997.

**Acquisitions and
Bibliographic Services**

**Acquisitions et
services bibliographiques**

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-26523-4

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

UTILISATION D'UN SYSTÈME D'INFORMATION
GÉOGRAPHIQUE DANS LA MODÉLISATION
D'UN BASSIN VERSANT

présentée par: THÉRIAULT Stéphanie

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

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

M. ROUSSELLE, Jean, Ph.D., président

M. MARCHE Claude, D.Sc.A., membre et directeur de recherche

M. TREMBLAY Pierre R., ing., M.Sc., membre et codirecteur de recherche

M. LECONTE Robert, Ph.D., membre

*À D.C. pour
son soutien
et sa patience.*

REMERCIEMENTS

Même s'il ne porte qu'une seule griffe, la réalisation d'un ouvrage comme celui-ci n'en demeure pas moins un travail d'équipe. C'est pourquoi je tiens à remercier spécialement tous ceux et celles qui, de près ou de loin, par leur appui, leur réconfort ou leur aide concrète, ont collaboré à la réalisation de ce travail. Je remercie plus particulièrement mon directeur M. Claude Marche ainsi que mon codirecteur M. Pierre R. Tremblay.

RÉSUMÉ

La modélisation du bassin versant de la rivière Samson se fait en établissant un lien entre un modèle hydrologique et le système d'information géographique SPANS. La rivière Samson, un affluent de la rivière Chaudière pour lequel il existe déjà une base de données dans SPANS, remplit les conditions requises pour l'étude à savoir un petit bassin versant naturel et non-jaugé. Après un examen des différents modèles hydrologiques existants et facilement disponibles, il ressort que ceux-ci sont d'utilisation trop complexe ou inaptes à exploiter les avantages des SIG. Le modèle est donc créé en intégrant des paramètres hydrologiques à un algorithme de propagation d'écoulement sur un bassin versant en deux dimensions. Cet algorithme utilise un modèle numérique d'élévation de terrain pour propager une hauteur de pluie sur l'ensemble du bassin versant. Avec les quantités d'eau à l'exutoire et les temps de concentration calculés par l'algorithme, il est possible de générer un hydrogramme du bassin et d'obtenir la variation du débit à l'exutoire pour une précipitation donnée. Cette méthode, d'abord testée sur des formes géométriques simples donne des résultats probants. Elle est ensuite appliquée au bassin versant à l'étude. Les résultats obtenus ne sont pas ceux escomptés, mais un ajustement au niveau de la précision du modèle numérique d'élévation de terrain et du calcul du temps de concentration permettrait d'obtenir des résultats plus concluants.

ABSTRACT

The modeling of the Samson River's watershed is made through the coupling of the geographical information system (GIS) SPANS with a hydrological model. The Samson River, a tributary of the Chaudière River for which a SPANS database already exists, possesses the required characteristics for the study: a small, natural and ungaged watershed. After a close look at existing and easily available hydrological models, it seems that none of them are appropriate due to their inability to take advantage of the GIS capabilities and because their use is too complex for the scope of this study. Thus, a hydrological model is built up from an existing two-dimensional routing algorithm modified to include hydrological parameters. The algorithm discretizes the watershed as a rectangular grid and uses a digital elevation model and the SCS curve numbers approach to compute the contributing areas. These contributing areas are indicators of discharge and can be used with the time to peak to construct a hydrograph of a rain event. This method is first tested with simple watersheds with good results. It is then applied to the watershed under study. The results obtained are not the ones expected, however with an adjustment to the accuracy of digital elevation model, and by modifying the way the time to peak is computed the algorithm should lead to conclusive results.

TABLE DES MATIÈRES

DÉDICACE.....	iv
REMERCIEMENTS.....	v
RÉSUMÉ.....	vi
ABSTRACT	vii
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX.....	xi
LISTE DES FIGURES.....	xiii
LISTE DES SIGLES ET ABRÉVIATIONS.....	xviii
LISTE DES ANNEXES	xx
INTRODUCTION	1
CHAPITRE 1 LES SYSTÈMES D'INFORMATION GÉOGRAPHIQUE.....	7
1.1 Principes de base des systèmes d'information géographique	8
1.1.1 Définition des systèmes d'information géographique.....	8
1.1.2 Fonctionnement et concepts de base.....	12
1.1.2.1 Fonctionnement d'un système d'information géographique	16
1.2 Le système d'information géographique SPANS	31
1.3 Les systèmes d'information géographique et les ressources hydriques	34

CHAPITRE 2 LES MODÈLES DE PRÉVISION HYDROLOGIQUE.....	40
2.1 Cadre de l'étude.....	42
2.2 Choix d'un modèle de prévision hydrologique	46
2.2.1 Modèles de prévision hydrologique.....	47
2.2.2 Lien entre un SIG et un modèle hydrologique	51
2.2.3 Modélisation d'un bassin par un système d'information géographique	54
CHAPITRE 3 MODÉLISATION DU BASSIN VERSANT.....	56
3.1 Processus hydrologiques.....	57
3.1.1 Méthode pluie-ruissellement du SCS.....	60
3.1.1.1 Estimation des CN	64
3.1.2 Implémentation de la méthode du SCS dans le système SPANS	69
3.2 Modèle d'écoulement.....	79
3.2.1 Le programme DEMON (Digital Elevation MOdel Network)	82
3.2.1.1 Représentation de l'écoulement dans le programme DEMON	85
3.2.1.2 L'algorithme du programme DEMON	87
3.2.2 Modifications à apporter au programme DEMON	95
CHAPITRE 4 MÉTHODE ET RÉSULTATS.....	100
4.1 Collecte des données	101
4.1.1 La topographie.....	101
4.1.2 Les coefficients de ruissellement	103
4.1.3 Les précipitations.....	104
4.1.4 Transfert des données vers le programme DEMONII	105
4.1.4.1 Opérations dans SPANS	105
4.1.4.2 Réorganisation des données dans la table.....	109
4.2 Résultats graphiques	110

4.2.1 Validation préliminaire des résultats de DEMONII.....	113
4.2.2 Application au bassin versant de la rivière Samson.....	120
4.2.2.1 Problèmes à l'exécution de DEMONII.....	120
4.2.2.2 Description des essais	126
4.2.2.3 Résultats graphiques pour la rivière Samson	127
4.3 Calcul de débit.....	151
4.3.1 Coefficients de Manning	152
4.3.2 Temps de concentration.....	153
CONCLUSION	160
BIBLIOGRAPHIE	166
ANNEXES	174

LISTE DES TABLEAUX

Tableau 2.1	Caractéristiques des modèles	48
Tableau 2.2	Modèles satisfaisant aux quatre premiers critères.....	51
Tableau 3.1	Facteurs influençant l'hydrogramme d'une précipitation	59
Tableau 3.2	Description des groupes de sol (McCuen, 1982)	64
Tableau 3.3	Classement selon le taux d'infiltration (McCuen, 1982)	65
Tableau 3.4	Coefficient de ruissellement pour le complexe hydrologique de couverture du sol (McCuen, 1982).	67
Tableau 3.5	Limites de pluie (McCuen, 1982)	68
Tableau 3.6	Ajustement du CN en fonction des conditions d'humidité antécédentes (McCuen, 1982)	68
Tableau 3.7	Composantes de la base de données du système d'information géographique (Tecsult, 1993a)	70
Tableau 3.8	Étude de modélisation du bassin versant de la rivière Chaudière. Clef d'interprétation des coefficients de ruissellement (CN) (Tecsult, 1993a).....	72
Tableau 3.9	Base de donnée du système d'information géographique pour obtenir le coefficient de ruissellement (CN).....	75
Tableau 3.10	Classes des cartes numériques.....	75
Tableau 3.11	Première matrice de superposition.....	77
Tableau 3.12	Deuxième matrice de superposition	79
Tableau 4.1	Matrice d'élévation du plan incliné	115
Tableau 4.2	Matrice d'élévation du biplan à double inclinaison	115

Tableau 4.3	Matrice des surfaces totales d'apport (TCA) pour le plan incliné	118
Tableau 4.4	Matrices des surfaces totales d'apport (TCA) pour le biplan à double inclinaison	118
Tableau 4.5	Volume total d'apport.....	150
Tableau 4.6	Coefficients de Manning utilisés pour la modélisation	153
Tableau 4.7	Temps de concentration obtenus du programme	154
Tableau 4.8	Temps de concentration obtenus par calcul manuel.....	155

LISTE DES FIGURES

Figure 1.1	Niveau d'abstraction des données dans les bases de données géographiques.	13
Figure 1.2	Comparaison entre les modèles vectoriel et matriciel de données. a) Réalité. b) Modèle vectoriel de la réalité. c) Modèle matriciel de la réalité.	14
Figure 1.3	Composantes fonctionnelles d'un système d'information géographique	17
Figure 1.4	Exemple de mosaïque régulière non structurée.....	22
Figure 1.5	Exemple de mosaïque régulière quelconque.....	22
Figure 1.6	Mosaïque hiérarchique, quadtree, avec sept niveaux de quad.	23
Figure 1.7	Exemple d'une structure de type liste inversée.....	26
Figure 1.8	Exemple d'une structure hiérarchique pour la base de données d'un parc national.....	26
Figure 1.9	Exemple d'une structure en réseau pour la base de données d'un parc national.....	28
Figure 1.10	Exemple d'une structure relationnelle pour une base de données d'un parc national.....	28
Figure 2.1	Localisation de la rivière Samson.	44
Figure 2.2	Hydrogramme type (Llamas, 1993).....	45

Figure 2.3	Deux idéologies en matière de lien entre un système d'information géographique et un modèle hydrologique (Stuart et Stocks, 1993)	52
Figure 3.1	Schéma du processus « précipitation-écoulement » (Llamas, 1993)	58
Figure 3.2	Relation entre la précipitation, le ruissellement et la rétention (McCuen, 1982)	62
Figure 3.3	Méthode SPANS pour le calcul du coefficient de ruissellement du SCS.....	78
Figure 3.4	Algorithme de résolution de la modélisation d'un petit bassin versant naturel et non jaugé.	81
Figure 3.5	Illustration des concepts de surface spécifique d'apport et de surface spécifique de dispersion d'un segment de courbe de niveau d'une longueur unitaire(Costa-Cabral et Burges, 1994).	83
Figure 3.6	Identification des points d'élévation sur une cellule.....	86
Figure 3.7	Organigramme du programme DEMON.	88
Figure 3.8	Organigramme de la fonction de calcul des surfaces totales d'apport et de dispersion: TCA_AND_TDA.....	89
Figure 3.9	(a) Exemple d'un écoulement bidimensionnel respectant l'orientation de la pente maximale des cellules. L'écoulement est généré sur la cellule (1,1). La surface en gris représente la STD de la cellule (1,1). (b) Matrice d'influence de la cellule (1,1). Le chiffre de chacune des cellules de la matrice d'influence représente la fraction de la cellule source qui est drainée par ces cellules. (c)	

	Signification physique de la valeur 0.58 de la cellule (4,3) de la figure (b) (Costa-Cabrall et Burges, 1994).	91
Figure 3.10	Illustration de la répartition de l'écoulement d'une cellule source dans les deux cellules voisines et cardinales (Costal et Burges, 1994).....	92
Figure 3.11	Représentation schématique des trajets de l'écoulement. Les embranchements correspondent à un écoulement réparti entre deux cellules voisines. Tous les trajets quittent le modèle d'élévation de terrain sous une forme concentrée indiquée par la lettre "C" (Costa-Cabral et Burges, 1994).....	93
Figure 3.12	Illustration des définitions des variables $F, f_1, f_2, f_3, x_1, x_2, x_3$ et I , utilisées pour la projection de l'écoulement à travers la cellule (4,2) (Costal et Burges, 1994).	96
Figure 4.1	(a) Format matriciel où chaque élément a_{mn} est une thématique. (b) Disposition de la matrice dans un fichier en code ASCII.....	106
Figure 4.2	Format de la table de points dans SPANS.	107
Figure 4.3	Équation de modélisation pour la commande <i>Model / Points / Grid Sample</i>	107
Figure 4.4	Schéma du processus du programme PREDEMON.....	111
Figure 4.5	(a) Plan incliné et (b) biplan à double inclinaison.	114
Figure 4.6	(a) Matrice des surfaces spécifiques d'apport (SCA) du plan incliné, (b) Matrice des surfaces spécifiques d'apport (SCA) du biplan à double inclinaison, (c) matrice des surfaces spécifiques de dispersion(SDA) du plan incliné et (d)	

	matrice des surfaces spécifiques de dispersion (SDA) du biplan à double inclinaison.....	116
Figure 4.7	Orientation des axes d'un plan cartésien vs une matrice.....	123
Figure 4.8	Topographie du bassin versant de la rivière Samson.....	129
Figure 4.9	Réseau hydrographique du bassin versant de la rivière Samson.....	130
Figure 4.10	Surfaces spécifiques d'apport (SCA) à une discrétisation de 200m x 200m, première série d'essais.	131
Figure 4.11	Surfaces totales d'apport (TCA) à une discrétisation de 200m x 200m, première série d'essais.....	132
Figure 4.12	Surfaces spécifiques de dispersion (SDA) à une discrétisation de 200m x 200m, première série d'essais.	133
Figure 4.13	Surfaces spécifiques d'apport (SCA) à une discrétisation de 100m x 100m, première série d'essais.	134
Figure 4.14	Surfaces totales d'apport (TCA) à une discrétisation de 100m x 100m, première série d'essais.....	135
Figure 4.15	Surfaces spécifiques de dispersion (SDA) à une discrétisation de 100m x 100m, première série d'essais.	136
Figure 4.16	Surfaces spécifiques d'apport (SCA) à une discrétisation de 100m x 100m, deuxième série d'essais.	137
Figure 4.17	Surfaces totales d'apport (TCA) à une discrétisation de 100m x 100m, deuxième série d'essais.	138
Figure 4.18	Surfaces spécifiques de dispersion (SDA) à une discrétisation de 100m x 100m, deuxième série d'essais.	139

Figure 4.19	Surfaces spécifiques d'apport (SCA) à une discrétisation de 50m x 50m, première série d'essais.....	140
Figure 4.20	Surfaces totales d'apport (TCA) à une discrétisation de 50m x 50 m, première série d'essais.....	141
Figure 4.21	Surfaces spécifiques de dispersion (SDA) à une discrétisation de 50m x 50m, première série d'essais.....	142
Figure 4.22	Surfaces spécifiques d'apport (SCA) à une discrétisation de 50m x 50m, deuxième série d'essais.....	143
Figure 4.23	Surfaces totales d'apport (TCA) à une discrétisation de 50m x 50m, deuxième série d'essais.....	144
Figure 4.24	Surfaces spécifiques de dispersion (SDA) à une discrétisation de 50m x 50m, deuxième série d'essais.....	145
Figure 4.25	Graphique des pentes des cellules pour une discrétisation de 100m par 100m.....	156
Figure 4.26	Graphique des pentes des cellules pour une discrétisation de 50m par 50m.	157

LISTE DES SIGLES ET ABRÉVIATIONS

ARC/INFO	Système d'information géographique
ASCII	American Standard Code for Information Interchange
AutoCAD	Logiciel de dessin assisté par ordinateur
CAD	Computer Asisted Design
CN	Curve Number
DAO	Dessin assisté par ordinateur
D-BASE	Logiciel de traitement de bases de données
DEM	Digital Elevation Model (Modèle numérique d'élévation de terrain)
DEMON	Programme de modélisation de ruissellement terrestre (Digital Elevation MOdél Network)
GIS	Geographical Information System
HEC	Hydrologic Engineering Center of the US Army Corp
HEC-1	Modèle hydraulique développé par le HEC
INRS	Institut National de Recherche Scientifique
InteraTYDAC	Compagnie qui produit le logiciel SPANS
LANDSAT	Nom d'un satellite
LDD	Local Drain Direction
MicroStation	Logiciel de cartographie informatisée
MTM	Projection cartographique : Modified Transverse Mercator
NAD27	North American Datum of 1927
OCR	Optic Character Reading
Oracle	Logiciel de traitement de bases de données
PC-RASTER	Système d'information géographique
RADARSAT	Nom d'un satellite
RTI	Réseau de triangles irréguliers

RHEA-CSU	Modèle de prévision des précipitations orographiques - Rhea, Colorado State University
SCA	Specific Contributing Area
SCS	Soil Conservation Service
SDA	Specific Distributing Area
SIG	Système d'Information Géographique
SPANS	Système d'information géographique (SPatial ANalysis System)
SSA	Surface spécifique d'apport
SSD	Surface spécifique de dispersion
STA	Surface totale d'apport
STD	Surface totale de dispersion
TCA	Total Contributing Area
TDA	Total Distributing Area
TIN	Triangular irregular network
TOPMODEL	Modèle de simulation hydrologique
UTM	Projection cartographique : Universal Transverse Mercator

LISTE DES ANNEXES

Annexe I	Copie de l'intégrale de l'article de Costa-Cabral et Burges	174
Annexe II	Listage du programme DEMON original.....	187
Annexe III	Paramètres de l'univers SPANS de la rivière Samson.....	224
Annexe IV	Listage du programme INITFICH.....	225
Annexe V	Listage du programme PREDEMON	228
Annexe VI	Listage des différentes déclarations de variables utilisées DEMONII utilisées pour les essais	238
Annexe VII	Listage des deux versions du programme DEMONII.....	241
Annexe VIII	Calcul de l'estimation du temps de concentration du bassin versant de la rivière Samson	278
Annexe IX	Exemple de calcul manuel de temps de concentration.....	280

INTRODUCTION

Dans la conjoncture actuelle du monde scientifique, la recherche, pour aboutir au progrès de la science ou à une certaine évolution des connaissances, semble emprunter deux avenues distinctes, mais tout à fait complémentaires et en interaction constante. L'une de ces avenues, la première, est instinctive et amène le progrès par le progrès. Chaque pas franchi dans une direction incite le second pas dans la même direction, une découverte dans un domaine spécifique entraîne et oriente les recherches dans le même domaine. Les connaissances deviennent de plus en plus pointues, de plus en plus spécialisées et participent à l'avancement de la science, avancement au sens propre et au sens figuré. Ce type d'avancement est au premier plan du développement de la science moderne. Il fixe la limite à atteindre, à supplanter, et le but ultime est de progresser, d'aller toujours plus loin. Il faut écraser les limites déjà fixées par d'autres, explorer des filons encore vierges, défricher.

Parallèlement à ce cheminement, se trouve une seconde avenue qui tente de relier et de mettre en commun les découvertes de la première. Le progrès par la symbiose. Il ne s'agit pas de devancer qui, ou quoi, que ce soit, mais de mettre à profit les théories, les technologies, les idées, les découvertes, les connaissances, le savoir-faire des uns avec ceux des autres. Le progrès sous cette forme n'est pas forcément une nouveauté absolue, mais la nouveauté provient de la relation

même qui naît entre, par exemple, deux connaissances ou deux technologies, et même entre une connaissance et une technologie. Cette pluridisciplinarité ouvre la porte à un progrès encore plus palpable, car elle permet à la science de s'infiltrer dans la vie quotidienne par le biais de l'innovation technologique ou, plus simplement, elle permet aux connaissances d'être appliquées et utilisées. Elle donne la vie aux sciences appliquées.

L'hydraulique et l'hydrologie sont de ces sciences appliquées. Même si le progrès auquel elles contribuent semble parfois tendre vers une spécialisation pure, il reste que leur développement est souvent fonction de l'avancement des sciences qui leur sont connexes ou interdépendantes. Plus particulièrement, le calcul de débits de rivière et la modélisation de bassins versants subissent une évolution, un recyclage, provoqué par la récente apparition, ou plutôt l'expansion, de la télédétection, des systèmes d'information géographique et de l'informatique en général. Ces innovations confèrent à l'analyse des ressources hydriques un nouveau souffle, une nouvelle inspiration. Des principes de modélisation et de calcul, jusqu'ici écartés à cause des difficultés de mise en oeuvre dues à la technologie, deviennent désormais réalisables. Des modèles relativement simples, mais nécessitant une base de données géographiques importante ou une compilation ardue des données en raison de leur quantité sont maintenant envisageables grâce entre autres à la télédétection et aux systèmes d'information géographique. La télédétection participe surtout à la

collecte des données. Les images satellitaires sont lues, interprétées puis intégrées dans le système d'information géographique où elles sont analysées, manipulées, dans un but précis, pour une utilisation spécifique. Le système d'information géographique est simplement un outil qui dote la base de données d'une nouvelle dimension: la dimension spatiale. Les données deviennent géoréférencées; on leur assigne une référence spatiale, et la télédétection est un moyen efficace d'établir une telle base de données géoréférencées.

De plus en plus, pour augmenter leur efficacité, les gouvernements fédéral et provinciaux, les municipalités régionales de comté (MRC) ainsi que différentes municipalités telles la Communauté urbaine de Montréal et Ville de Laval constituent et mettent régulièrement à jour une base de données de l'ensemble de leur territoire. Cette pratique ouvre la porte à une gestion plus efficace de leur territoire en accord avec les tendances pro-environnementales des dernières années.

Qui dit gestion du territoire dit gestion des ressources hydriques et toute bonne gestion se sert d'outils de prévision et de modélisation. C'est dans cette optique que se place cette étude. Comment peut-on utiliser ces bases de données pour modéliser les bassins versants d'un territoire quelconque et évaluer le débit d'une rivière pour prévoir les crues, les étiages et l'adduction en eau? Le domaine est vaste et l'étude ne prétend aucunement résoudre un problème de

calcul de débit de rivière. L'objectif est plutôt d'explorer une relation possible entre un système d'information géographique et un modèle de calcul de débit de rivière, de tirer profit des possibilités qu'offre un système d'information géographique pour les intégrer à l'intérieur d'une modélisation de bassin versant.

La réalisation du projet se fait par le biais de l'étude d'un bassin versant en particulier, celui de la rivière Samson. Cette rivière est un affluent de la rivière Chaudière situé en Beauce au nord-est du lac Mégantic, tout près de la frontière avec l'état du Maine. Ce bassin versant est naturel, c'est-à-dire sans développement urbain de grande importance. De petite dimension, il est considéré comme étant non jaugé pour qu'il soit le plus représentatif possible de la majorité des petits bassins versants du sud du Québec. Certaines informations sont tout de même disponibles sur les précipitations et les débits sur ce bassin versant et elles serviront lors de l'analyse des résultats. Le bassin versant de la rivière Samson représente un cas type qui s'insère bien dans l'objectif de créer un outil de gestion de bassin versant pour les bases de données des municipalités régionales de comté (MRC) ou autres organismes pour lesquelles les études traditionnelles de modélisation de bassin versant sont difficilement accessibles faute de budget.

Le système d'information géographique (SIG) est choisi pour des raisons d'ordre monétaire, de disponibilité des ressources et de base de données. Il s'agit du logiciel SPANS (*SPatial ANalysis System*) de la compagnie InteraTYDAC opéré sur le système d'exploitation OS/2. Les ministères fédéraux Parcs Canada et Pêches et Océans Canada, les ministères provinciaux de l'Environnement, des Ressources naturelles, de la sécurité publique ainsi que l'INRS-Eau (Institut national de la recherche scientifique - Eau) et plusieurs autres organismes possèdent déjà de bonnes bases de données établies à l'aide de ce logiciel.

Beaucoup de gens connaissent les systèmes d'information géographique par l'entremise de leurs différentes applications concrètes, mais l'idée qu'ils s'en font manque souvent de précision et de détails techniques. Le chapitre premier se veut une exploration du monde des systèmes d'information géographique, de leurs définitions, de leur principes de base et de leur fonctionnement. Il examine les caractéristiques du système SPANS ainsi que les différentes applications des systèmes d'information géographique dans le domaine des ressources hydriques.

Après une définition détaillée du cadre de l'étude, le chapitre deux aborde le problème du choix du modèle de calcul de débit de rivière parmi les modèles hydrologiques existants. Ce choix est éclairé par une étude de compatibilité entre les modèles dénombrés et le système d'information géographique. La

plupart des modèles existants ne sont pas faits pour être adaptés aux technologies d'aujourd'hui. Ils ont plutôt été astucieusement conçus pour esquiver les carences technologiques d'hier. Il est donc plus intéressant de regrouper une modélisation des processus hydrologiques et un modèle d'écoulement tous deux basés sur l'utilisation des données géomorphologiques pour construire une nouvelle modélisation. Le chapitre troisième étudie la modélisation des différents processus, retient et présente l'utilisation de la méthode pluie-ruissellement du SCS et explique comment le système SPANS peut intervenir dans cette modélisation.

La modélisation du ruissellement terrestre est prise en charge par le programme DEMON. Le fonctionnement de ce programme ainsi que les modifications qui lui sont apportées pour le mettre en relation avec les données fournies par le système SPANS font l'objet de la deuxième partie du chapitre 3. Au chapitre 4, les résultats de cette union sont présentés et analysés après un exposé sur la collecte des données et les difficultés survenues en cours d'exécution. Ce dernier chapitre est suivi d'une conclusion sur le travail réalisé puis des annexes présentant différents calculs et les listages des programmes.

CHAPITRE 1

LES SYSTÈMES D'INFORMATION GÉOGRAPHIQUE

La modélisation de bassin versant présentée dans cet ouvrage constitue un moyen de mettre à profit les bases de données géographiques dont se dotent de plus en plus les organismes gouvernementaux et para-gouvernementaux. Elle est une application d'un système d'information géographique, issue de l'ère des données géoréférencées. Le système d'information géographique est un gestionnaire de bases de données, mais il est davantage, car il permet d'accéder à une nouvelle dimension de l'information: la dimension spatiale. Ce chapitre dresse un portrait des systèmes d'information géographique; leur fonctionnement, leurs principes de base, la façon dont ils structurent les données et l'acquisition des données par ces systèmes. Les notions sont générales, mais utiles et suffisantes pour faire la lumière sur ce concept et évaluer la portée de tels systèmes. Cette démarche s'inspire d'un ouvrage exhaustif, en deux volumes, sur les systèmes d'information géographique de Maguire *et al.*, *Geographical Information Systems: Principles and Applications*, publié en 1991.

La deuxième section du chapitre porte sur le système d'information géographique qui a été retenu pour réaliser l'étude et les motifs qui ont justifié ce choix. Les différentes caractéristiques du système choisi y sont examinées

pour ensuite jeter un coup d'oeil sur l'utilisation des systèmes d'information géographique dans le domaine des ressources hydriques.

1.1 Principes de base des systèmes d'information géographique

1.1.1 Définition des systèmes d'information géographique

Nés il y a une trentaine d'années, les systèmes d'information géographique se sont développés et ont évolué très rapidement, surtout durant les dix dernières années grâce aux progrès technologiques. Leur origine récente et leur progression rapide rendent difficile l'établissement d'une définition des systèmes d'information géographique. Cependant, l'utilisation du terme « système d'information géographique » pour exprimer plusieurs réalités différentes, telles la technologie informatique liée à la géographie, les systèmes pour faire de la gestion intégrée des données ou tout simplement une nouvelle discipline contribue à compliquer davantage cette définition.

Au cours de leur travaux, plusieurs auteurs ont défini, à leur façon et selon l'orientation de leur recherche, la perception qu'ils ont des systèmes d'information géographique. Voici quelques-unes de ces définitions tirées de Maguire *et al.* (1991):

1. Tout ensemble de procédures manuelles ou basées sur l'utilisation de l'ordinateur pour emmagasiner et manipuler des données à références géographiques.
2. Un outil puissant pour la cueillette, l'emmagasinage, l'extraction et l'affichage des données spatiales du monde réel.
3. Une forme de gestion de l'information qui permet un affichage cartographique de l'information générale.
4. Un système pour la cueillette, l'emmagasinage, la manipulation, l'analyse et l'affichage de données géographiquement référées à la Terre.
5. Un système d'aide à la décision impliquant l'intégration de données à références spatiales dans un environnement de résolution de problèmes.

Toutes ces définitions sont vastes et générales, mais elles englobent les trois composantes clefs des systèmes d'information géographique. La première composante, la technologie, regroupe tout le matériel informatique et les logiciels nécessaires à l'exploitation et l'utilisation d'un système. La base de données, géographiques ou connexes, constitue la deuxième composante et le coeur des systèmes. Chaque définition comporte une mention, plus ou moins explicite, sur les deux dimensions de l'information, concernant les données, utilisées par les systèmes d'information géographique. La première dimension est géographique et la seconde est l'attribut. Vu de l'angle opposé, à chaque donnée est associée une référence géographique. Une référence géographique

est une coordonnée de latitude et de longitude, ou dans une projection terrestre quelconque, qui permet de situer un point sur la terre. Pour les systèmes d'information géographique, c'est la dimension géographique de l'information qui est la plus importante.

L'organisation de l'information dans les systèmes d'information géographique permet de faire la distinction entre ceux-ci et les autres systèmes de cartographie informatisée, de dessin assisté par ordinateur, de gestion de base de données et de télédétection. L'organisation de l'information donne la possibilité de faire de l'analyse spatiale, ce que les autres systèmes ne peuvent faire d'une façon efficace. Faire de l'analyse spatiale, c'est générer de l'information en questionnant la base de données. Pour faire de l'analyse spatiale, on doit pouvoir poser des questions au système. Maguire *et al.* (1991), définit les six types de questions suivantes qu'il est possible de poser à un système d'information géographique.

1. Question de localisation: Que trouve-t-on à ...?
2. Question de condition: Où se trouve ...?
3. Question de tendance: Quel est le changement ...?
4. Question de cheminement: Quelle est la meilleure façon de ...?
5. Question de motif: Pour quelles raisons...?
6. Question de modélisation: Qu'obtient-on si...?

La question sur la localisation nécessite le questionnement, l'interrogation d'une base de données afin de déterminer le type de caractéristiques existant en un point donné. Un exemple de ce type de question est « Quelle est la population pour telle étendue lors de tel recensement? »

Le type de question sur la condition propose une logique contraire à la question de localisation. Par exemple, on peut rechercher la localisation d'un site ayant des caractéristiques précises au moyen de la question suivante: « Où sont toutes les terres recouvertes de forêt à moins de 200 mètres d'une route? » est un exemple du type de question de condition.

Quant à la question de tendance, elle nécessite un contrôle des changements dans le temps. La question pourrait être: « Quelle est la variation de débit de véhicules sur telle route? ».

Les autres questions sont beaucoup plus complexes et requièrent une analyse spatiale poussée. Une question sur un itinéraire demande, en effet, des calculs pour les routes les plus rapides, les plus courtes ou les plus panoramiques entre deux points. Les questions d'itinéraire sont du type: « Quelle est la clinique médicale la plus près? ».

Les questions à propos des motifs permettent aux spécialistes en environnement ou en sociologie et aux planificateurs de décrire et de comparer les phénomènes

de distribution. Ils peuvent ainsi comprendre certains processus qui sont basés sur ces distributions. Une question type ressemble à celle-ci. « Existe-t-il un motif, une raison à la distribution des décès, plus précisément, une surface dans la distribution où les décès ont pu être causés par une trop grande exposition à des radiations? »

Il existe actuellement sur le marché peu de systèmes qui permettent de résoudre chacune des questions énumérées. Le choix judicieux d'un système d'information géographique se fait sur la base de certains compromis.

Finalement, l'infrastructure, le personnel, les éléments de support et les installations, sans lesquels l'exploitation et l'utilisation est impossible, forment la troisième et dernière composante. L'examen des différentes composantes des systèmes d'information géographique constitue en quelque sorte une définition assez complète qui laisse place à l'interprétation de l'étendue et de la possibilité des systèmes.

1.1.2 Fonctionnement et concepts de base

Les données intégrées dans un système d'information géographique, qu'on nomme caractéristiques géographiques, sont constituées d'un élément géographique qui sert de référence à l'élément attribut. Elles représentent différents aspects de la réalité et deviennent donc une abstraction de cette réalité

qui s'établit sur trois niveaux distincts (figure 1.1). Le premier niveau d'abstraction, les modèles conceptuels, revêt une forme vectorielle ou matricielle (figure 1.2).

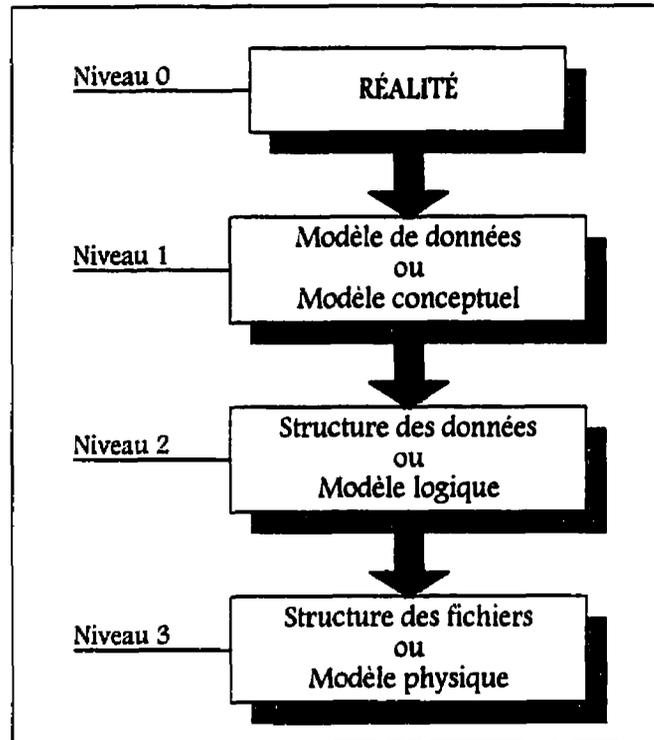


Figure 1.1 Niveau d'abstraction des données dans les bases de données géographiques.

Dans un modèle vectoriel, les caractéristiques géographiques, représentées par une série de coordonnées (x,y) ou (x,y,z) , sont de type objet, car chaque entité, composée d'un seul vecteur ou d'un ensemble de vecteurs, est associée à un objet. Par exemple, à la figure 1.2, il faut quatre vecteurs pour représenter l'objet « bâtiment ».

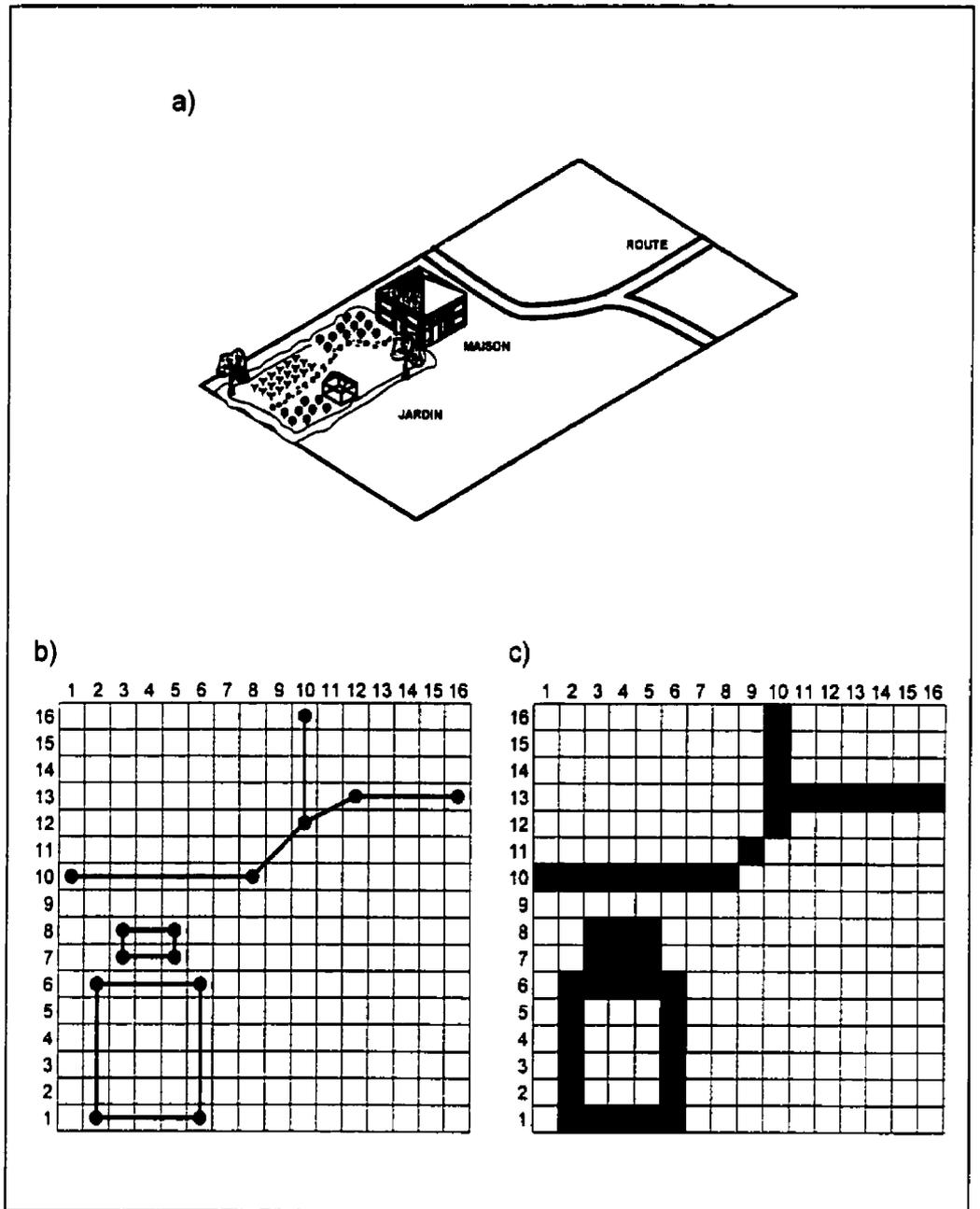


Figure 1.2 Comparaison entre les modèles vectoriel et matriciel de données.
 a) Réalité. b) Modèle vectoriel de la réalité. c) Modèle matriciel de la réalité.

Dans le modèle matriciel, les caractéristiques géographiques sont décrites comme étant les unités polygonales d'une matrice. Chaque caractéristique occupe l'espace d'un polygone quelconque dans une matrice. En général, les unités polygonales sont des carrés réguliers et correspondent à un pixel. Il existe cependant d'autres types d'unités polygonales tels les triangles réguliers et irréguliers ainsi que les hexagones. Le modèle matriciel est de type surfacique, car l'emphase est mise sur le contenu des surfaces plutôt que sur les frontières entre elles. La plupart du temps, les mots mosaïque et matrice sont utilisés comme des synonymes. Strictement parlant, une mosaïque est une structure de données géographiques et elle est le seul moyen d'implémenter le modèle de données géographiques des matrices dans un système d'information géographique.

Les autres niveaux d'abstraction de la figure 1.1 sont moins conceptuels et plus reliés à la réalité informatique. La structure des données (niveau 2) est spécifique à chaque système d'information géographique et constitue la méthode d'intégration des données dans le système d'information géographique. Chaque modèle de données possède une ou plusieurs structures qui lui est propre. Les différentes structures de données sont abordées un peu plus loin dans le chapitre.

La structure des fichiers qui emmagasinent l'information est le dernier niveau d'abstraction de la réalité et le plus technique des trois. Tous les systèmes d'information géographique utilisent et génèrent des fichiers selon une structure spécifique des plus rigoureuses. En général, un système se sert de plusieurs structures de fichiers selon le type de données. Certaines structures sont en code ASCII, elles sont accessibles et utilisées pour transférer des données entre systèmes. Les structures en code binaire sont internes au système et leur accès, pour modification ou consultation, est restreint aux possibilités d'édition de fichiers qu'offrent les commandes du système.

1.1.2.1 Fonctionnement d'un système d'information géographique

Un système d'information géographique offre la possibilité d'opérer un certain nombre d'actions sur des données composées de deux éléments, un géographique et un attribut. L'ensemble de ces actions, qui en compose le fonctionnement, se résume en dix catégories d'actions, nommées composantes fonctionnelles et illustrées à la figure 1.3. L'exploration de chacune d'elles confère une bonne notion du déroulement d'une utilisation type des systèmes d'information géographique. Les composantes sont abordées ici en suivant une progression logique, mais pas nécessairement chronologique; de la saisie de données à la présentation des résultats en passant par le transfert de données, la validation et l'édition des données, l'emmagasinage et la structure des données,

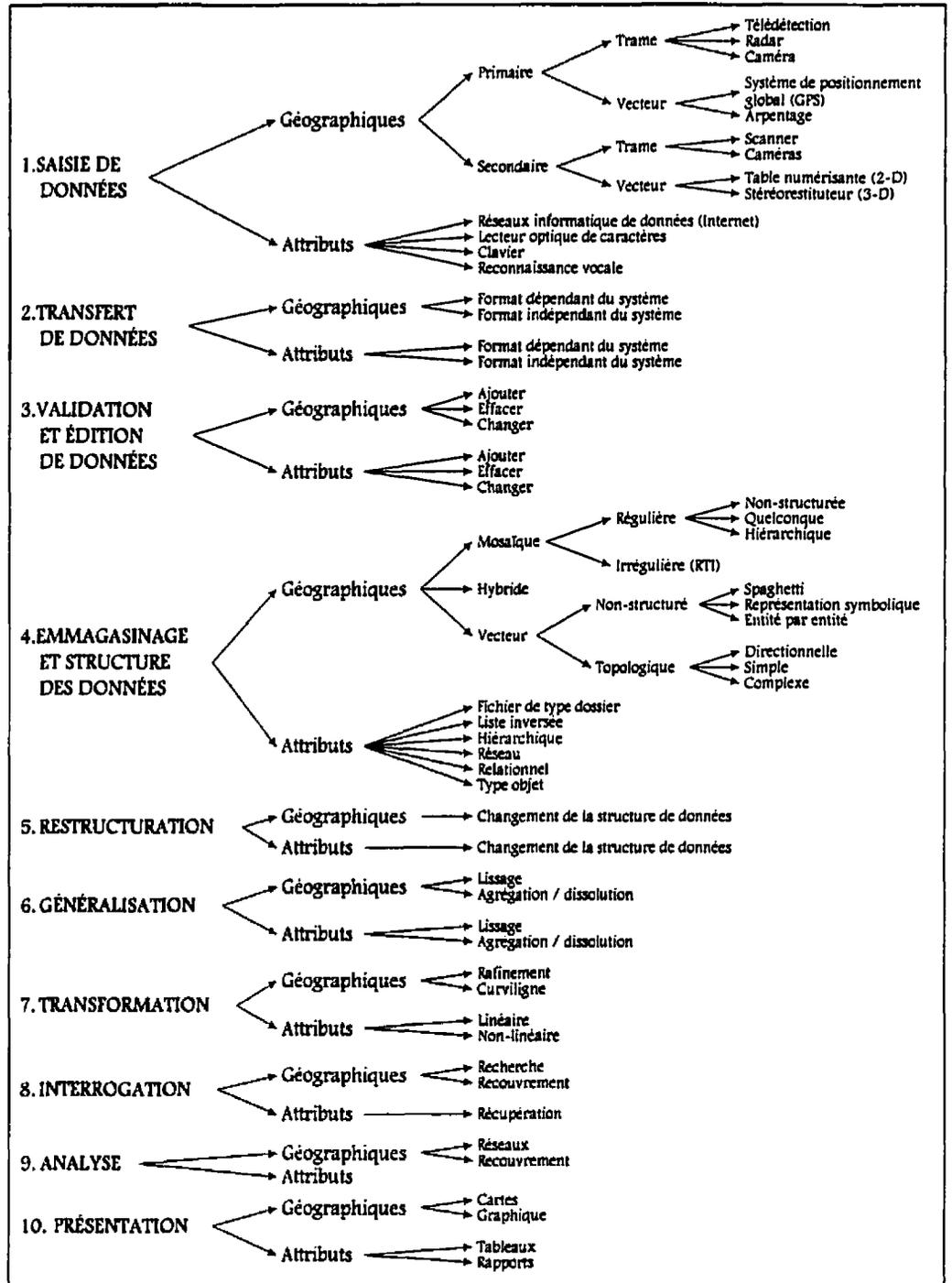


Figure 1.3 Composantes fonctionnelles d'un système d'information géographique

la restructuration, la généralisation, la transformation, l'interrogation, et finalement l'analyse des données. Pour chacune d'elles, on examine la façon dont s'applique la composante à l'élément géographique ainsi qu'à l'élément attribut des données.

La saisie, le transfert, la validation et l'édition concernent l'acquisition et l'enregistrement des données numériques dans le système. La saisie des données géographiques est soit primaire, lorsqu'elle s'effectue directement à partir de la réalité, soit secondaire, lorsqu'elle se fait à partir d'une représentation de la réalité comme à partir d'une carte topographique, d'une photographie aérienne ou d'une image satellite. Les instruments de saisie sont, pour la saisie primaire de données, le radar, les caméras, la télédétection par images satellites, les systèmes de positionnement global et l'arpentage. De plus en plus, et surtout depuis le lancement, à l'automne 1995, du premier satellite radar RADARSAT, la télédétection s'impose comme une méthode des plus efficaces pour obtenir rapidement des données précises et variées. Le traitement et l'interprétation des images satellites, bien maîtrisés par les spécialistes de télédétection, donnent des résultats impressionnants. Pour la saisie secondaire des données, on utilise plutôt les scanners, les caméras, les tables numérisantes ainsi que les stéréorestituteurs (3-D).

La saisie des données attributs, à l'image de la variété des attributs, peut s'effectuer d'une multitude de façons. Les méthodes les plus courantes sont les réseaux informatiques de données, comme l'Internet, ou l'entrée directe à partir du clavier. Le lecteur optique de caractères (OCR) et la reconnaissance vocale peuvent aussi servir à recueillir des données, mais leur technologie pointue les rend moins accessibles.

Le transfert de données implique la migration de données, déjà saisies dans un système quelconque, vers le système utilisé en se servant des réseaux électroniques ou des médiums magnétiques; disquettes, cassettes, disques compacts, etc. Le transfert peut se faire, pour les deux éléments des données, dans un format dépendant du système ou indépendant du système. Le format indépendant du système le plus utilisé est le *American Standard Code for Information Interchange*, c'est-à-dire le code ASCII. Par contre, lorsque les données existent dans un format propre à un système en particulier, ARC/INFO, AUTOCAD ou MicroStation par exemple, les formats dépendent des systèmes et les données doivent être exportées d'un système à l'autre.

Peu importe si les données ont été saisies ou transférées, elles doivent être validées et, si nécessaire, éditées. Les systèmes de gestion de base de données (SGBD), ainsi que les éditeurs standards d'ordinateur, sont habituellement utilisés pour exécuter ce travail. Souvent, pour vérifier la validité des données,

la carte contenant les données est imprimée puis comparée avec la carte originale.

L'emmagasinage et la structuration des données constituent une étape cruciale dans la création d'une base de données d'un système d'information géographique, car elles ont une répercussion sur la gamme de fonctions qui serviront, par la suite, à transformer, interroger et analyser les données. Ainsi, lors d'un projet, il est impératif de bien cibler le genre d'information nécessaire ainsi que les traitements qui seront effectués sur les données afin d'établir, avant de commencer la saisie des données, la ou les structures de données qui s'imposent pour la réalisation du projet. Le choix d'une structure de données implique aussi le choix d'une catégorie de systèmes d'information géographique disponible car, en général, un système est basé sur une seule structure. Les données, surtout leur élément géographique, ne peuvent subir les mêmes traitements, les mêmes transformations, les mêmes interrogations ni les mêmes analyses selon leur structure et les résultats escomptés s'avèrent inaccessibles si un mauvais choix de structure est fait.

Dans la représentation de la réalité, la structure des données se situe au deuxième rang de l'abstraction, juste derrière les modèles de données abordés à la section précédente. Les structures servent à implémenter les modèles de données dans un système et elles sont étroitement liées à ces modèles. Par

conséquent, le choix d'une structure découle de la représentation la plus appropriée de la réalité, entre le modèle vectoriel ou le modèle matriciel.

Lorsque le modèle est matriciel, la structure qui s'impose est la mosaïque régulière ou irrégulière. Les mosaïques régulières sont non structurées dans le cas des grilles simples (figure 1.4) et quelconques lorsque le quadrillage n'est pas orthogonal (figure 1.5). Les mosaïques hiérarchiques, ou quadtree, sont des quadrillages orthogonaux où chaque carré se subdivise à répétition en quatre carrés pour atteindre le niveau de précision requis par la densité des données (figure 1.6). Chaque degré de précision de la mosaïque est appelé niveau de quad. Cette structure est très économique pour la mémoire et la vitesse d'exécution des ordinateurs, car elle permet d'atteindre un degré de précision intéressant là où l'information est dense et, lorsque l'information devient plus uniforme, le niveau de quad rétrograde. À l'opposé, pour une structure à quadrillage orthogonal non structuré, le plus haut niveau de précision de l'information impose le degré de précision de toute la grille.

Parmi les mosaïques irrégulières, les réseaux de triangles irréguliers (*Triangular Irregular Network, TIN*) sont les plus courants. Ils servent souvent à la construction d'un modèle numérique d'élévation de terrain à partir d'une série de points dont l'attribut est l'élévation.

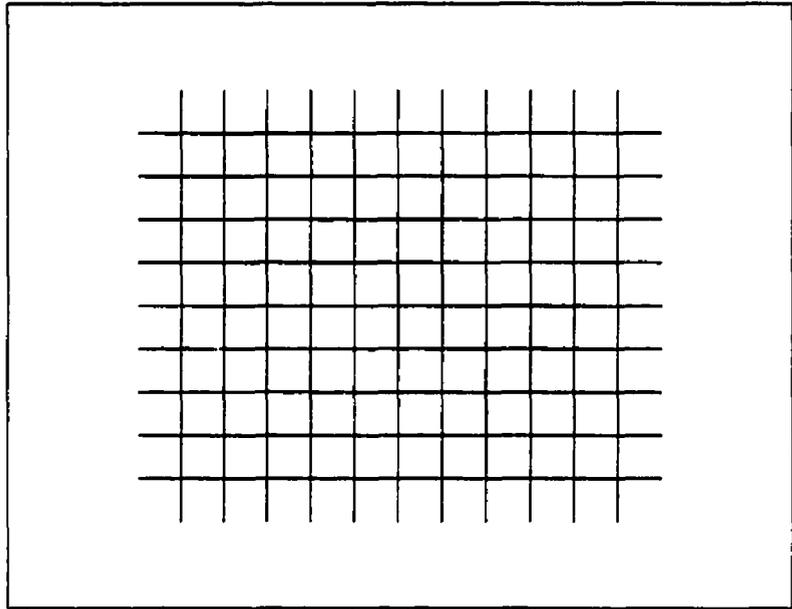


Figure 1.4 Exemple de mosaïque régulière non structurée

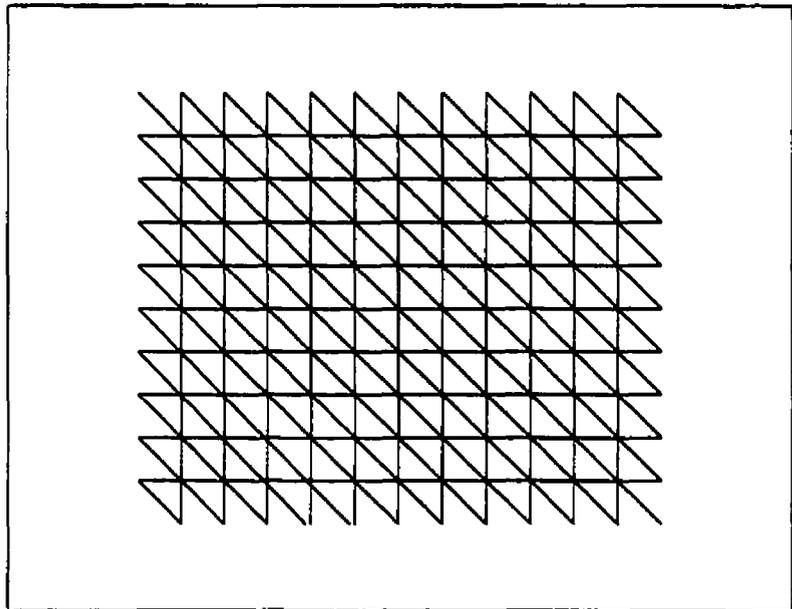


Figure 1.5 Exemple de mosaïque régulière quelconque.

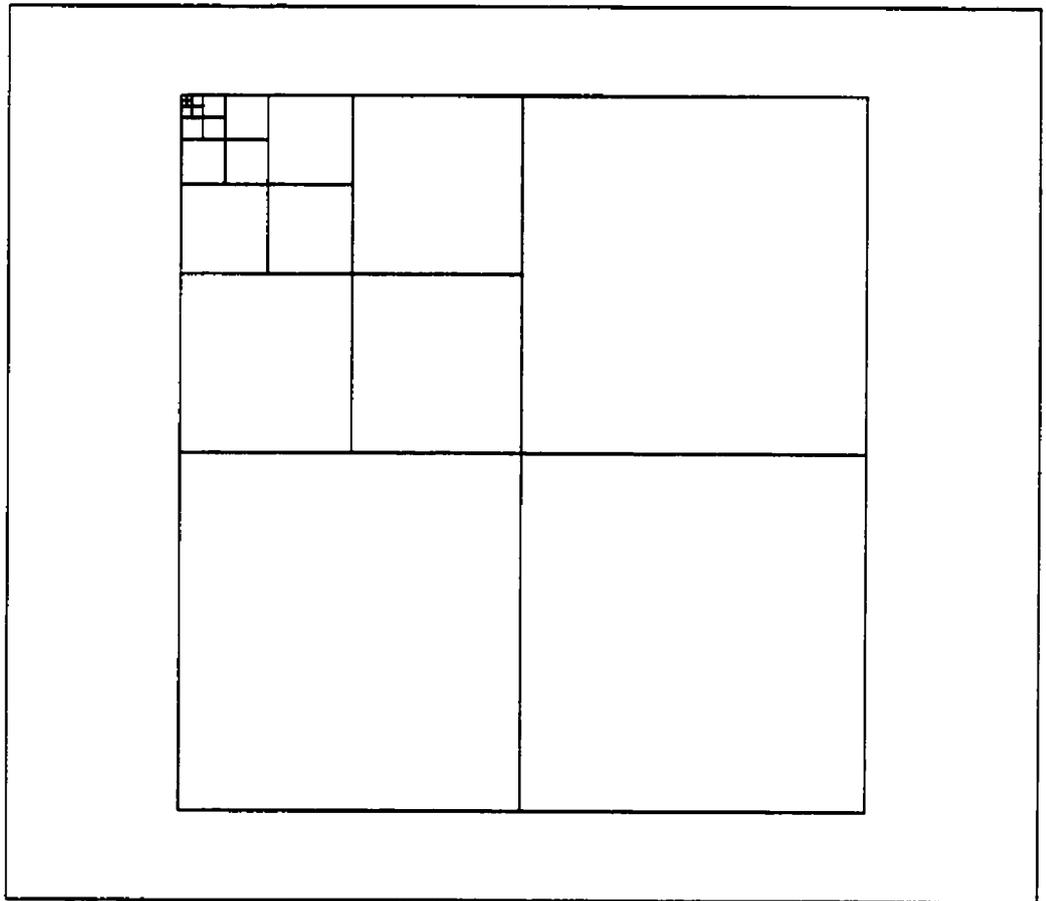


Figure 1.6 Mosaïque hiérarchique, quadtree, avec sept niveaux de quad.

Les structures de vecteurs, implémentant les modèles vectoriels, se divisent en deux types: le type structuré et le type topologique. Le premier type se compose de la structure "spaghetti", qui est simplement une accumulation de points et de lignes, de la représentation symbolique, pour laquelle les éléments de base sont des symboles représentant des édifices, des routes, des feux de circulation, etc. pouvant être déplacés de façon interactive et repositionnés à l'endroit approprié sur une carte. C'est le principe des "Design assisté par ordinateur", (DAO ou

CAD en anglais). La structure entité-par-entité, appartenant aussi au type non structuré, codifie les caractéristiques géographiques en tant qu'unités complètes tel un polygone fermé. Cependant, aucune information topologique n'est incluse dans les données.

La topologie est vue comme une structure lien-noeud entre les vecteurs et fait référence à la capacité des systèmes d'information géographique d'agencer et de mettre en relation des vecteurs pour former des entités tout en identifiant leur voisinage, c'est-à-dire la position relative des entités par rapport aux segments de lignes; à gauche ou à droite. Lorsqu'on bâtit une topologie simple dans un fichier de vecteurs, on crée un ensemble de polygones positionnés les uns par rapport aux autres. Si des polygones sont inclus dans d'autres polygones, c'est le cas des îles dans un lac par exemple, la topologie est dite complexe. La topologie directionnelle établit des réseaux de vecteurs à partir desquels on peut identifier le chemin le plus court vers un point quelconque. Les cartes de métro sont des exemples de topologie directionnelle.

La structure hybride est un mélange de la structure mosaïque et de la structure vecteur. Elle a été créée pour tirer les avantages de chacune des structures.

Généralement, les bases de données géographiques sont construites séparément des bases de données attributs, puis associées l'une à l'autre dans le système d'information géographique. Il est donc possible et avantageux de construire la

base de données attributs dans un gestionnaire de base de données. Il existe plusieurs structures de base de données attributs. Voici une définition des principales structures.

La structure de liste inversée est un tableau fait de colonnes et de rangées qui décrit quelles entités possèdent chacune des caractéristiques. Dans un système de liste normale, le tableau décrit plutôt les caractéristiques de chacune des entités. La figure 1.7 donne un exemple de liste inversée.

La structure hiérarchique définit un ensemble d'entités comme étant la racine de l'arbre hiérarchique puis un ensemble de pointeurs de type parent-enfant, c'est-à-dire d'un niveau supérieur vers un niveau inférieur, pour représenter les liens entre différents types d'entités. Chaque lien conduit à un niveau inférieur jusqu'à ce qu'on atteigne la racine (figure 1.8). Dans ce genre de structure, la relation de plusieurs à plusieurs ne peut être représentée.

Dans les réseaux, par opposition aux hiérarchies, chaque « enfant » a plusieurs « parents ». Les relations entre les entités ou les ensembles d'entités sont représentées par des liens de type pointeur. Des clefs établissent des relations entre les niveaux d'information et permettent, à partir d'un niveau d'information quelconque, de pointer vers n'importe quel autre niveau qui reconnaît la clef spécifiée par la recherche. La représentation des relations dans

# de la caractéristique	# de l'entité
1	3
	4
2	2
	4
3	1
	2
4	1
	3
5	3
	4

Figure 1.7 Exemple d'une structure de type liste inversée.

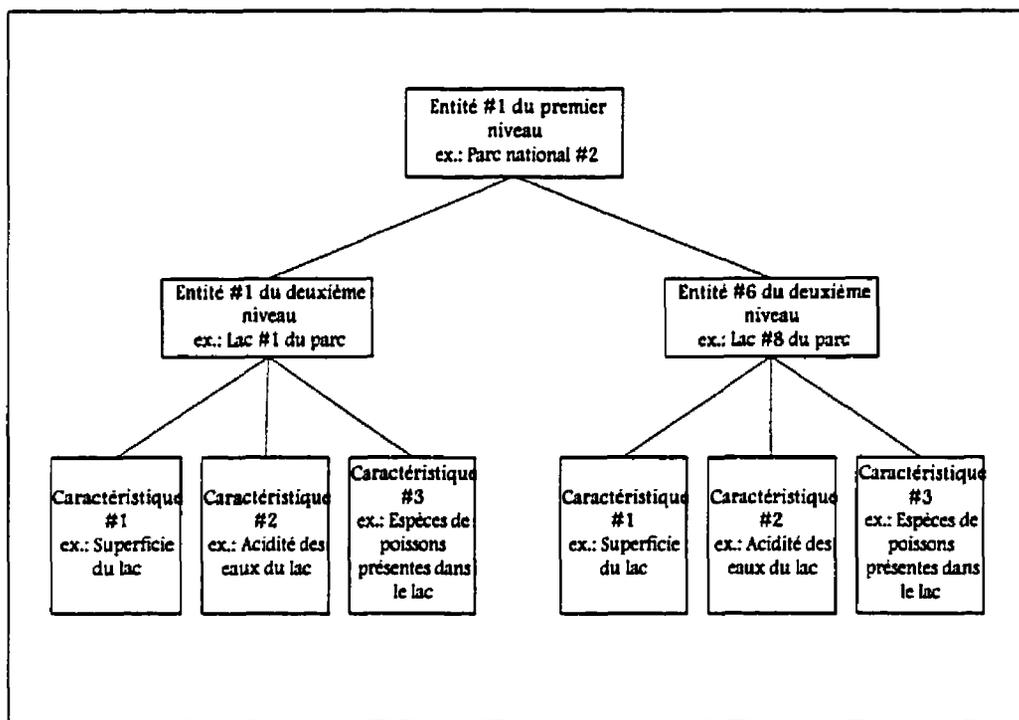


Figure 1.8 Exemple d'une structure hiérarchique pour la base de données d'un parc national.

une base de données construite selon cette structure aboutit à un nombre impressionnant de pointeurs (figure 1.9).

La structure relationnelle est un moyen de décrire les données selon leur structure « naturelle ». Elle est simple, car toutes les données sont représentées par des tableaux formés de lignes et de colonnes et reliés entre eux, non pas par un pointeur ou un lien, mais par une clef qui constitue la principale caractéristique de cette structure. Chaque tableau de cette structure est relié aux autres par un joint relationnel. À la figure 1.10, les colonnes contenant les numéros d'entités et de caractéristiques sont les clefs, les joints relationnels, entre les tableaux. La souplesse de cette structure en fait sa popularité.

Les systèmes d'information géographique ne fonctionnent pas avec toutes ces structures ni tous avec la même structure. En général, un système est basé sur une structure en particulier, une pour les données géographiques et une pour les attributs, et gère moins bien les autres structures. Les systèmes qui se prétendent polyvalents le sont très rarement. La plupart du temps, le système est basé sur une structure et offre la possibilité de convertir les données à partir d'une structure quelconque vers la structure de base du système pour en faire la manipulation et l'analyse.

Les structures, tant des données géographiques que des attributs, sont intimement liées à la nature du projet à réaliser et orientent le choix du système

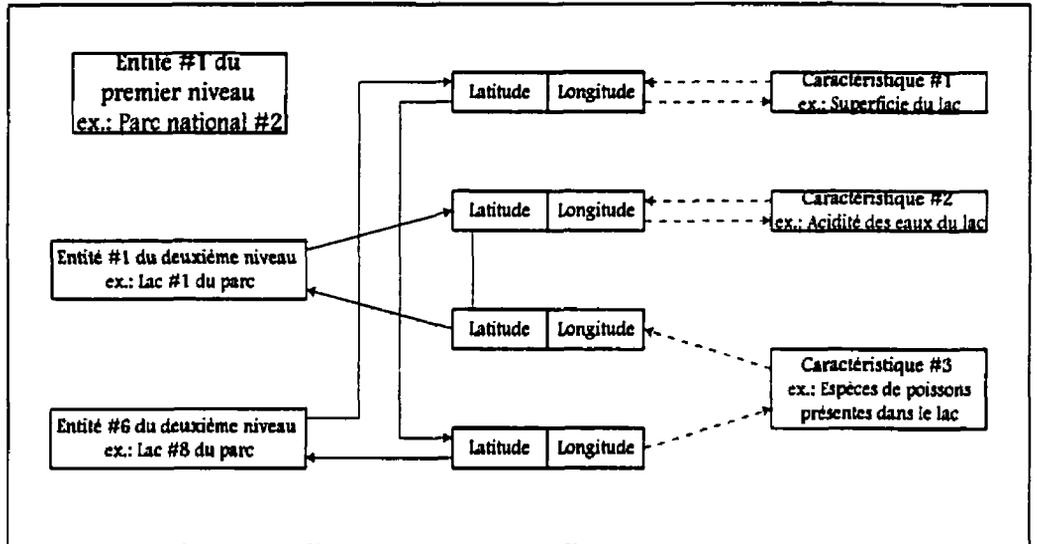


Figure 1.9 Exemple d'une structure en réseau pour la base de données d'un parc national.

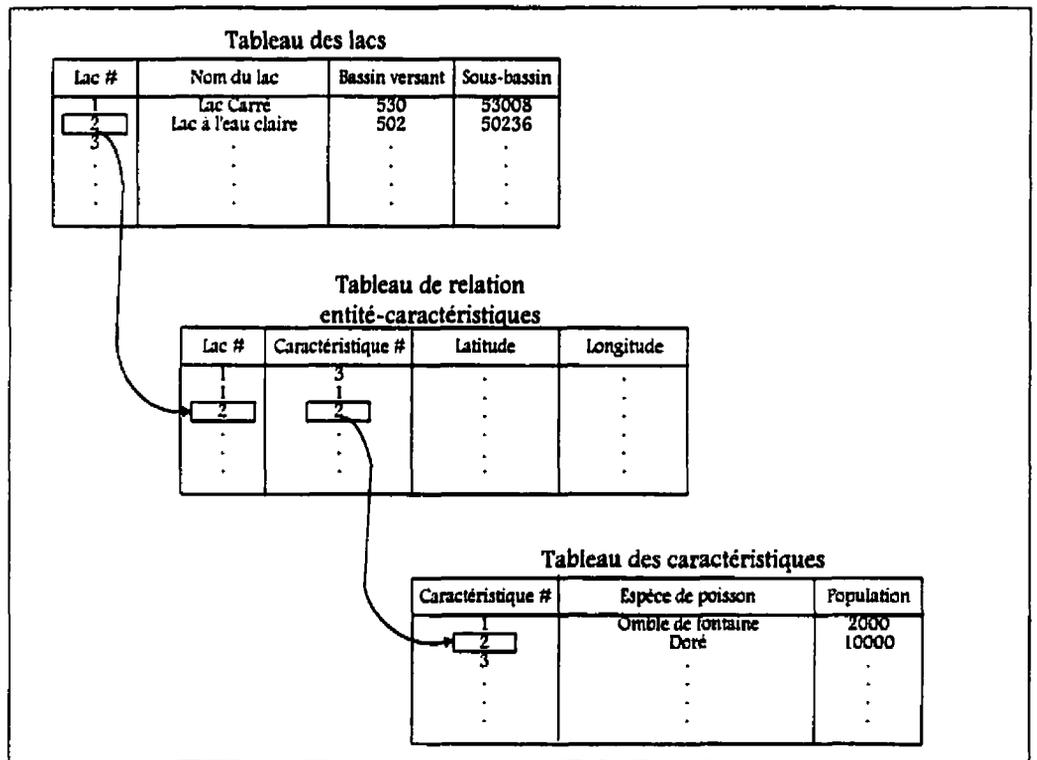


Figure 1.10 Exemple d'une structure relationnelle pour une base de données d'un parc national.

d'information géographique le plus approprié. Cependant, il arrive que la saisie des données ne peut se faire dans la structure choisie, soit parce que les outils de saisie nécessaires ne sont pas disponibles, soit parce que les données existent déjà dans une structure spécifique. Dans ce cas, par des fonctions de transformation qu'offrent certains systèmes, on peut convertir les données et modifier leur structure pour les adapter au système choisi. Il faut cependant être vigilant pour conserver une certaine précision des données ou pour ne pas créer une fausse précision.

La restructuration, la généralisation ainsi que la transformation sont des manipulations de données clés dans le fonctionnement d'un système d'information géographique, car elles permettent de prendre des données de sources disparates et de les convertir dans un format commun pour l'analyse de ces données. La restructuration implique un changement dans la structure des données. Elle consiste donc en un transfert d'une structure à une autre. La généralisation comprend le lissage et l'agrégation des données géographiques et attributs. Une des fonctions définies dans la généralisation consiste à dissoudre la frontière entre deux polygones adjacents pour former un autre polygone plus grand où l'attribut devient la moyenne des deux attributs antécédents.

La transformation des données géographiques se résume à la modification de l'échelle, à la rotation, la translation et l'inversion, ainsi qu'à la transformation

curviligne pour passer d'une projection cartographique à une autre. La transformation des attributs est une application d'une fonction linéaire ou non linéaire aux données, comme celle qu'on trouve dans les ensembles de base d'analyse statistique.

L'interrogation, et plus spécifiquement l'analyse des données, constitue le coeur des systèmes d'information géographique. C'est leur capacité d'analyser des modèles et des relations géographiques qui différencie un système d'information géographique d'un autre système de cartographie, de dessin assisté par ordinateur, ou autres. L'interrogation et l'analyse spatiale disposent d'opérateurs tels le voisinage, l'inclusion, la distance, la direction et la superposition pour générer de l'information supplémentaire à partir de l'information contenue dans les bases de données. Par exemple, il est possible à partir de la superposition des cartes de types de sols, de couvert végétal, de l'utilisation du sol et de la topographie, de calculer le coefficient de ruissellement (*Curve Number, CN*) d'un terrain. C'est par l'analyse spatiale qu'on peut « poser des questions » à la base de données géographiques et obtenir des réponses. Supposons qu'on dispose d'une carte du réseau routier avec les infrastructures et les services d'une municipalité, on peut interroger le système et obtenir, à partir d'une position précise, la localisation de l'hôpital le plus proche, la distance entre cet hôpital et la position initiale, ainsi que le trajet le plus court pour s'y rendre.

Une analyse spatiale d'une carte du réseau hydrographique d'une région donnée permet aussi de déterminer quelle portion du territoire est occupée par les lacs.

Ces deux exemples ne sont qu'une infime portion des applications des systèmes d'information géographique qui se multiplient très rapidement à mesure que les bases de données se développent.

1.2 Le système d'information géographique SPANS

Généralement, on choisit un système en fonction du modèle de données, matriciel ou vectoriel, qui sied aux exigences du projet à réaliser et qui correspond aux méthodes de saisie des données. La conversion des données d'un modèle à l'autre entraîne inévitablement une perte de précision de l'information qui peut être importante. Les systèmes matriciels conviennent davantage à la gestion, l'inventaire et l'analyse des ressources qui sont continues sur le terrain tels les types de sol et la végétation. Ils sont aussi plus appropriés lorsque la saisie des données se fait dans un format matriciel comme à partir d'une image satellite. Les systèmes vectoriels sont, pour leur part, mieux adaptés aux données pour lesquelles la localisation des objets est définie avec exactitude. C'est le cas des routes, des égouts et des frontières politiques.

Dans le cas présent, le système est en quelque sorte imposé pour des raisons économiques. Économie de temps, d'argent et de démarches. En effet, l'idée de

créer une application d'un modèle de calcul de débit de rivière dans un système d'information géographique fait suite à une étude réalisée pour le compte du ministère de la Sécurité publique. Une partie de cette étude devait déterminer si le développement anthropique pouvait être à l'origine d'un accroissement de l'occurrence des inondations dans les municipalités riveraines le long du tronçon des eaux mortes de la rivière Chaudière entre Saint-Maxime de Scott et Beauceville (Tecsult, 1993a). Il fallait donc déterminer la sensibilité du milieu à l'érosion et au ruissellement, en fonction du développement anthropique et des pratiques qui y sont associées, deux processus physiques qui peuvent être associés aux inondations de la rivière Chaudière.

Pour l'étude du bassin versant de la rivière Samson, nous disposons d'une base de données considérable déjà implémentée dans un système d'information géographique. Ce système a été développé par la compagnie canadienne InterATYDAC et se nomme SPANS, pour *SPatial ANalysis System*. Il est basé sur un modèle matriciel des données géographiques implémenté par une structure quadtree quoiqu'il offre une gestion appréciable des données en modèle vectoriel. Les données attributs y sont organisées selon la structure relationnelle décrite au chapitre précédent. Ces structures constituent des atouts dans le domaine des ressources naturelles où la télédétection et l'imagerie satellite occupent une place grandissante pour la saisie des données.

Au niveau des ressources hydriques, les données sont vectorielles dans le cas des limites des bassins versants et du réseau hydrographique, mais beaucoup de données, telles l'intensité de la pluie, le type de sol, la végétation, l'utilisation du sol, les cultures, la température et la topographie, sont distribuées et continues sur la surface terrestre, et le modèle matriciel les représente plus adéquatement (Leipnik, 1993). Comparativement aux autres systèmes d'information géographique sur le marché, SPANS est un des rares systèmes matriciels qui offrent une gestion des données vectorielles et le passage du modèle vectoriel au modèle matriciel, et vice-versa. De plus, la gamme de fonctions de modélisation qu'il possède est parmi les plus complètes chez les systèmes matriciels. Le système SPANS être celui qui convient le mieux au travail envisagé, d'autant plus que sa structure en quadtree atteint une précision qui peut aller jusqu'au mètre lors de la conversion de données vectorielles en données matricielles.

Peu importe le système qu'on choisit, il faut faire face aux inconvénients qu'il présente et s'adapter aux limites et aux possibilités qui lui sont inhérentes. L'essentiel est d'avoir un système dont l'utilisation est suffisamment répandue pour avoir la possibilité d'échanger de l'information et d'entrevoir la contribution du projet au développement des applications à ce système d'information géographique. L'utilisation du système SPANS au sein de la fonction publique fédérale et provinciale, et des groupes de recherche tel

l'Institut national de la recherche scientifique - Eau (INRS-Eau), lui confère un avantage certain.

Certaines limites, qui touchent de près le domaine des ressources hydriques, sont des dénominateurs communs pour presque tous les systèmes sur le marché. Il n'existe pas de systèmes qui fonctionnent vraiment en trois dimensions; ils émulent plutôt la troisième dimension en la considérant comme un attribut au même titre que le type de sol. Ils représentent les trois dimensions graphiquement, mais l'analyse spatiale se fait seulement en deux dimensions. La variation temporelle ou la dynamique des phénomènes est un autre concept qui n'est pas traité par les systèmes d'information géographique. On contourne le problème en présentant une série de « photos » des phénomènes prises à différents temps pour former une séquence (Leipnik, 1993).

1.3 Les systèmes d'information géographique et les ressources hydriques

On relève des exemples d'utilisation des systèmes d'information géographique dans tous les domaines. Brilly *et al.* (1993) affirment qu'après vingt ans de développement, les systèmes d'information géographique sont devenus des outils courants et presque indispensables dans les domaines tels la construction, les services gouvernementaux, la santé, le commerce, l'industrie, la politique, la lutte contre le crime, les services d'urgence et les transports.

Au niveau des ressources hydriques, la capacité des systèmes d'information géographique d'emmagasiner, d'analyser et de récupérer l'information requise par les différents modèles hydrologiques (Van Deursen et Kwadijk, 1993) contribue au développement de nouvelles applications, même si l'acquisition et la manipulation de données demeurent, pour l'instant, difficiles et coûteuses (DeVantier et Feldman, 1993). Dans leur article sur l'utilisation du système ARC/INFO pour la modélisation hydrologique, Warwick et Hanes (1994) ajoutent que « [...] a GIS is perfectly suited for continual updating and rapid re-computation of alternative scenarios as is typically required in water resources projects ». Voici donc quelques exemples d'utilisation des systèmes d'information géographique au niveau des ressources hydriques.

La rivière Gunnison au sud du Colorado est une source d'eau importante et fait l'objet d'étude et de modélisation hydrologique. La superficie du bassin est de 20 770 km² et se situe entre 1400 et 4400 mètres d'altitude. La variation du climat a beaucoup d'effet sur les ressources de ce bassin et la simulation de la distribution spatiale et temporelle des précipitations joue un rôle important dans l'utilisation des modèles de prévision hydrologique. Hay *et al.* (1993) ont couplé un système d'information géographique à leur modèle de prévision des précipitations orographiques (RHEA-CSU) pour automatiser le développement des modèles numériques d'élévation de terrain et obtenir une distribution spatiale de l'information, nécessaire pour exécuter le modèle.

De leur côté, Van Deursen et Kwadijk (1993) ont couplé un modèle conceptuel d'évaluation du bilan hydrique du bassin versant du Rhin avec le système d'information géographique PC-RASTER. Le modèle RHINEFLOW est conçu pour détecter les variations du bilan hydrique des différentes sections du bassin du Rhin en fonction des variations du climat. Pour contourner les problèmes de compatibilité des formats des données spatiales entre le système d'information géographique et le modèle et aussi pour faciliter les opérations, les chercheurs ont créé à l'aide des utilitaires de PC-RASTER une série de modules qui préparent les intrants et traitent les résultats. Les modules ainsi que le modèle RHINEFLOW sont écrits dans le langage du progiciel PC-RASTER et sont directement accessibles à partir de ce système. Les modules s'occupent de l'acquisition des données et de la production des cartes représentant la variation spatiale des données (TIMEINP), de la production de cartes supplémentaires par la modélisation arithmétique des cartes initiales (CALC), de la création de la matrice du schéma de drainage, de la matrice d'orientation locale du drainage (*Local Drain Direction*, LDD), à partir des données d'élévation (WATERSHED), de la propagation de l'eau en fonction de la matrice d'orientation du drainage, de l'exécution du modèle RHINEFLOW et finalement de la présentation des résultats du modèle sous forme de cartes thématiques et de tables pour chaque pas de temps calculé. L'approche est intéressante, mais peu de progiciels offrent la gamme d'utilitaires nécessaires pour implémenter un modèle de prévision

hydrologique et créer des modules de calcul directement accessibles dans le progiciel. En général, les langages de modélisation des systèmes ne sont pas aussi flexibles que les langages de haut niveau comme le C ou le FORTRAN. Ils ne permettent pas, par exemple, les constructions de boucles itératives et répétitives (Stuart et Stocks, 1993). Le système SPANS possède un langage de modélisation qui contient des constructions conditionnelles, mais aucune construction itérative ou répétitive.

Muzik et Chang (1993) ont créé leur propre système d'information géographique adapté à l'hydrologie qui peut effectuer la simulation d'écoulement avec un hydrogramme, générer des courbes de récurrence sur des bassins non jaugés, simuler un écoulement en utilisant un modèle agrégé basé sur la méthode du coefficient de ruissellement du *Soil Conservation Service* (SCS) ou en utilisant la méthode de l'hydrogramme unitaire adimensionnel, évaluer la performance des méthodes standard de la SCS et la modifier pour améliorer les résultats de la simulation, et finalement, évaluer la performance d'un hydrogramme unitaire adimensionnel utilisé pour la simulation d'un écoulement. La région étudiée se situe sur le versant est des montagnes Rocheuses en Alberta. Le lien entre le système et le modèle est idéal, car les fonctions du SIG répondent parfaitement au modèle. Par contre, la construction d'un système d'information géographique est une entreprise qui nécessite une

connaissance approfondie des SIG, beaucoup de ressources informatiques, surtout en programmation, et de nombreuses heures de travail.

La partie supérieure du bassin versant de la rivière Arkansas possède un passé minier et métallurgique qui continue d'affecter la qualité de l'eau et qui en limite l'utilisation. Kern et Stednick (1993) ont couplé un système d'information géographique à un modèle hydrochimique pour prédire et identifier les concentrations en métaux lourds dans l'eau de surface.

Stuart et Stocks (1993) présentent dans leur article un couplage entre le modèle semi-discrétisé TOPMODEL et le système SPANS GIS. L'intégration du modèle dans le système n'est pas possible, car la modélisation de SPANS ne permet pas les boucles itératives et récursives. Ils insistent sur le fait que jusqu'à ce que les systèmes d'information géographique deviennent plus flexibles, il est plus avantageux de coupler les modèles et les systèmes d'une façon autonome, c'est-à-dire en exécutant la modélisation hydrologique en parallèle au système d'information géographique.

Plusieurs chercheurs (Fedra, 1993; Walsh, 1993; Feldman, 1992; McKinney *et al.*, 1993) ne se contentent pas de coupler un modèle quelconque avec un système d'information géographique. Ils réalisent plutôt la mise en relation de plusieurs modèles hydraulique, hydrologique, hydrogéologique et de qualité de l'eau avec un système d'information géographique et un système expert pour

faire de la gestion intégrée de bassins versants. Dans ce genre de montage, toutes les composantes doivent être intégrées dans un même système pour que ce soit efficace. Dans son article sur la gestion des ressources hydriques, les systèmes d'information géographique et les systèmes experts, Walsh (1993) conclut en présentant un nouveau terme pour le concept. On pourrait traduire cette expression par système expert à référence spatiale.

«SDSS [Spatial Decision Support System] is the new term given to the merging of GIS [Geographical Information System], DSS [Decision Support System], and water resource models to assist decision makers with decisions regarding water resources .»

D'autres se sont penchés sur des problèmes plus pragmatiques de l'utilisation des systèmes d'information géographique dans le domaine des ressources hydriques tels l'acquisition des données et les problèmes de transfert, de projection cartographique, d'exactitude et de précision des données. Clark (1993) considère que la disponibilité des données représente une contrainte majeure dans le développement des applications des systèmes d'information géographique dans le domaine des ressources hydriques.

L'utilisation des systèmes d'information géographique dans le domaine des ressources hydriques en est à ses débuts seulement et les études démontrant les avantages de données saisies par télédétection dans les modèles hydrologiques (Tao et Kouwen, 1989) laissent présager une généralisation de l'utilisation de ces systèmes.

CHAPITRE 2

LES MODÈLES DE PRÉVISION HYDROLOGIQUE

Le but de l'étude actuelle est de réaliser la modélisation d'un bassin versant en utilisant un système d'information géographique, c'est-à-dire de coupler un modèle de prévision hydrologique et un système d'information géographique. Le système d'information géographique SPANS est déjà choisi, il reste à déterminer le modèle de prévision hydrologique le plus approprié. Le choix d'un modèle de prévision hydrologique doit se baser sur des critères bien précis qui sont fonction de la problématique hydrologique et de la nature du site étudié. Singh (1988) définit en quatre points la façon d'aborder un problème de prévision hydrologique. Tout d'abord, il faut bien définir l'utilisation ou les utilisateurs éventuels du modèle. L'utilisateur fournit un indice sur la précision nécessaire du modèle, sur la simplicité d'utilisation et sur la quantité d'information désirée. Il faut ensuite bien cerner le problème, puis choisir l'approche en fonction de la disponibilité des données, des outils et du matériel informatique. Finalement, on doit tenir compte d'un point très important les coûts. Les modèles hydrologiques sont souvent d'utilisation complexe et requièrent des connaissances approfondies en hydrologie, mathématiques et programmation pour être utilisés efficacement et intelligemment. Lorsque ces connaissances ne sont pas suffisantes, la courbe d'apprentissage d'un modèle peut être très longue et très coûteuse.

Il est possible de raffiner davantage cette façon d'aborder un problème de prévision hydrologique. C'est ce qu'a fait Environnement Canada (1986) en proposant une méthodologie d'utilisation d'un modèle hydrologique en onze points.

1. Définir le problème.
2. Spécifier les objectifs.
3. Étudier la disponibilité des données.
4. Déterminer le matériel informatique disponible.
5. Spécifier les contraintes économiques et sociales.
6. Choisir une classe particulière de modèles hydrologiques.
7. Sélectionner dans cette classe un type de modèle en particulier.
8. Calibrer le modèle.
9. Évaluer la performance du modèle.
10. Utiliser le modèle pour des fins de prédiction.
11. Insérer le modèle dans un contexte plus général.

La démarche de modélisation de bassin versant présentée dans ce chapitre suit implicitement cette méthodologie en présentant d'abord le cadre spécifique de l'étude où l'on retrouve l'utilisation et les utilisateurs potentiels, les données disponibles, le matériel informatique, la problématique hydrologique ainsi que le site choisi pour réaliser le projet. On présente ensuite le choix du modèle de prévision hydrologique basé sur des critères bien spécifiques.

2.1 Cadre de l'étude

Les domaines publics fédéral, provincial et municipal se dotent de bases de données géoréférencées pour faire la gestion du territoire et des ressources. Cette gestion nécessite des outils qu'il faut concevoir en fonction des utilisateurs qui ne sont pas toujours des spécialistes en hydrologie ou en hydraulique même s'ils possèdent certaines notions dans ces domaines. Il est possible de concevoir, pour ces gestionnaires, un outil, qui s'adapte à leur système d'information géographique et qui l'exploite, pour estimer le débit de crue d'une rivière suite à une précipitation.

La modélisation des grands bassins versants est souvent complexe et plusieurs grands bassins font déjà l'objet d'études approfondies, ce qui n'est pas le cas des petits bassins versants. L'étude vise donc les petits bassins versants naturels et, a priori, non-jaugés pour répondre à un besoin de gestion des municipalités moins peuplées qui ne possède pas les moyens de commander des études et de faire des mesures de leurs bassins versants. Un bassin naturel signifie un bassin où le pourcentage de la superficie occupée par une zone urbaine est négligeable.

Pour réaliser cette modélisation, on couple un système d'information géographique à un modèle de prévision hydrologique. Le système utilisé est SPANS GIS, car on possède déjà beaucoup de données intégrées dans ce système sur le bassin versant de la rivière Chaudière. De plus, malgré sa base

matricielle, il permet de manipuler les données vectorielles, il offre beaucoup de fonctions de modélisation et sa conception canadienne en fait le système privilégié des ministères du gouvernement fédéral. Le bassin versant de la rivière Chaudière étant très vaste, on choisit d'étudier un de ses sous-bassins, celui de la rivière Samson (figure 2.1). La rivière Samson prend sa source dans les montagnes aux abords de la frontière américaine et se jette dans la rivière Chaudière après avoir traversé une région forestière puis une plaine agricole. Le bassin versant fait moins de 200 km² de superficie, possède peu de zones urbaines et présente un relief accidenté et varié.

Le débit à calculer est un débit de crue pour prévoir des inondations en vue d'un certain contrôle plutôt que pour faire de la gestion de réservoirs. La partie de l'hydrogramme de débit à considérer dans ce cas est celle allant de l'écoulement de base jusqu'à la pointe de l'hydrogramme (figure 2.2). La partie de l'hydrogramme constituée de la courbe de décrue, de la fin du ruissellement de surface et de la courbe de tarissement résultant de l'écoulement souterrain n'est pas considérée. Lors de la modélisation, il sera donc possible de négliger le processus d'écoulement souterrain. A priori, l'analyse est faite à partir d'un cas simple, tel une pluie uniforme, et les précipitations sont représentées par un hyétogramme.

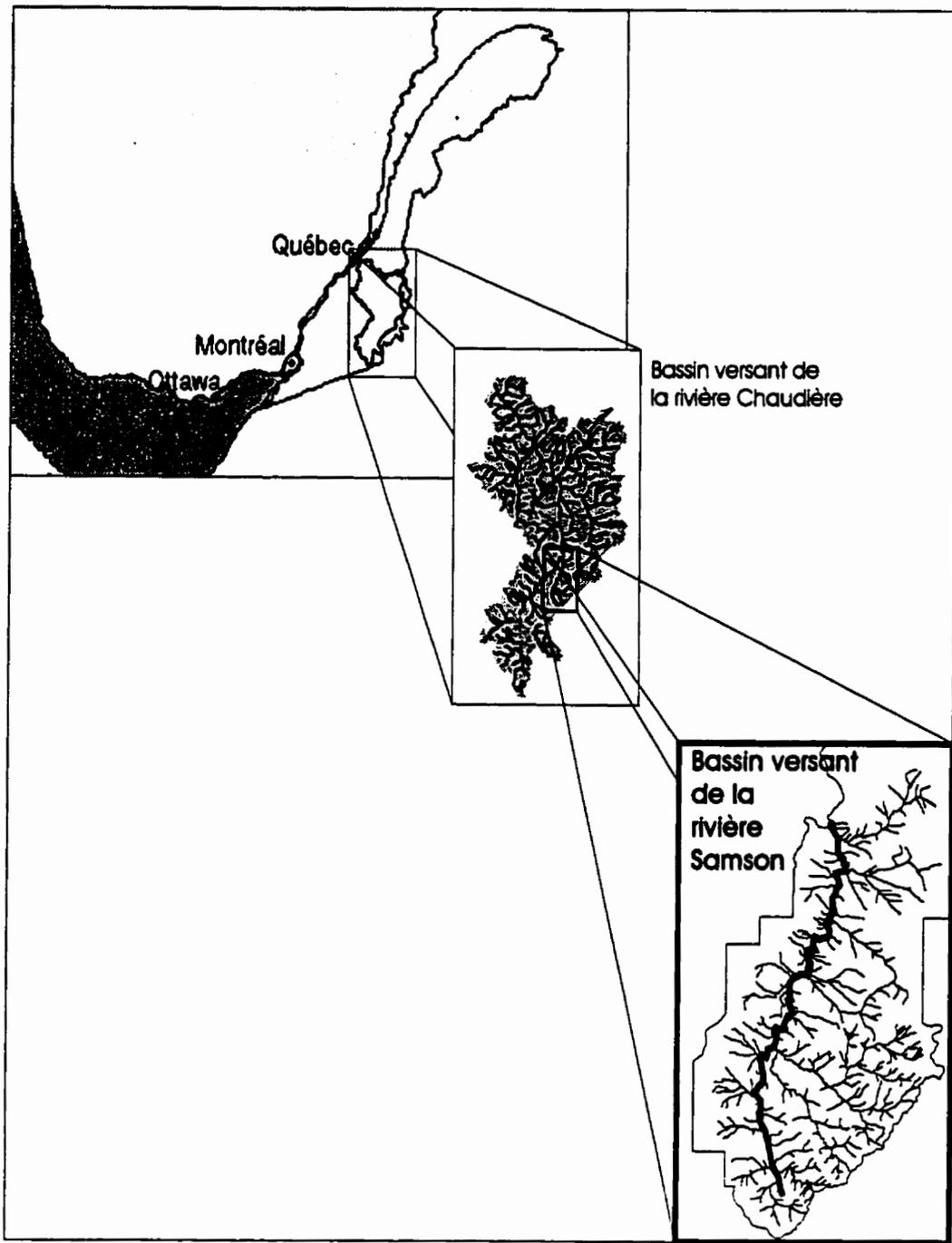


Figure 2.1 Localisation de la rivière Samson.

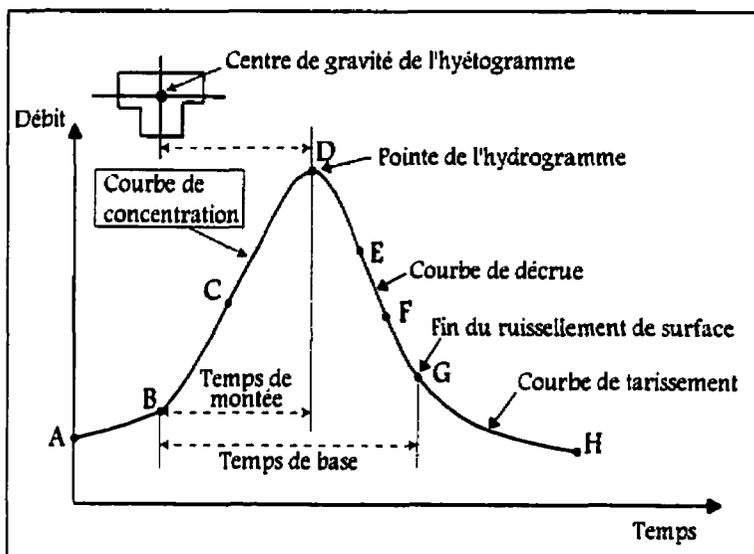


Figure 2.2 Hydrogramme type (Llamas, 1993).

Cette problématique hydrologique permet d'établir des critères de sélection. Les modèles ne répondant pas à tous ces critères à la fois seront rejetés, car ils sont inaptes à traiter la problématique hydrologique exposée ci-haut. Voici ces critères de sélection:

1. Le modèle doit être applicable au site à l'étude, c'est-à-dire être conçu pour des bassins versants présentant des caractéristiques géomorphologiques similaires à celles retrouvées sur le bassin de la rivière Samson.
2. Il doit être conçu ou adaptable aux données de télédétection.
3. Il doit modéliser le bassin versant de façon à estimer le débit direct résultant d'une précipitation.

4. Il doit être suffisamment documenté et être du domaine public.
5. Il doit exploiter les capacités du système d'information géographique auquel il sera couplé.

2.2 Choix d'un modèle de prévision hydrologique

Pour réaliser le couplage entre un modèle de prévision hydrologique et un système d'information géographique, deux alternatives se présentent. La première consiste à choisir un modèle hydrologique existant et suffisamment documenté et à créer le lien entre le modèle et le système d'information géographique. Le système sert principalement à fournir les intrants nécessaires à l'exécution du modèle. Il fait le lien entre la saisie des données par télédétection et le modèle de prévision hydrologique. Cette approche a été utilisée par plusieurs auteurs dont Kern et Stednick (1993), Stuart et Stocks (1993) et Hay et al. (1993). La seconde alternative se sert des différentes fonctions du système d'information géographique pour construire un modèle de prévision hydrologique. Chacune des alternatives présente des avantages et des inconvénients qui méritent d'être évalués afin de déterminer la méthode qui convient le mieux au cadre de l'étude. Elles seront expliquées plus en détail lors de la sélection d'un modèle hydrologique faisant partie de la liste établie à la section suivante.

2.2.1 Modèles de prévision hydrologique

Dans une étude sur les techniques de télédétection et de prévision de débit des cours d'eau, Environnement Canada (1986) a établi une liste de modèles mathématiques, déterministes et stochastiques, qui s'insèrent bien dans un contexte canadien, qui utilisent, ou peuvent utiliser, des données de télédétection, qui sont du domaine public et suffisamment documentés. Un tableau qui décrit une série de caractéristiques pour chaque modèle est construit à partir de cette liste. Il est repris au tableau 2.1 en conservant seulement les principales caractéristiques qui permettront de déterminer le ou les modèles qui conviennent à l'étude. Les caractéristiques sont les suivantes:

1. La structure du modèle.
2. La valeur cognitive du modèle.
3. Le caractère du modèle.
4. La distribution du modèle.

La structure du modèle correspond au niveau d'abstraction hydrologique du modèle. Le modèle est à processus unique s'il simule un seul processus hydrologique. Il est dit à composantes lorsqu'il traite individuellement les composantes du processus hydrologique et de type bassin lorsque les composantes du processus hydrologique traitées interagissent par des liens

Tableau 2.1 Caractéristiques des modèles

Nom du modèle	Type de modèle			
	Structure du modèle	Valeur cognitive	Caractère du modèle	Distribution
Canada:				
SIMFLO	Type bassin	Conceptuel	Déterministe	Agrégé
MOEHYDR2	Type bassin	Physique	Déterministe	Agrégé
CEQUEAU	Type bassin	Conceptuel	Déterministe	Discretisé
QFORECAST	Composantes	Conceptuel	Déterministe	Agrégé
QUFM	Global	Tendanciel	Déterministe	Agrégé
MANTHORN	Global	Tendanciel	Déterministe	Agrégé
SASK6	Type bassin	Conceptuel	Déterministe	Agrégé
SLURP	Type bassin	Conceptuel	Déterministe	Agrégé
UBC	Type bassin	Physique Conceptuel	Déterministe	Agrégé
FLOCAST	Type bassin	Conceptuel	Déterministe	Agrégé
États-Unis:				
SWMM III	Type bassin	Conceptuel	Déterministe	Agrégé
HEC-1	Type bassin	Conceptuel	Déterministe	Agrégé
HSPF	Type bassin	Physique	Déterministe	Agrégé
Stanford IV	Type bassin	Physique	Déterministe	Agrégé
SSARR-4	Type bassin	Conceptuel Tendanciel	Déterministe	Agrégé
LSBR	Type bassin	Conceptuel Tendanciel	Déterministe	Agrégé
SCS-BSM	Type bassin	Conceptuel Tendanciel	Déterministe	Agrégé
NWSRFS	Type bassin	Conceptuel Tendanciel	Déterministe	Agrégé
USDAHL-74	Type bassin	Conceptuel Tendanciel	Déterministe	Agrégé
HYMO-VUH	Type bassin	Physique Conceptuel	Déterministe	Agrégé
USGS	Type bassin	Conceptuel	Déterministe	Agrégé
Autres nationalités:				
MARTINEC	Composantes	Conceptuel	Déterministe	Discretisé
HBV	Type bassin	Conceptuel	Déterministe	Discretisé
SHE	Type bassin	Physique Conceptuel	Déterministe	Discretisé
TANK	Type bassin	Conceptuel	Déterministe	Agrégé
Régression:				
PREVIN	Global	Tendanciel	Stochastique	Agrégé
VOLCAST	Global	Tendanciel	Stochastique	Agrégé
MANAPI	Global	Tendanciel	Stochastique	Agrégé

internes souvent complexes. Les modèles globaux regroupent toutes les composantes en un seul opérateur. C'est le cas des modèles de régression.

La valeur cognitive des modèles les regroupe selon qu'ils sont physiques, conceptuels ou tendanciels. Un modèle physique représente le phénomène rigoureusement par une ou plusieurs équations mathématiques. Les modèles conceptuels simplifient les descriptions mathématiques des phénomènes hydrologiques et les modèles tendanciels n'ont pas vraiment une base rigoureuse. Ils sont généralement composés de relations empiriques existant entre différents phénomènes, entre la pluie et le débit par exemple. Cette valeur cognitive est associée à la façon dont les modèles traitent chaque composante du cycle hydrologique, soit la fonte de la neige, l'infiltration, l'évapotranspiration et le laminage.

Un modèle mathématique possède un caractère stochastique lorsque ses variables sont aléatoires et présentent une distribution de probabilité. Lorsque les paramètres du modèle ne possèdent pas de variables aléatoires, celui-ci est déterministe. Les modèles mathématiques déterministes regroupent les modèles physiques et les modèles conceptuels. Ils permettent de suivre l'évolution des processus dans le temps. Les changements opérationnels et physiographiques d'un bassin peuvent être incorporés dans un modèle de ce genre. Ils sont flexibles et nécessitent beaucoup moins de données historiques.

La distribution d'un modèle classifie les modèles selon qu'ils sont discrétisés ou agrégés. Un modèle discrétisé tient compte de la variation spatiale des phénomènes tandis qu'un modèle agrégé en fait une représentation sur l'ensemble du bassin, ou sur chacun des sous-bassins. Un modèle déterministe et discrétisé bénéficie davantage du lien avec un système d'information géographique, car ce dernier permet de traiter les données de télédétection pour les adapter à l'intrant au modèle. Il permet aussi de répartir dans l'espace les données ponctuelles. Cependant, un modèle agrégé profite aussi de l'apport des SIG, mais par la méthode opposée. Le bassin est subdivisé par un quadrillage, en sous-bassins ou en toutes autres unités, telles les unités de paysage (Ducruc et De Sède, 1991), et une moyenne des données de télédétection est calculée sur chacune des ces surfaces.

En plus des modèles canadiens plus récents, tels WATFLOOD, HYDROTEL et TOPMODEL, plusieurs modèles tirés du tableau 2.1 satisfont aux quatre premiers critères de sélection définis dans le cadre de l'étude. Ils sont répartis au tableau 2.2 selon deux groupes:

1. Les modèles discrétisés.
2. Les modèles agrégés.

Le critère décisif de sélection devient l'exploitation des capacités du système d'information géographique auquel le modèle sera couplé.

Tableau 2.2 Modèles satisfaisant aux quatre premiers critères

<p style="text-align: center;">GROUPE 1</p> <p style="text-align: center;">Les modèles distribués</p>	<p style="text-align: center;">SHE CEQUEAU HBV WATFLOOD HYDROTEL TOPMODEL</p>	
<p style="text-align: center;">GROUPE 2</p> <p style="text-align: center;">Les modèles agrégés</p>	<p style="text-align: center;">STANFORD-IV HSP-F UBC HYMO-VUH SWMM-III SASK6 FLOCAST USGS SLURP</p>	<p style="text-align: center;">TANK HEC-1 SSARR-4 USDAHL-74 SCS-BSM NWSRFS LSBR MOEHYDR2 SIMFLO</p>

2.2.2 Lien entre un SIG et un modèle hydrologique

Selon Stuart et Stock (1993), il existe deux types de relation entre les modèles et les systèmes d'information géographique: la combinaison et l'intégration (figure 2.3). Ces deux types de lien sont souvent perçus comme deux idéologies de couplage entre les modèles et les systèmes d'information géographique. Ces idéologies ont tendance à évoluer en fonction du développement des systèmes d'information géographique qui deviennent de plus en plus versatiles au niveau du langage et de la modélisation.

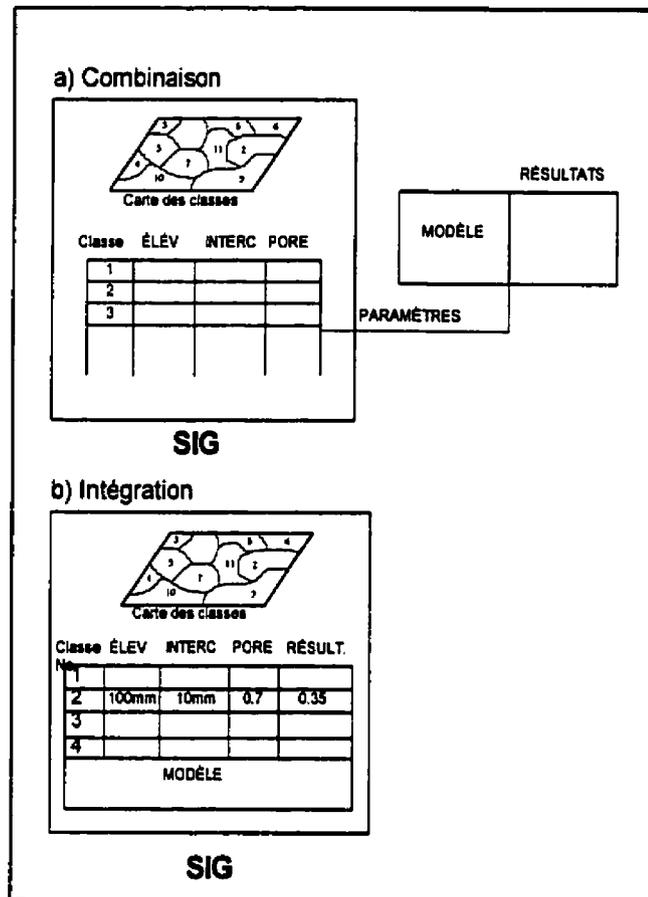


Figure 2.3 Deux idéologies en matière de lien entre un système d'information géographique et un modèle hydrologique (Stuart et Stocks, 1993)

À court terme, la combinaison semble être la meilleure solution, car on utilise les systèmes et les modèles pour effectuer les opérations pour lesquelles ils sont respectivement conçus (Stuart et Stocks, 1993). Les systèmes d'information géographique modélisent et analysent les données pour produire les intrants aux modèles. Ces derniers sont exportés vers le modèle de prévision hydrologique où on effectue la simulation. Les résultats de la simulation sont importés de

nouveau dans le système d'information géographique pour y être affichés. La combinaison est flexible pour les concepteurs, cependant il faut déployer des efforts constants pour maintenir le lien à jour, car chaque nouvelle version du modèle hydrologique ou du système d'information géographique vient compromettre le lien voire le détruire complètement. De plus, les données typologiques sont exportées dans des fichiers en format ASCII; elles sont dissociées de leur contexte géographique.

Avec le perfectionnement des systèmes d'information géographique qui offrent de plus en plus de possibilités au niveau de leurs fonctions internes et de leur versatilité, on entrevoit une autre stratégie pour relier un modèle de prévision hydrologique à un système d'information géographique. L'intégration d'un modèle dans un système consiste à créer, à partir des fonctions de modélisation et du langage de programmation du système, le modèle d'écoulement. Le modèle est conçu, construit et implémenté à l'intérieur même du système d'information géographique. Cette relation permet de pousser encore plus loin le couplage jusqu'à la création d'un système expert qui intègre un système d'information géographique, un modèle de prévision hydrologique et autres modèles, ainsi qu'un système d'aide à la décision. C'est ce que proposent plusieurs auteurs (Walsh, 1993; Leipnik *et al.*, 1993; McKinney *et al.*, 1993; Feldman, 1992) pour faire de la gestion et de la planification intégrée des ressources hydriques.

L'intégration est cependant limitée par l'incapacité des systèmes d'information géographique à gérer la quatrième dimension qu'est le temps. En effet, ils sont inaptes à modéliser la variation des phénomènes dans le temps ce qui constitue un obstacle majeur (Brilly *et al.*, 1993). De plus, le langage de modélisation ne permet pas, en général, de construire des structures récursives avec des boucles d'itérations (Stuart et Stock, 1993). C'est le cas du système SPANS utilisé dans ce projet.

Ces contraintes ont poussé certains chercheurs, comme Muzik et Chang (1993) qui estiment que les systèmes d'information géographique ne sont pas conçus pour traiter des problèmes hydrologiques, à créer leur propre système d'information géographique hydrologique. Le système possède les fonctions de base qui permettent de manipuler des données géoréférencées, des fonctions d'analyse spatiale pour déterminer les paramètres des bassins versants, les fonctions du modèle hydrologique ainsi que certains utilitaires.

2.2.3 Modélisation d'un bassin par un système d'information géographique

En couplant un des modèles sélectionnés au tableau 2.2 à un système d'information géographique, la combinaison est le seul type de relation possible, ce qui est tout à fait normal puisque tous les modèles ont été conçus indépendamment des systèmes d'information géographique. Les modèles plus récents ont été conçus en fonction des données à répartition spatiale qui peuvent

être aisément manipulées par les SIG, mais pas vraiment en fonction des possibilités de modélisation qu'offrent les SIG. Il serait possible d'intégrer l'algorithme codé d'un de ces modèles à même le programme d'un SIG de sorte que le modèle devienne une fonction de modélisation du SIG. Cet exercice est cependant plus un défi informatique qu'un défi hydrologique. L'alternative de l'intégration qui consiste à créer un modèle à partir d'un système d'information géographique représente un défi hydrologique plus intéressant et c'est celle qui sera adoptée dans le cadre de l'étude.

L'utilisation d'un modèle existant est donc écartée, car il ne permet pas d'établir le lien d'intégration privilégié dans le cadre de l'étude. De plus, le fait de fournir des paramètres intrants à un modèle n'exploite pas les possibilités de modélisation qu'offre les systèmes d'information géographiques. On ne peut pas non plus envisager l'intégration d'un modèle dans un système d'information géographique, car SPANS comporte trop de lacunes au niveau des commandes récursives et itératives de modélisation. Il est plus rationnel de modéliser le bassin versant en construisant un modèle d'écoulement qui maximise l'utilisation des fonctions du système d'information géographique en favorisant le lien le plus étroit possible entre le système d'information géographique et le modèle hydrologique qui sera construit.

CHAPITRE 3 MODÉLISATION DU BASSIN VERSANT

Le choix d'un modèle de prévision hydrologique existant ne permettant pas d'atteindre les objectifs visés, il est préférable de concevoir un modèle qui exploite mieux les capacités des systèmes d'information géographique. À cause des lacunes inhérentes au système SPANS, l'intégration du modèle dans le système n'est pas possible. Il faut cependant tenter de maximiser la combinaison de façon à ce que l'intégration, lorsqu'elle sera envisageable, nécessite un minimum d'efforts et de manipulations.

La modélisation du bassin versant, basée sur le cadre de l'étude, c'est-à-dire le calcul d'un débit direct de rivière suite à une précipitation sur un petit bassin versant naturel et non jaugé, se fait en deux étapes. En premier lieu, chaque composante du cycle hydrologique est étudiée pour élaborer une méthode de modélisation du processus dans le système d'information géographique.

La modélisation de l'écoulement et du réseau de drainage fait l'objet de la deuxième partie de la modélisation du bassin versant. C'est la partie la plus problématique, car elle ne s'exécute pas dans le système d'information géographique. Plusieurs essais de modélisation dans SPANS sont tentés, mais ils se soldent tous par des échecs. Il faut se résigner à recourir à un programme externe pour la réaliser. Les principes sur lesquels repose le calcul du débit à

l'exutoire du bassin versant sont établis en détail et suivent la présentation de l'algorithme utilisé pour établir le réseau de drainage dans le modèle d'écoulement.

3.1 Processus hydrologiques

La modélisation d'un bassin versant consiste à estimer le ruissellement total en fonction des précipitations donc à modéliser le processus précipitation-écoulement schématisé à la figure 3.1. L'évapotranspiration et la sublimation, la fonte de la neige, l'infiltration profonde et hypodermique ainsi que le ruissellement superficiel sont des composantes du processus hydrologique qui viennent influencer le ruissellement total selon des échelles de temps et de quantités différentes.

L'évaporation, l'évapotranspiration et la sublimation sont en général regroupés dans un même calcul ou dans la même estimation, mais malgré l'importance notable de ces phénomènes, le calcul des quantités d'eau qu'ils représentent n'est pas inclus dans le modèle actuel. En effet, le bassin est petit, le temps de réponse court et il y a précipitation ce qui minimise l'importance de ces phénomènes. De plus, pour le moment, le modèle ne traite pas les précipitations de neige et la sublimation fait très peu varier l'épaisseur du manteau nival en quelques heures. L'évaluation de la valeur de l'évapotranspiration peut se faire en se basant sur des données de bassin se situant dans une région voisine où les

caractéristiques et conditions sont suffisamment similaires pour être transposées.

La validation des résultats devrait venir confirmer cette hypothèse.

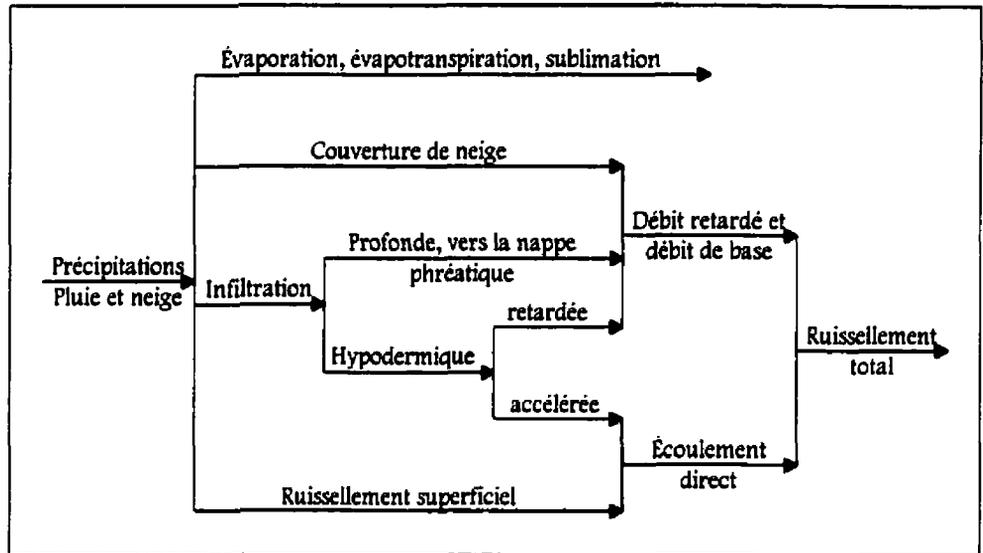


Figure 3.1 Schéma du processus « précipitation-écoulement » (Llamas, 1993)

L'infiltration profonde et hypodermique sont des processus dont l'échelle de temps est de l'ordre de la journée, du mois, voire même de l'année pour les très grands bassins ce qui est de beaucoup supérieur au temps de réponse pour un petit bassin. Le but étant de calculer un débit quotidien, ils n'interviennent pas en tant que tel dans le modèle. Le processus d'infiltration sert principalement à déterminer la quantité d'eau qu'il faut retrancher de la pluie réelle pour obtenir la pluie effective ou nette, celle qui ruisselle.

Le phénomène de la fonte de neige est pour l'instant négligé, car l'étude se limite aux précipitations sous forme de pluie, mais il serait pertinent de l'inclure éventuellement dans le modèle.

Le ruissellement superficiel devient le processus clef du modèle d'écoulement. Sa modélisation se fait souvent par une estimation de l'hydrogramme à partir d'un hyétogramme définissant l'averse. Les facteurs influençant la forme de l'hydrogramme sont très nombreux et sont répertoriés dans le tableau 3.1.

Tableau 3.1 Facteurs influençant l'hydrogramme d'une précipitation

Précipitations	Sol	Climat	Facteurs du bassin
<ul style="list-style-type: none"> • Intensité • Durée • Distribution spatiale • Cheminement de l'averse • Type de précipitation 	<ul style="list-style-type: none"> • Couvert végétal • Couche de labour • Perméabilité du sol • Profondeur du sol • Profil du sol • Drainage souterrain • Techniques culturales 	<ul style="list-style-type: none"> Évapotranspiration • Température • Précipitations antérieures 	<ul style="list-style-type: none"> • Longueur d'écoulement et temps de cheminement • Superficie • Pente • Forme • Lacs, étangs et dépressions • Densité de drainage • Géologie du sous-sol • Aménagement du bassin

Parmi les méthodes de modélisation du ruissellement citons:

- les méthodes d'estimation de la pointe de l'hydrogramme: méthode rationnelle ou hydrogramme en échelons;
- la méthode de l'hydrogramme unitaire (HU);
- la méthode de l'hydrogramme unitaire instantané (HUI);

- la méthode de l'hydrogramme standard à décroissance exponentielle ou à décroissance hyperbolique;
- la méthode du SCS (*Soil Conservation Service*);
- et la méthode de l'hydrogramme synthétique.

Le ruissellement superficiel est responsable de l'écoulement direct et, par conséquent, du débit de crue, mais pour que le ruissellement se produise, le sol doit être saturé d'eau en surface et les dépressions du terrain doivent être comblées. L'eau s'écoule ensuite sur le terrain en fonction de la pente, du type de sol et du couvert végétal jusqu'au cours d'eau. Après un examen sommaire des différentes méthodes énumérées ci-dessus, la relation pluie-ruissellement du SCS est retenue. Cette méthode est utilisée pour l'analyse de milliers de bassins versants, dans différents milieux et pour différentes utilisations du sol, et donne des résultats satisfaisants (Tecsult, 1993a). Elle semble être la plus intéressante, car son utilisation des caractéristiques géomorphologiques du bassin pour estimer les valeurs de ruissellement permet de mettre à profit une modélisation dans un système d'information géographique et l'acquisition de données par télédétection.

3.1.1 Méthode pluie-ruissellement du SCS

La description de la méthode d'évaluation du coefficient de ruissellement du SCS s'appuie sur la consultation de plusieurs documents dont le rapport sur le projet

de la rivière Chaudière produit par TecSult (1993a), le document du service de l'hydrométrie du gouvernement du Québec par Monfet (1979), le livre de Ponce (1989) ainsi que le guide sur la méthode du SCS de McCuen (1982). Cette description est accompagnée de la méthode d'estimation du coefficient de ruissellement et des opérations d'implémentation du calcul du coefficient dans le système SPANS.

La première équation décrivant la relation pluie-ruissellement établit le lien entre le ruissellement, le volume de précipitation et la capacité d'emmagasinage du sol, disponible pour la rétention.

$$\frac{F}{S} = \frac{Q}{P - I_a} \quad (3.1)$$

F représente la rétention actuelle c'est-à-dire la différence entre le volume de la précipitation et le ruissellement. S est le potentiel maximum de rétention d'emmagasinage (*storage*), I_a les pertes initiales, P le volume de précipitation et Q le ruissellement. Le diagramme de la figure 3.2 schématise cette équation.

De plus, en considérant les pertes initiales, la rétention actuelle (F) s'exprime par l'expression:

$$F = (P - I_a) - Q \quad (3.2)$$

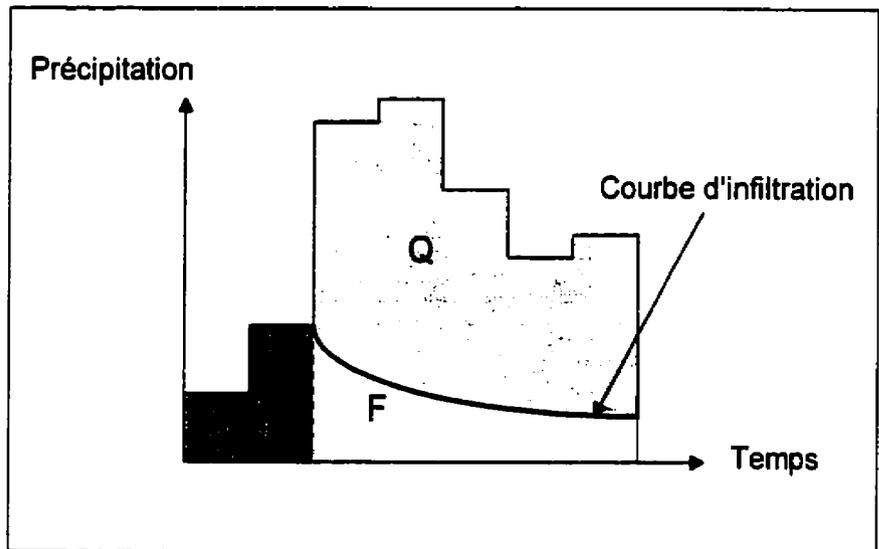


Figure 3.2 Relation entre la précipitation, le ruissellement et la rétention (McCuen, 1982)

La combinaison des deux premières équations donne l'expression du ruissellement (Q).

$$Q = \frac{(P - I_a)^2}{(P - I_a) + S} \quad (3.3)$$

On peut estimer, toujours selon le SCS, la valeur des pertes initiales (I_a) par l'équation empirique suivante:

$$I_a = 0,2 S \quad (3.4)$$

En fait, les pertes initiales sont fonction de l'utilisation et du traitement du sol, du type de sol, de l'interception, de l'infiltration, de l'emmagasinage dans les dépressions et de l'humidité antécédente du sol. Cette équation tient compte de

la rétention de l'eau dans les dépressions et de l'absorption des premiers millimètres de pluie due aux conditions d'humidité antécédente du sol.

La substitution de cette dernière équation des pertes initiales dans l'équation 3.3 réduit cette dernière à deux inconnues au lieu de trois.

$$Q = \frac{(P - 0,2 S)^2}{P + 0,8 S} \quad (3.5)$$

Il reste maintenant à trouver une équation pour exprimer le potentiel maximal de rétention (S). Des études empiriques de la SCS ont démontré qu'on pouvait estimer la valeur de S par l'équation suivante:

$$S = \frac{1000}{CN} - 10 \quad (3.6)$$

Dans cette équation *CN* signifie "*runoff Curve Number*", numéro de courbe de ruissellement. Ce numéro sert d'index de classification des sols selon leur perméabilité. Par exemple, un CN égal à 100 indique que le sol est complètement imperméable et à l'opposé, un CN nul indique une absorption totale de la pluie. Effectivement, si le sol est tout à fait perméable c'est donc dire que le ruissellement est nul. Cet index dépend en grande partie du type de sol, de l'utilisation du sol, de la condition hydrologique du couvert végétal, ou condition de drainage, et de l'humidité antécédente du sol.

3.1.1.1 Estimation des CN

L'index de ruissellement CN, *Curve Number*, sert à l'estimation du volume et du taux de ruissellement d'une parcelle de terrain. Cet index est évalué en classifiant le terrain selon le groupe de sol auquel il appartient puis selon son complexe de couverture du sol. Finalement, le résultat est ajusté en fonction de la condition d'humidité antécédente du sol.

La classification du sol, développée par la SCS, se fait en quatre groupes. Les groupes sont identifiés par les lettres A, B, C, D et répartis comme suit.

Tableau 3.2 Description des groupes de sol (McCuen, 1982)

Groupes de sol du SCS	Type de sol correspondant
Groupe A	sable en profondeur, loess en profondeur et silts agrégés.
Groupe B	loess en surface et terreau sableux.
Groupe C	terreau d'argile, terreau sableux superficiel, sols faiblement organiques et sols fortement argileux.
Groupe D	sols qui gonflent d'une façon significative lorsqu'ils sont mouillés, argiles très plastiques et certains sols salins.

Lorsque le type de sol du terrain est connu, le classement se fait aisément en suivant la classification ci-dessus. Dans le cas contraire, il est possible d'utiliser une carte de la région qui indique les groupes de sols. Plusieurs comtés aux États-Unis possèdent ce genre de carte. Le taux minimum d'infiltration du sol est un autre moyen pour effectuer la classification du sol en se servant du tableau 3.3.

Tableau 3.3 Classement selon le taux d'infiltration
(McCuen, 1982)

Groupe de sol	Taux d'infiltration minimum (mm/h)
A	0.0762 - 0.1143
B	0.0381 - 0.0762
C	0.0127 - 0.0381
D	0.0000 - 0.0127

La seconde classification se fait selon le complexe de couverture du sol. Le complexe de couverture regroupe trois facteurs: l'utilisation du sol, le traitement ou pratique qu'on y exerce et les conditions hydrologiques. Les conditions hydrologiques représentent la qualité de drainage du terrain en question. La subdivision selon le genre de pratique se fait surtout pour l'agriculture selon que la culture se fait en rangée ou en contour. Cette subdivision permet de faire une distinction entre les différents potentiels de ruissellement associés aux variations dans la pratique des cultures. La condition hydrologique du sol est un indicateur de la qualité du drainage du terrain. Il est partagé en deux ou trois catégories selon le cas: pauvre et bonne ou pauvre, passable et bonne.

Une fois que les caractéristiques du terrain sont bien identifiées, le tableau 3.4 est utilisé pour déterminer la valeur de l'index CN. Par exemple, pour une région boisée avec une bonne couverture et dont le sol appartient au groupe B, le CN obtenu est de 55. Il reste à faire l'ajustement en fonction de l'humidité antécédente du sol.

Selon le SCS, il existe trois conditions d'humidité du sol qu'on décrit comme suit:

- Condition I: sol sec, mais non desséché; une culture satisfaisante a déjà été pratiquée dans ce sol.
- Condition II: conditions moyenne d'humidité; les valeurs du tableau 3.4 correspondent à cette condition.
- Condition III: sol saturé; une pluie dense ou une pluie légère à une basse température s'est produite dans les cinq derniers jours.

Le tableau 3.5 donne les limites de pluie pour les trois conditions d'humidité antécédente du sol en fonction de la saison. L'ajustement des valeurs de l'index obtenues dans le tableau 3.4 se fait selon le tableau 3.6. Les valeurs obtenues dans le tableau 3.4 donne le CN pour la condition d'humidité II.

Tableau 3.4 Coefficient de ruissellement pour le complexe hydrologique de couverture du sol (McCuen, 1982).

Description de l'utilisation du terrain / pratique / condition hydrologique			Groupe hydrologique de sol			
			A	B	C	D
Résidentiel ¹						
Grandeur moyenne du lot	% moyen d'imperméabilité ²					
1/8 d'acre ou moins	65		77	85	90	92
1/4 d'acre	38		61	75	83	87
1/3 d'acre	30		57	81	81	86
1/2 acre	25		54	70	80	85
1 acre	20		51	68	79	84
Aires de stationnement pavées, toits, entrées, etc. ³			98	98	98	98
Rues et routes:						
	pavées avec trottoirs et égouts pour les orages ³		98	98	98	98
	en gravelle		76	85	89	91
	de terre		72	82	87	89
Zones d'affaires et commerciales (85% imperméables)			89	92	94	95
Parcs industriels (72% imperméables)			81	88	91	93
Espaces ouverts, gazons, parcs, terrains de golf, cimetières:						
	bonnes conditions: couverture de gazon sur 75% de la surface ou plus		39	61	74	80
	conditions passables: couverture de gazon sur 50% à 75% de la surface		49	69	79	84
Jachère	Rangée droite	- - -	77	86	91	94
Cultures en rangée	Rangée droite	Pauvre	72	81	88	91
	Rangée droite	Bonne	67	78	85	89
	Contour	Pauvre	70	79	84	88
	Contour	Bonne	65	75	82	86
	Contour et en terrasse	Pauvre	66	75	80	82
	Contour et en terrasse	Bonne	62	71	78	81
Céréales	Rangée droite	Pauvre	65	76	84	88
		Bonne	63	75	83	87
	Contour	Pauvre	63	74	82	85
		Bonne	61	73	81	84
	Contour et en terrasse	Pauvre	61	72	79	82
		Bonne	59	70	78	81
Semis - rapprochés légumes ⁴ ou rotation de pré	Rangée droite	Pauvre	66	77	85	89
		Bonne	58	72	81	85
	Contour	Pauvre	64	75	83	85
		Bonne	55	69	78	83
	Contour et terrasse	Pauvre	63	73	80	83
		Bonne	51	67	76	80
Pâturage ou prairie		Pauvre	68	79	86	89
		Passable	49	69	79	84
		Bonne	39	61	74	80
	Contour	Pauvre	47	67	81	88
		Passable	25	59	75	83
		Bonne	6	35	70	79
Pré		Bonne	30	58	71	78
Bois ou forêt		Pauvre	45	66	77	83
		Passable	36	60	73	79
		Bonne	25	55	70	77
Fermes		- - -	59	74	82	86

Notes relatives au tableau 3.4 :

¹ Les CN "curve number" sont calculés en assumant que le ruissellement de la maison et de l'entrée est acheminé vers la rue avec un minimum d'eau provenant du toit. Cette eau du toit se dirige vers les gazons où peut se produire une infiltration additionnelle.

¹ Les surfaces perméables restantes (les gazons) sont considérées en bonne condition de pâture pour ces CN « curve number ».

¹ Dans les pays où le climat est plus clément, il est possible d'utiliser un CN de 95.

¹Rapproché: semé en sillons ou semé à la volée.

Tableau 3.5 Limites de pluie (McCuen, 1982)

Condition d'humidité antécédente	Total de la pluie antécédente sur cinq jours (mm)	
	Saison de repos	Saison de croissance
I	Moins de 12.7	Moins de 35.6
II	de 12.7 à 27.9	de 35.6 à 53.3
III	plus de 27.9	plus de 53.3

Tableau 3.6 Ajustement du CN en fonction des conditions d'humidité antécédentes (McCuen, 1982)

CN pour la condition II	CN correspondant pour la condition	
	I	III
100	100	100
95	87	99
90	78	98
85	70	97
80	63	94
75	57	91
70	51	87
65	45	83
60	40	79
55	35	75
50	31	70
45	27	65
40	23	60
35	19	55
30	15	50
25	12	45
20	9	39
15	7	33
10	4	26
5	2	17
0	0	0

3.1.2 Implémentation de la méthode du SCS dans le système SPANS

Le calcul du coefficient de ruissellement a déjà été effectué dans le système SPANS pour le bassin versant de la rivière Chaudière (Tecsult, 1993a). Pour les fins de l'étude de la rivière Chaudière, le calcul du coefficient se fait en utilisant un découpage du bassin versant en 112 unités de paysage au lieu du maillage orthogonal (Ducruc et De Sède, 1991). Une unité de paysage sur le bassin est une surface qui présente une certaine homogénéité des principales caractéristiques du milieu influençant le ruissellement, soit:

- les dépôts de surface: nature des matériaux, texture, épaisseur;
- les pentes;
- et l'altitude.

Cette méthode de division du territoire est avantageuse, car elle dissocie la précision du calcul de la résolution des unités. Dans une division conventionnelle du territoire, comme le maillage orthogonal, la valeur d'une caractéristique du terrain est une moyenne de l'ensemble des valeurs sur une maille. Plus la maille est petite moins l'erreur est grande. Dans le cas des unités de paysage, la valeur des caractéristiques est uniforme sur toute la surface de l'unité peu importe la dimension de celle-ci.

Le découpage en unités de paysage est effectué sur une carte papier par des géomorphologues puis numérisé et intégré dans SPANS. Pour compléter la base

de données, il faut une carte de l'utilisation des sols, des points d'altitude, une carte des sols et des dépôts de surface ainsi que les pentes du terrain. Le tableau 3.7 présente la liste de ces données, leur source et le traitement effectué pour les intégrer au système d'information géographique.

Tableau 3.7 Composantes de la base de données du système d'information géographique (Tecsult, 1993a)

Composante	Source de l'information	Traitement pour l'intégration des données dans SPANS
Utilisation des sols	Images satellitaires LANDSAT fournies par l'INRS-Eau	Fichier numérique obtenu de l'image
Unités de paysage	Cartes topographiques au 1:50 000 et photos SPOT au 1:250 000	Numérisation des périmètres des unités à partir des cartes
Points d'altitude	INRS-Eau à partir des cartes topographiques au 1:250 000	Fichier numérique des points d'altitude
Sols et dépôts de surface	Cartes géologiques et pédologiques au 1:50 000	Fichier numérique des superficies par unités de paysage
Pentes	Cartes topographiques au 1:50 000	Fichier numérique des pentes moyennes par unités de paysage

Certaines composantes de la base de données sont intégrées en tant qu'attribut de la carte des unités de paysage et ne possèdent pas de référence spatiale. Elles ont été incorporées dans une base de données en se servant du logiciel D-Base et, parallèlement, associées à la carte des unités de paysage. Les pentes, par exemple, sont mesurées sur les cartes topographiques selon un quadrillage

orthogonal de points de 1 km², les moyennes pour chaque unité de paysage sont ensuite calculées puis entrées dans un fichier D-Base et associées aux unités de paysage. Les superficies de chaque type de sols ou dépôts de surface par unités de paysage sont calculées par le système d'information géographique puis entrées dans la base de données D-BASE.

La méthode de calcul du coefficient de ruissellement (*Curve Number*) pour chaque unité de paysage est assez simple, mais elle n'est pas complètement intégrée dans le système SPANS

Les sols du bassin versant sont classifiés, à partir des cartes pédologiques à l'échelle 1:50 000, selon les groupes de sol A, B, C ou D, tel que définis au tableau 3.2. La base de données D-BASE des groupes de sol est construite en prélevant des points sur la carte selon un quadrillage de 1 km². Les pentes sont ensuite évaluées à partir des cartes topographiques et la pente moyenne pour chaque unité de paysage est calculée par ordinateur au moyen du logiciel D-BASE. Les coefficients de ruissellement (CN) de chaque unité de paysage, sont déterminés en utilisant la clef d'interprétation, présentée au tableau 3.8, tirée du tableau 3.4 présenté à la section précédente.

Le calcul se fait comme suit, dans un utilitaire programmé dans le langage D-BASE:

Tableau 3.8 Étude de modélisation du bassin versant de la rivière Chaudière.
Clef d'interprétation des coefficients de ruissellement (CN)
(Tecsult, 1993a)

Utilisation des sols	Pente	Condition hydrologique	Classe de sol			
			A	B	C	D
Cultures intensives:	<3%	pauvre	63	74	80	82
• maïs	3-8%	pauvre	65	76	84	88
• céréales	>8%	pauvre	72	81	88	91
• cultures						
Cultures extensives	<3%	bonne	25	40	70	78
• friche	3-8%	bonne	39	61	74	80
	>8%	bonne	49	69	79	84
Boisé	<3%		22	53	65	74
• forêt	3-8%		25	55	70	77
	>8%		41	63	75	81
Résidentiel, commercial peu dense:	note		59	74	82	86
• urbain						
Zones humides:			22	22	22	22
• tourbière						
• marécage						

Note: Indépendant de la pente et des conditions hydrologiques.

1. Le coefficient de ruissellement (CN_i) pour l'utilisation de sol i dans l'unité de paysage K se calcule d'après la formule suivante:

$$CN_i = \frac{1}{A_{TS}} \sum_{j=1}^4 CN_j A_j \quad (3.7)$$

où j est le type de sol selon la classification SCS (1=A, 2=B, 3=C, 4=D),
A_j la superficie de chaque type de sol dans l'unité de paysage K pour l'utilisation du sol i et A_{TS} la superficie totale de l'utilisation des sols i

dans l'unité de paysage K. Tous les calculs de superficies sont faits par le système d'information géographique et intégrés dans la base de données D-BASE.

2. Le coefficient de ruissellement moyen pour l'unité de paysage est alors calculé par l'équation suivante:

$$\overline{CN}_k = \frac{1}{A_{TK}} \sum_{i=1}^n CN_i A_i \quad (3.8)$$

Dans cette équation, CN_i représente les résultats de l'équation précédente pour chaque utilisation du sol i de l'unité de paysage k . Le terme A_i est la superficie de l'utilisation du sol i et A_{TK} la superficie totale de l'unité de paysage. Ces superficies ont été calculées par le système d'information géographique.

Les calculs de coefficients de ruissellement ont déjà été effectués pour le bassin de la rivière Samson et ils sont utilisés dans le modèle d'écoulement. Cependant, il serait aisé d'intégrer le calcul du coefficient de ruissellement dans le système d'information géographique en éliminant la base de données dans le logiciel D-BASE.

En effet, une simple superposition matricielle des différentes couches d'information, présentes sous forme de cartes (quadtree) dans SPANS, permet d'obtenir le coefficient de ruissellement. Le procédé nécessite moins d'intrants et de manipulations, car le bassin versant n'est plus divisé en unités de paysage et

le calcul du coefficient devient une association d'un coefficient à chaque quad de la carte en utilisant l'utilisation du sol, la pente, le type de sol et le tableau 3.8.

Le niveau de quad, qui représente la densité du maillage orthogonal ou la discrétisation du bassin, doit être suffisamment élevé (14 ou 15) pour ne pas favoriser le nivelage de l'information. La discrétisation du bassin devrait être la même pour tout le processus de modélisation afin de conserver une certaine cohérence au niveau de la précision de l'information. En général, le niveau de quad le plus approprié est sélectionné et utilisé pour les différentes opérations.

Les données disponibles, leur source ainsi que leur mode d'intégration dans le système SPANS sont présentées au tableau 3.9. Ces données initiales doivent être transformées pour obtenir la base de données nécessaire au calcul du coefficient de ruissellement dans le système SPANS soit, les cartes numériques (quadtree) de l'utilisation du sol, du type de sol et de la pente avec les classes du tableau 3.10. Dans SPANS, chaque carte possède une légende où chaque item de la légende est une classe de la carte.

À la suite de leur intégration dans SPANS, les cartes d'utilisation du sol et du type de sol sont reclassifiées en fonction des classes définies dans le tableau 3.10. Par exemple, la carte pédologique numérisée est classée, au départ, selon des attributs de type de sol conventionnels (argile, affleurement, sable, etc.) et elle

Tableau 3.9 Base de donnée du système d'information géographique pour obtenir le coefficient de ruissellement (CN)

Composante	Source de l'information	Traitement pour l'intégration des données dans SPANS
Utilisation des sols	Images satellitaires LANDSAT	Fichier numérique obtenu de l'image
Type de sol	Carte pédologique	Numérisation des types de sol
Élévation	Cartes topographiques 1:50 000	Numérisation des courbes de niveau, extraction de points d'élévation et modèle numérique d'élévation de terrain

Tableau 3.10 Classes des cartes numériques

Cartes	Classes pour chaque carte	
Carte de l'utilisation des sols	Classe 1	Culture intensive: maïs, céréales, cultures et condition hydrologique pauvre
	Classe 2	Culture extensive: friche et bonne condition hydrologique
	Classe 3	Boisé: forêt et bonne condition hydrologique
	Classe 4	Résidentiel, commercial peu dense: urbain
	Classe 5	Zones humide: tourbière, marécage
Carte des types de sol	Classe 1	A
	Classe 2	B
	Classe 3	C
	Classe 4	D
Carte des pentes	Classe 1	<3%
	Classe 2	3-8%
	Classe 3	>8%

doit être reclassifiée pour que les attributs deviennent les groupes de sol (A, B, C et D) du SCS. La reclassification se fait en deux étapes. La première étape consiste à créer un patron de relation entre les types de sol et les groupes de

solc'est-à-dire une table qui contient d'un coté le type de sol et de l'autre, le groupe de sol. Ensuite SPANS procède à la reclassification en fonction du patron et affiche la nouvelle carte qui contient les quatre nouvelles classes des groupes de sol

L'obtention de la carte d'élévation n'est pas une reclassification, mais une modélisation à partir d'une carte topographique numérique. Les courbes de niveau de la carte papier sont numérisées sous forme vectorielle et l'élévation de chaque courbe de niveau est l'attribut des vecteurs. Une série de points (géographiques et attributs) est prélevée sur chaque vecteur, selon une densité spécifiée, et sert à créer un modèle numérique d'élévation de terrain à l'aide d'un réseau de triangle irrégulier (*Triangular Irregular Network, TIN*) qui devient la carte numérique de la topographie.

La carte de la pente est obtenue à partir de la carte de topographie en calculant la pente maximale pour chaque quad à partir de l'élévation des huit quads voisins. Cette carte de pente est reclassifiée pour obtenir les trois classes décrites au tableau 3.10. Les fonctions d'interpolation par réseau de triangles irréguliers, de calcul de pente et de reclassification font partie des fonctions standards du système SPANS.

La base de données étant complète, il est possible de déterminer le coefficient de ruissellement pour chaque quad. La figure 3.3 explique la séquence des

opérations. La première opération consiste à superposer, selon une matrice de superposition (tableau 3.11), la carte d'utilisation des sols et la carte de la pente. La carte intermédiaire obtenue contient onze classes définies dans la matrice du tableau 3.11. Les classes numéro dix et onze sont répétées car, pour ces deux utilisations du sol, il n'existe pas de variation du coefficient de ruissellement en fonction de la pente (voir le tableau 3.8).

Tableau 3.11 Première matrice de superposition

Classes de la carte intermédiaire	<3%	3-8%	>8%
Culture intensive	1	2	3
Culture extensive	4	5	6
Boisé	7	8	9
Zones urbaines	10	10	10
Zones humides	11	11	11

La deuxième et dernière opération est aussi une superposition matricielle, mais cette fois ce sont la carte intermédiaire obtenue de l'opération précédente et la carte du type de sol qui sont superposées. La matrice de superposition est présentée au tableau 3.12. Les classes de la carte résultante correspondent directement au coefficient de ruissellement tel que défini dans le tableau 3.8.

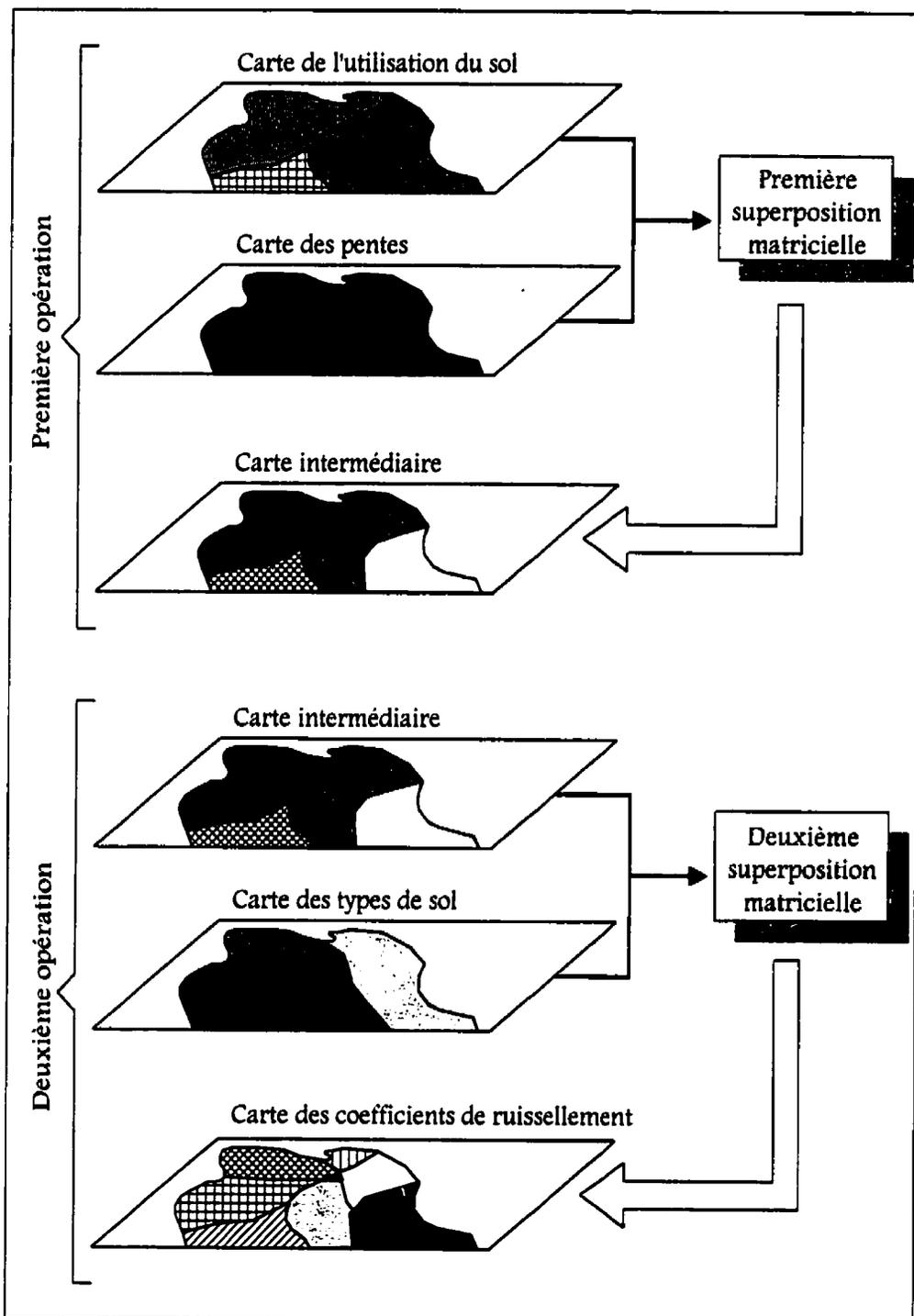


Figure 3.3 Méthode SPANS pour le calcul du coefficient de ruissellement du SCS.

Tableau 3.12 Deuxième matrice de superposition

Coefficients de ruissellement (CN)	A	B	C	D
1	63	74	80	82
2	65	76	84	88
3	72	81	88	91
4	25	40	70	78
5	39	61	74	80
6	49	39	79	84
7	22	53	65	74
8	25	55	70	77
9	41	63	75	81
10	59	74	82	86
11	22	22	22	22

3.2 Modèle d'écoulement

La modélisation des processus hydrologiques, par son calcul du coefficient de ruissellement, permet de déterminer la pluie effective, ou la quantité d'eau, qui ruisselle sur le bassin. Le modèle d'écoulement à élaborer doit incorporer les données de précipitations et les coefficients de ruissellement, établir le réseau de drainage, simuler la propagation de la pluie effective jusqu'à l'exutoire du bassin versant et calculer le débit de la rivière. Cette problématique est illustrée à la figure 3.4.

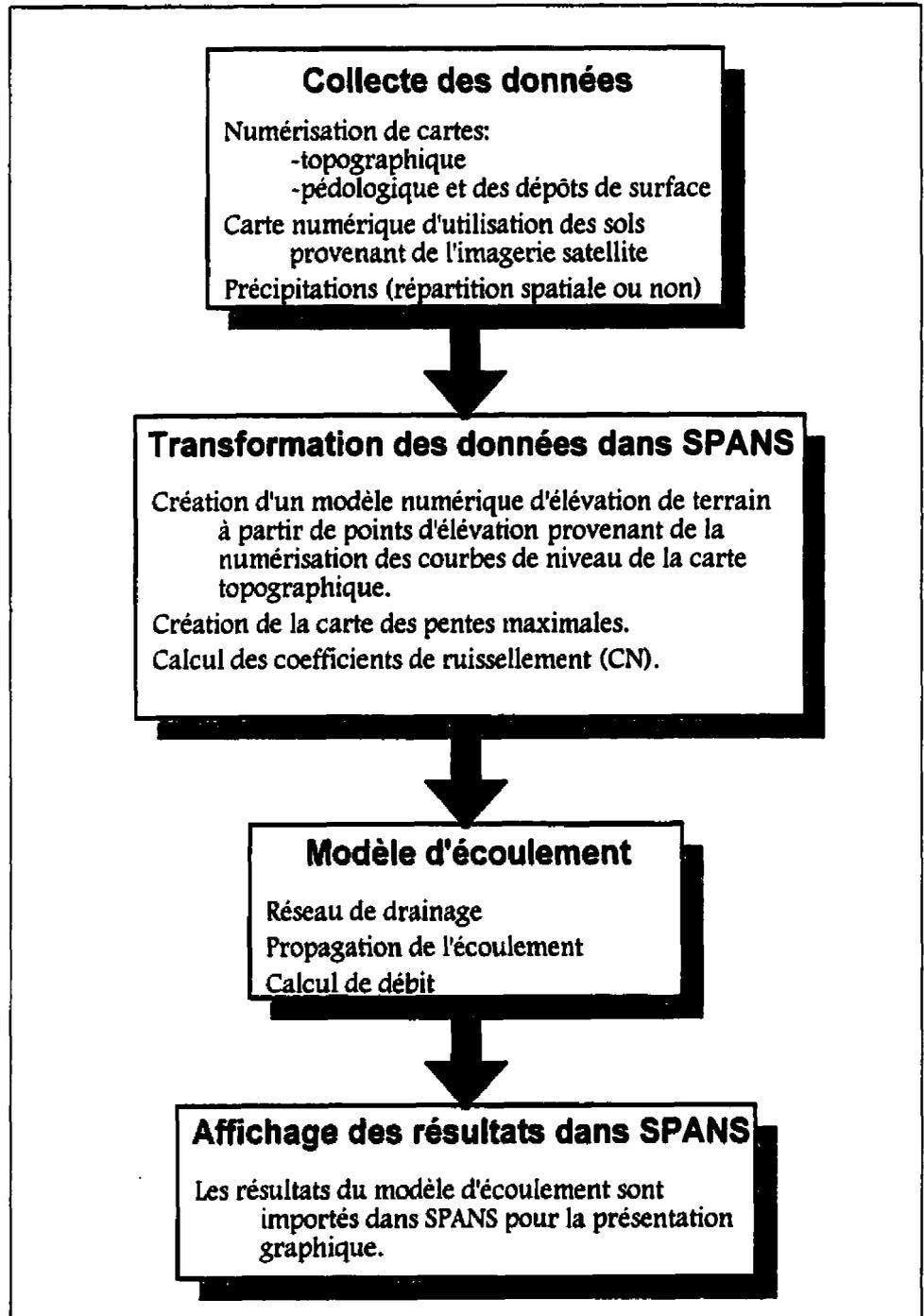


Figure 3.4 Algorithme de résolution de la modélisation d'un petit bassin versant naturel et non jaugeé.

L'utilisation des systèmes d'information géographique donne accès à des données à référence spatiale pour les coefficients de ruissellement et les données sur les précipitations lorsque celles-ci proviennent de la télédétection. La répartition spatiale des données de précipitation est certes un atout, mais pas vraiment une nécessité dans le cas d'un calcul de débit événementiel sur un petit bassin versant.

Un modèle d'écoulement discrétisé, et dont la discrétisation correspond à celle des données d'intrant, permet de tirer profit des systèmes d'information géographique, de conserver le caractère spatial des données lors du traitement et facilite son intégration éventuelle à l'intérieur du système. De plus, les résultats obtenus peuvent être réimportés dans le système d'information géographique pour la présentation.

Plusieurs auteurs (Martz et Garbrecht, 1993; Tachikawa *et al.*, 1994; Garg et Sen, 1994; Costa-Cabral et Burges, 1994) proposent des méthodes d'extraction des réseaux de drainage à partir des modèles numériques d'élévation de terrain, donc en faisant une discrétisation du territoire à l'étude. Cependant la méthode proposée par Costa-Cabral et Burges, est très intéressante parce qu'elle s'adapte bien aux données de télédétection et à la méthode des coefficients de ruissellement utilisée, elle utilise un modèle numérique d'élévation de terrain et

l'écoulement est généré en deux dimensions. De plus, les auteurs fournissent sur demande les codes, dans le langage FORTRAN, de leur programme DEMON, pour *Digital Elevation Model Network*.

3.2.1 Le programme DEMON (*Digital Elevation Model Network*)

Conçu par Mariza C. Costa-Cabral et Stephen J. Burges du département de génie civil de l'Université de Washington à Seattle (Costa-Cabral et Burges, 1994), ce programme divise le territoire en un maillage orthogonal et calcule les surfaces d'apport à partir d'un modèle d'élévation de terrain pour chaque élément de surface du maillage nommé cellule. L'intégrale de l'article se retrouve à l'annexe I. Ce nouvel algorithme évite les problèmes rencontrés par les concepteurs des algorithmes précédents en représentant l'écoulement en deux dimensions et en dirigeant cet écoulement selon l'orientation de la pente maximale des cellules du modèle d'élévation de terrain. Il calcule non seulement les surfaces d'apport, mais aussi les surfaces de dispersion.

La surface totale d'apport (STA ou TCA: *Total Contributing Area*) d'une courbe de niveau quelconque correspond à la superficie de la projection sur un plan du terrain qui se draine vers cette courbe de niveau (figure 3.5).

La surface spécifique d'apport (SSA ou SCA: *Specific Contributing Area*) est équivalente à une surface totale d'apport pour une unité de longueur de courbe

de niveau, ω . En d'autre terme le *SSA* est égal au *STA* divisé par la largeur de l'écoulement.

$$SSA = \frac{STA}{\omega} \quad (3.9)$$

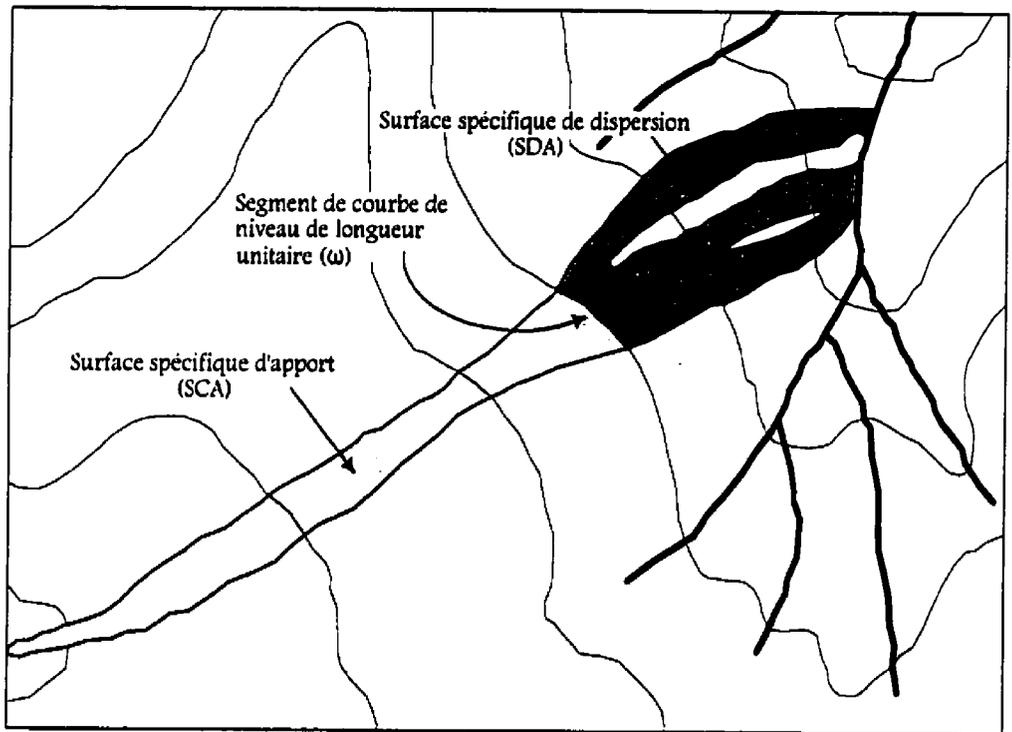


Figure 3.5 Illustration des concepts de surface spécifique d'apport et de surface spécifique de dispersion d'un segment de courbe de niveau d'une longueur unitaire (Costa-Cabral et Burges, 1994).

De même, la surface totale de dispersion (STD ou TDA: *Total Dispersal Area*) d'une courbe de niveau est la superficie de la projection plane du terrain qui draine l'écoulement passant par cette courbe de niveau. La surface spécifique de

dispersion (SSD ou SDA: *Specific Dispersal Area*) correspond à la surface totale de dispersion pour une unité de longueur de la courbe de niveau.

$$SSD = \frac{STD}{\omega} \quad (3.10)$$

Les surfaces d'apport, totales ou spécifiques, se situent en amont d'une courbe de niveau tandis que les surfaces de dispersion correspondantes se situent en aval de cette même courbe de niveau.

Dans les équations 3.9 et 3.10 la largeur totale de l'écoulement (ω) correspond à la largeur de la bande de fluide mesurée perpendiculairement à la direction de l'écoulement. La direction de l'écoulement correspond à l'orientation de la frontière de la cellule par laquelle l'eau sort de la cellule. La portion d'un segment par laquelle l'eau s'écoule est appelée segment mouillé. En supposant un modèle d'élévation de terrain où l'axe des x et l'axe des y pointent respectivement vers l'est et le nord, et une cellule de dimension Δx et Δy , l'équation de la largeur de l'écoulement est la suivante:

$$\begin{aligned} \omega &= \int_{l_v} \frac{\partial \omega}{\partial l} dl + \int_{l_e} \frac{\partial \omega}{\partial l} dl + \int_{l_s} \frac{\partial \omega}{\partial l} dl + \int_{l_o} \frac{\partial \omega}{\partial l} dl \\ &= \int_{l_v} \sin \alpha(x, y = \Delta y) dx + \int_{l_e} \cos \alpha(x = \Delta x, y) dy \\ &\quad - \int_{l_s} \sin \alpha(x, y = 0) dx - \int_{l_o} \cos \alpha(x = 0, y) dy \end{aligned} \quad (3.11)$$

Les variables l_N , l_E , l_S et l_O représentent la longueur du segment mouillé qui repose sur la frontière *nord*, *sud*, *est* ou *ouest* et $\alpha(x, y)$ est l'angle de l'écoulement mesuré dans le sens contraire des aiguilles d'une montre à partir de l'*est*. La longueur des segments au *nord* et au *sud* se situe donc entre zéro et Δx tandis que celle des segments à l'*est* et à l'*ouest* varie entre zéro et Δy .

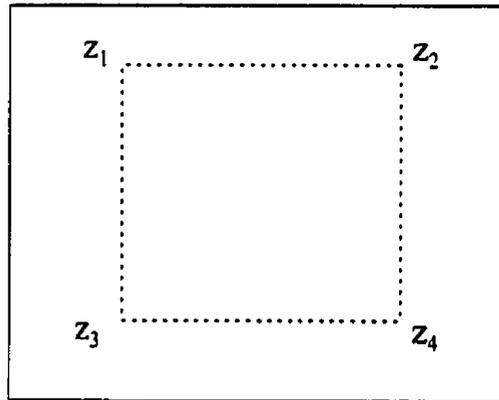
Il est important de souligner que le fait de travailler sur des projections planes du terrain soutient l'hypothèse d'une direction d'écoulement qui n'est pas fonction de la profondeur de l'écoulement, i.e. de la hauteur d'eau, mais seulement fonction de la topographie. La hauteur d'eau intervient seulement dans le calcul de la vitesse et non dans le calcul de l'orientation de l'écoulement.

3.2.1.1 Représentation de l'écoulement dans le programme DEMON

Dans l'algorithme du programme DEMON, l'écoulement sur une cellule se fait selon l'orientation de la pente maximale de cette cellule. L'orientation de la pente maximale d'une cellule, $\alpha(x, y)$, un point correspond à l'opposé de la direction du tenseur d'élévation en ce même point, calculé à partir des points du modèle numérique d'élévation de terrain (*DEM*). Le tenseur d'élévation est $a\bar{i} + b\bar{j}$ où les vecteurs \bar{i} et \bar{j} sont orientés selon l'axe des x et des y respectivement et où les variables a et b sont définies comme suit :

$$\begin{aligned}
 a &= \frac{1}{2\Delta x} (-z_1 + z_2 + z_3 - z_4) \\
 b &= \frac{1}{2\Delta y} (z_1 + z_2 - z_3 - z_4)
 \end{aligned}
 \tag{3.12}$$

Les variables z sont représentées à la figure suivante



. **Figure 3.6** Identification des points d'élévation sur une cellule.

L'orientation du vecteur d'écoulement est donc donnée par l'équation $-(a\bar{i} + b\bar{j})$ et l'angle que forme ce vecteur avec l'abscisse dans le sens contraire des aiguilles d'une montre est désigné par la lettre α , qui est l'orientation de la pente maximale sur une cellule. La connaissance de cet angle α permet de calculer la largeur de l'écoulement sur la cellule par l'équation suivante.

$$\omega = |\sin\alpha|\Delta x + |\cos\alpha|\Delta y
 \tag{3.13}$$

L'algorithme crée alors deux matrices initiales. Celle des angles d'écoulement α pour chacune des cellules et celle des largeurs d'écoulement. Elles sont toutes deux nécessaires au calcul des *SSA*, *SSD*, *STA* et *STD*, qui se fait de l'amont vers l'aval. C'est alors que commence la définition du réseau de drainage. La figure 3.7 illustre l'algorithme du programme DEMON et la figure 3.8 représente l'organigramme de la fonction *TCA_AND_TDA*.

3.2.1.2 L'algorithme du programme DEMON

La première fonction du programme lit le fichier contenant le modèle d'élévation de terrain (*DEM*) et l'emmagasine dans une variable matrice. Ensuite, la fonction *PITS* repère les puits dans le modèle d'élévation de terrain. Une cellule est un puits si ses huit voisines, cardinales et diagonales, ont une élévation supérieure à la sienne. Cette fonction crée aussi un cadre de cellule puits entourant l'ensemble de la matrice d'élévation pour éviter un manque d'information ou l'entrée dans une boucle infinie lors de l'exécution du programme.

La fonction *ANGLES* s'occupe de préparer la matrice des angles de direction de l'écoulement selon la méthode décrite ci-haut. Tous les intrants sont maintenant en place pour le calcul des surfaces d'apport et de dispersion.

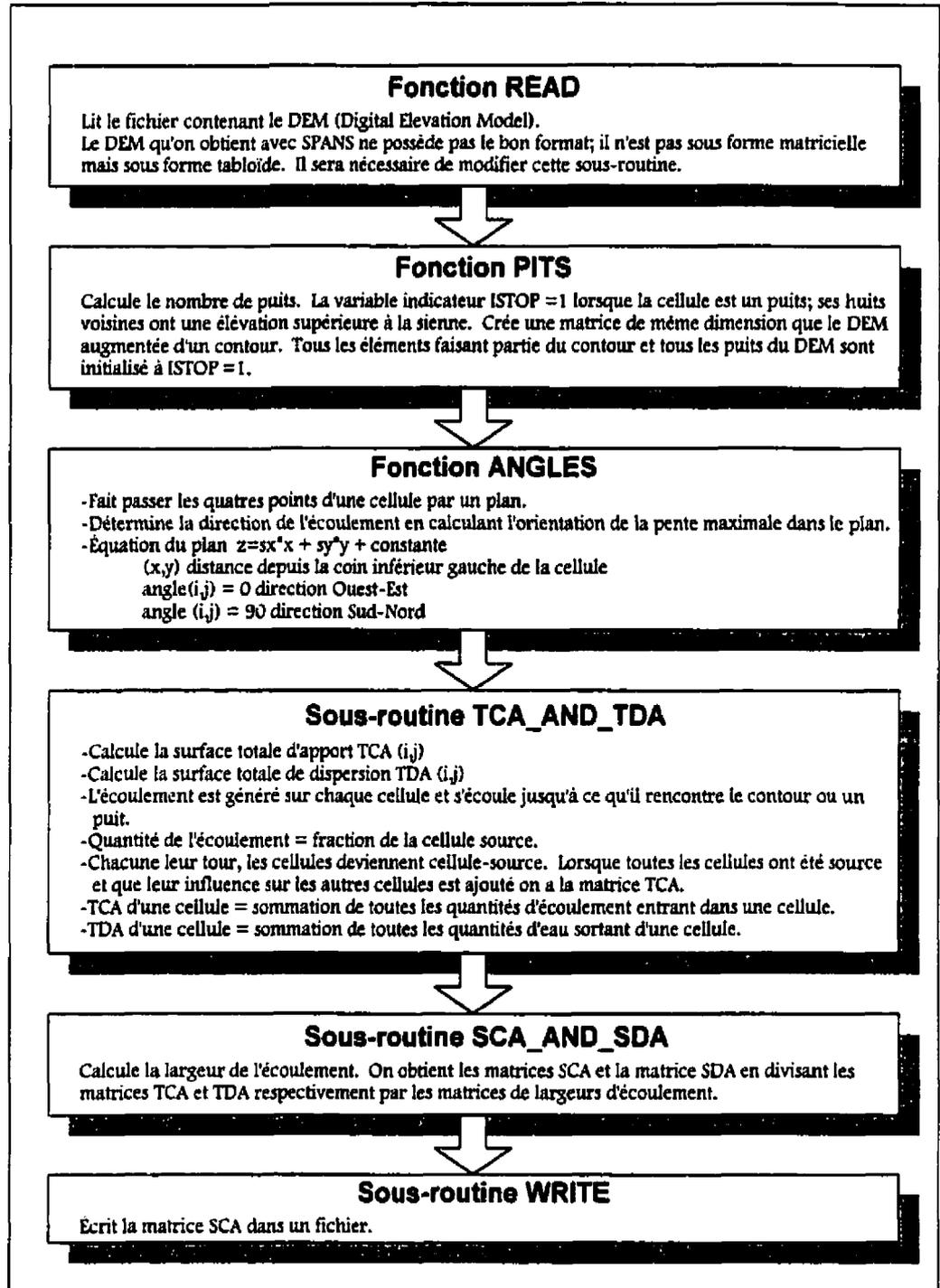


Figure 3.7 Organigramme du programme DEMON.

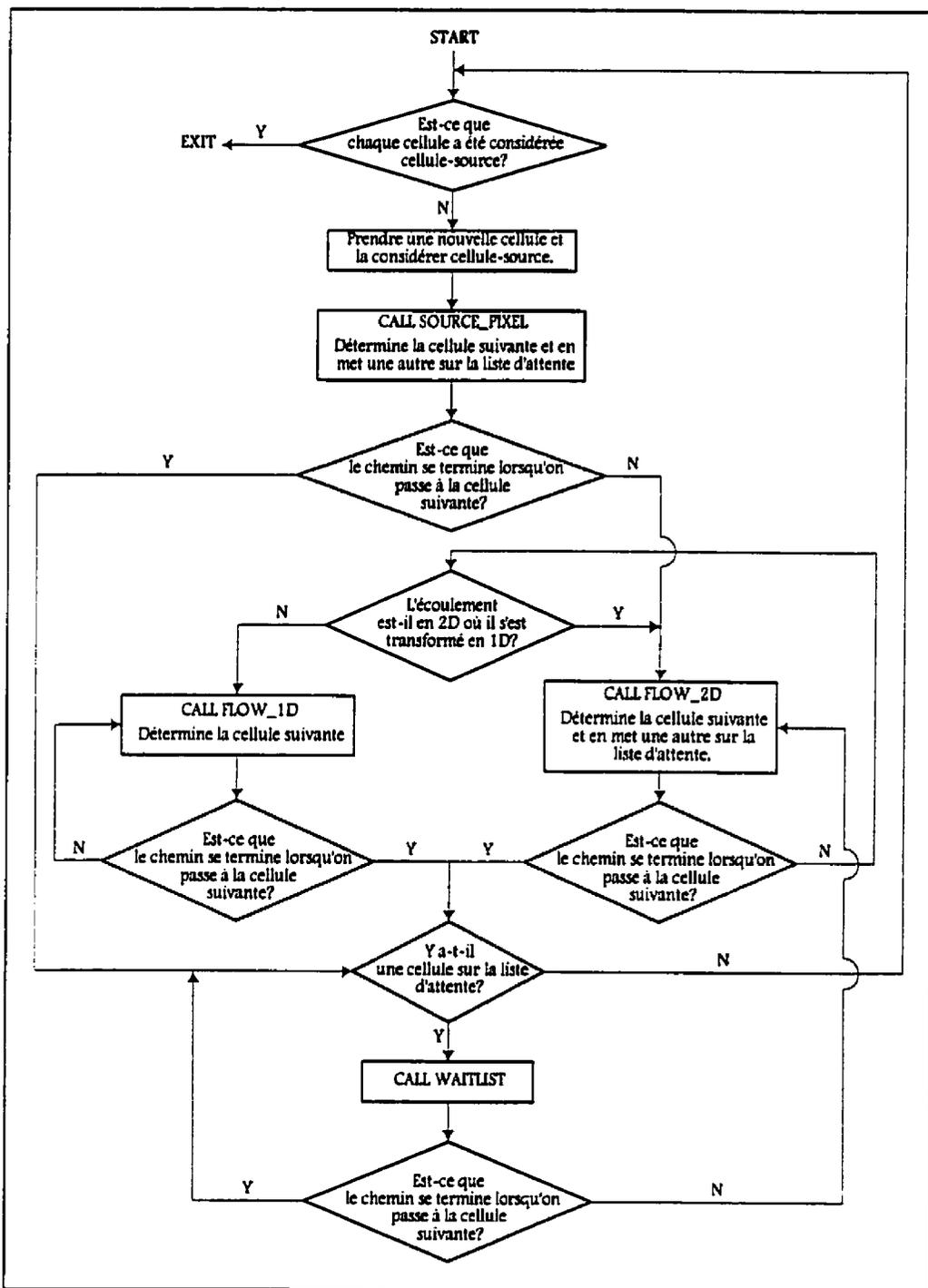


Figure 3.8 Organigramme de la fonction de calcul des surfaces totales d'apport et de dispersion: TCA_AND_TDA.

La fonction qui se charge de ce calcul s'appelle *TCA_AND_TDA*. Elle définit, en même temps qu'elle calcule les surfaces d'apport et de dispersion, le réseau de drainage. Elle génère d'abord, sur la cellule source, une surface d'écoulement. Elle suit, de l'amont vers l'aval, cette surface d'écoulement en deux dimensions, tel un ruban d'écoulement, en respectant l'orientation de la pente maximale de chacune des cellules. Elle crée ainsi la matrice d'influence de la cellule source. Son trajet se termine lorsqu'elle rencontre une cellule puits. La figure 3.9(a) donne un exemple d'écoulement en deux dimensions respectant l'orientation de la pente maximale des cellules. La figure 3.9(b) est un exemple d'une matrice d'influence de la cellule source (1,1) tandis que la figure 3.9(c) indique ce que signifient les chiffres dans cette matrice d'influence. La matrice d'influence contient la portion d'écoulement que chaque cellule reçoit de la cellule source. La sommation de toutes les matrices d'influence donne la matrice des surfaces totales d'apport. La matrice *SSA* est simplement la matrice *STA* divisée par la largeur de l'écoulement calculée selon l'équation 3.13.

L'écoulement généré sur la surface d'une cellule se déverse dans une seule cellule réceptrice si l'angle d'écoulement α est un multiple de 90° . Dans le cas contraire, le déversement se fait dans deux cellules réceptrices et cardinales. Une cellule diagonale n'est jamais une cellule potentiellement réceptrice, car les deux cellules n'ont qu'un seul point en commun et un écoulement bidimensionnel à travers un point est une aberration.

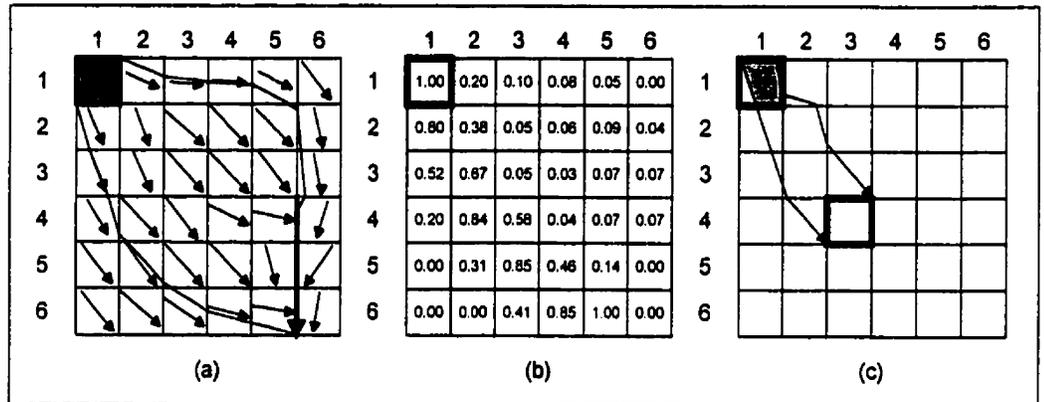


Figure 3.9 (a) Exemple d'un écoulement bidimensionnel respectant l'orientation de la pente maximale des cellules. L'écoulement est généré sur la cellule (1,1). La surface en gris représente la STD de la cellule (1,1). (b) Matrice d'influence de la cellule (1,1). Le chiffre de chacune des cellules de la matrice d'influence représente la fraction de la cellule source qui est drainée par ces cellules. (c) Signification physique de la valeur 0.58 de la cellule (4,3) de la figure (b) (Costa-Cabrall et Burges, 1994).

La portion de l'écoulement généré se dirigeant dans chacune des cellules réceptrices se calcule en fonction de l'angle α . La figure 3.10 donne un exemple d'écoulement avec deux cellules réceptrices; la cellule *est* et la cellule *sud*. L'angle α mesure 292° .

La fraction *est* (f_E) de l'écoulement correspond à l'aire du triangle formé par la droite d'angle α passant par le coin inférieur de la cellule et les segments *est* et *nord* de la cellule, sur l'aire totale de la cellule.

$$f_E = \frac{A_\Delta}{A} = -\frac{1}{2} \frac{\Delta y}{\Delta x} \frac{1}{\tan(292^\circ)} = 0.20 \quad (3.14)$$

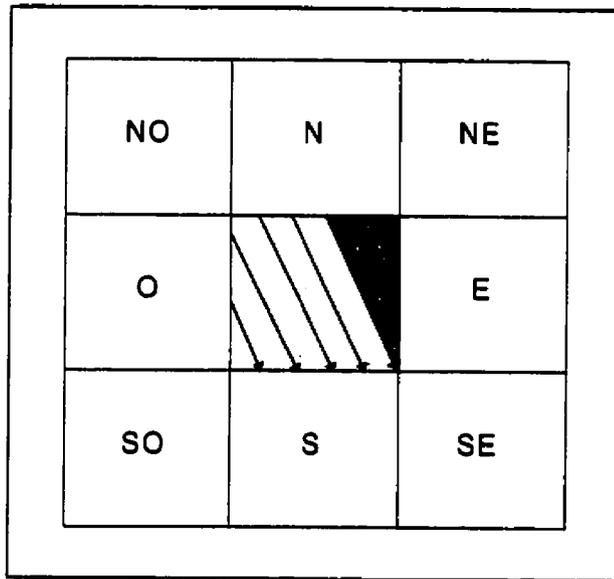


Figure 3.10 Illustration de la répartition de l'écoulement d'une cellule source dans les deux cellules voisines et cardinales (Costal et Burges, 1994).

L'aire de la partie *sud*, ayant la forme d'un trapèze quelconque, peut se calculer par la simple soustraction de l'aire totale moins l'aire du triangle.

$$f_s = 1 - f_e = 0.80 \quad (3.15)$$

Tout en exécutant cette opération, le programme repère les cellules suivantes dans le réseau de drainage. Comme un seul chemin peut être traité à la fois, une des deux cellules est placée sur une liste d'attente. Un chemin se termine à la sortie du modèle d'élévation de terrain ou en présence d'un puits. Lorsqu'un chemin est terminé, le programme reprend un autre chemin en utilisant la

première cellule sur la liste d'attente. Ce cheminement se poursuit jusqu'à ce qu'il n'y ait plus de cellule sur la liste d'attente. La figure 3.11 représente symboliquement les différents chemins empruntés lors du calcul de la matrice d'influence de la cellule source (1,1).

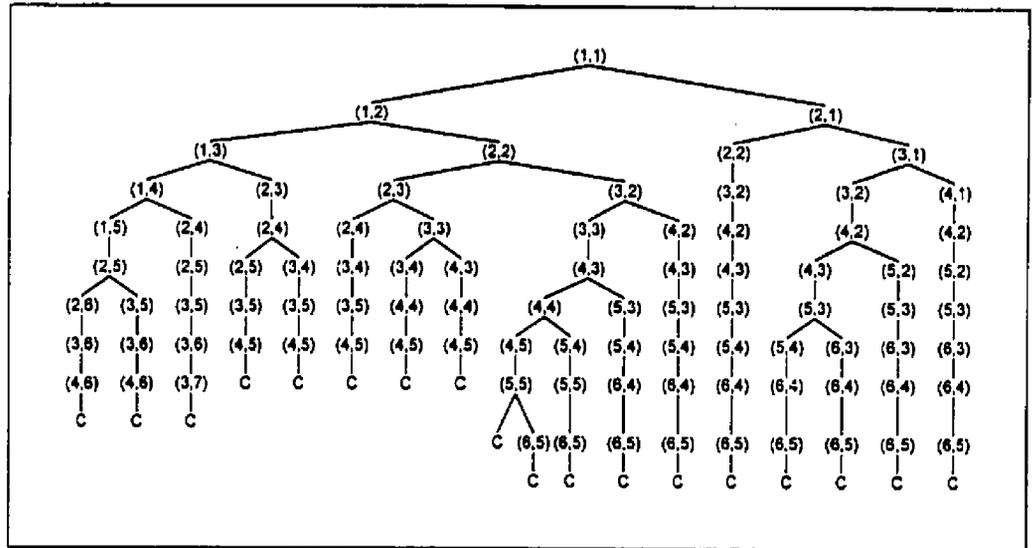


Figure 3.11 Représentation schématique des trajets de l'écoulement. Les embranchements correspondent à un écoulement réparti entre deux cellules voisines. Tous les trajets quittent le modèle d'élévation de terrain sous une forme concentrée indiquée par la lettre "C" (Costa-Cabral et Burges, 1994).

Lorsque la liste d'attente est vide, le programme revient au début de la fonction et refait le même cheminement avec une nouvelle cellule source. À tour de rôle, chacune des cellules devient une cellule source. Une fois que toutes les cellules ont été cellule source, le programme se termine et la matrice *STA* des surfaces totales d'apport est obtenue. En suivant l'écoulement de cellule en cellule, le

programme incrémente la matrice contenant le volume d'eau drainée par les cellules. La quantité totale d'eau drainée est égale au *STA* de cette cellule.

Au cours du cheminement de cellule en cellule, il faut vérifier la convergence de l'écoulement. En effet, l'écoulement peut devenir unidimensionnel s'il y a convergence topographique. Cette convergence ne se situe pas nécessairement à la frontière d'une cellule. La flèche en caractère gras de la figure 3.9(a) est un exemple d'écoulement devenu unidimensionnel. Le programme évalue plusieurs variables décrivant la géométrie du mouvement de l'écoulement d'une cellule à l'autre.

La connaissance de ces variables permet d'adapter le calcul des surfaces d'apport et de dispersion à la géométrie de l'écoulement. La figure 3.12 illustre ces variables qui sont nécessaires à la projection de l'écoulement vers l'aval. La lettre *F* désigne l'écoulement entrant dans la cellule (4,2). Il est représenté par la zone ombragée de la cellule source (1,1). Une partie de l'écoulement de la cellule (4,2) se déverse dans la cellule (5,2) et l'autre partie dans la cellule (4,3). Les fractions de l'écoulement allant d'un côté ou de l'autre seront calculées en utilisant les variables f_1 , f_2 et f_3 . Les variables f_1 et f_2 sont donc les longueurs drainées par les lignes d'écoulement formant la bande. Lorsque $f_2 = f_3$, tous les points à l'intérieur de la bande d'écoulement et à droite de la ligne pointillée drainent la même quantité d'eau tandis que l'écoulement à gauche de la ligne

pointillée varie linéairement à travers la bande d'écoulement. L'écoulement entrant par la frontière *nord, est, sud* ou *ouest* est désigné par la lettre $I = 1, 2, 3$ ou 4 respectivement. x_1 et x_2 indiquent les points d'entrée dans la cellule des lignes délimitant l'écoulement. Finalement, x_3 indique le point d'entrée de la ligne centrale de l'écoulement (la ligne pointillée) générée à la cellule source. Le listage du programme est à l'annexe II. On y retrouve les équations dans lesquelles apparaissent ces différentes variables.

Le calcul de la surface totale de dispersion se fait parallèlement au calcul de la surface totale d'apport. Lors de la projection de l'écoulement vers l'aval, la surface qu'occupe la bande d'écoulement dans chacune des cellules est additionnée. Ainsi la matrice *STD* est obtenue. Pour obtenir la matrice des *SSD*, il suffit de diviser la matrice *STD* par la matrice des largeurs d'écoulement, ω .

3.2.2 Modifications à apporter au programme DEMON

Dans sa forme actuelle, le seul intrant nécessaire au programme DEMON est un fichier contenant une matrice de l'élévation de chaque cellule qui constitue le modèle numérique d'élévation de terrain. Le programme calcule ensuite dans la sous-routine *ANGLES* l'orientation de la pente maximale sur chaque cellule. Dans le but de maximiser l'intégration du modèle dans le système d'information géographique, le calcul de l'orientation des pentes maximales sur chaque cellule est fait dans SPANS, car il possède une commande qui exécute ce calcul. La

routine *ANGLES* peut être éliminée et les commandes permettant de lire la matrice de l'orientation des pentes du terrain ajoutées à la routine *READ*.

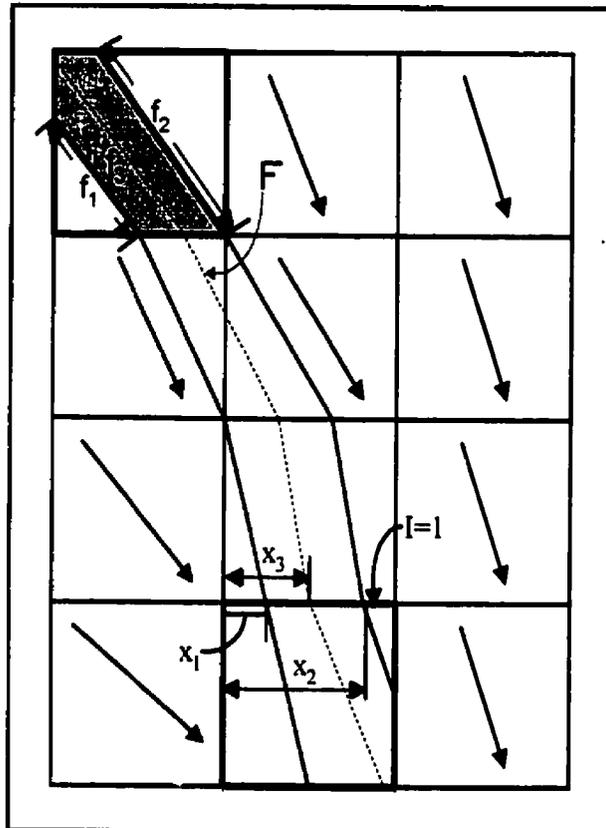


Figure 3.12 Illustration des définitions des variables F , f_1 , f_2 , f_3 , x_1 , x_2 , x_3 et I , utilisées pour la projection de l'écoulement à travers la cellule (4,2) (Costal et Burges, 1994).

La sous-routine *PIXEL_SOURCE* génère un écoulement unitaire, qui correspond à la surface de la cellule, soit Δx multiplié par Δy , sur la cellule source. Cette quantité est ensuite acheminée selon les règles et les calculs décrits à la section précédente vers les autres cellules jusqu'à un puits quelconque. Il est possible de

multiplier le calcul de cette surface par une hauteur d'eau de précipitation pour obtenir un volume d'eau de pluie qui tombe sur la cellule.

Par cette multiplication, les surfaces d'apport deviennent des volumes d'eau d'apport et la matrice des surfaces totales d'apport la matrice des volumes d'eau totaux d'apports. En principe, le résultat de la matrice des volumes totaux sur la cellule qui se trouve à l'exutoire contient le volume d'eau total de ruissellement. Pour tenir compte de la capacité de rétention du terrain, la matrice des coefficients de ruissellement est intégrée au calcul et chaque volume d'eau qui arrive sur une cellule est multiplié par ce coefficient pour simuler l'infiltration.

Cependant, pour obtenir un débit, il faut une dimension temporelle au calcul. Après l'entière exécution du programme, c'est-à-dire après que chaque cellule a été cellule source, il s'est écoulé un temps de concentration du bassin versant. Le calcul du temps de concentration du bassin peut se faire en faisant la somme du temps de parcours de l'écoulement sur chaque cellule appartenant au trajet le plus long, en supposant que ce dernier correspond au trajet de la cellule la plus éloignée de l'exutoire, jusqu'à l'exutoire.

Le calcul du temps de parcours de l'écoulement sur chaque cellule est le résultat de l'équation de Manning en supposant un écoulement uniforme sur les cellules.

$$V = \frac{R^{2/3} S^{1/2}}{n} \quad (3.16)$$

où V est la vitesse de l'écoulement, R le rayon hydraulique estimé par la hauteur d'eau sur une cellule, S la pente de l'écoulement qui correspond à la pente maximale du terrain sur chaque cellule et n est le coefficient de Manning. Deux coefficients de Manning sont utilisés. Le premier pour les écoulements sur le terrain dans la sous-routine *FLOW-2D* et le second dans les canaux dans la sous-routine *FLOW-1D*. Les effets de laminage qu'entraîne la présence d'un lac dans un réseau hydrologique sont négligés, mais l'impact de cette simplification est quand même petit sur l'ensemble du phénomène. La valeur précise des coefficients sera déterminée lors du calage du modèle, ils serviront d'ajustement pour les phénomènes qui sont négligés ou qui ne sont pas modélisés.

Dans l'équation 3.16, R est égal au volume d'eau total divisé par la largeur de l'écoulement et la longueur du segment d'écoulement sur chaque cellule.

$$R = \frac{Vol}{\omega * L} \quad (3.17)$$

Le temps de parcours du fluide sur une cellule est donc égal à la longueur du segment sur la vitesse de Manning.

$$T = \frac{L}{V} \quad (3.18)$$

Ce calcul est effectué sur chacune des cellules faisant partie de tous les trajets. La sommation des temps sur chaque chemin donne le trajet le plus long en

temps qui correspond au temps de concentration du bassin. La longueur du segment fait déjà partie des variables calculées par DEMON, c'est la variable f_3 de la figure 3.12.

Dans ce modèle d'écoulement, la délimitation du bassin n'est plus nécessaire puisque les cellules qui ne font pas partie du bassin ne drainent pas vers l'exutoire du bassin. Elles n'ont pas d'influence sur les cellules faisant partie du bassin, car l'orientation de leur pente maximale est externe au bassin. La délimitation du bassin versant est automatique dans cet algorithme.

Il serait aussi possible d'ajouter une matrice contenant des données sur la répartition spatiale des précipitations. Il suffit d'ajouter une carte des précipitations dans SPANS et d'inclure un champ des précipitations dans le fichier de points. Dans le programme DEMON, la multiplication par la variable *pluie* est remplacée par une multiplication par la variable matricielle *pluie(i,j)*. Dans le cas présent, les précipitations sont uniformes sur l'ensemble du bassin versant.

En plus des matrices de l'élévation et de l'orientation de la pente maximale, le modèle a besoin des matrices des coefficients de ruissellement et des pentes sur chacune des cellules. Il faut donc modifier la routine de lecture pour intégrer ces données au programme.

CHAPITRE 4 MÉTHODE ET RÉSULTATS

Dans l'algorithme de résolution de la modélisation du bassin versant, présenté à la figure 3.4, la première étape est la collecte des données. La première opération à réaliser dans cette étape est la création de l'univers du bassin versant de la rivière Samson. Créer un univers dans SPANS signifie définir l'étendue et la projection cartographique pour l'ensemble des données à intégrer dans le système. En considérant l'échelle du bassin et les données matricielles nécessaires pour le programme DEMON, il faut que la projection soit la plus orthogonale possible. La projection MTM (*Modified Transverse Mercator*) dans le datum de 1927 (NAD27) est presque parfaitement orthogonale et c'est la projection des cartes à l'échelle 1: 20 000 qui seront numérisées. L'utilisation de cette projection avec le datum de 1927, plutôt qu'avec celui de 1983, minimise la conversion des données, qui peut devenir très complexe. Les données de la rivière Chaudière, en format numérique, ainsi que les cartes topographiques, en format papier, de la région sont basées sur le datum de 1927.

Pour définir l'étendue des données, un fichier de vecteurs contenant un simple rectangle formé de quatre points et englobant le bassin versant de la rivière Samson est utilisé. Ce rectangle est par la suite « enquadé » et sert de carte de base de l'univers. Pour faciliter la discrétisation du bassin, la grandeur en

mètres des quads est fixée à un nombre exact. Par exemple, le niveau de quad 9 est un carré de 100 mètres par 100 mètres, le niveau de quad 8 de 200 mètres par 200 mètres, le niveau de quad 10 de 50 mètres par 50 mètres, etc. Le calcul d'un nombre de quad fixe se fait en fonction du nombre de pixels contenus dans l'écran.

Une fois l'univers défini, les données peuvent être importées et aucune modification subséquente des paramètres de l'univers ne peut être faite sans qu'elle n'entraîne la corruption de l'ensemble des données de l'univers. Les paramètres de l'univers de la rivière Samson sont définis à l'annexe III.

4.1 Collecte des données

4.1.1 La topographie

L'étude du bassin versant de la rivière Chaudière réalisée par TecSult (1994c) fournit un bon nombre de données notamment sur les coefficients de ruissellement calculés en fonction des unités de paysage. Cependant, les données disponibles sur l'élévation, une carte avec une courbe de niveau à tous les cinquante mètres, ne sont pas suffisamment précises pour l'échelle du bassin de la rivière Samson. Les cartes topographiques couvrant le bassin de la rivière Samson, trois cartes au total, sont numérisées en utilisant une table numérisante

et le logiciel PMTydig. Chaque courbe de niveau est un arc qui n'est pas fermé sur le cadre de la carte et les fichiers ne contiennent aucune topologie.

Par la suite, les trois fichiers de vecteurs sont importés dans SPANS séparément et sont regroupés en un seul fichier à l'aide d'un utilitaire de SPANS: l'éditeur de vecteur. Une seconde fonction de l'éditeur de vecteur est utilisée pour extraire un fichier de points qui servira à créer une carte « enquadée ». Le fichier de points est importé à son tour dans SPANS en deux étapes. La première étape convertit le fichier de points en table d'attributs et la seconde donne une géoréférence à chaque attribut de la table. La modélisation numérique du terrain par un réseau de triangles irréguliers (TIN, *Triangular Irregular Network*) sert à créer la carte « enquadée » de l'élévation. La fonction SPANS qui fait cette modélisation numérique de terrain est la suivante:

*TRANSFORM / Data Types / Points to Map / Contouring (INTERA TYDAC,
SPANS Reference Manuel, 1993)*

À partir d'une carte d'élévation, il est possible, dans SPANS, de générer une carte des pentes maximales ainsi qu'une carte de l'orientation de ces pentes. Les commandes SPANS pour ces deux opérations sont respectivement:

MODEL / Surface / Neighbor / Slope

MODEL / Surface / Neighbor / Aspect

4.1.2 Les coefficients de ruissellement

La carte des coefficients de ruissellement en format SPANS existe déjà, mais dans un autre univers, celui de la rivière Chaudière qui est en coordonnées UTM (*Universal Transverse Mercator*) dans le datum de 1927. Elle doit être récupérée puis intégrée dans l'univers de la rivière Samson.

La carte de base de l'univers du bassin de la rivière Samson, le rectangle, est transformée en format vectoriel pour être exportée dans un fichier ASCII en latitude/longitude. Ce même fichier est importé dans l'univers du bassin versant de la rivière Chaudière et « enquadé » pour devenir une carte. Cette dernière est utilisée pour découper la carte des coefficients de ruissellement du bassin de la rivière Chaudière. Le résultat est une nouvelle carte possédant la même forme que la carte de base du bassin de la rivière Samson avec les attributs de la carte des coefficients de ruissellement. Il reste à faire le cheminement inverse pour intégrer la carte des coefficients de ruissellement dans l'univers de la rivière Samson. La carte des coefficients de ruissellement est transformée en format vectoriel, exportée en ASCII en latitude/longitude, puis importée dans l'univers Samson, où les vecteurs sont de nouveau « enquadés » en une carte contenant les classes de coefficient de ruissellement pour les différentes unités de paysage. Le passage des vecteurs en coordonnées de

latitude et de longitude permet de passer d'une projection à une autre sans fausser les données.

La méthode de calcul des coefficients de ruissellement est complètement indépendante du fonctionnement du programme DEMON. En effet, le calcul peut être fait en utilisant un découpage orthogonal, en unités de paysage ou tout autre découpage puisque le format des données qui entrent dans DEMON provient d'opérations effectuées sur la carte résultante des coefficients de ruissellement.

4.1.3 Les précipitations

Le bassin versant de la rivière Samson est suffisamment petit, moins de 200 km², pour émettre l'hypothèse d'une pluie uniforme sur l'ensemble du bassin dans le cas des averses ne provenant pas de cellules orageuses. Des données de pluie des différentes stations sur le bassin versant de la rivière Chaudière sont disponibles, mais les données de télédétection sur la répartition spatiale des précipitations sont plus difficiles à obtenir et moins accessibles. L'intégration de données de précipitation à répartition spatiale dans le logiciel SPANS se fait facilement en y important, à partir d'un format matriciel compatible, des données de télédétection qui sont ensuite « enquadées ». La carte des précipitations peut être intégrée dans le programme DEMONII (la version modifiée de DEMON) au niveau du calcul de la même manière que sont

intégrées les données de pente ou de coefficients de ruissellement. La répartition spatiale des précipitations ne constitue pas un élément clef dans la réalisation du calcul et n'est pas utilisée dans les essais. Son intégration éventuelle constituerait cependant un atout au calcul.

4.1.4 Transfert des données vers le programme DEMONII

4.1.4.1 Opérations dans SPANS

L'ensemble des données nécessaires fait maintenant partie de la base de données de SPANS sous forme de cartes thématiques « enquadées ». L'étape qui suit doit convertir ces données dans le format requis par le programme DEMONII, c'est-à-dire des fichiers contenant une matrice numérique en code ASCII. Il faut une matrice pour chaque thématique: élévation, pente, orientation de la pente maximale et coefficient de ruissellement. La figure 4.1 illustre la disposition de la matrice dans un fichier. Pour inclure une répartition spatiale des précipitations, il faudrait avoir une carte thématique des précipitations, en extraire une matrice correspondante et l'intégrer au programme DEMONII. La fonction *Read* du programme fait la lecture de ces fichiers et emmagasine les données de chaque fichier dans des variables matricielles.

Il est possible de construire dans SPANS, une table de points qui contient, en plus des coordonnées x et y en projection MTM, une colonne pour chaque thématique nécessaire. L'en-tête de la table de points est illustré à la figure 4.2.

La commande *Model / Points / Grid Sample*, utilisée pour créer la table de base, se sert d'une carte « enquadée » et d'une équation pour extraire des points sur la carte selon une certaine densité spécifiée dans l'équation. Pour obtenir une matrice qui soit rectangulaire, m rangées par n colonnes, il faut utiliser une carte de forme rectangulaire qui englobe le bassin versant soit la carte de base de l'univers.

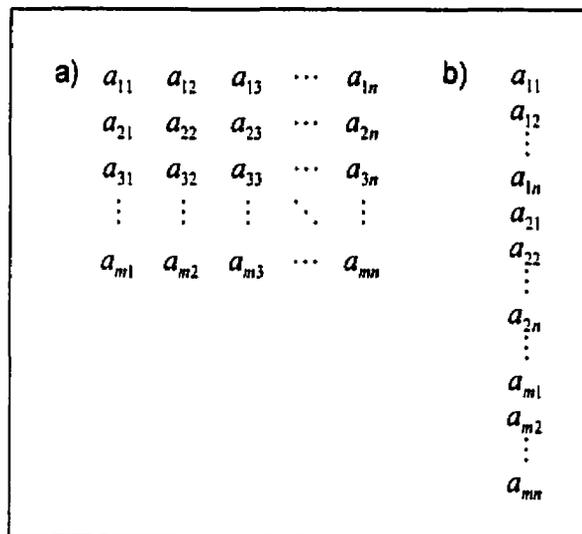


Figure 4.1 (a) Format matriciel où chaque élément a_{mn} est une thématique. (b) Disposition de la matrice dans un fichier en code ASCII.

L'équation, écrite dans le langage de modélisation de SPANS, spécifie d'extraire, selon une densité en km^2 , un point ainsi que les attributs (la classe) des cartes mentionnées, lorsque la classe de la carte de base (*basemap*) est différente de zéro. Par exemple, une densité de 1 signifie qu'un point est extrait à tous les

km² et une densité de 0.01 signifie qu'un point est extrait à tous les 10 000 mètres carrés.

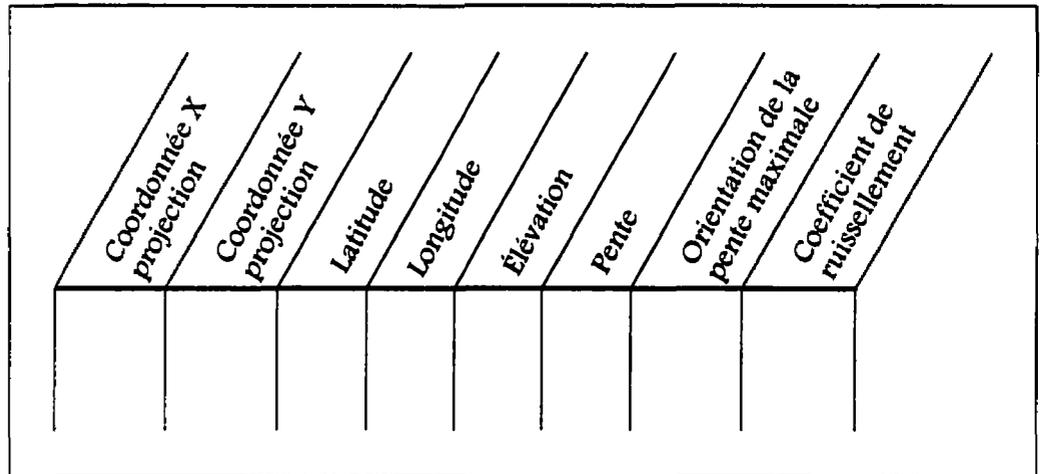


Figure 4.2 Format de la table de points dans SPANS.

Dans l'équation de la figure 4.3, le nombre 0,01 indique la densité en km², « topoq9c » est la carte d'élévation au quad 9, « pente9 » la carte de la pente au quad 9, « aspect9 » la carte de l'orientation maximale des pentes au quad 9 et « cn9 » la carte des coefficients de ruissellement au quad 9.

E Grille

```
result ({0.01 if class ('basemap'), 0}, (class('topoq9c'),
class ('pente9'), class ('aspect9'), class('cn9')));
```

Figure 4.3 Équation de modélisation pour la commande *Model / Points / Grid Sample*.

La définition d'un quad fixe lors de la création de l'univers prend toute son importance dans cette opération. L'extraction des attributs étant ponctuelle, il est essentiel pour éviter des erreurs de précision que le niveau de quad soit conforme à la densité spécifiée. En effet, si le niveau de quad est plus précis que la densité, il se peut que la valeur de l'attribut prélevé par l'équation sur une des cartes ne soit pas représentative de la valeur de l'ensemble des attributs sur une surface de 10 000 mètres carrés. Dans l'exemple donné ci-dessus, le niveau de quad 9 des cartes est de 100 mètres par 100 mètres ce qui équivaut à la densité de 0,01 km².

Le résultat de cette modélisation est une table de points en format SPANS qui contient les champs suivants: latitude, longitude, densité, attribut de la carte de base, élévation, pente, orientation de la pente maximale, coefficient de ruissellement. L'en-tête de cette table est édité pour supprimer les colonnes de densité et d'attribut de la carte de base qui ne sont pas nécessaires. Par défaut, les coordonnées sont en latitude et longitude, mais sont transformées en coordonnées de projection, par la commande *Transform / Coordinates / Latitude Longitude to projection*, une fois la table exportée en format ASCII. La table ainsi obtenue contient dans l'ordre les champs suivants: coordonnée x en projection, coordonnée y en projection, latitude, longitude, élévation, pente, orientation de la pente maximale et coefficient de ruissellement.

4.1.4.2 Réorganisation des données dans la table

L'ordre des données dans la table correspond à l'ordre croissant des nombres de Morton, un nombre hexadécimal associé au fonctionnement interne de SPANS qui ne correspond pas du tout au format matriciel requis par le programme DEMONII. Un préprocesseur au programme DEMONII est développé pour réorganiser les données en format matriciel. Le programme PREDEMON, listé à l'annexe V, lit les données dans la table provenant de SPANS, les enregistre et, à l'aide des coordonnées de projection en x et en y, place les données en ordre croissant selon l'axe des x et l'axe des y de sorte qu'elles forment une matrice. La projection MTM étant orthogonale, elle se prête bien à l'exercice. L'utilisation des latitudes et longitudes ou des coordonnées d'une projection qui n'est pas orthogonale, est plus complexe et n'assure pas l'obtention d'une matrice bien rectangulaire, à moins d'imposer une distorsion aux données.

Par la suite, chaque matrice résultant des opérations de classification est transcrite dans un fichier en code ASCII. La figure 4.4 illustre l'intrant et l'extrant du programme PREDEMON. L'extrant de PREDEMON devient l'intrant de DEMONII. Il n'existe pas de postprocesseur au programme DEMONII, car il est possible, à même le programme, de créer directement une table avec le format d'importation SPANS.

4.2 Résultats graphiques

À leur sortie du préprocesseur, les données possèdent le format requis par le programme DEMONII. Il ne reste qu'à éditer le fichier contenant la déclaration des variables du programme DEMONII pour fournir les bons noms de fichiers ainsi que les valeurs des différents paramètres avant de le recompiler et de le lancer. Il est nécessaire de recompiler le programme chaque fois que ce fichier est édité. Une copie des différentes déclarations de variables utilisées pour les essais est insérée à l'annexe VI.

Le programme DEMONII génère deux types de résultats. Le premier type est un fichier contenant une table en code ASCII dans laquelle on retrouve un champ pour les coordonnées en x, un champ pour les coordonnées en y, un pour les surfaces totales d'apport, un pour les surfaces spécifiques d'apport et un pour les surfaces spécifiques de dispersion. Cette table possède le format requis pour être directement importées dans SPANS telle quelle, sans édition.

Dans SPANS, la table est importée deux fois; une première fois en tant que table d'attributs et une seconde fois en tant que table de points géoréférencés. Elle est utilisée pour fabriquer trois cartes qui permettent de mieux analyser les résultats. La représentation graphique des matrices de TCA, de SCA et de SDA est rendue nécessaire à cause de la grande quantité de données à l'échelle du bassin versant.

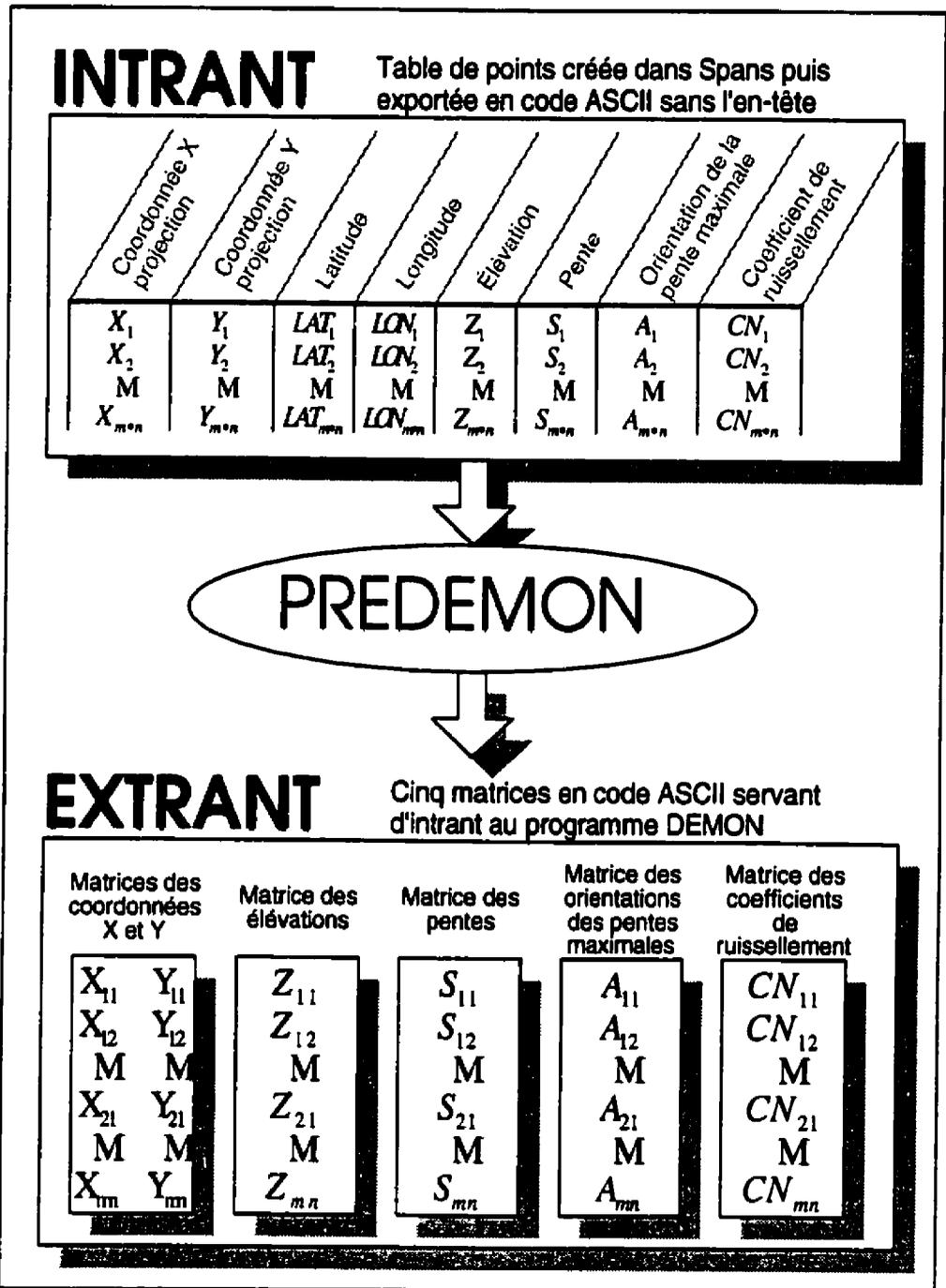


Figure 4.4 Schéma du processus du programme PREDEMON.

Le processus pour « enquader » les cartes à partir de points géoréférencés est le même que celui utilisé pour « enquader » la carte topographique c'est-à-dire la triangulation. Dans ce cas-ci, la commande est lancée à deux reprises, car l'éventail des valeurs apparaissant dans la table de points est supérieur à 256 valeurs différentes alors que les palettes de couleurs ne contiennent que 256 couleurs. Dans un premier temps, la commande établit une classification par quantiles, qui définit les intervalles inégaux des classes de façon à ce que chaque classe occupe une superficie équitable sur la carte. Le nombre de classes est déterminé par l'utilisateur. Dans un deuxième temps, la commande « enquade » la carte en suivant cette classification. La classification se fait sur environ 250 classes et une palette de couleur en noir et blanc est utilisée pour afficher la carte afin d'en augmenter la clarté.

Le deuxième type de résultat est un calcul de temps de concentration provenant de l'addition des temps de parcours sur chaque cellule appartenant au chemin le plus long. À partir de ce résultat, jumelé aux surfaces totales d'apport des cellules à l'exutoire, il est possible de déduire un débit moyen et de construire l'hydrogramme pour la rivière à l'étude. Les débits calculés et les hydrogrammes construits seront analysés par comparaison avec des données de précipitation - débit prises sur le terrain.

4.2.1 Validation préliminaire des résultats de DEMONII

Avant de réaliser la modélisation du bassin versant à l'étude, les résultats graphiques du programme DEMONII sont testés sur des formes géométriques simples de faible superficie soit un plan incliné et un biplan à double inclinaison (figure 4.5). Les matrices d'élévation, présentées aux tableaux 4.1 et 4.2 respectivement, ne proviennent pas de données SPANS, il s'agit de deux matrices créées dans un tableur et sauvegardées dans des fichiers en code ASCII. Une petite routine en langage Fortran (annexe IV) est écrite pour produire, à partir de la matrice d'élévation, les autres matrices d'intrant requises par le programme DEMONII. Le fichier contenant la déclaration de variables de DEMONII est modifié et le programme recompilé.

Afin de s'assurer que les modifications apportées au programme DEMON original ne changent en rien la modélisation, les résultats obtenus pour les deux programmes (DEMON original et DEMONII modifié) sont comparés en utilisant à tour de rôle les deux matrices tests.

La figure 4.6 donne les résultats obtenus en utilisant l'un ou l'autre des programmes. Sur les figures, l'intensité de la couleur augmente avec les valeurs. Les figures a) et b) représentent les matrices des surfaces spécifiques d'apport (SCA) pour le plan incliné et le biplan à double inclinaison respectivement, tandis que les figures c) et d) illustrent respectivement les matrices spécifiques de dispersion (SDA) des deux mêmes formes géométriques.

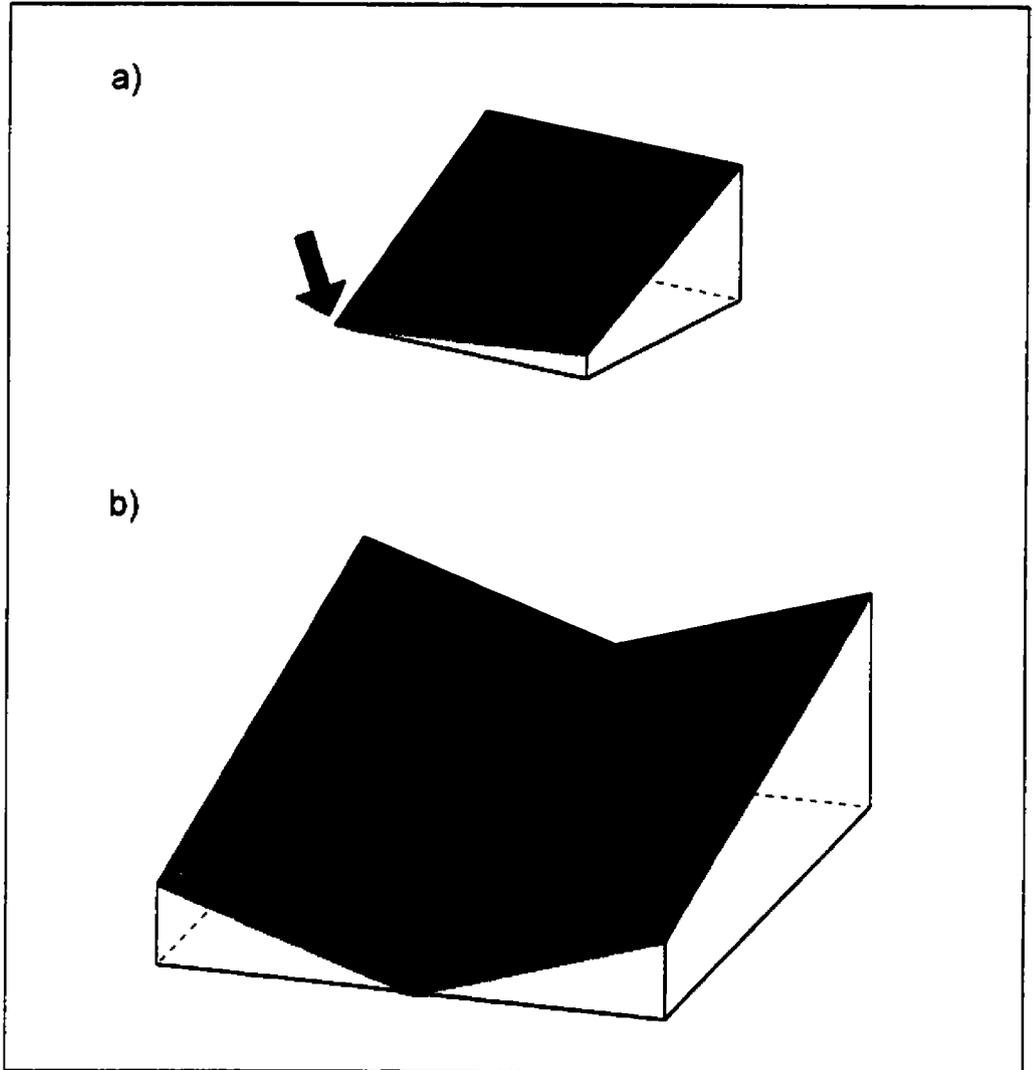


Figure 4.5 (a) Plan incliné et (b) biplan à double inclinaison.

Tableau 4.1 Matrice d'élévation du plan incliné

300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
279	280	281	282	283	284	285	286	287	288	289	290	291	292	293
258	260	262	264	266	268	270	272	274	276	278	280	282	284	286
237	240	243	246	249	252	255	258	261	264	267	270	273	276	279
216	220	224	228	232	236	240	244	248	252	256	260	264	268	272
195	200	205	210	215	220	225	230	235	240	245	250	255	260	265
174	180	186	192	198	204	210	216	222	228	234	240	246	252	258
153	160	167	174	181	188	195	202	209	216	223	230	237	244	251
132	140	148	156	164	172	180	188	196	204	212	220	228	236	244
111	120	129	138	147	156	165	174	183	192	201	210	219	228	237
90	100	110	120	130	140	150	160	170	180	190	200	210	220	230
69	80	91	102	113	124	135	146	157	168	179	190	201	212	223
48	60	72	84	96	108	120	132	144	156	168	180	192	204	216
27	40	53	66	79	92	105	118	131	144	157	170	183	196	209
6	20	34	48	62	76	90	104	118	132	146	160	174	188	202

Tableau 4.2 Matrice d'élévation du biplan à double inclinaison

445	410	375	340	305	270	235	200	235	270	305	340	375	410	445
421	388	355	322	289	256	223	190	223	256	289	322	355	388	421
397	366	335	304	273	242	211	180	211	242	273	304	335	366	397
373	344	315	286	257	228	199	170	199	228	257	286	315	344	373
349	322	295	268	241	214	187	160	187	214	241	268	295	322	349
325	300	275	250	225	200	175	150	175	200	225	250	275	300	325
301	278	255	232	209	186	163	140	163	186	209	232	255	278	301
277	256	235	214	193	172	151	130	151	172	193	214	235	256	277
253	234	215	196	177	158	139	120	139	158	177	196	215	234	253
229	212	195	178	161	144	127	110	127	144	161	178	195	212	229
205	190	175	160	145	130	115	100	115	130	145	160	175	190	205
181	168	155	142	129	116	103	90	103	116	129	142	155	168	181
157	146	135	124	113	102	91	80	91	102	113	124	135	146	157
133	124	115	106	97	88	79	70	79	88	97	106	115	124	133
109	102	95	88	81	74	67	60	67	74	81	88	95	102	109

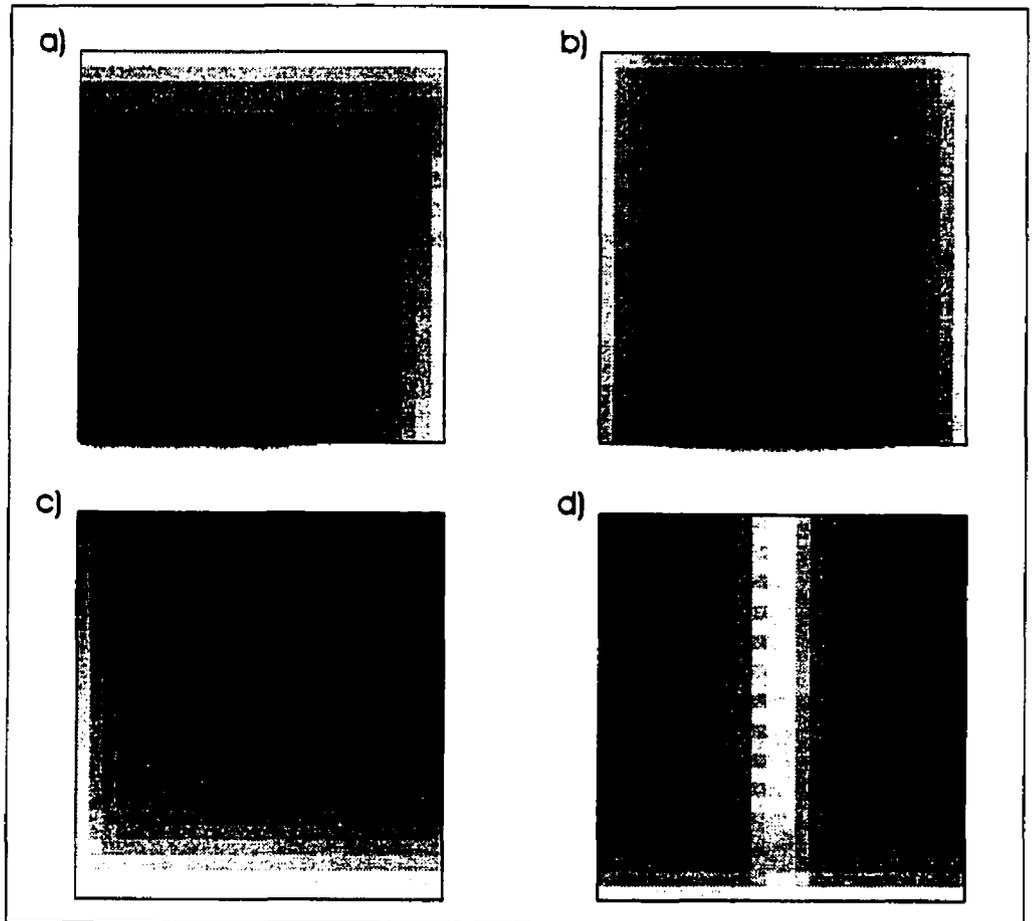


Figure 4.6 (a) Matrice des surfaces spécifiques d'apport (SCA) du plan incliné, (b) Matrice des surfaces spécifiques d'apport (SCA) du biplan à double inclinaison, (c) matrice des surfaces spécifiques de dispersion (SDA) du plan incliné et (d) matrice des surfaces spécifiques de dispersion (SDA) du biplan à double inclinaison.

Les matrices des surfaces totales d'apport (TCA), contenues dans les tableaux 4.3 et 4.4, ne sont pas illustrées, car elles correspondent aux surfaces spécifiques d'apport multipliées par la largeur de l'écoulement de chaque cellule. Leur représentation graphique est même que celle des SCA. Voir à ce sujet les explications du programme DEMON au chapitre 3.

D'après les matrices d'élévation et les dessins de la figure 4.5, l'écoulement, qui suit l'orientation des pentes maximales sur le terrain et se dirige vers la cellule dont l'élévation est la plus faible, se termine dans le coin gauche du plan incliné et au centre inférieur de la figure pour le biplan. Dans ces régions, les surfaces spécifiques d'apport sont les plus élevées avec une grande intensité de couleur (figure 4.6 a) et b)), car l'apport des cellules dans cette région provient de l'ensemble des cellules du plan ou du biplan.

À l'inverse, les surfaces spécifiques de dispersion sont les plus faibles dans ces mêmes régions et sont illustrées par des couleurs très claires (figure 4.6 c) et d)). Les cellules qui possèdent les surfaces de dispersion les plus élevées correspondent aux cellules en altitude.

Les résultats obtenus aux tableaux 4.3 et 4.4 pour les matrices des surfaces totales d'apport suivent naturellement l'allure des matrices des surfaces spécifiques d'apport. Il indiquent précisément laquelle des cellules se trouve à l'exutoire. Ainsi, l'exutoire du biplan se trouve bien au centre de la matrice du tableau 4.4 tandis que l'exutoire du plan incliné est décalé de cinq cellules vers la droite par rapport au coin inférieur gauche de la matrice du tableau 4.3. Ce décalage indique qu'une partie de l'eau qui ruisselle est acheminée au-delà de la limite inférieure du plan incliné et ne parvient pas à la cellule du coin inférieur gauche. Il y a une perte de quantité d'eau aux frontières des matrices.

Tableau 4.3 Matrice des surfaces totales d'apport (TCA) pour le plan incliné

10128	10135	10143	10152	10161	10172	10185	10200	10217	10238	10263	10294	10333	10000
21160	21223	21293	21372	21461	21563	21679	21815	21974	22164	22393	22678	23039	19667
33268	33448	33649	33874	34130	34422	34757	35148	35609	36159	36830	37663	38726	26667
46501	46864	47270	47727	48245	48838	49521	50318	51261	52391	53772	55495	57373	28333
60910	61528	62220	63000	63886	64900	66073	67444	69139	71275	74003	77569	82000	22333
76545	77495	78569	79824	81287	83001	85033	87472	90443	94065	98579	101056	63400	18714
93740	95239	96945	98895	101109	103665	106660	110237	114867	120882	128863	104395	49153	16926
112653	114705	117025	119668	122835	126670	131334	137077	144294	151114	144264	88072	42490	15788
133004	135914	139286	143224	147865	153273	160052	168742	180061	183355	131333	71411	38649	15000
155820	159712	164141	169153	176021	183963	194044	207488	220749	175133	103633	63941	36053	14412
180564	185893	192146	199554	208726	220204	235353	256398	218938	139223	91412	59084	34146	13947
208302	215349	223822	234043	247234	264140	282238	263298	178690	120858	83863	55583	32656	13571
239241	248706	260354	274629	293892	320426	307851	223382	152117	110206	78510	52880	31460	13261
274275	286944	303225	323983	353304	352367	269407	185091	137958	102765	74420	50733	30479	13000

Tableau 4.4 Matrices des surfaces totales d'apport (TCA) pour le biplan à double inclinaison

10000	16618	17992	18782	19737	20913	22370	22370	20913	19737	18782	17992	16618	10000
13382	28414	37839	42368	44410	46724	49442	77874	46724	44410	42368	37839	28414	13382
13594	29547	43650	55008	62201	66294	69200	167628	66294	62201	55008	43650	29547	13594
13833	30137	44444	57036	68141	76998	83155	280149	76998	68141	57036	44444	30137	13833
14107	30802	45324	57970	69050	78645	86949	403079	78645	69050	57970	45324	30802	14107
14423	31553	46299	59008	70014	79478	87566	527397	79478	70014	59008	46299	31553	14423
14792	32412	47384	60163	71063	80366	88181	649608	80366	71063	60163	47384	32412	14792
15226	33406	48598	61423	72211	81283	88799	769310	81283	72211	61423	48598	33406	15226
15741	34569	49986	62790	73458	82236	89398	885939	82236	73458	62790	49986	34569	15741
16359	35922	51584	64321	74753	83224	89943	998846	83224	74753	64321	51584	35922	16359
17114	37510	53381	66034	76146	84189	90405	1107261	84189	76146	66034	53381	37510	17114
18057	39392	55393	67854	77619	85123	90711	1210029	85123	77619	67854	55393	39392	18057
19270	41643	57714	69850	79081	85973	90806	1305943	85973	79081	69850	57714	41643	19270
20885	44331	60288	71945	80531	86629	90596	1393143	86629	80531	71945	60288	44331	20885

Ces matrices des surfaces totales d'apport permettent aussi de vérifier la quantité d'eau qui arrive à l'exutoire. En principe, la somme des surfaces totales d'apport des cellules formant l'exutoire équivaut à la superficie totale de la matrice. Il en va autrement des matrices présentées en raison des pertes aux frontières.

Initialement, une matrice contient 196 cellules (14 X 14) de 10 000 m² chacune pour une superficie totale de 1 960 000 m². Cette quantité ne se retrouve pas dans une seule cellule, car la convergence n'est pas absolue et il y a des pertes le long des frontières des matrices. Ces pertes se traduisent par des accumulations dans les cellules frontières définies comme des puits. L'orientation des pentes des cellules voisines de la cellule à l'exutoire n'étant pas de 90°, 180° ou 270° une partie de l'écoulement est déviée vers d'autres cellules et accumulée dans les cellules aux frontières. Pour obtenir la superficie totale de la matrice, il faut additionner une fraction de la superficie de chacune des cellules formant le voisinage de l'exutoire. La fraction de la superficie correspond à la portion de l'écoulement qui est orienté au-delà des frontières de la matrice.

Si la somme des superficies des cellules voisines l'exutoire est supérieure à la superficie totale, on peut alors supposer que la superficie totale se retrouve à l'exutoire de la matrice. Dans les deux cas, cette somme est bien supérieure à la superficie totale (2 757 951 m² pour le plan incliné et 2 212 957 m² pour le

biplan). Plus cette valeur tend vers la superficie totale plus la convergence est absolue.

Ces résultats viennent appuyer l'hypothèse posée au chapitre 3 selon laquelle la surface totale d'apport de la ou des cellules à l'exutoire représente la quantité d'eau provenant de l'ensemble du bassin versant qui chemine jusqu'à l'exutoire.

4.2.2 Application au bassin versant de la rivière Samson

L'informatique étant une science du « cas particulier », il est peu probable, malgré le fait que le programme fonctionne très bien pour les matrices tests, que la même réussite se répète automatiquement avec la matrice d'élévation du bassin versant. Le but premier à atteindre est l'obtention d'une représentation graphique des matrices de surfaces spécifiques d'apport et de dispersion qui soient cohérentes, significatives et sur lesquelles il est possible de distinguer le réseau hydrographique. La validité du calcul du temps de concentration et du calcul de débit est conditionnelle à l'obtention de résultats graphiques valides.

4.2.2.1 Problèmes à l'exécution de DEMONII

À l'origine la fonction « Retrace_Chemin » doit enregistrer tous les chemins partant de toutes les cellules dans une matrice où le premier indice indique le numéro du chemin et le second l'ordre de la cellule dans ce chemin. De plus, chaque élément de la matrice comporte cinq champs qui décrivent les indices i

et j, garants de la position de la cellule dans la matrice d'élévation du bassin, le temps de parcours sur la cellule, la longueur du segment de l'écoulement ainsi que la vitesse de l'écoulement de la cellule. La mémoire requise pour générer un tel mastodonte dépasse les capacités du système central, il faut donc déjouer le système et transcrire la matrice dans un fichier à chaque nouvelle cellule source. Le fichier obtenu est gigantesque (plus de 300 Mega-octets) et aucun éditeur de code ASCII n'arrive à le charger. Il faut donc renoncer à toute cette montagne d'information, qui tout compte fait n'est pas vraiment essentielle, et se résigner à ne conserver que les paramètres du chemin dont le temps de parcours est maximal. En effet, pour le calcul de débit et du temps de concentration, seul le chemin le plus long, en temps, est important.

Suite à cette correction, l'exécution du programme est lancée, mais est stoppée assez rapidement, dès l'évaluation des premières cellules. Cet arrêt prématuré est causé par une mauvaise définition de la variable « $istop(i,j)$ » dans la routine « PITS ».

La routine « PITS » initialise l'indice d'arrêt (« $istop$ ») de chaque cellule à la valeur un (1) s'il s'agit d'un puits ou s'il s'agit d'une cellule sur la frontière de la matrice. Une cellule est un puits si ses huit voisines ont une élévation plus grande ou égale à la sienne. Lorsque la cellule ne remplit aucune des deux conditions, l'indice d'arrêt est initialisé à zéro.

Cette variable est essentielle au fonctionnement du programme, car elle indique la fin d'un chemin. Chaque fois qu'une cellule suivante est identifiée le programme vérifie la valeur du « istop » de la cellule. Lorsqu'il est égal à zéro, les calculs se poursuivent et une cellule suivante est identifiée, mais lorsque qu'il est égal à un, il s'agit d'un puits ou d'une frontière de la matrice et le chemin est terminé. Le programme poursuit alors avec un autre chemin en utilisant une cellule sur la liste d'attente ou une nouvelle cellule source lorsque la liste d'attente est vide.

L'emmagasinage des matrices dans les variables (fonction READ) du programme se fait dans une boucle débutant au chiffre 2 et se terminant au nombre de colonnes plus un (nc1) et au nombre de rangées plus un (nr1). La première cellule de la matrice dans le coin en haut à gauche porte donc les indices 2,2. Cette astuce évite d'attribuer un indice nul aux cellules voisines des cellules dont un des deux indices est égal à 2, car lors de l'évaluation de la cellule suivante pour un angle entre 90° et 270°, l'indice *i* de la nouvelle cellule est calculé comme suit:

$$i_n = i_o - 1 \quad (4.1)$$

En d'autres termes, l'indice *i* de la cellule suivante (*i_n*) est égal à l'indice *i* de la cellule originale moins un. La même relation s'établit pour l'indice *j* des cellules dont l'orientation de la pente maximale se situe entre 0° et 180°. L'orientation

des axes d'une matrice est différente de l'orientation des axes x et y du plan cartésien (figure 4.7).

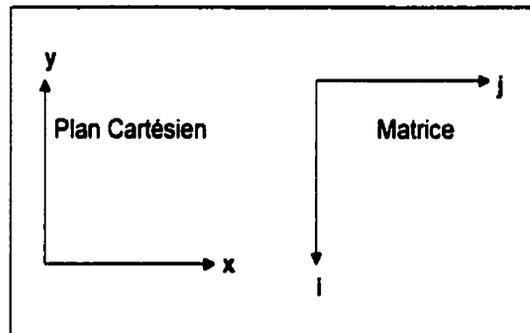


Figure 4.7 Orientation des axes d'un plan cartésien vs une matrice.

Dans les matrices tests, l'écoulement étant toujours orienté vers le centre et le bas de la matrice, il n'aboutit jamais dans une cellule dont un des deux indices est le chiffre un, à l'extérieur du domaine de calcul. C'est cependant ce qui arrive dans le cas du bassin, car les pentes maximales des cellules sur le pourtour de la matrice sont orientées vers l'extérieur de la matrice puisque ces cellules ne font pas partie du bassin versant à l'étude.

Elles n'en font pas partie, car la carte de base choisie est rectangulaire et sert à former une matrice orthogonale qui englobe tout le bassin. De plus, la délimitation du bassin devient superflue car, par définition, les cellules ne faisant pas partie du bassin ne contribuent pas au calcul de la surface totale d'apport de la cellule à l'exutoire.

Dans la routine « PITS », l'initialisation de l'indice d'arrêt des cellules sur la frontière se fait dans des boucles débutant aussi au chiffre 2. Toutes les cellules dont un des deux indices de position est le chiffre 1 ont un indice d'arrêt (« $istop(i,j)$ ») établi par défaut à zéro et il n'existe aucune autre valeur d'élévation, de pente, de surface d'apport ou autres, associée à ces indices. Dans ces cas précis, la condition aux frontières n'est pas respectée et, par conséquent, le programme s'interrompt abruptement, incapable de trouver les variables associées à ces cellules.

Le problème est complexe à retracer, mais simple à résoudre. Il suffit d'associer la valeur un aux variables « $istop(1,j)$ » et « $istop(i,1)$ » dans la routine « PITS » pour indiquer l'arrêt de l'écoulement sur ces cellules.

La correction est effectuée et le programme relancé. Il roule un bon moment puis s'interrompt à nouveau. Il faut plusieurs heures de dépistage pour s'apercevoir que le programme est entré dans une boucle infinie. L'orientation des pentes maximales le ramène à une cellule appartenant au chemin en cours, plusieurs positions en amont de la cellule actuelle. L'écoulement tourne en rond!

Avant de conclure à une erreur de programmation, les données utilisées sont analysées. Des aberrations dans le modèle numérique d'élévation de terrain et une mauvaise définition de l'orientation des pentes sur le bassin est peut-être à

l'origine du problème. L'importance d'utiliser, pour la création des cartes dans SPANS, un niveau de quad dont la dimension correspond à la discrétisation du bassin dans le programme DEMONII est mise en évidence. La dimension des quads des cartes, la densité des points extraits des cartes ainsi que la discrétisation utilisée dans le programme DEMONII doivent obligatoirement correspondre au même niveau de quad pour éviter les erreurs de représentativité des valeurs sur les cellules. L'extraction des données est ponctuelle, aucun calcul de moyenne sur les cellules ne s'exécute lors de cette extraction des valeurs et il faut y remédier en conservant une cohérence entre les données tout au long du processus.

Les essais avec des données cohérentes repoussent l'interruption du programme sans toutefois assurer son entière exécution. Le problème des boucles infinies persiste. La seconde solution consiste à interpréter ces boucles comme des puits de plus grande dimension. Cette hypothèse permet d'inclure dans le programme une deuxième condition de fin de chemin qui prend la forme d'un deuxième indice d'arrêt (« i2stop »). Contrairement au premier indice qui est défini préalablement dans le programme, ce deuxième indice est évalué pour chaque nouvelle cellule en utilisant la variable « chemin ». Chaque fois que le programme détermine une cellule suivante, la routine "Check_loop" passe en revue tous les indices des cellules en amont de la cellule suivante. Si la même

cellule est repérée dans le chemin en cours le deuxième indice de chemin est initialisé à un et le chemin se termine.

Ce dernier ajustement au programme permet son entière exécution. Le fichier de résultats contient une en-tête de table en format SPANS ainsi que cinq colonnes: la coordonnée de projection en x, la coordonnée de projection en y, la valeur du SCA, la valeur du SDA et la valeur du TCA. Cette table est importée dans SPANS et transformée en trois cartes, une pour chaque thématique (SCA, SDA et TCA).

4.2.2.2 Description des essais

Deux séries d'essais ont été réalisées avec une pluie unitaire pour la validation des surfaces totales d'apport. La première série d'essais est effectuée pour trois discrétisations différentes, soit 200m par 200m, 100m par 100m et 50m par 50m, et la seconde série pour deux discrétisations différentes de 100m par 100m et 50m par 50m. Des essais avec des discrétisations plus fines n'ont pas été réalisés, car la grosseur des fichiers générés par le logiciel SPANS s'élevait à plusieurs centaines de Mega-octets et les temps de calculs étaient de l'ordre de plusieurs jours.

Dans la première série d'essais, le coefficient de ruissellement agit sur chacune des cellules des chemins en multipliant la quantité d'eau de chaque cellule par le coefficient de ruissellement. Cette méthode tend à maximiser la capacité

d'infiltration du sol provoquant ainsi un retard du temps de montée de l'hydrogramme et une diminution du débit de pointe. Cet écrêtement de la pointe de l'hydrogramme est peu recommandable dans un calcul de débit, car il donne des résultats trop optimistes qui sous-évaluent le ruissellement direct et surévaluent le ruissellement souterrain.

En ce sens, la deuxième méthode est plus valable et plus réaliste, car la multiplication par le coefficient de ruissellement se fait sur la cellule source seulement. L'infiltration qui se fait durant l'ensemble de la simulation est de beaucoup inférieure dans ce cas-ci diminuant ainsi le phénomène d'écrêtement de l'hydrogramme.

4.2.2.3 Résultats graphiques pour la rivière Samson

Les résultats des deux séries d'essais, un plan de la topographie de la région ainsi qu'un plan du réseau hydrographique du bassin versant sont présentés aux figures 4.8 à 4.24 sous forme de cartes où les valeurs des surfaces spécifiques ou totales augmentent du blanc vers le noir.

Un premier coup d'oeil sur les cartes représentant les surfaces d'apport, spécifiques et totales, de chaque essai permet de constater une nette ressemblance entre elles. Ce lien de parenté est logique puisqu'elles proviennent du seul et même calcul où intervient la multiplication par le coefficient de

ruissellement. La différence entre les cartes de surfaces d'apport totales et spécifiques découle de la définition de la surface spécifique d'apport qui correspond à la valeur de la surface totale d'apport divisée par la largeur de l'écoulement (ω). Cependant, l'allure des cartes des surfaces de dispersion est différente des autres, car le calcul des surfaces de dispersion est complètement différent dans l'algorithme du programme. De plus, ce calcul n'est pas soumis à la multiplication par le coefficient de ruissellement ni à la multiplication par une hauteur d'eau, car il n'intervient pas au niveau du calcul de débit et de la modélisation du bassin versant faite dans cette étude. La présentation des cartes des surfaces de dispersion sert surtout de référence dans l'analyse des cartes des surfaces d'apport.

Essai #1 (200m x 200 m)

Ces figures représentent respectivement les surfaces spécifiques d'apport et de dispersion ainsi que les surfaces totales d'apport pour la première série d'essais seulement avec une discrétisation de 200 mètres par 200 mètres (figures 4.10 à 4.12). Sur la carte des surfaces spécifiques d'apport (figure 4.10), le réseau hydrographique du bassin (figure 4.9) n'est pas très clair, quoique que la rivière Samson soit facilement repérable, et la figure présente de grandes zones noires. Le calcul du temps de concentration est refait dans le tableur en utilisant le TCA

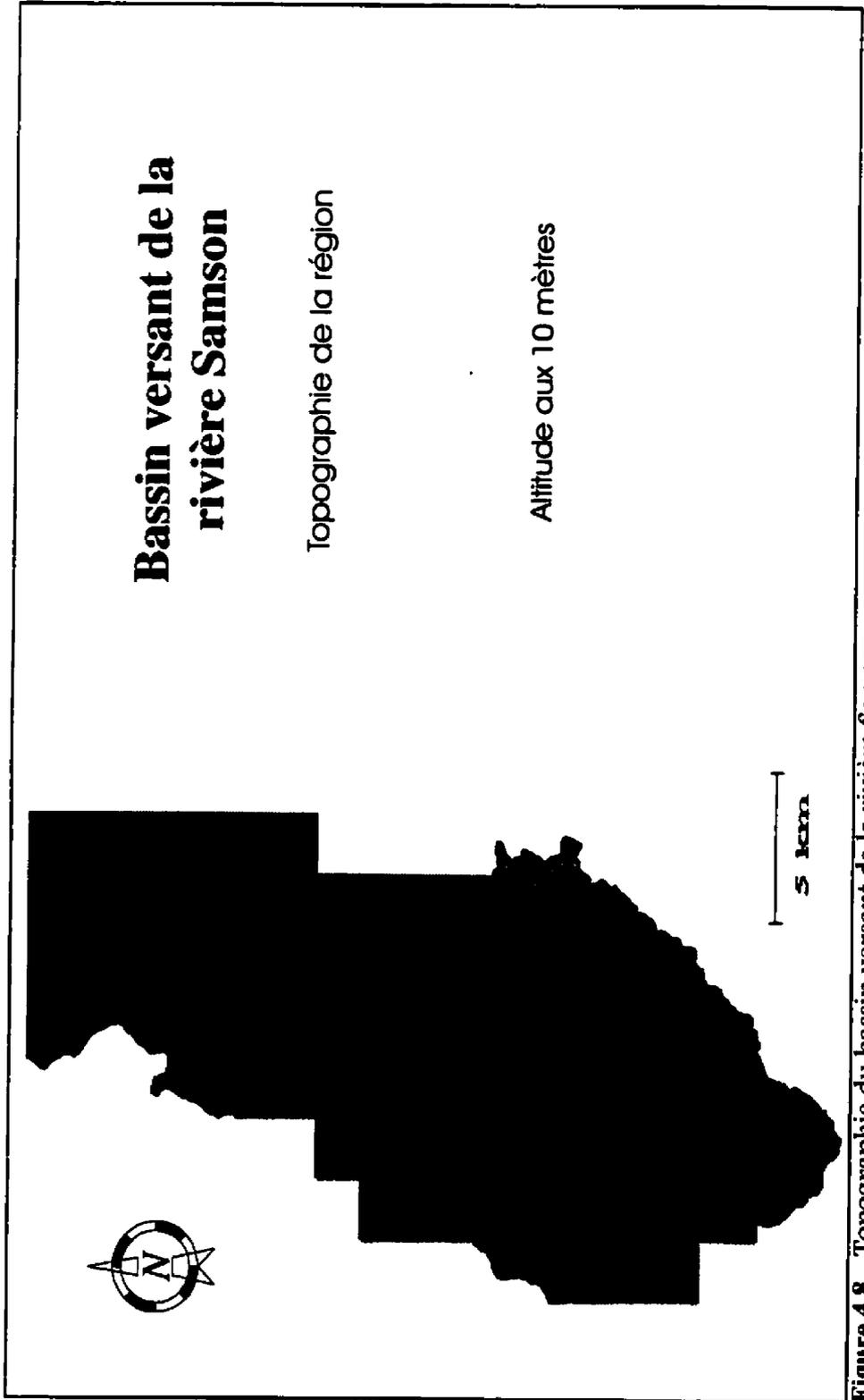


Figure 4.8 Topographie du bassin versant de la rivière Samson.

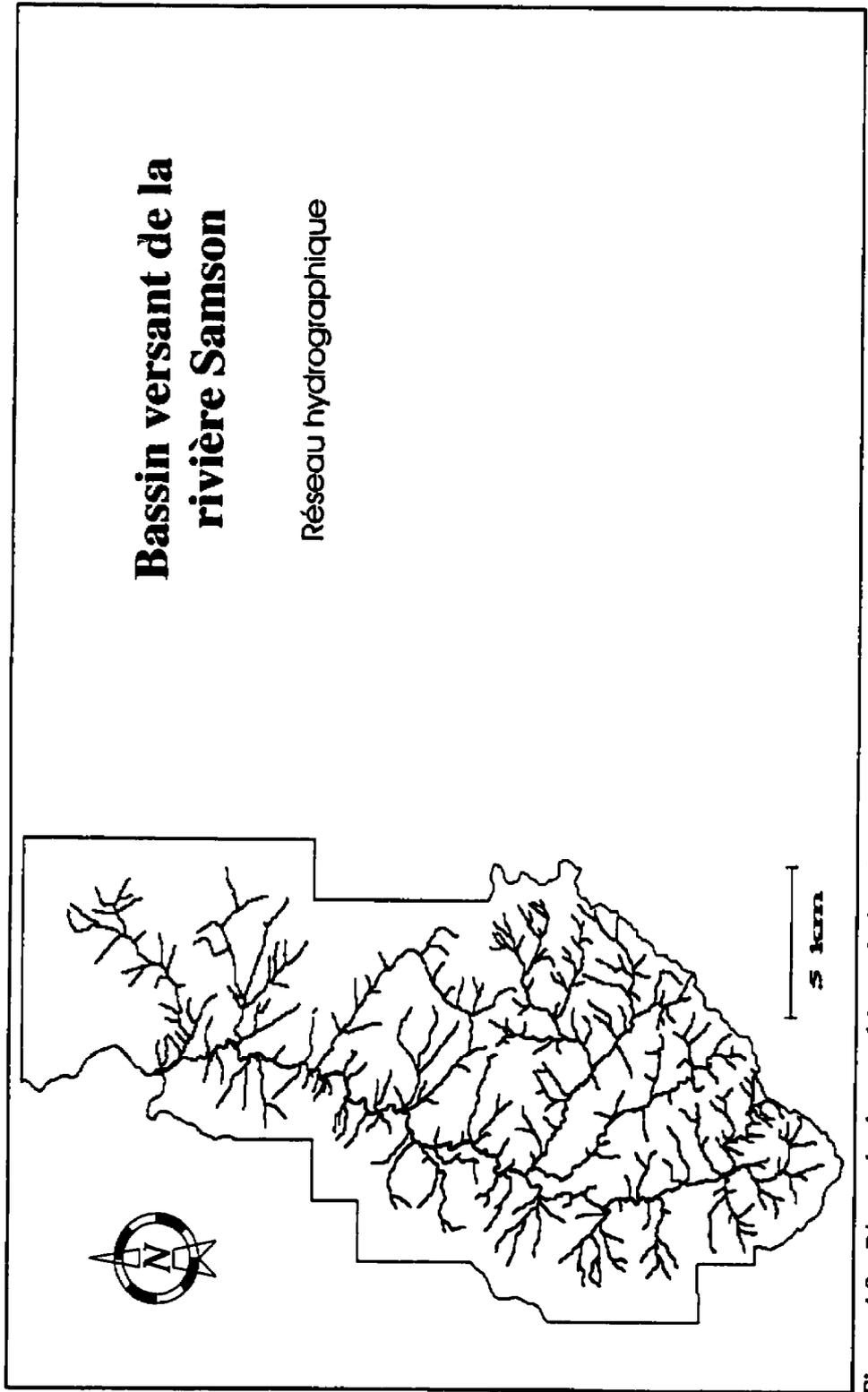


Figure 4.9 Réseau hydrographique du bassin versant de la rivière Samson.

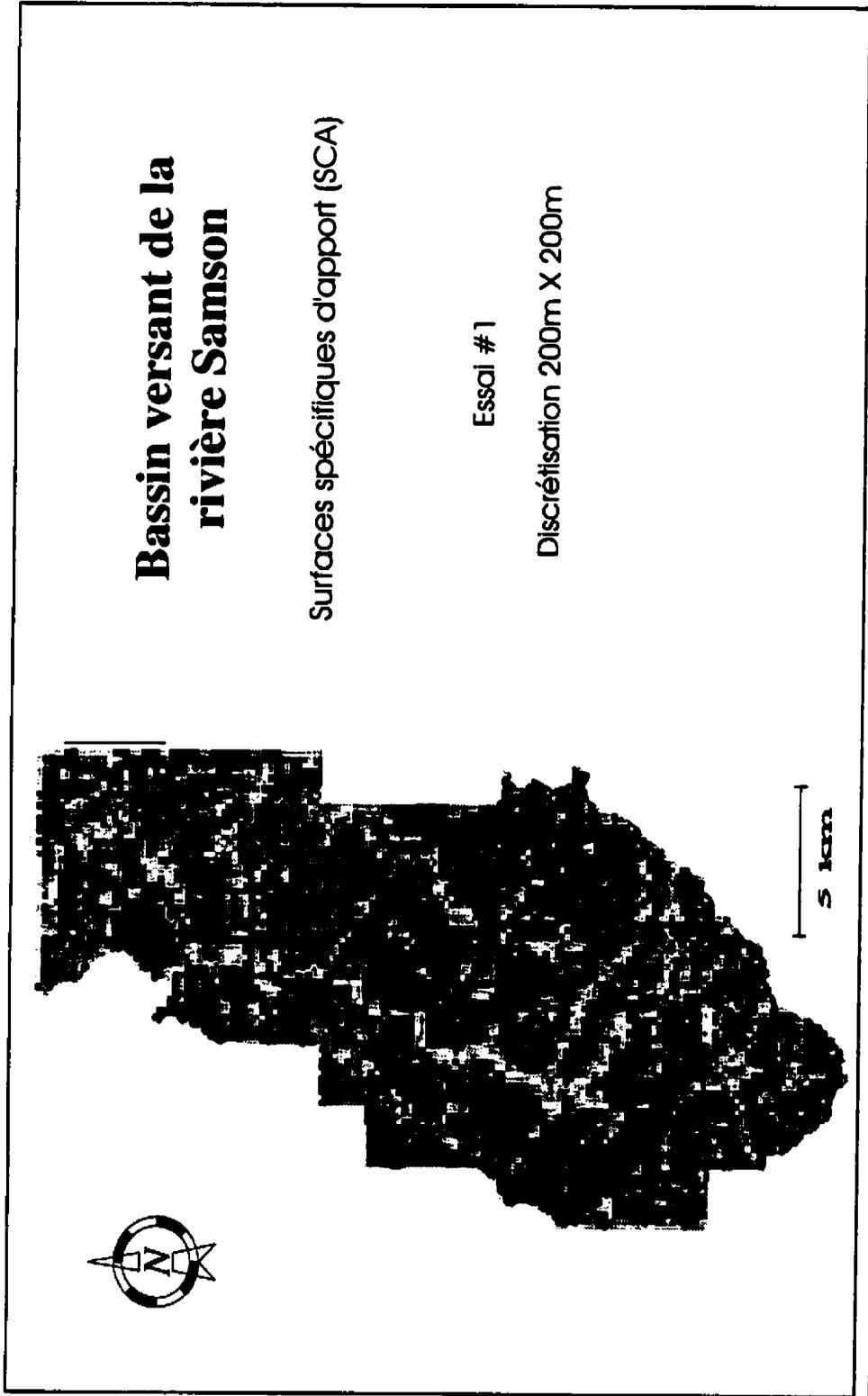


Figure 4.10 Surfaces spécifiques d'apport (SCA) à une discrétisation de 200m x 200m, première série d'essais.

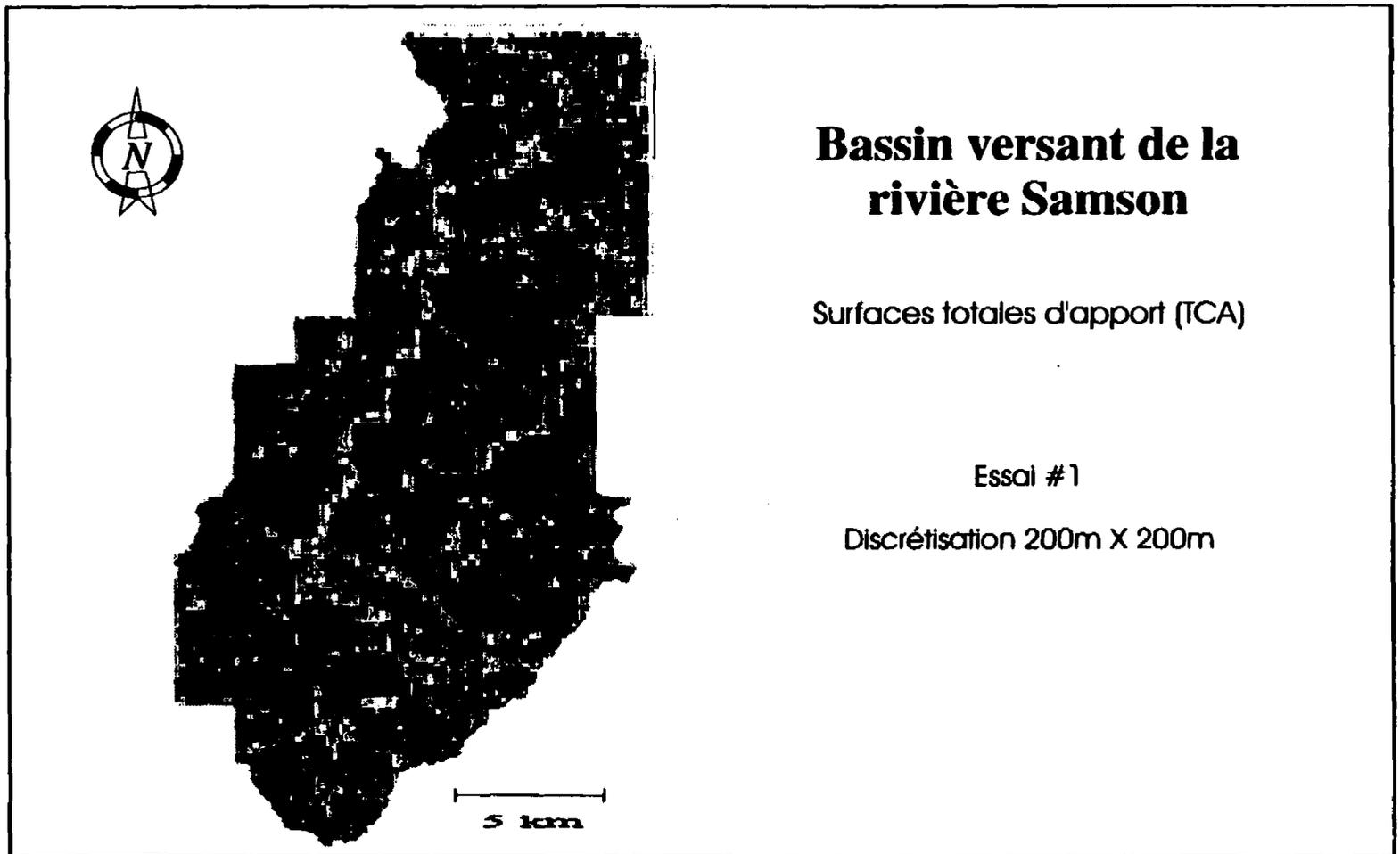


Figure 4.11 Surfaces totales d'apport (TCA) à une discrétisation de 200m x 200m, première série d'essais.

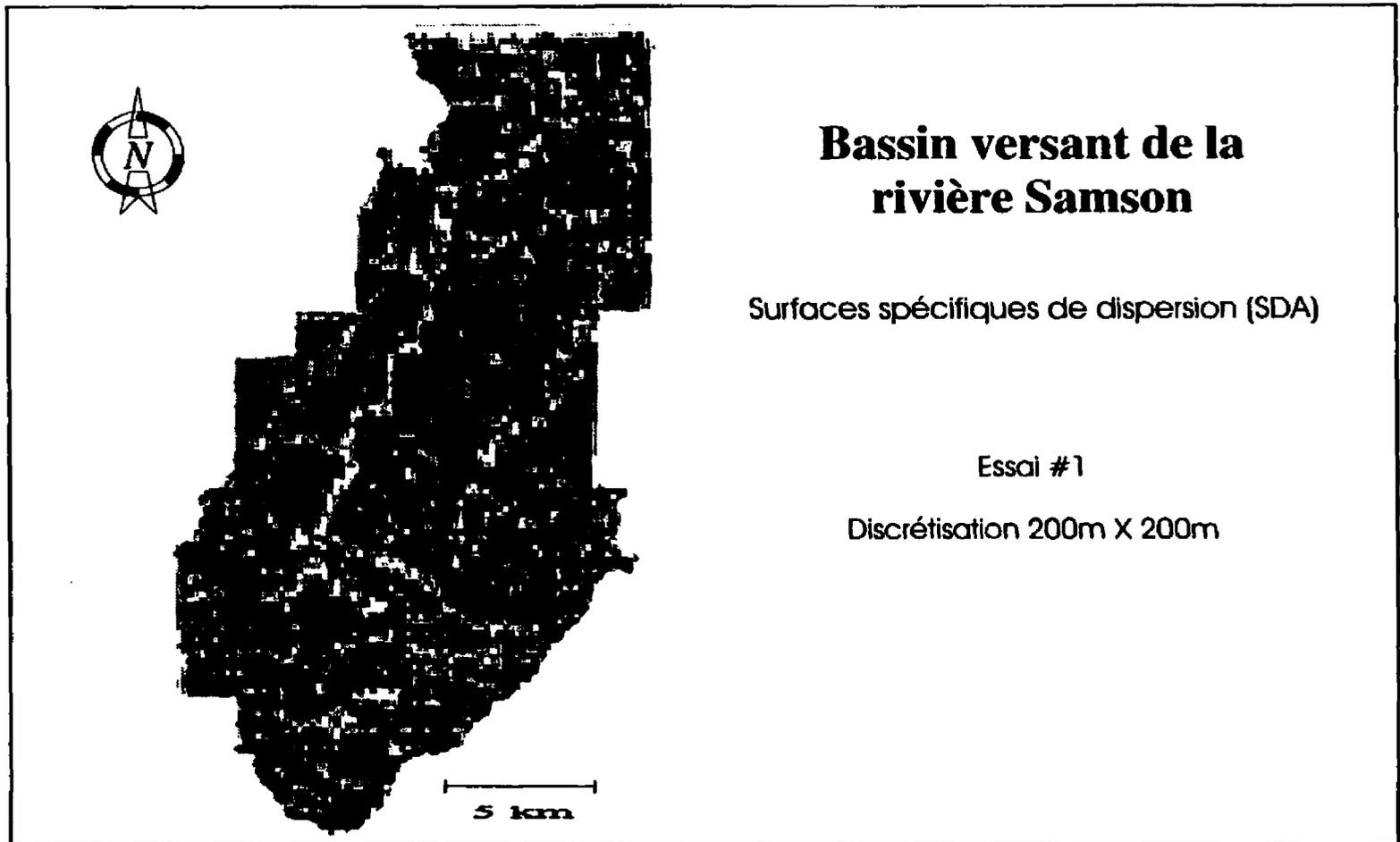


Figure 4.12 Surfaces spécifiques de dispersion (SDA) à une discrétisation de 200m x 200m, première série d'essais.

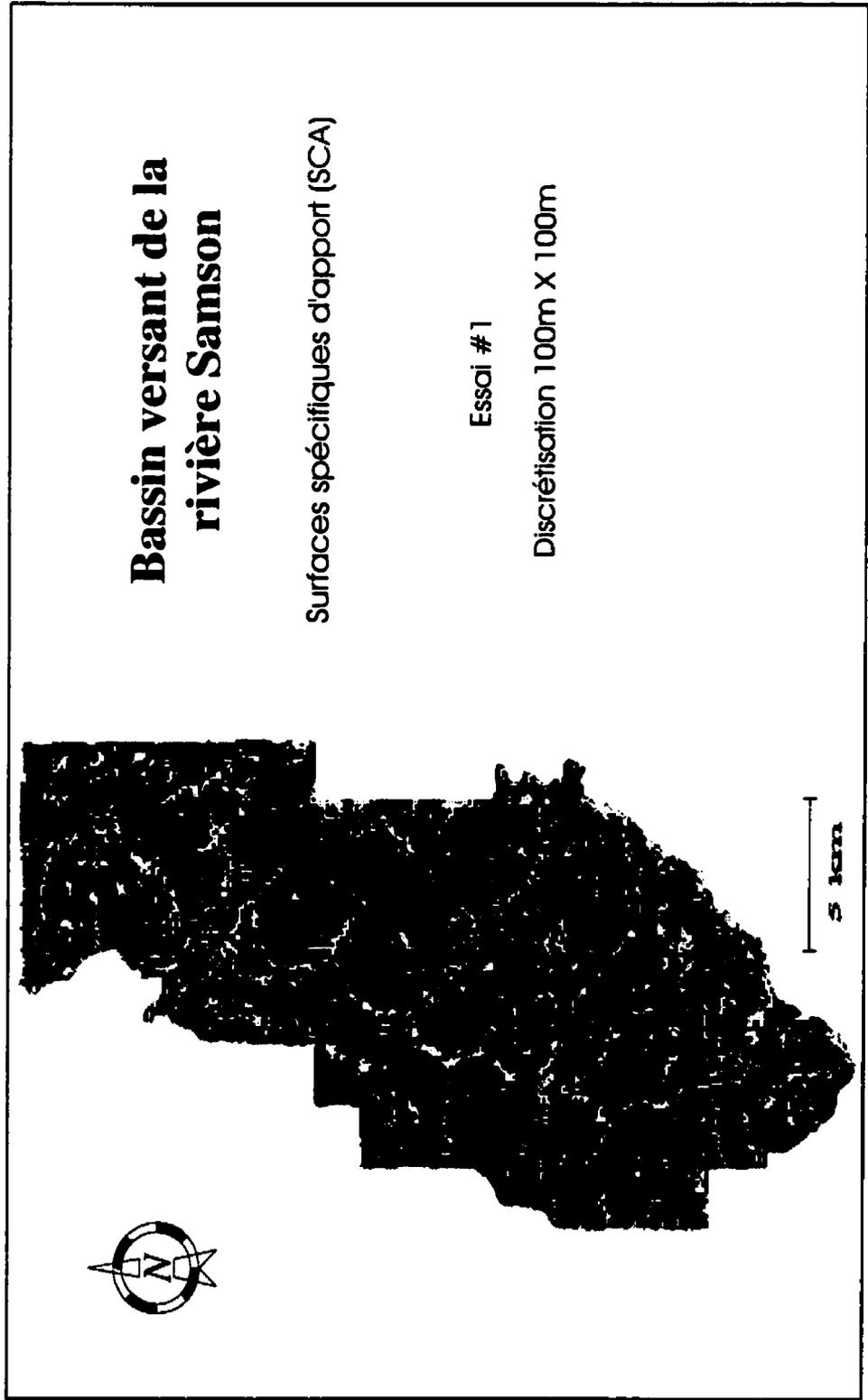


Figure 4.13 Surfaces spécifiques d'apport (SCA) à une discrétisation de 100m x 100m, première série d'essais.

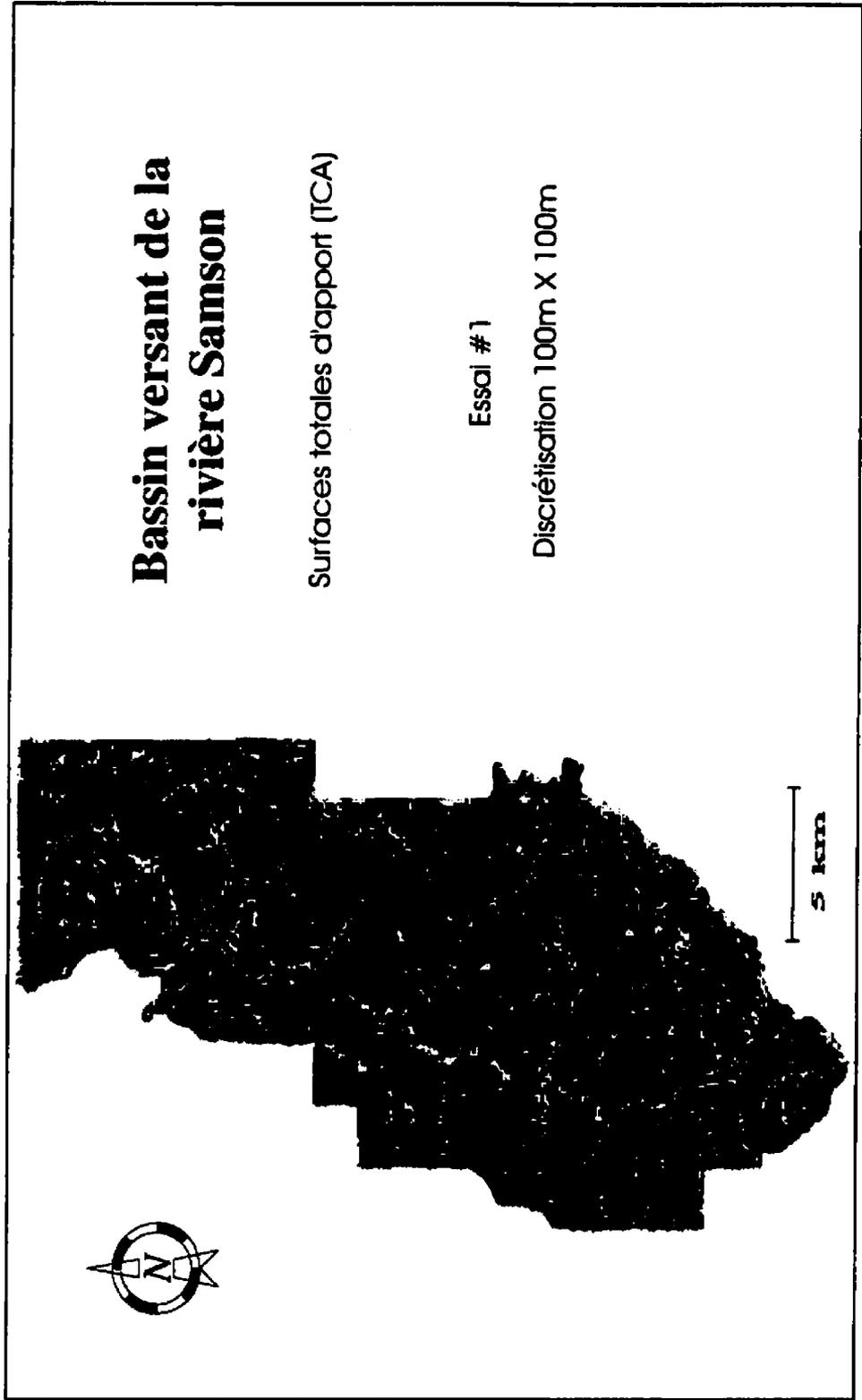


Figure 4.14 Surfaces totales d'apport (TCA) à une discrétisation de 100m x 100m, première série d'essais.

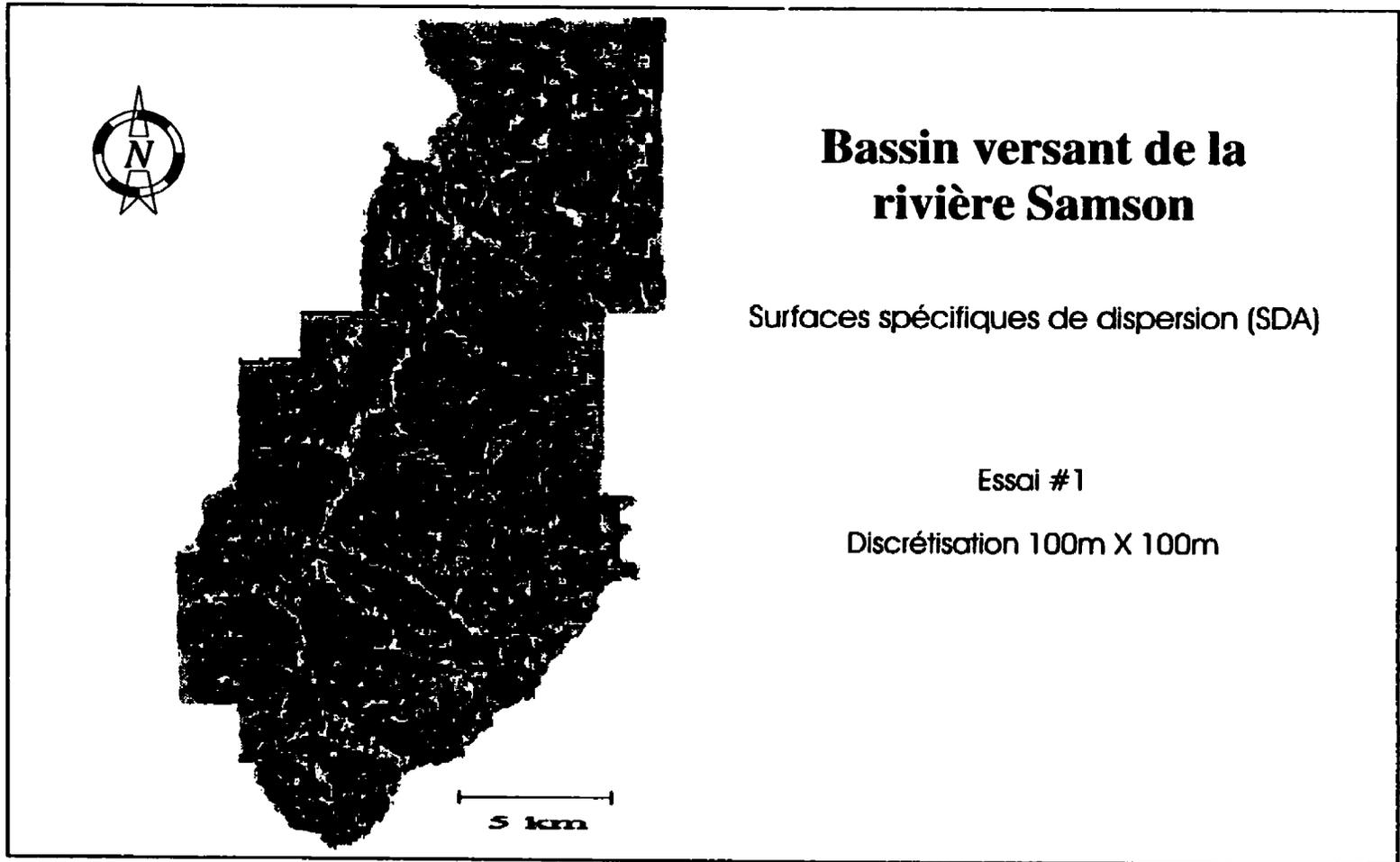


Figure 4.15 Surfaces spécifiques de dispersion (SDA) à une discrétisation de 100m x 100m, première série d'essais.

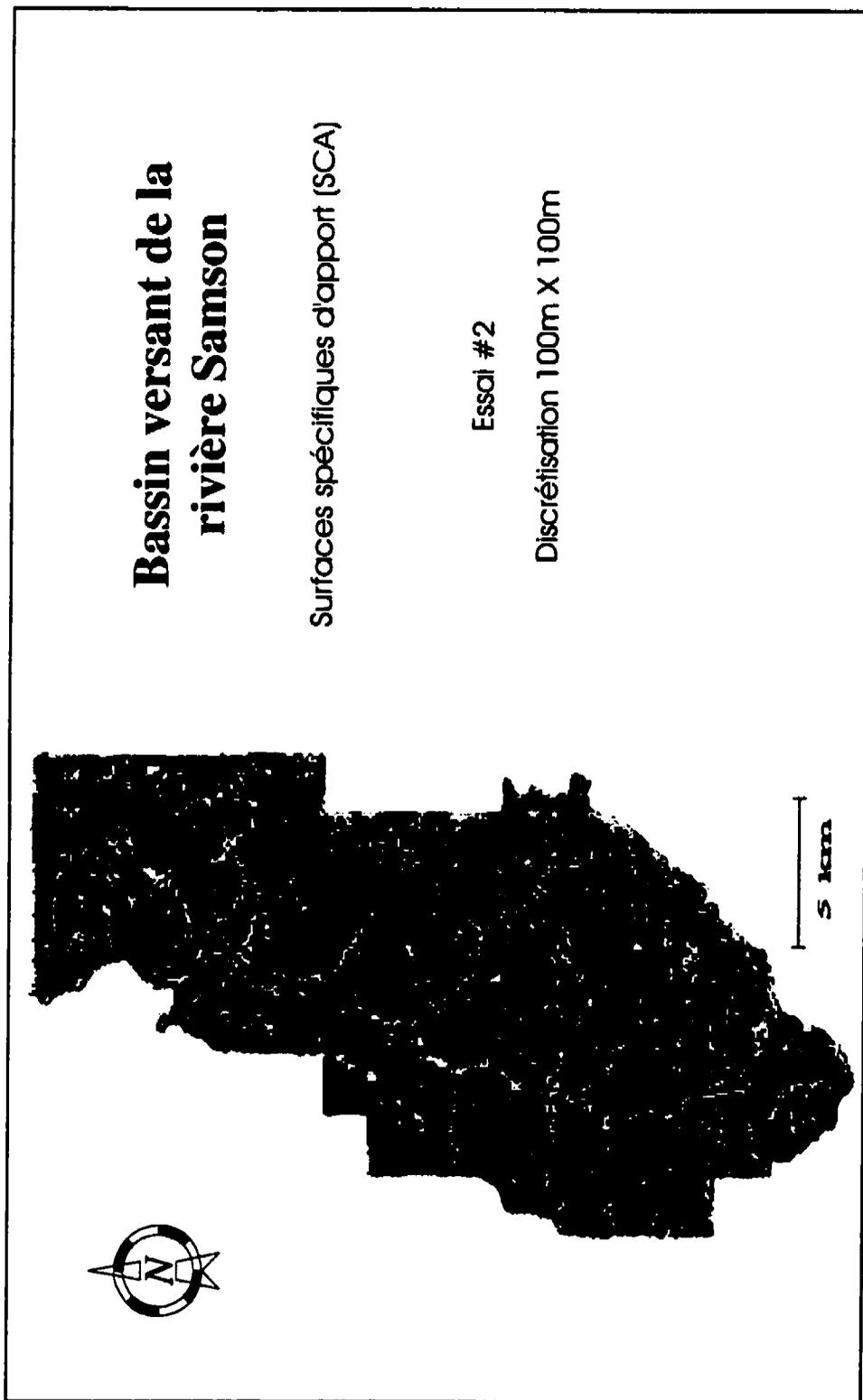


Figure 4.16 Surfaces spécifiques d'apport (SCA) à une discrétisation de 100m x 100m, deuxième série d'essais.

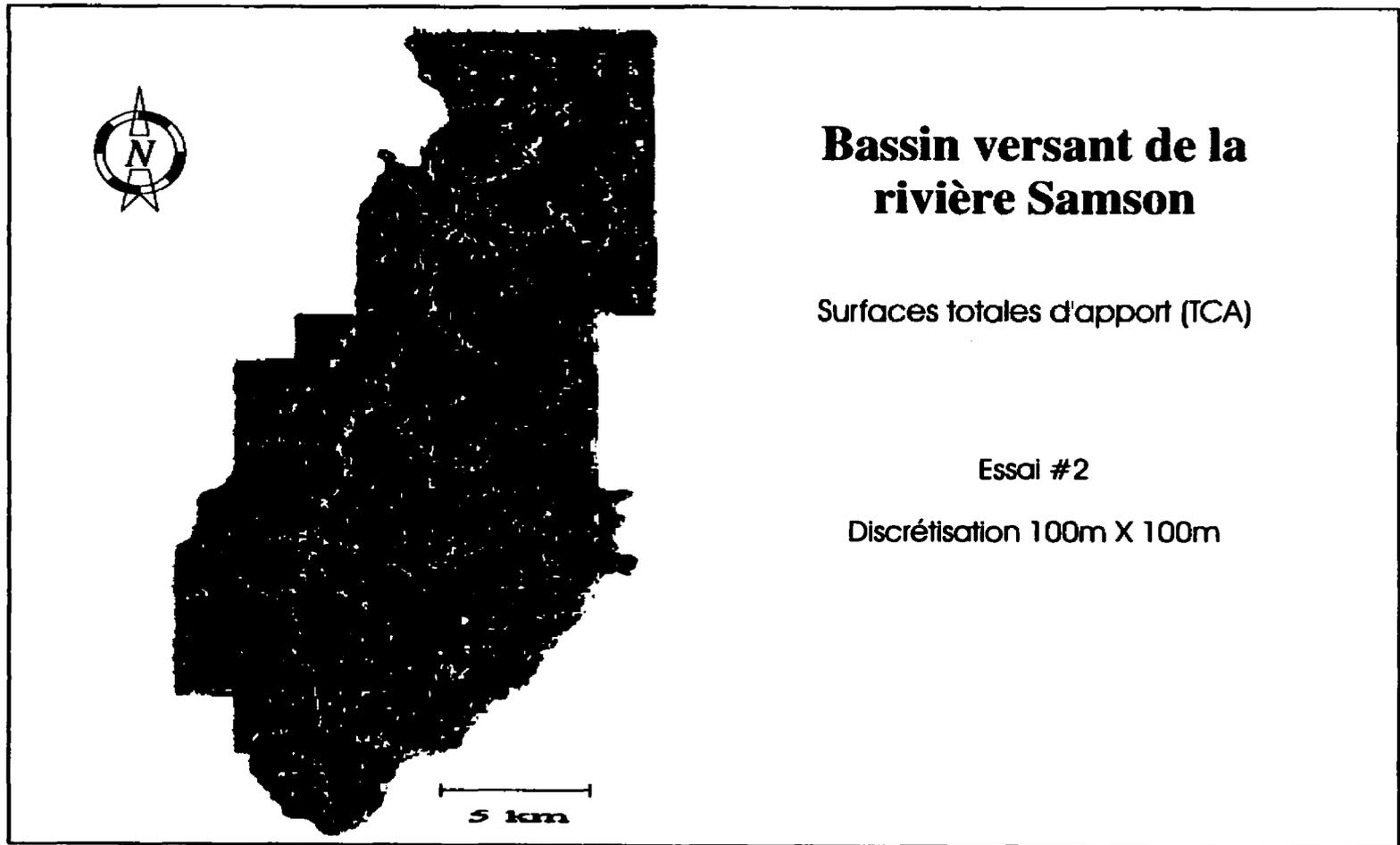


Figure 4.17 Surfaces totales d'apport (TCA) à une discrétisation de 100m x 100m, deuxième série d'essais.

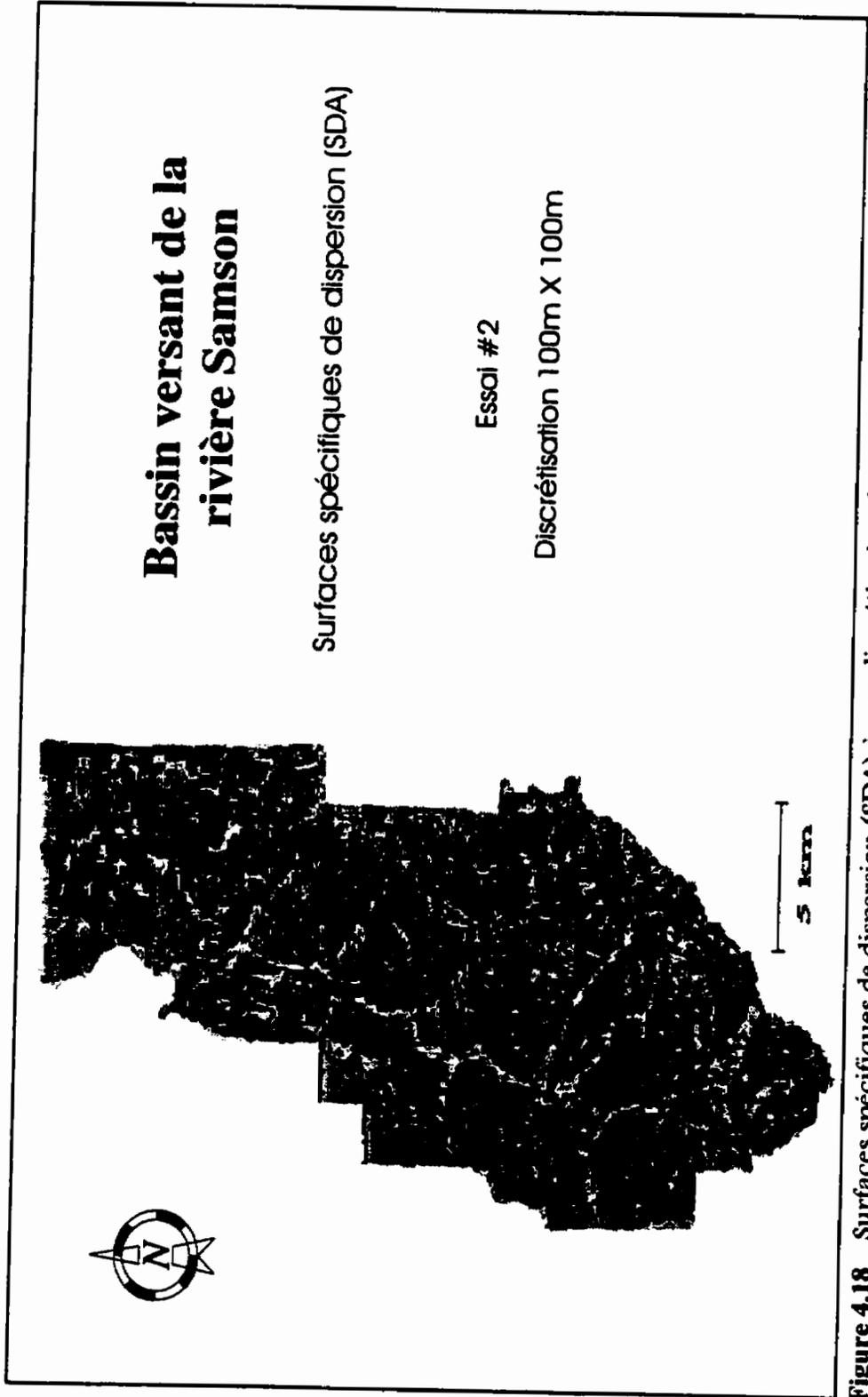


Figure 4.18 Surfaces spécifiques de dispersion (SDA) à une discrétisation de 100m x 100m, deuxième série d'essais.

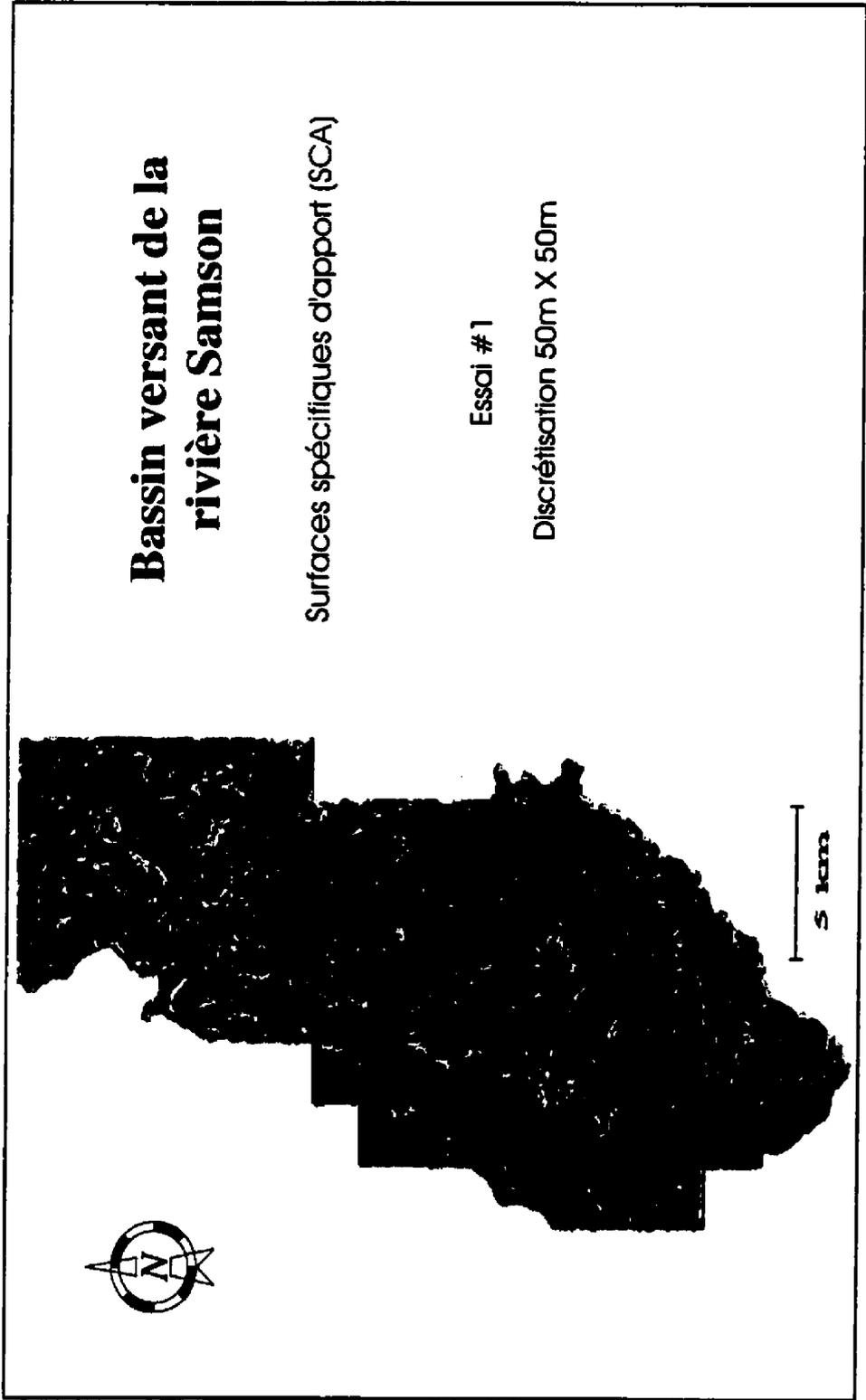


Figure 4.19 Surfaces spécifiques d'apport (SCA) à un discrétisation de 50m x 50m, première série d'essais.

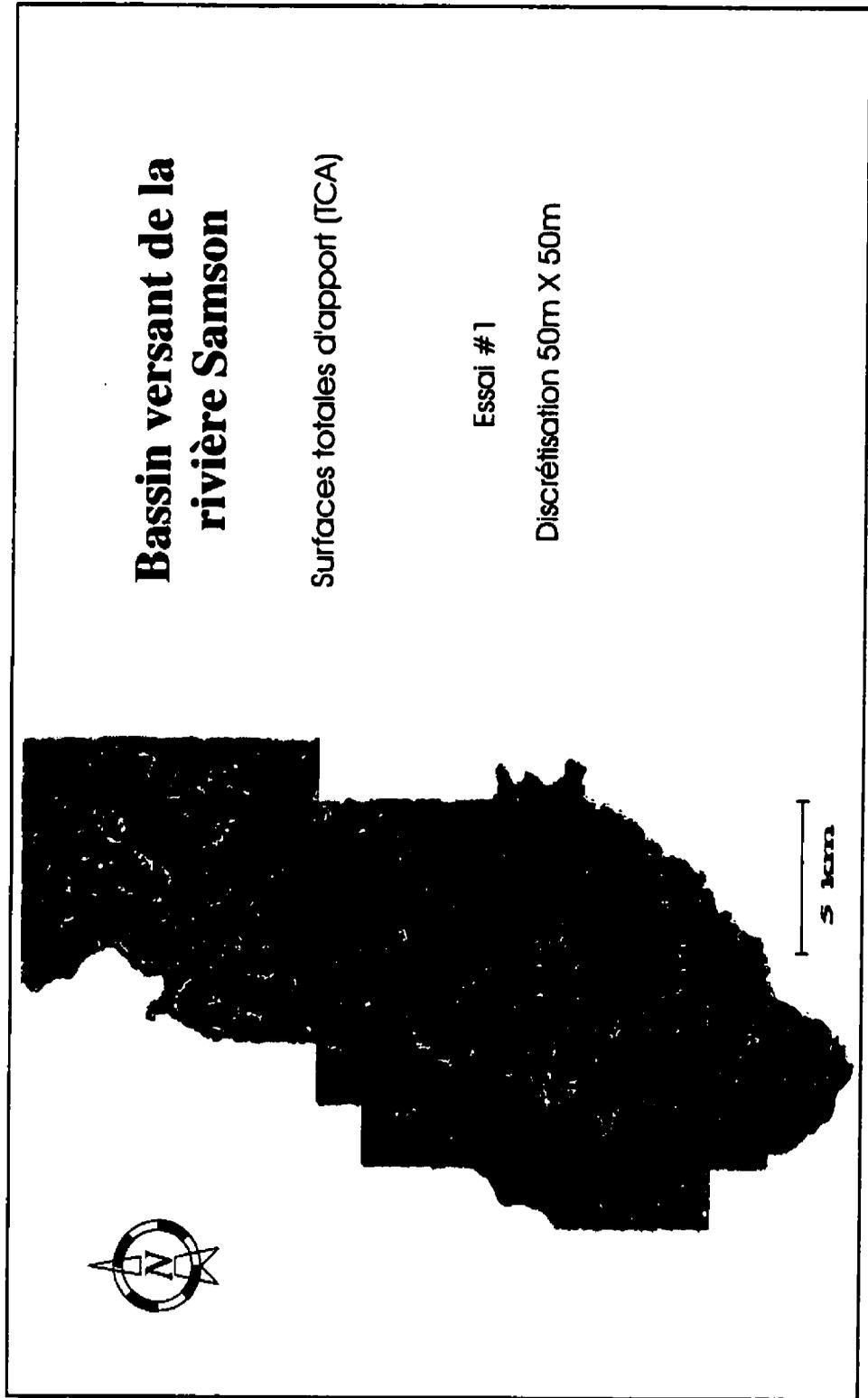


Figure 4.20 Surfaces totales d'apport (TCA) à une discrétisation de 50m x50 m, première série d'essais.

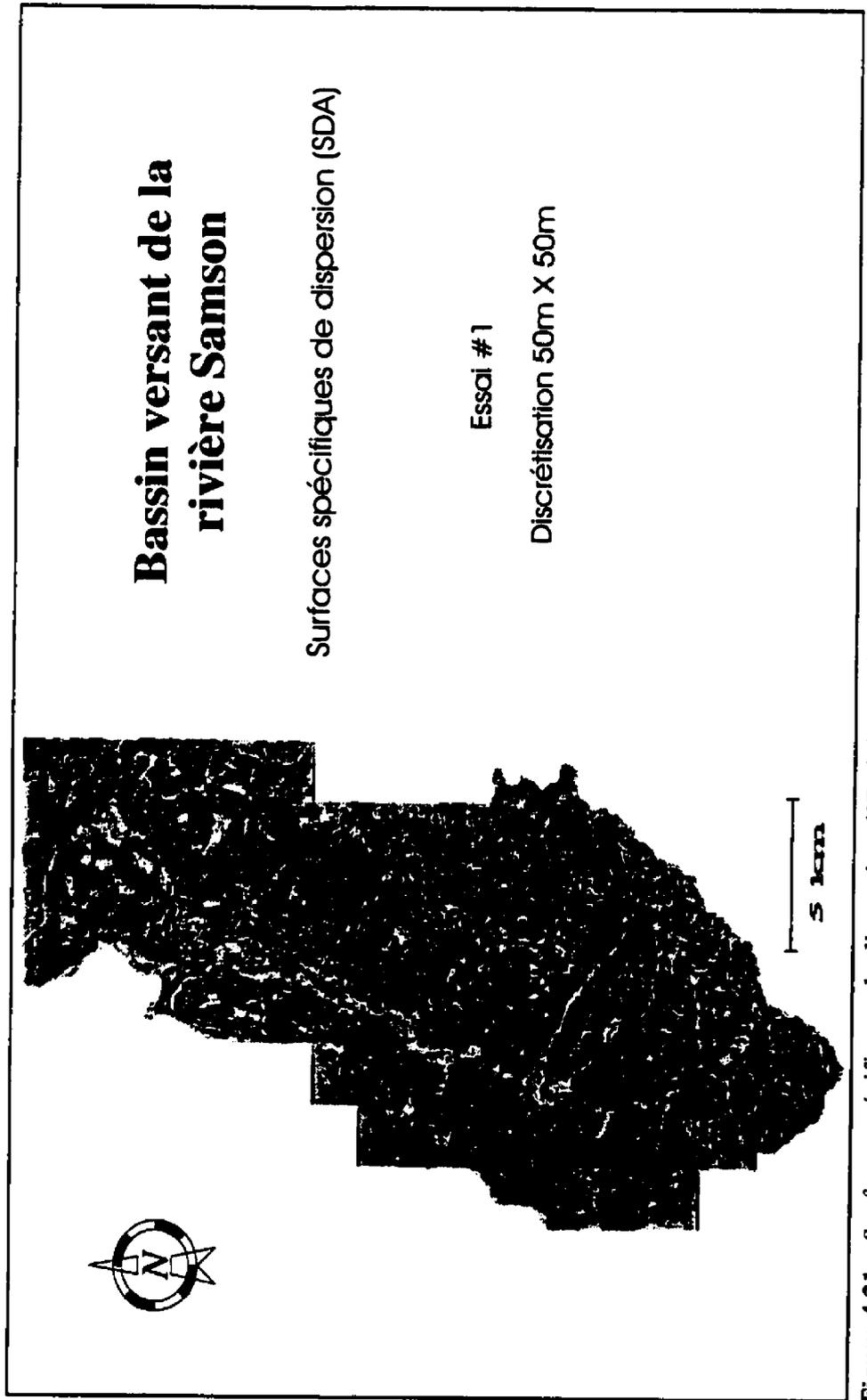


Figure 4.21 Surfaces spécifiques de dispersion (SDA) à une discrétisation de 50m x 50m, première série d'essais.

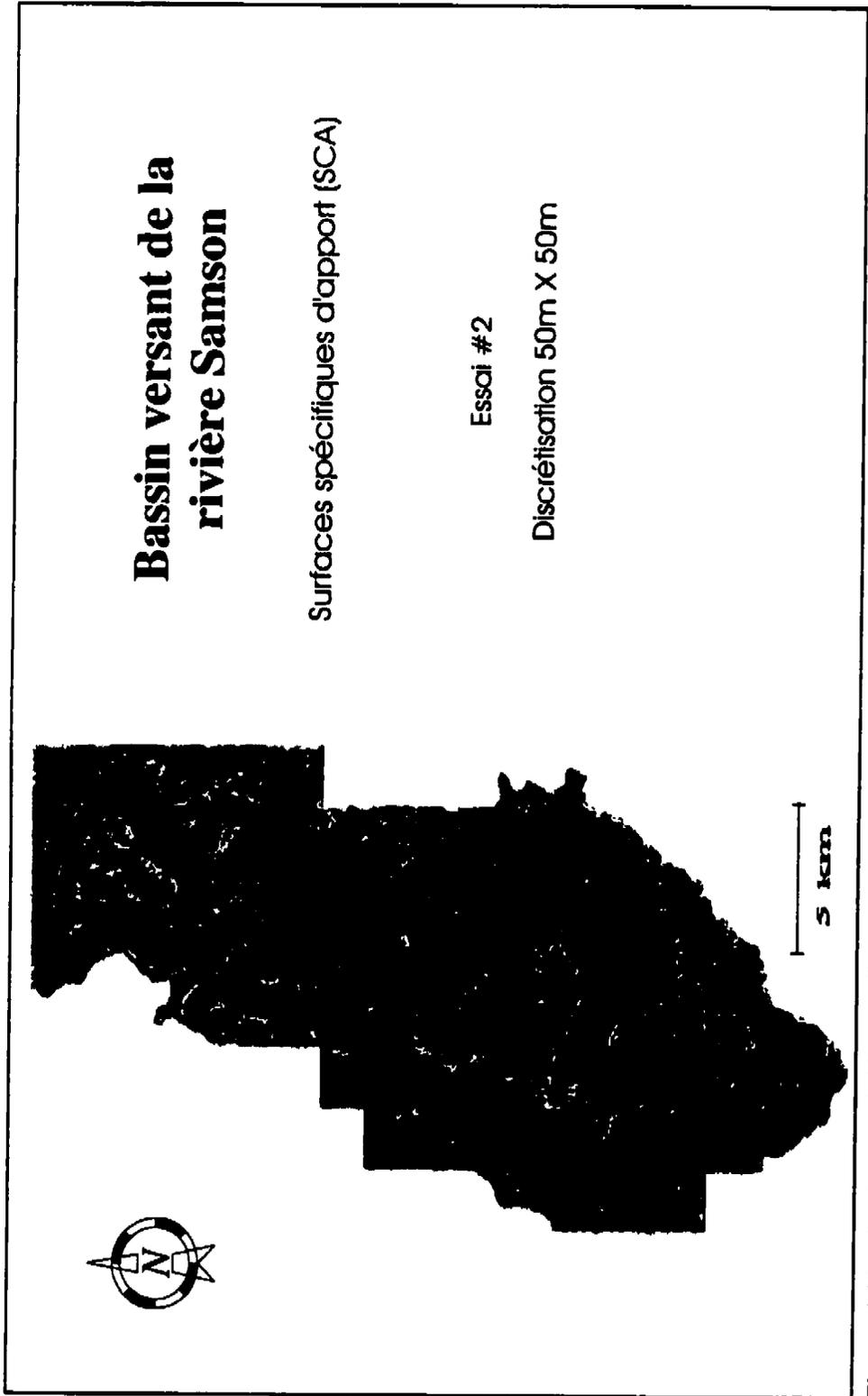


Figure 4.22 Surfaces spécifiques d'apport (SCA) à une discrétisation de 50m x 50m, deuxième série d'essais.

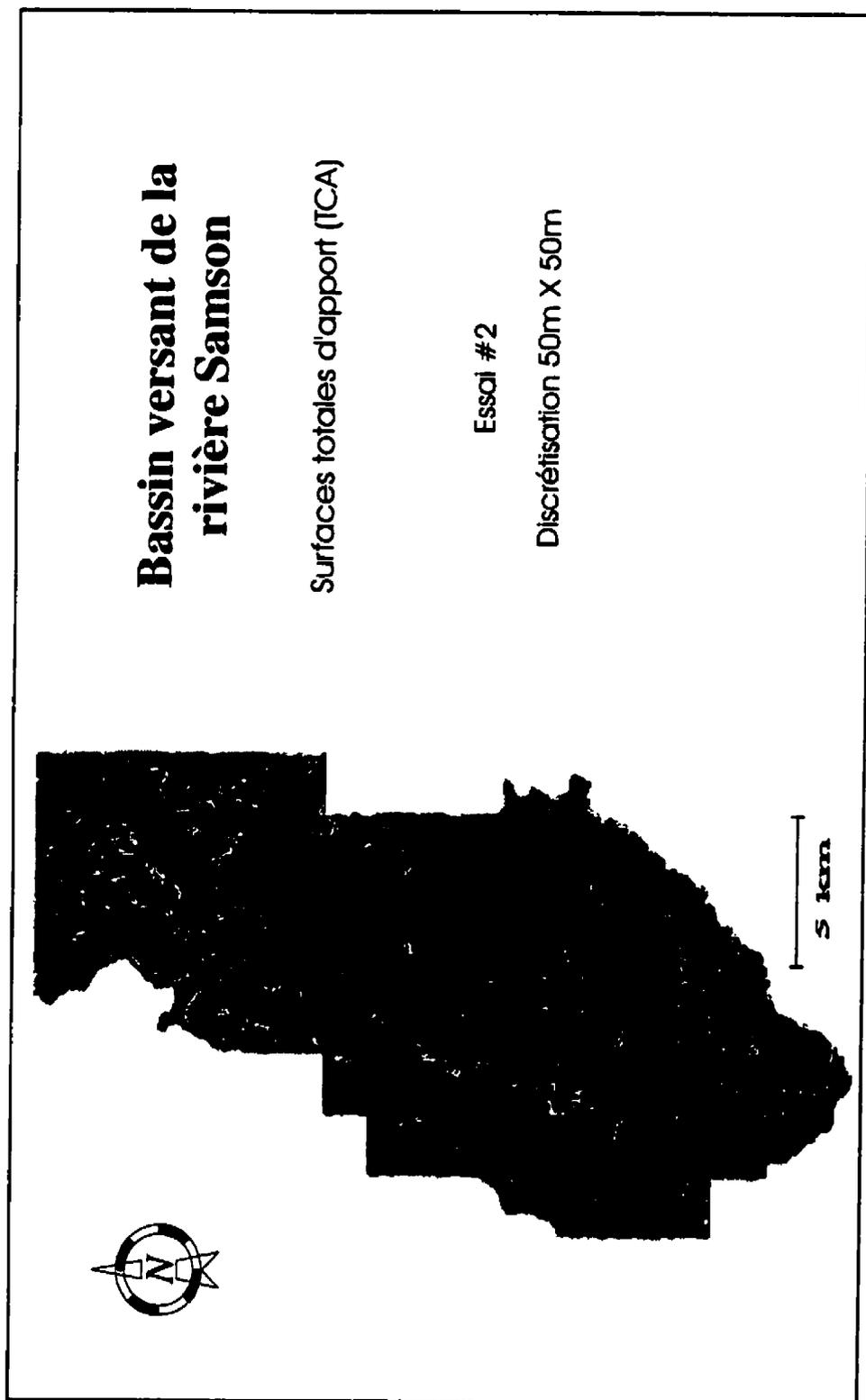


Figure 4.23 Surfaces totales d'apport (TCA) à une discrétisation de 50m x 50m, deuxième série d'essais.

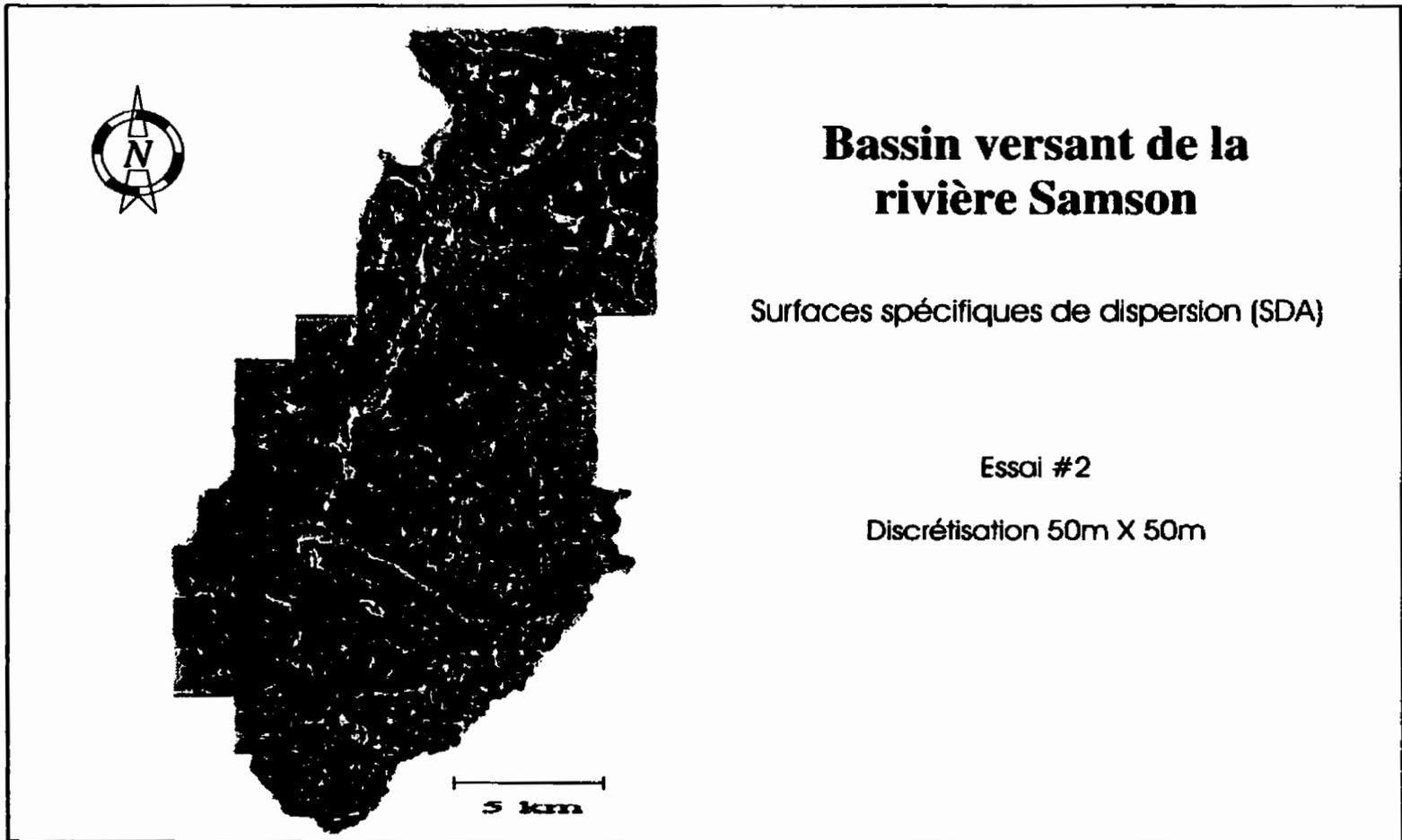


Figure 4.24 Surfaces spécifiques de dispersion (SDA) à une discrétisation de 50m x 50m, deuxième série d'essais.

ou puits qui indique une région des lacs ou de très grandes zones inondables lors des pluies. Selon la topographie et le réseau hydrographique, il existe bien des lacs dans cette région, à l'ouest de la rivière Samson, mais pas de l'envergure suggérée par la carte des SCA.

Même si sur la carte des surfaces spécifiques de dispersion les grandes zones noires ne se retrouvent pas et que la rivière Samson y est mieux définie, il est clair que cette discrétisation est insuffisante et elle n'est pas utilisée dans la deuxième série d'essais.

Essai #2 (100m x 100m)

Les figures 4.13 à 4.18 représentent les surfaces d'apport et de dispersion avec une discrétisation de 100 mètres par 100 mètres pour les deux séries d'essais confondues.

La netteté des cartes s'est grandement améliorée par rapport à la discrétisation précédente. Le réseau hydrographique est plus facilement identifiable et les zones noires se précisent. Dans la carte des surfaces spécifiques de dispersion, le réseau hydrographique ressort en teintes pâles signifiant des surfaces où la dispersion est faible. En effet, ces cellules qui font partie du réseau hydrographique se trouvent généralement en aval des chemins de ruissellement et possèdent donc une dispersion faible.

À l'inverse, les cellules dont l'altitude est supérieure à celle de leurs voisines possèdent une dispersion élevée, qui se traduit par une couleur plus sombre sur la carte des SDA. En se basant sur le plan topographique, il devrait être possible de retracer précisément ces cellules sur la carte des SDA. La discrétisation n'est cependant pas suffisante pour permettre cet exercice.

Sur les cartes des surfaces d'apport, le réseau hydrographique est plus apparent qu'avec la discrétisation précédente et, quoique les cellules en bordure des cours d'eau soient assez sombres, les cellules définissant les cours d'eau sont étrangement pâles. Les cellules des cours d'eau devraient posséder les apports les plus élevés et donc être très foncées. Le modèle numérique de terrain n'étant pas assez raffiné, l'altitude associée aux cellules dans les cours d'eau est probablement équivalente à celles de leurs voisines. Dans un chemin, une cellule dont toutes les voisines sont des puits ne sera jamais évaluée, car l'exécution du programme se termine toujours sur le premier puits rencontré et n'atteint jamais les cellules qui se trouvent cernées par des cellules puits. Moins le modèle numérique d'élévation de terrain est précis, plus le nombre de puits est important à cause de l'erreur sur l'altitude des cellules. Il faut donc se rabattre sur une meilleure discrétisation du modèle de terrain pour que la modélisation soient valable au niveau des cours d'eau.

Essai #3 (50m x 50m)

Ces résultats sont présentés aux figures 4.19 à 4.24. Sur les cartes de surfaces d'apport (figures 4.19 et 4.22), la quantité de cellules pâles au centre du cours d'eau a diminué, mais elles n'ont pas complètement disparu. Pour que la modélisation soit valable, il faut que les cours d'eau se distinguent par une couleur sombre, s'assombrissant vers l'exutoire. Ce qui n'est pas le cas, car la discrétisation, quoique présentant une nette amélioration comparativement à celle de 200 m par 200 m, n'est pas suffisante.

Des essais sont tentés avec une discrétisation de 25 m par 25 m mais, tel que mentionné précédemment, les fichiers générés par le logiciel SPANS sont énormes et le temps de calcul de plusieurs jours. De plus, il n'est pas assuré que cette nouvelle discrétisation donne de bons résultats car, même si la discrétisation de la matrice dans DEMONII est accrue, le modèle de terrain duquel sont extraites la données de la matrice demeure le même pour tous les essais.

Pour améliorer la modélisation du ruissellement dans DEMONII, il faudrait raffiner davantage le modèle numérique d'élévation de terrain, le rendre plus représentatif de la réalité. L'augmentation de la discrétisation du modèle de terrain n'est pas nécessairement le meilleur moyen, car il faut conserver la cohérence entre la dimension des quad des cartes dans le SIG et la dimension des

cellules de la matrice de DEMONII. Cependant, des fonctions de lissage et de krigeage peuvent accroître significativement la qualité du modèle numérique d'élévation de terrain et minimiser les problèmes causés par le trop grand nombre de cellules puits.

Une modification du traitement des cellules puits dans le programme DEMONII est un autre moyen qui permettrait d'améliorer la modélisation du bassin versant. Une routine qui enregistre le niveau d'eau dans les puits peut être intégrée dans le programme. L'altitude sur une cellule puits est additionnée à la hauteur d'eau qui s'y trouve. Lorsque cette somme est supérieure à l'altitude des cellules voisines la cellule n'est plus un puits et le chemin poursuit son cours. Cet ajout rapprocherait davantage la modélisation du phénomène réel.

La différence entre les deux séries d'essais n'est pas apparente sur les résultats graphiques à cause de la méthode de classification des données utilisée lors de création des cartes. L'ensemble des données doit être réparti sur un maximum de 256 classes correspondant au nombre de couleurs disponibles dans la palette. Le logiciel SPANS définit lui-même les intervalles inégaux formant ces classes tout en assurant une répartition de la superficie équitable pour chaque classe.

Même si les valeurs numériques diffèrent entre le premier et le second essai, les conditions similaires de bassin versant et de discrétisation, jumelées à une méthode de classification par proportion de superficie, donnent des résultats

graphiques très ressemblants. Les valeurs de surfaces totales d'apport relevées à l'exutoire sont cependant significativement différentes pour chaque essai tel qu'en témoigne le tableau 4.5. Les valeurs présentées dans ce tableau sont les valeurs maximales de surface totale d'apport sur le cours d'eau et le plus près possible de l'exutoire. Dans les calculs, les surfaces étant multipliées par une hauteur de pluie unitaire sont donc exprimées en mètre cube d'eau par millimètre de pluie et le terme surface est remplacé par volume. Dans ces cas-ci, même si la convergence n'est pas vraiment absolue, la largeur de la rivière à l'exutoire est inférieure à la dimension d'une cellule (50 m ou 100 m) et la surface totale d'apport d'une seule cellule est suffisante.

Tableau 4.5 Volume total d'apport

	Première série	Deuxième série
Essai #1 (200m x 200m)	46.93 m ³ /mm	---
Essai #2 (100m x 100m)	58 730.6 m ³ /mm	121 959.5 m ³ /mm
Essai #3 (50m x 50m)	28 993.5 m ³ /mm	74 305.5 m ³ /mm

La dimension réelle du bassin versant est 110 km² et son coefficient de ruissellement moyen est de 0,65 (Monfet, 1979). Par le petit calcul ci-après, on obtient la quantité qu'il devrait y avoir à l'exutoire.

Superficie du bassin	*	hauteur de pluie	*	Coefficient de ruissellement	=	Volume d'eau à l'exutoire
110*10 ⁶ m ²	*	1	*	0,65	=	71 500 000 m ³ /mm

Les valeurs de volume total d'apport du tableau 4.5 sont de beaucoup inférieures au volume calculé ci-dessus. Ce résultat indique que la plupart des chemins n'aboutissent pas à l'exutoire et sont interrompus en cours de route. Il vient appuyer les conclusions tirées des résultats graphiques qui montrent que les chemins ne se rendent pas au cours d'eau principal par la présence de cellules claires au centre de celui-ci. Les chemins qui ne se rendent pas au cours principal ne peuvent contribuer au TCA ce qui explique les faibles valeurs du tableau 4.5.

4.3 Calcul de débit

Les résultats de la modélisation du bassin versant n'étant pas très concluants, il est évident que le calcul de débit ne peut être exact. Il est quand même nécessaire d'examiner le calcul du temps de concentration effectué par le programme pour se rendre compte de l'ampleur de l'erreur et analyser la corrélation qui existe entre les deux types de résultats.

4.3.1 *Coefficients de Manning*

Le calcul du temps de concentration se fait en additionnant, sur le chemin le plus long, les temps de parcours de l'écoulement sur chaque cellule. Les temps de parcours sont évalués par l'équation de Manning avec deux coefficients différents selon que l'écoulement se fait en deux dimensions ou en une seule.

L'écoulement en deux dimensions se fait principalement sur le sol par ruissellement terrestre. Les profondeurs d'eau étant très faibles pour le ruissellement terrestre, Bedient et Huber (1988) suggèrent un coefficient de Manning qui varie de 0,05 pour des sols couverts d'une végétation clairsemée à 0,5 pour des sols couverts d'une végétation dense. Le bassin versant de la rivière Samson est tantôt forêt dense, tantôt pâturage, sols dénudés ou zone urbaine. Le coefficient utilisé pour l'écoulement terrestre est donc 0,3 environ à mi-chemin entre le minimum et le maximum. Cette valeur est au-dessus de la valeur médiane, car la superficie occupée par la végétation et les cultures est de beaucoup supérieure à celle des zones urbaines. Éventuellement, une matrice des coefficients de Manning pourrait servir d'intrant au programme DEMONII.

Lorsque l'écoulement à deux dimensions converge, il devient unidimensionnel formant ainsi un canal. Selon Chaudry (1993), les valeurs moyennes du coefficient de Manning varient de 0,030 pour un canal naturel propre et rectiligne à 0,050 pour un canal naturel donc le lit est jonché de grosses pierres.

Le coefficient de Manning utilisé par le programme pour l'écoulement dans un canal est de 0,04. Il est assez élevé puisque les canaux formés ont souvent une faible profondeur d'eau et qu'il n'existe pas de canaux à fond lisse dans le bassin versant de la rivière Samson.

Tableau 4.6 Coefficients de Manning utilisés pour la modélisation

Écoulement à une dimension	0,04
Écoulement à deux dimensions	0,30

Dans le cas d'une modélisation réussie, c'est deux coefficients peuvent être utilisés pour faire le calage du modèle avec des données de pluie-débit.

4.3.2 Temps de concentration

Les résultats précédents démontrent que la modélisation du bassin versant n'est pas une réussite. Il est donc normal que les calculs de temps de concentration du bassin, assujettis à cette modélisation, ne correspondent pas exactement au temps de concentration estimé dans le calcul présenté à l'annexe VIII. Le temps de concentration du bassin versant de la rivière Samson est estimé à cinq heures environ.

Cependant, il est moins normal que les valeurs, obtenues par le programme et présentées au tableau 4.7, excèdent l'approximation du temps de concentration de plusieurs ordres de grandeur. Pour obtenir des temps si grands, il faut des

vitesse très proches de zéro. La vitesse étant directement proportionnelle à la pente et à la hauteur d'eau dans l'équation de Manning, il faut s'attarder sur ces deux termes de l'équation pour expliquer les résultats complètement erronés.

Tableau 4.7 Temps de concentration obtenus du programme

Discrétisation	Première série	Deuxième série
	Temps (heures)	Temps (heures)
100m X 100m	$2,5828 \times 10^9$	$1,0316 \times 10^9$
50m X 50m	$75,5983 \times 10^6$	$542,1894 \times 10^6$

Pour les analyser plus en profondeur, le calcul du temps de concentration est refait manuellement en utilisant le parcours qui emprunte le cours d'eau principal pour se rendre au point le plus éloigné de l'exutoire comme chemin le plus long. Ce chemin est tracé dans le logiciel SPANS, sous forme vectorielle, à partir du réseau hydrographique. Le vecteur est transformé en deux cartes aux niveaux de quads 9 et 10 qui correspondent à une discrétisation de 50m par 50m et 100m par 100m, respectivement. Deux tables contenant les coordonnées, les pentes et les TCA sont ensuite créées par superposition de cartes puis importées dans un tableur.

Les graphiques de pente sur chaque cellule (figures 4.25 et 4.26) du chemin le plus long pour les essais de la deuxième série indiquent que plusieurs cellules ont une pente nulle. De plus, la précision des chiffres, de l'ordre de 0.01, est insuffisante. Malheureusement, dans le logiciel SPANS la précision des données

prélevées sur les cartes est fixe et ne peut être modifiée. Le calcul devrait être fait dans le programme DEMONII pour obtenir plus de chiffres significatifs pour les pentes. Probablement que plusieurs valeurs nulles seraient alors éliminées. Ce manque de précision fausse probablement les données, mais il n'est pas responsable d'une si grande erreur sur les temps de concentration. Il n'y a pas de chiffres inférieurs à 0,01 et les vitesses nulles provenant des pentes nulles sont éliminées dans l'algorithme du calcul du temps de parcours sur une cellule dans DEMONII, car la division par zéro n'étant pas possible dans un programme.

Le calcul du temps de concentration est refait dans le tableur en utilisant le TCA et les pentes prélevées dans SPANS et en éliminant les cellules pour lesquelles la pente est nulle. Le tableau 4.8 fournit les résultats obtenus.

Tableau 4.8 Temps de concentration obtenus par calcul manuel

Discrétisation	Deuxième série
	Temps (heures)
100m X 100m	103.8
50m X 50m	85.0

L'ordre de grandeur des temps de concentration est nettement plus réaliste. La différence entre ce calcul et le calcul fait à l'exécution du programme c'est le moment où intervient le calcul. Lors du calcul manuel le programme est

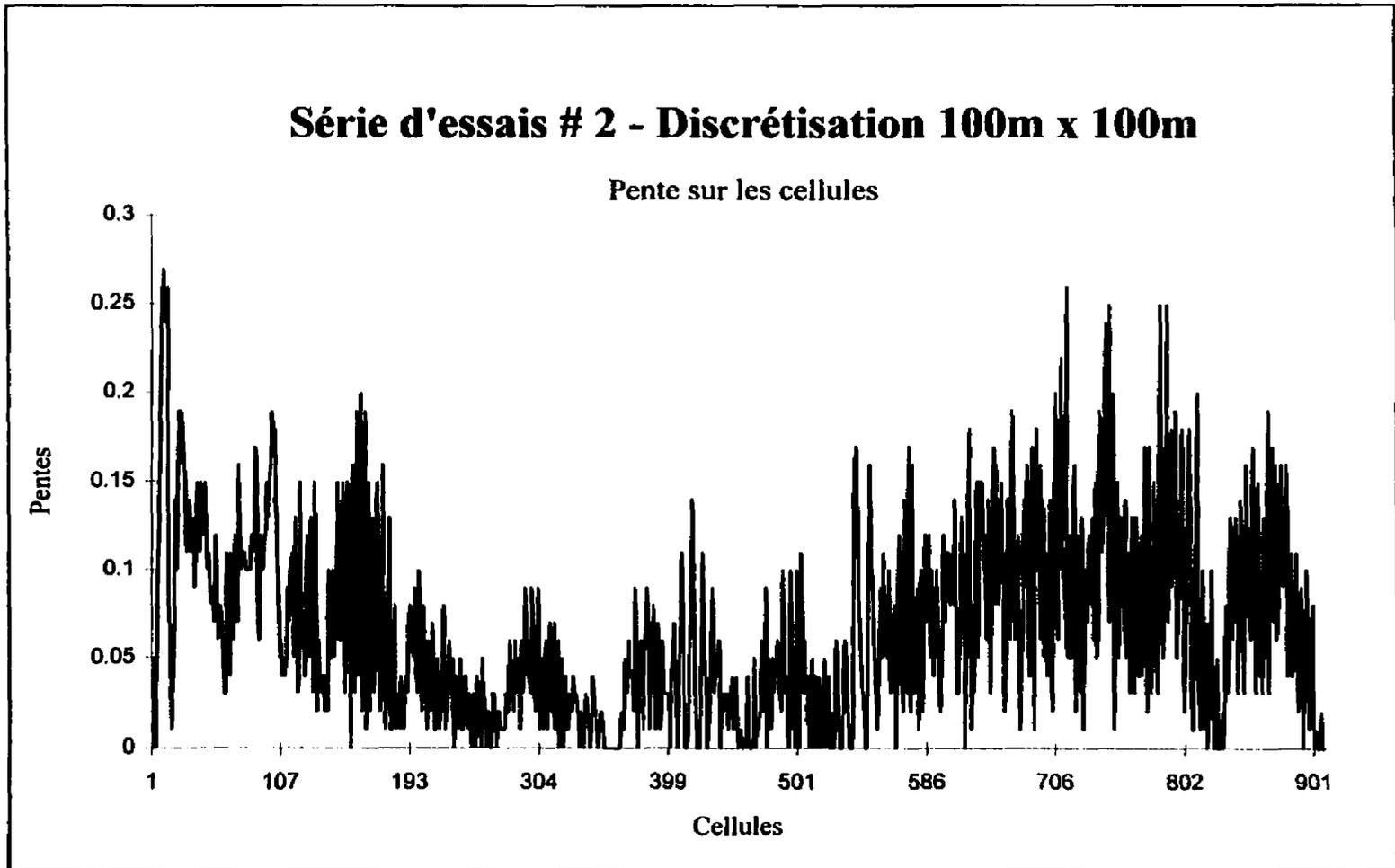


Figure 4.25 Graphique des pentes des cellules pour une discrétisation de 100m par 100m.

Série d'essais # 2 - Discrétisation 50m x 50 m

Pente sur les cellules

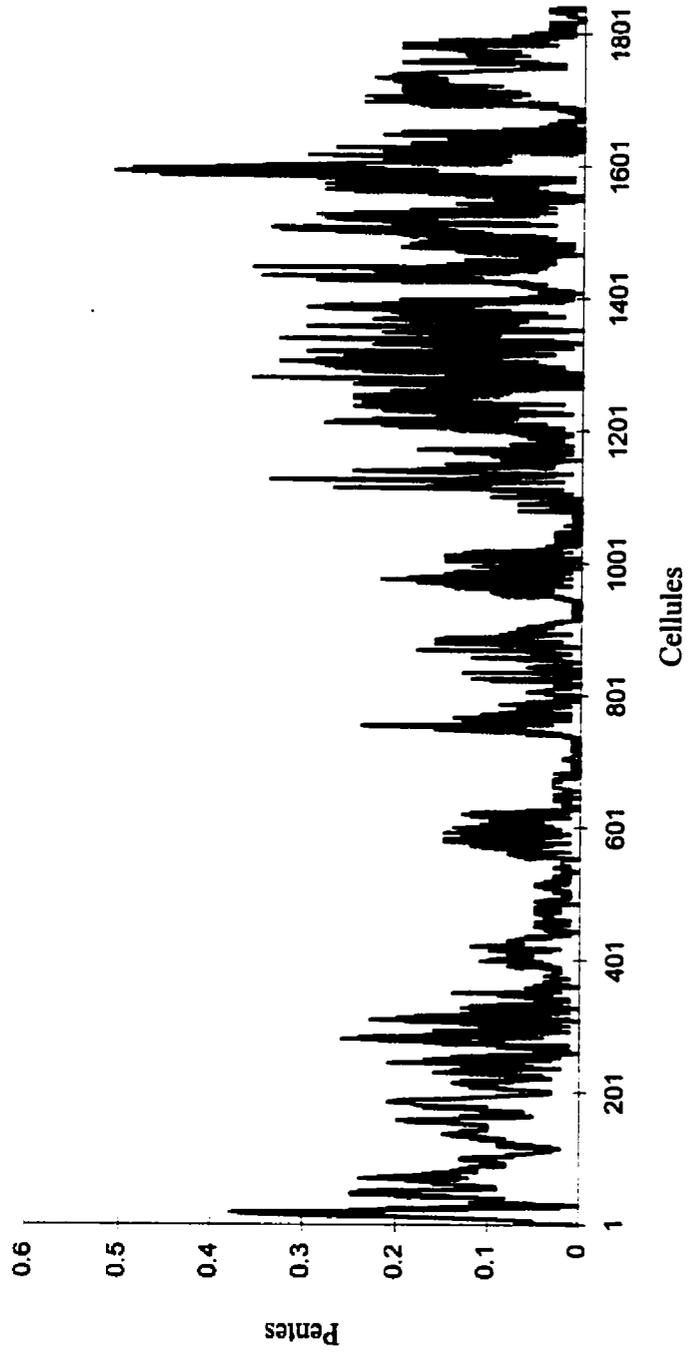


Figure 4.26 Graphique des pentes des cellules pour une discrétisation de 50m par 50m.

terminé et la valeur de TCA provienne de la version finale de la matrice. Ce n'est pas le cas dans le programme puisque le calcul sur les cellules des chemins se fait au fur et à mesure que se construit la matrice TCA. Les hauteurs d'eau utilisées sont souvent très faibles surtout dans les premiers chemins ce qui engendre des vitesses presque nulles et des temps de concentration très élevés.

Pour résoudre ce problème, il faudrait faire une première simulation sans calcul de temps de concentration et emmagasiner la matrice TCA dans un fichier. La simulation est reprise une deuxième fois en faisant le calcul du temps de parcours avec les TCA enregistrés dans le fichier pendant la première simulation. Il est moins coûteux en terme d'espace disque de refaire la simulation que d'enregistrer tous les parcours.

La très faible hauteur d'eau sur certaines cellules, due à un mauvais synchronisme des calculs dans le programme, est la cause principale des mauvais résultats. Le calcul manuel du temps de concentration démontre qu'une modification dans l'algorithme du programme permettrait d'obtenir des temps de concentration qui se rapprochent du temps estimé. L'augmentation de la précision des pentes ainsi que la manipulation des coefficients de Manning pour le calibrage du modèle peuvent par la suite aider à parfaire la modélisation du bassin versant.

Comme pour les résultats graphiques, il est peu probable que le fait d'augmenter la discrétisation du bassin dans le programme produise de meilleurs résultats à cause du caractère adimensionnel de la pente. Cependant, un raffinement du modèle numérique d'élévation de terrain par des fonctions de lissage ou de krigeage permettrait sûrement une meilleure modélisation et un meilleur calcul de temps de concentration. Il permettrait aussi de remédier à certains problèmes rencontrés lors de l'exécution initiale du programme. À cette occasion, une des causes de l'arrêt prématuré du programme était la présence de boucles sur plusieurs cellules qui empêchaient le programme de stopper son exécution par la rencontre d'une cellule puits. Même si le problème a été contourné par l'ajout d'une fonction qui permet de détecter ces boucles et d'arrêter le programme, le problème persiste et le chemin emprunté par le ruissellement n'est probablement pas conforme à la réalité.

Le rôle que joue le modèle numérique d'élévation de terrain est évidemment très important pour obtenir une bonne modélisation du bassin versant et un calcul temps de concentration acceptable. Et même si les résultats de temps de concentration obtenus par les simulations sont erronés, le calcul manuel du temps de concentration laisse supposer que l'hypothèse de l'onde cinématique est vérifiée pour ce bassin versant.

CONCLUSION

Le but de l'étude était d'utiliser un système d'information géographique pour modéliser le ruissellement direct sur un bassin versant afin d'en calculer le débit à l'exutoire. La modélisation et le calcul à même le système d'information géographique était souhaitable, mais le système disponible, SPANS, ne possède ni les fonctions de modélisation nécessaires, ni la possibilité de gérer la dimension temporelle. Il faut donc recourir à un programme externe qui utilise une répartition spatiale des données géomorphologiques d'un bassin, assez souple pour s'adapter aux données que peut lui fournir le système SPANS. Le programme DEMONII permet de modéliser un ruissellement terrestre sur un bassin versant à partir d'un modèle numérique d'élévation de terrain (DEM). Le programme est modifié pour resserrer le lien avec le système d'information géographique qui lui fournit en plus du modèle numérique d'élévation de terrain, des matrices des pentes maximales et de l'orientation des pentes maximales sur chaque cellule.

Le fonctionnement du programme DEMONII est testé sur deux formes géométriques simples soit un plan incliné et un biplan à double inclinaison (voir la figure 4.5). Le programme fonctionne comme prévu, c'est-à-dire que la surface d'apport la plus grande se trouve à l'exutoire de ces formes géométriques et il y a conservation des surfaces d'apport.

Après plusieurs embûches, la modélisation du bassin versant de la rivière Samson est réalisée. À l'examen des cartes résultantes, le réseau hydrographique est bien repérable mais, dans le cours d'eau principal et spécialement vers le nord, les cellules, très claires au centre, sont bordées de cellules très foncées. Il apparaît que l'écoulement n'atteint pas le cours d'eau, que sa course est interrompue juste avant. Pour comprendre le phénomène, il faut analyser la qualité du modèle numérique d'élévation de terrain et la topographie de la région.

Au centre et au nord du bassin, la rivière principale coule dans une plaine et le manque de précision dans le modèle numérique d'élévation de terrain représente mal la topographie. La pente du terrain étant très faible, plusieurs cellules aux abords des cours d'eau possèdent la même élévation et sont considérées comme des puits. La propagation de l'écoulement se fait, mais elle est interrompue juste avant son arrivée dans le cours d'eau ce qui explique les zones claires au centre du cours d'eau principal. Pour un bassin en région montagneuse le problème de la discrétisation aurait été beaucoup moins important à cause des fortes pentes.

En plus de faire une mauvaise représentation de la topographie, le modèle numérique d'élévation de terrain contient plusieurs aberrations comme

les puits qui explique aussi les boucles sur plusieurs cellules rencontrées lors de l'exécution de DEMONII.

L'écoulement qui ne parvient pas au cours d'eau principal ne peut évidemment pas être acheminé jusqu'à l'exutoire. Par conséquent, le volume total d'apport de la cellule qui représente l'exutoire n'est pas du tout représentatif de la quantité d'eau qui ruisselle jusqu'à l'exutoire après une précipitation. De plus, il est fort peu probable, voire impossible, que le chemin le plus long qui sert au calcul du temps de concentration produise de bons résultats. La plupart des chemins ne parvenant pas à l'exutoire, il devrait être inférieur au temps estimé par une équation empirique.

Contrairement à toute attente, le temps de concentration calculé par le programme dépasse de plusieurs ordres de grandeur le temps estimé. Le mauvais synchronisme du calcul du temps de concentration et, encore une fois, la qualité du modèle numérique d'élévation de terrain en sont responsables. En effet, le calcul du temps de parcours sur chaque cellule se fait à mesure que se construit la matrice des surfaces totales d'apport (TCA) et les hauteurs d'eau calculées à partir du TCA sont parfois très faibles, surtout dans les premiers chemins. La vitesse étant directement proportionnelle à la hauteur de l'écoulement, les vitesses calculées sont près de zéro et les temps de parcours très grands. Dans ces conditions, le chemin le plus long calculé par le programme

n'est pas du tout le chemin qui prend sa source au point le plus éloigné de l'exutoire pour aboutir à celui-ci. C'est un chemin dont les cellules ont des hauteurs d'eau très faibles. Le modèle numérique d'élévation de terrain avec ses puits et ses pentes nulles ne fait qu'empirer les résultats.

En définitive, la modélisation ne donne pas de mauvais résultats sur l'ensemble du territoire. Elle est réussie pour les cas tests et, selon les résultats graphiques, semble aussi réussie sur quelques sous-bassins. Elle peut être applicable dans certains cas particuliers, spécialement dans les régions les plus accidentées là où les bassins versant ont une forme qui ressemble à celle du biplan utilisé dans les tests. Par contre, il y aurait lieu d'améliorer la modélisation et la qualité du modèle numérique d'élévation de terrain pour la rendre applicable pour tous les types de bassins versants. Une plus grande précision et une meilleure qualité du modèle numérique d'élévation de terrain, obtenues par l'application de fonctions de lissage et de krigeage, aurait permis d'obtenir de meilleurs résultats graphiques et numériques dans le cas du bassin versant de la rivière Samson. Surtout dans les plaines où les pentes sont faibles.

Le programme DEMON est très performant pour des bassins versants d'environ un kilomètre carré de superficie, mais il devient extrêmement lourd à gérer lorsque la taille du bassin augmente. Afin de rendre le programme plus performant et applicable à différents types de bassins versants, il faudrait

séparer le bassin en plusieurs sous-bassins de petite taille et inclure un modèle d'écoulement en rivière pour faire le calcul de débit. Ce modèle d'écoulement tiendrait compte des différents apports des tributaires au cours d'eau principal.

L'utilisation d'un système d'information géographique pour la modélisation d'un bassin est une bonne innovation, surtout pour le calcul du coefficient de ruissellement. Une fois la méthode de calcul et les équations développées, le calcul des coefficients de ruissellement devient une simple routine. Il suffit d'intégrer dans le système les cartes de type de sol et d'utilisation du sol à partir d'une ou de plusieurs images satellites classifiées. Pour la topographie, elle doit être numérisée dans un logiciel de cartographie à partir d'une carte topographique balayée électroniquement. Elle peut aussi être captée à partir de photos aériennes pour obtenir des données plus récentes et plus précises. Par la suite, il suffit de suivre la routine pour exécuter le calcul. Les systèmes d'information géographique rendent possible la gestion et la manipulation des données géomorphologiques tout en conservant leur référence spatiale, un atout dans le domaine de l'hydraulique et de l'hydrologie.

Par contre, le système SPANS n'est peut-être pas le plus approprié. Il est très peu convivial et devient vite essoufflé lorsque les fichiers dépassent les quelques Mega-octets. Il existe sûrement un système d'information géographique plus convivial qui manipule bien les données matricielles, ne bronche pas devant les

données volumineuses, et qui, surtout, offre la possibilité d'intégrer une routine de modélisation de bassin versant et de calcul de débit à même ses fonctions. Beaucoup de manipulations et de transferts de données seraient ainsi éliminés.

Malgré tout, la réalisation de cette étude, même si elle n'est pas en tout point une réussite, constitue un pas en avant dans le domaine de la modélisation des bassins versants. Elle est un essai de plus dans le coffret des tentatives de modélisation à l'aide des systèmes d'information géographique. Elle ouvre la porte à d'autres réalisations et permettra peut-être à certains de ne pas répéter les mêmes erreurs afin de poursuivre plus en avant le développement des techniques de modélisation de bassin versant.

BIBLIOGRAPHIE

ABTEW, W., OBEYSEKERA, J. et SHIH, G. (1993 avril). « Spatial Analysis for Monthly Rainfall in South Florida », Water Resources Bulletin, vol. 29, n°2, p. 179-188.

BATHURST, J.C. (1986). "Sensitivity analysis of the Systeme Hydrologique Europeen for an upland catchment." J. Hydrol., 87(1/2), 103-123.

BEDIENT, P.B. et HUBER, W.C. (1988). Hydrology and Floodplain Analysis, Reading, Addison-Wesley Publishing Company, 650 pages.

BEVEN, K.J. (1989). Changing ideas in hydrology - the case of physically based models. J. Hydrol. 87(1/2), 79-102

BRILLY, M., SMITH, M. et VIDMAR A. (1993). « Spatially oriented surface water hydrological modelling and GIS », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p.547-557.

CHAUDRY, M.H. (1993). Open-Channel Flow, Englewood Cliffs, Prentice Hall, 483 pages.

CHOW, V.T., MAIDMENT, D.R. et MAYS, L.W. (1988). Applied Hydrology, McGraw-Hill Book Company, New York, 572 pages.

CLARK, M. J. (1993). « Data constraints on GIS applications development for water resources management », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p.451-463.

COSTA-CABRAL, M. et BURGESS, S.J. (1994 June). « Digital elevation model networks (DEMON): A model of flow over hillslopes for computation of contributing and dispersal areas », Water Resources Research, vol. 30, n°6, p.1681-1692.

CRISMAN, R.D., CHIU, C.L., YU, W., MIZUMURA, K. et CORBU, I. (1993 novembre). « Uncertainties in Flow Modeling and Forecasting for Niagara River », Journal of Hydraulic Engineering, vol. 119, n°11, p. 1231-1250.

DELCLAUX, F. et BOYER, G. (1993). « Exemple d'utilisation d'un SIG pour la gestion des données d'un modèle hydrologique à mailles carrés », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p.475-484.

DEVANTIER, B.A. et FELDMAN, A.D. (1993 mars/avril). « Review of GIS Application in Hydrologic Modeling », Journal of Water Resources Planning and Management, vol. 119, no. 2, p. 246 -261.

DUCRUC, J.P. et De SÈDE, M.H. (1991 juin). « Contribution de la cartographie écologique à la modélisation hydrologique spatiale du bassin supérieur de la rivière Eaton. Partie I : la carte écologique », Gouvernement du Québec, Ministère de l'environnement Planification écologique - Série de l'inventaire du capital-nature, no. 10.

EASH, D.A. (1994 février). « A Geographic Information System Procedure to Quantify Drainage-Basin Characteristics », Water Resources Bulletin, vol. 30, n°1, p. 1-8.

ENVIRONMENT CANADA, INLAND WATERS DIRECTORATE, WATER RESOURCES BRANCH (1986 August). Phase I Review of Operational Remote

Sensing Techniques and Streamflow Forecasting Techniques, Study of Methodologies of Streamflow Forecasting Incorporating Remotely Sensed Data,
A.J. Robinson & Associates Inc.

FARISSIER, P., et GIVONE, P. (1993). « Mapping and management of flood plains », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p.485-489.

FEDRA, K. (1993). « Models, GIS, and expert systems: integrated water resources models », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p. 297-308.

FELDMAN, A.D. (1992 mai/juin). « Systems Analysis Applications at the Hydrologic Engineering Center », Journal of Water Resources Planning and Management, vol. 118, n°3, p. 249-261.

GARBRECHT, J. et LAWRENCE, W.M. (1993 décembre). « Network and Subwatershed Parameters Extracted from Digital Elevation Models: the Bills Creek Experience », Water Resources Bulletin, vol. 29, n°6, p. 909-916.

GARG, N. K. et SEN, D. J. (1994 avril). « Determination of Watershed Features for Surface Runoff Models », Journal of Hydraulic Engineering, vol. 120, n°4, p. 427-447.

HAY, L.E., BATTAGLIN, W.A., BRANSON, M.D. et LEAVESLEY, G.H. (1993). « Application of GIS in modelling winter orographic precipitation, Gunnison River Basin, Colorado, USA », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management,

Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p.491-500.

HROMADKA II, T.V., McCUEN, R.H. et YEN, C.C. (1986 novembre). « Comparison of Overland Flow Hydrograph Models », Journal of Hydraulic Engineering, vol. 113, n°11, p. 1422-1440.

INTERA TYDAC Technologies Inc. (1993). SPANS GIS Reference Manuel, 6^e éd.

JOHNSON, H.P. et BRAKENSIEK, D.L. (1982). Hydrologic Modeling of Small Watershed, Michigan, American Society of Agricultural Engineers James A. Basselman technical editor, C.T. Haan editor, 533 pages.

KERN T.J. et STEDNICK, J.D. (1993). « Identification of heavy metal concentrations in surface waters through coupling of GIS and Hydrochemical models », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p.559-567.

KIENZLE, S.W. (1993). « Application of a GIS for simulating hydrological responses in developing regions », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p. 309-318.

KOUWEN, N., SOULIS, E.D., PIETRONIRO, A., DONALD, J. et HARRINGTON, R.A. (1993 mai/juin). « Grouped Response Units for Distributed Hydrologic Modeling », Journal of Water Resources Planning and Management, vol. 119, n°3, p. 289-305.

LECLERC, G., et GOLAY, F. (1988). « Les systèmes d'information urbains et leur potentiel pour la planification, le dimensionnement et l'entretien des infrastructures urbaines », Cycle de conférences: Informatisation municipale et systèmes d'information géographique, Foire nationale des municipalités, Braga, p. 1-18.

LEIPNIK, M.R., KEMP, K.K. et LOAICIGA, H.A. (1993 mars/avril). « Implementation of GIS for Water Resources Planning and Management », Journal of Water Ressources Planning and Management, Vol. 119, n° 2, p.184-205.

LLAMAS, J. (1993). Hydrologie générale, Principes et applications, Boucherville, 2^e édition, Gaëtan Morin éditeur, 527 pages.

MAGUIRE, D.J., GOODCHILS, M.F. et RHIND, D.W. (1991). Geographical Information Systems: Principles and Applications, Harlow, Essex, England, New York: Longman Scientific & Technical; John Wiley & Sons, 2 vol., 649 pages et 447 pages.

MAIDMENT, D.R. (1991). « GIS and hydrological modeling », Proc. of First Int. symposium/Workshop on GIS an Environmental Modeling, held in Boulder, Colorado, september 1991, (ed. by M.F. Goodchild, B.O. Parks & L.T. Steyaert), Oxford University Press, New York.

MAIDMENT, D.R. (1993). « Developing a spatially distributed unit hydrograph by using GIS », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p. 181-192.

MARTZ, L.W. et GARBRECHT, J. (1993 décembre). « Automated Extraction of drainage Network and Watershed Data from Digiatl elevation Models », Water Resources Bulletin, vol. 29, n°6, p. 901-908.

McCUEN, R.H. (1982). A Guide to Hydrologic Analysis Using SCS Methods, Englewood Cliffs, Prentice-Hall, 145 pages.

McKINNEY, D.C., MAIDMENT, D.R. et TANRIVERDI, M. (1993 mars/avril). « Expert Geographic Information System for Texas Planning », Journal of Water Resources Planning and Management, vol. 119, n°2, p. 170-183.

MONFET, J. (1979). Évaluation du coefficient de ruissellement à l'aide de la méthode SCS modifiée, Gouvernement du Québec, Ministère des Richesses naturelles, Service de l'hydrométrie, Rapport HP-51, Québec, 35 pages.

MUZIK, I. et CHANG, C. (1993). « Flood simulation assisted by a GIS », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p.531-540.

PONCE, V.M. (1989). Engineering Hydrology : principles and practices, Englewood Cliffs, Prentice Hall, 640 pages.

ROSS, M.A. et TARA, P.D. (1993 mars/avril). « Integrated Hydrologic Modeling with Geographic Information Systems », Journal of Water Ressources Planning and Management, vol. 119, n° 2, p. 129-140.

SHEA, C., GRAYMAN, W., DARDEN, D., MALES, R.M. et SUSHINSKY, P. (1993 mars/avril). « Integrated GIS and Hydrologic Modeling for Countywide Drainage Study », Journal of Water Ressources Planning and Management, vol. 119, no. 2, p.112-128.

SINGH, V.P. (1988). Hydrologic Systems - Rainfall-Runoff Modeling, Volume 1, Englewood Cliffs, Prentice Hall, 480 pages.

SORMAN, A.U. et ABDULRAZZAK, M.J. (1993 janvier/février). « Flood Hydrograph Estimation for Ungaged Wadis in Saudi Arabia », Journal of Water Resources Planning and Management, vol. 119, n°1, p.45-63.

STUART, N. et STOCKS C. (1993). « Hydrological modelling within GIS: an integrated approach », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p.319-329.

TACHIKAWA, Y., SHIIBA, M. et TAKASAO, T. (1994 février). « Developement of a Basin Geomorphic Information System Using a Tin-Dem Data Structure », Water Resources Bulletin, vol. 30, n°1, p.9-17.

TAO, T. et KOUWEN, N. (1989 novembre). « Remote Sensing and Fully Distributed Modeling for Flood Forecasting », Journal of Water Resources Planning and Management, vol. 115, n°6, p. 809-823.

TECSULT (1993a). Étude de modélisation du bassin versant de la rivière Chaudière, Rapport d'étape n°1, Les effets du développement sur le comportement de la rivière, Volume 1 État de la situation, document réalisé pour Environnement Canada, Région du Québec et Gouvernement du Québec, Ministère de la Sécurité publique, Direction générale de la sécurité civile.

TECSULT (1993b). Modélisation hydrologique du bassin versant de la rivière Chaudière au moyen du logiciel SSARR, document réalisé pour Environnement Canada, Région du Québec et Gouvernement du Québec, Ministère de la Sécurité publique, Direction générale de la sécurité civile.

TECSULT (1994a). Étude de modélisation du bassin versant de la rivière Chaudière, Rapport d'étape n°2, Solutions aux inondations en eau libre, document réalisé pour Environnement Canada, Région du Québec et

Gouvernement du Québec, Ministère de la Sécurité publique, Direction générale de la sécurité civile, 112 pages.

TECSULT (1994b). Étude de modélisation du bassin versant de la rivière Chaudière, Rapport final, document réalisé pour Environnement Canada, Région du Québec et Gouvernement du Québec, Ministère de la Sécurité publique, Direction générale de la sécurité civile, 94 pages.

VAN DEURSEN, W.P.A. et KWADIJK, J.C.J. (1993). « RHINEFLOW: an integrated GIS water balance model for the river Rhine », HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceeding of the Vienna Conference, avril 1993, IAHS, Publication n°211, p.507-518.

VENTURA, S.J. et KIM, K. (1993 avril). « Modeling Urban Nonpoint Source Pollution with a Geographic Information System », Water Resources Bulletin, vol. 29, n°2, p. 189-198.

VISSMAN, W., LEWIS, G.L. et KNAPP, J.W. (1989). Introduction to Hydrology, New York, 3^e édition, Harper & Row, Publishers, 780 pages.

WALSH, M.R. (1993 mars/avril). « Toward Spatial Decision Support System in Water Resources », Journal of Water Resources Planning and Management, vol. 119, n°2, p.158-169.

WANG, D. et MAYER, L. (1993). « Wbandz: A C Program to Calibrate the Hydrologic Model and Calculate Monthly Water Balance and Palmer's Z-index », Computers & Geosciences, vol. 19, n°9, p. 1267-1283.

WARWICK, J.J. et HANESS, S.J. (1994 mai/juin). « Efficacy of ARC/INFO Application to Hydrologic Modeling », Journal of Water Resources Planning and Management, vol.120, n°3, p. 366-381.

**ANNEXE I : COPIE DE L'INTÉGRALE DE L'ARTICLE
DE COSTA-CABRAL ET BURGÉS**

NOTE TO USERS

Copyrighted materials in this document have not been filmed at the request of the author. They are available for consultation in the author's university library.

Pages 175-186

UMI

ANNEXE II : LISTAGE DU PROGRAMME DEMON ORIGINAL

A. Déclaration des variables

```

*
* -----
* DEMON_DOWNSLOPE.VAR
* -----
*
* THE PRESENT FILE IS AN INTEGRAL PART OF THE FORTRAN PROGRAM
* 'DEMON_DOWNSLOPE.F', AND IS REQUIRED FOR COMPILATION OF THIS
* PROGRAM.
*
* The present file contains the parameter specification and variable
* declaration of the program 'DEMON_DOWNSLOPE.F'. The present file
* is inserted in the code 'DEMON_DOWNSLOPE.F' at the beginning of
* each subroutine, via the following statement:
*
*       include 'demon_downslope.var'
*
* The present file may be eliminated if each entry of the above
* 'include' statement in the code 'DEMON_DOWNSLOPE.F' is replaced
* by a copy of the present file.
*
* Since the above 'include' statement is placed in every subroutine
* of the 'DEMON_DOWNSLOPE.F' program, the variables in the 'common'
* statement in the present file are shared by all subroutines.
* All variables used in the program 'DEMON_DOWNSLOPE.F' are included
* in the common statement, excepting local 'dummy' variables.
* For the meaning of each variable, consult the program Glossary.
*
*       - * -
*
* All parameters in the present file are to be specified by the user,
* for each DEM matrix. The meaning of each parameter is the following:
*
* nr:   Number of rows in DEM matrix
* nc:   Number of columns in DEM matrix
* nrl:  nrl = nr + 1
* ncl:  ncl = nc + 1
* ne:   ne = nr x nc
* dx:   Grid spacing, or pixel side, in the xx direction [L]
* dy:   Grid spacing, or pixel side, in the yy direction [L]
* anglecut: Arctangent of (dy/dx) in degrees. E.g., if dy=dx, then
*         anglecut = 45.
*
* The names of the input and output files are to be supplied to the
* user in the present file: DEMfile is the name of the DEM file to
* be read; SCAfile is the name of the output file containing the
* computed sca values; SDAfile is the name of the output file
* containing the sda values.
* The name of the DEM file is to be supplied by the user in variable
* DEMfile. The program DEMON_DOWNSLOPE.F assumes that the DEM file
* is in ASCII format and contains nr lines each with nc values. If

```

* the DEM file is in a format different from this, the subroutine
 * READ of DEMON_DOWNSLOPE.F should be changed accordingly.

 * EACH TIME THIS FILE IS EDITED, THE PROGRAM DEMON_DOWNSLOPE.F *
 * MUST BE RECOMPILED ! *

```

parameter ( nr = ,      nc = ,
&          nr1 = ,     ncl = ,
&          ne = ,
&          dx = ,      dy = ,
&          anglecut = )

common z(nr1, ncl), angle(nr1, ncl),
&      istop(nr1, ncl), iflat(nr1, ncl),
&      tca(nr1, ncl), sca(nr1, ncl),
&      tda(nr1, ncl), sda(nr1, ncl),
&      iw(ne), jw(ne), fw(ne), idirw(ne),
&      xlw(ne), x2w(ne), x3w(ne), i3w(ne), flw(ne), f2w(ne),
&      is, js,
&      in, jn, fn, idirn, x1n, x2n, x3n, i3n, f1n, f2n,
&      io, jo, fo, idiro, x1o, x2o, x3o, i3o, f1o, f2o,
&      ic, ic2

```

character*20 DEMfile, SCAfile, SDAfile

```

DEMfile = ' '
SCAfile = ' '
SDAfile = ' '

```

B. Programme

```
PROGRAM DEMON_DOWNSLOPE
```

```
PROGRAM DEMON_DOWNSLOPE, VERSION 1, APRIL 1993.
```

```
by Mariza C. Costa-Cabral and Stephen J. Burges  
Dpt. Civil Engineering, Univ. of Washington, Seattle, WA
```

```
Reference: Costa-Cabral and Burges, Digital Elevation Model  
Networks (DEMON): a model of flow over hillslopes for  
computation of contributing and dispersal areas,  
Water Resources Research, 20(6), 1994.
```

```
This program computes the specific contributing area (SCA) and  
the specific dispersal area (SDA) for each pixel of a  
rectangular DEM.
```

```
This program cannot be compiled without the file  
'demon_downslope.var', which is an integral part of this  
program, containing all parameters, the common variable  
statement, and the name of the DEM file to be read.  
The user should edit the file 'demon_downslope.var' and provide  
the name of the DEM file and the values of all parameters.  
Each time 'demon_downslope.var' is edited this program must be  
compiled again.  
Subroutine READ assumes that the DEM file is in ASCII format  
and that it contains nr rows each with nc values. For a  
different format, modify subroutine READ.
```

```
Note: This program treats all points of elevation lesser than  
or equal to those of its neighbors as sinks. Flow entering a  
sink pixel stops. If some of such local minima or flat spots  
in your data are believed not to correspond to enclosed  
depressions, then you should modify your data prior to using  
this code.
```

```
Units: If pixel sides (parameters dx and dy) are provided in  
meters (m), then the SCA and SDA are expressed in m**2 / m.
```

```
call READ  
call PITS  
call ANGLES  
call TCA_AND_TDA  
call SCA_AND_SDA  
call WRITE  
end
```

```
SUBROUTINE READ
```

```
* | This subroutine assumes that the DEM file is in ASCII format
* | and that it contains nr rows each with nc values. For a
* | different format, modify this subroutine.
```

```
* | INPUT: DEMfile, nr1, nc1
* | OUTPUT: z(,)
```

```
include 'demon_downslope.var'

open(1, file = DEMfile, status='old')

do 10 i = 2, nr1
10 read(1, *) ( z(i, j), j = 2, nc1 )

close(1)

end
```

```
* | SUBROUTINE PITS
```

```
* | This subroutine sets the indicator variable istop(,) to 1 for
* | pixels where flow paths end: the DEM boundary and pits.
* | Pits are not filled in! They are treated as sinks.
* | INPUT: nr1, nc1, z(,)
* | OUTPUT: istop(,), npits
```

```
include 'demon_downslope.var'

do 10 i = 1, nr1
10 istop(i, 1) = 1
istop(i, nc1) = 1
continue

do 20 j = 1, nc1
20 istop(1, j) = 1
istop(nr1, j) = 1
continue

do 30 i = 3, nr
do 30 j = 3, nc
z0 = z(i, j)
z1 = z(i-1, j-1)
z2 = z(i-1, j)
z3 = z(i-1, j+1)
z4 = z(i, j-1)
z5 = z(i, j+1)
z6 = z(i+1, j-1)
z7 = z(i+1, j)
z8 = z(i+1, j+1)
```



```

z4 = z(i+1, j+1)

sx = 0.5 * (- z1 + z2 - z3 + z4) / dx
sy = 0.5 * ( z1 + z2 - z3 - z4) / dy

iA = 0
iB = 0
absx = abs(sx)
absy = abs(sy)
if(sx.ne.0) iA = sx / absx
if(sy.ne.0) iB = sy / absy

factor = - iA + iB - iA*iB + 3*iA*iA + 2*iB*iB
&          + iA*iB*iB - 3*iA*iA*iB*iB

dum = 90. * factor
if(iA*iB.ne.0) dum = dum + iA*iB*atand(absy/absx)

if(abs(dum-360.) .lt. 1E-3) dum = 0.
angle(i, j) = dum
iflat(i, j) = 0

if(iA.eq.0.and.iB.eq.0) iflat(i, j) = 1
10 continue

end

```

*
SUBROUTINE TCA_AND_TDA
*

```

*
* | This subroutine computes the total contributing area, tca(i,j),
* | and the total dispersal area, tda(i,j), of each pixel.
* | Flow is generated at each pixel at a time (the "source" pixel)
* | and is followed downstream until either the edge of the DEM is
* | reached or a "pit". The amount of flow (expressed as a
* | fraction of the area of the source pixel) entering each pixel
* | along the way is added to the tca of that pixel. The area
* | spanned by the flow inside each pixel along the way is added to
* | the tda of the source pixel. After flow has been generated at
* | every pixel (each pixel has been made a source pixel), and its
* | impact on each of the other pixels has been added, the final
* | tca values will have been computed.
*

```

```

* | INPUT: nr, nc, istop(,)
* | OUTPUT: tca(,), tda(,)
*

```

```

* |
* | START
* | |
* |

```



```

10  continue
    end

```

```

*-----*
SUBROUTINE SOURCE_PIXEL
*-----*

```

```

*
* This subroutine generates flow over the source pixel and
* determines the receiving pixel or pixels.  If the flow angle in
* the source pixel is 0, 90, 180 or 270 degrees, then there is a
* single receiving pixel.  Otherwise, there are two receiving
* pixels, which are cardinal neighbors of the source pixel.  If
* there are two receiving pixels (the most frequent case), flow
* is split among them.
*

```

```

* INPUT:

```

```

*   in, jn, ic, nr, nc, dx, dy, tca(,), angle(,)

```

```

* OUTPUT:

```

```

*   tca(,)

```

```

*   Regarding a receiving pixel indicated by subscript 'n', for
*   'next' because it will be the next one to step to:

```

```

*   in, jn, idirn, fn, fln, f2n, x1n, x2n, x3n, i3n;

```

```

*   If there are two receiving pixels, the output variables
*   regarding the receiving pixel indicated by 'w', for 'waiting'
*   because it is put in waiting list (in position no. ic) are:

```

```

*   ic, iw(ic), jw(ic), idirnw(ic), fw(ic), flw(ic), f2w(ic),
*   x1w(ic), x2w(ic), x3w(ic), i3w(ic).
*

```

```

include 'demon_downslope.var'

```

```

io = in

```

```

jo = jn

```

```

* 1. GENERATE FLOW OVER THE SOURCE PIXEL :
*-----*

```

```

fo = dx * dy

```

```

tca(io, jo) = tca(io, jo) + fo

```

```

* 2. "GOTO" STATEMENTS DEPENDING ON FLOW ANGLE :
*-----*

```

```

a = angle(io, jo)

```

```

if(a.eq.0.) goto 20

```

```

if(a.eq.90.) goto 22

```

```

if(a.eq.180.) goto 24

```

```

if(a.eq.270.) goto 26

```

```

goto 30

```

* 3. ANGLES THAT ARE 90-DEGREE MULTIPLES (1 RECEIVING ELEMENT ONLY):

* 3.1 ANGLE = 0 DEGREES:

```
20   jn = jo + 1
      fn = fo
      f1n = dx
      f2n = dx
      idirn = 4
      x1n = 0.
      x2n = dy
      x3n = 0.
      i3n = 1
      RETURN
```

* 3.2 ANGLE = 90 DEGREES:

```
22   in = io - 1
      fn = fo
      f1n = dy
      f2n = dy
      idirn = 3
      x1n = 0.
      x2n = dx
      x3n = 0.
      i3n = 1
      RETURN
```

* 3.3 ANGLE = 180 DEGREES:

```
24   jn = jo - 1
      fn = fo
      f1n = dx
      f2n = dx
      idirn = 2
      x1n = 0.
      x2n = dy
      x3n = 0.
      i3n = 1
      RETURN
```

* 3.4 ANGLE = 270 DEGREES:

```
26   in = io + 1
      fn = fo
      f1n = dy
      f2n = dy
      idirn = 1
      x1n = 0.
      x2n = dx
      x3n = 0.
      i3n = 1
      RETURN
```

* 4. ANGLES THAT ARE *NOT* 90-DEGREE MULTIPLES (2 RECEIVING ELEMENTS):

```

30   ic = ic + 1
      x1n = 0.
      x2n = dy
      x1w(ic) = 0.
      x2w(ic) = dx
      tanda = tand(a)
      if(a.lt.90.) goto 32
      if(a.lt.180.) goto 34
      if(a.lt.270.) goto 36
      goto 38

```

• 4.1 ANGLE IN 1st QUADRANT:

```

32   jn = jo + 1
      idirn = 4
      iw(ic) = io - 1
      jw(ic) = jo
      idirw(ic) = 3
      fln = 0.
      f2w(ic) = 0.

      if(a.le.anglecut) then
        f = 1. - 0.5 * dx/dy * tanda
        f2n = dx / cosd(a)
        x3n = dx * tanda
        i3n = 2
        x3w(ic) = 0.
        i3w(ic) = 1

      else
        f = 0.5 * dy/dx / tanda
        f2n = dy / sind(a)
        x3n = 0.
        i3n = 1
        x3w(ic) = dx - dy / tanda
        i3w(ic) = 1
      endif

      fn = f * fo
      fw(ic) = fo - fn
      flw(ic) = f2n
      RETURN

```

• 4.2 ANGLE IN 2nd QUADRANT:

```

34   jn = jo - 1
      idirn = 2
      iw(ic) = io - 1
      jw(ic) = jo
      idirw(ic) = 3
      f2n = 0.

```

```

flw(ic) = 0.

if(a.le.180.-anglecut) then
f = - 0.5 * dy/dx / tanda
fln = dy / sind(a)
x3n = 0.
i3n = 1
x3w(ic) = - dy / tanda
i3w(ic) = 2

else
f = 1. + 0.5 * dx/dy * tanda
fln = - dx / cosd(a)
x3n = dy + dx * tanda
i3n = 1
x3w(ic) = 0.
i3w(ic) = 1
endif

fn = f * fo
fw(ic) = fo - fn
f2w(ic) = fln
RETURN

```

• 4.3 ANGLE IN 3rd QUADRANT:

```

36  jn = jo - 1
     idirn = 2
     iw(ic) = io + 1
     jw(ic) = jo
     idirw(ic) = 1
     fln = 0.
     f2w(ic) = 0.

if(a.le.180.+anglecut) then
f = 1. - 0.5 * dx/dy * tanda
f2n = - dx / cosd(a)
x3n = dx * tanda
i3n = 2
x3w(ic) = 0.
i3w(ic) = 1

else
f = 0.5 * dy/dx / tanda
f2n = - dy / sind(a)
x3n = 0.
i3n = 1
x3w(ic) = dx - dy / tanda
i3w(ic) = 1
endif

fn = f * fo
fw(ic) = fo - fn
flw(ic) = f2n
RETURN

```

• 4.4 ANGLE IN 4th QUADRANT:

```

38   jn = jo + 1
      idirn = 4
      iw(ic) = io + 1
      jw(ic) = jo
      idirw(ic) = 1
      f2n = 0.
      flw(ic) = 0.

      if(a.le.360.-anglecut) then
        f = - 0.5 * dy/dx / tanda
        fln = - dy / sind(a)
        x3n = 0.
        i3n = 1
        x3w(ic) = - dy / tanda
        i3w(ic) = 2

      else
        f = 1. + 0.5 * dx/dy * tanda
        fln = dx / cosd(a)
        x3n = dy + dx * tanda
        i3n = 1
        x3w(ic) = 0.
        i3w(ic) = 1
      endif

      fn = f * fo
      fw(ic) = fo - fn
      f2w(ic) = fln
      RETURN

end

```

* SUBROUTINE FLOW_2D

```

*
* This subroutine projects a 2D flow "tube" across a pixel. It
* takes the variables describing the flow entering the pixel and
* determines which pixel or pixels ('receiving pixels') the flow
* will move to; and computes the variables describing the flow
* entering each receiving pixel.
*
* INPUT: is, js, in, jn, fn, idirn, x1n, x2n, x3n, i3n, fln,
*        f2n, angle(,), tca(,), tda(,)
* OUTPUT:
*         tca(,), tda(,)
* Regarding a receiving pixel indicated by subscript 'n', for
* 'next' because it will be the next one to step to:
* in, jn, fn, idirn, x1n, x2n, x3n, i3n, fln, f2n,
* If there are two receiving pixels, the output variables

```

```

*|      regarding the receiving pixel indicated by 'w', for 'waiting'
*|      because it is put in waiting list in position number 'ic' are:
*|      ic, iw(ic), jw(ic), idirnw(ic), fw(ic), xlw(ic), x2w(ic),
*|      x3w(ic), i3w(ic), flw(ic), f2w(ic).
*|

```

```

include 'demon_downslope.var'

```

```

io = in
jo = jn
fo = fn
idiro = idirn
x1o = x1n
x2o = x2n
x3o = x3n
i3o = i3n
f1o = f1n
f2o = f2n

```

```

tca(io,jo) = tca(io,jo) + fo
a = angle(io,jo)

```

```

goto(10, 20, 30, 40) idiro

```

```

* 1. IF FLOW ENTERED THROUGH THE NORTHERN FACE (idiro = 1):

```

```

-----

```

```

10 continue

```

```

** 1.1 IF CONVERGENCE IS FOUND:

```

```

if(a.le.180.) then
tca(io,jo) = tca(io,jo) - fo
call CONVERGENCE
RETURN
endif

```

```

* 1.2 IF DIRECTION IS PARALLEL THROUGH:

```

```

if(a.eq.270.)then
in = io + 1
area = dy*(x2o-x1o)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

```

** 1.3 IF FLOW IS AT AN ANGLE:

```

```

tanda = tand(a)
p1 = x1o - dy / tanda
p2 = x2o - dy / tanda
p3 = x3o - dy / tanda

```

```

** 1.3.1 IF SOME FLOW GOES WEST:

```

```

if(p1.lt.0.)then
in = io
jn = jo - 1
idirn = 2
x1n = x1o * tanda
f1n = f1o

```

* 1.3.1.1 IF ALL FLOW GOES WEST:

```

if(p2.le.0.)then
x2n = x2o * tanda
x3n = x3o * tanda
area = 0.5*(x2o-x1o)*(x1n+x2n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 1.3.1.2 IF FLOW GOES WEST AND SOUTH:

```

x2n = dy
ic = ic + 1
iw(ic) = io + 1
jw(ic) = jo
idirw(ic) = 1
x1w(ic) = 0.
x2w(ic) = p2

if(p3.le.0.) then
x3n = x3o * tanda
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = - p3 / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(-p3)
    g2 = 0.5*(f2n+f2o)*p2
  else
    f2n = f2o
    g1 = 0.5*(f1o+f2o)*(p3-p1) + f2o*(-p3)
    g2 = f2o*p2
  endif
else
x3n = x1n
i3n = 1
x3w(ic) = p3
i3w(ic) = i3o
  if(i3o.eq.1) then
    f2n = f1o
    g1 = f1o*(-p1)
    g2 = f1o*p3 + 0.5*(f1o+f2o)*(p2-p3)
  else
    f = p3 / (p3-p1)
    f2n = f*f1o + (1.-f)*f2o
    g1 = 0.5*(f1o+f2n)*(-p1)

```

```

        g2 = 0.5*(f2n+f2o)*p3 + f2o*(p2-p3)
    endif
endif

f1w(ic) = f2n
f2w(ic) = f2o
g = g1/(g1+g2)
fn = g * fo
fw(ic) = fo - fn

area = dy*(x2o-x1o) + 0.5*p1*(x2n-x1n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 1.3.2 IF SOME FLOW GOES SOUTH:

```

if(p1.lt.dx)then
in = io + 1
jn = jo
idirn = 1
x1n = p1
f1n = f1o

```

* 1.3.2.1 IF ALL FLOW GOES SOUTH:

```

if(p2.le.dx)then
x2n = p2
x3n = p3
area = dy*(x2o-x1o)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 1.3.2.2 IF FLOW GOES SOUTH AND EAST:

```

x2n = dx
ic = ic + 1
iw(ic) = io
jw(ic) = jo + 1
idirw(ic) = 4
x1w(ic) = 0.
x2w(ic) = (dx-p2) * tanda

if(p3.le.dx) then
x3n = p3
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
    if(i3o.eq.1) then
        f = (dx - p3) / (p2 - p3)
        f2n = f*f2o + (1.-f)*f1o
        g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(dx-p3)
        g2 = 0.5*(f2n+f2o)*(p2-dx)
    else

```

```

        f2n = f2o
        g1 = 0.5*(f1o+f2o)*(p3-p1) + f2o*(dx-p3)
        g2 = f2o*(p2-dx)
        endif
    else
        x3n = x1n
        i3n = 1
        x3w(ic) = (dx-p3) * tanda
        i3w(ic) = i3o
        if(i3o.eq.1) then
            f2n = f1o
            g1 = f1o*(dx-p1)
            g2 = f1o*(p3-dx) + 0.5*(f1o+f2o)*(p2-p3)
        else
            f = (p3-dx) / (p3-p1)
            f2n = f*f1o + (1.-f)*f2o
            g1 = 0.5*(f1o+f2n)*(dx-p1)
            g2 = 0.5*(f2n+f2o)*(p3-dx) + f2o*(p2-p3)
        endif
    endif

    endif

    f1w(ic) = f2n
    f2w(ic) = f2o
    g = g1/(g1+g2)
    fn = g * fo
    fw(ic) = fo - fn

    area = dy*(x2o-x1o) - 0.5*x2w(ic)*(p2-dx)
    tda(is, js) = tda(is, js) + area
    RETURN
    endif

```

* 1.3.3 IF ALL FLOW GOES EAST:

```

    jn = jo + 1
    idirn = 4
    x1n = (dx-p1) ** tanda
    x2n = (dx-p2) ** tanda
    x3n = (dx-p3) ** tanda
    area = (x2o-x1o)**(dy-0.5*(xin+x2n))
    tda(is, js) = tda(is, js) + area
    RETURN

```

* 2. IF FLOW ENTERED THROUGH THE EASTERN FACE (idiro = 2):
 *-----

20 continue

* 2.1 IF CONVERGENCE IS FOUND:

```

    if(a.le.90..or.a.ge.270.) then
        tca(io,jo) = tca(io,jo) - fo
        call CONVERGENCE
        RETURN
    endif

```

- 2.2 IF DIRECTION IS PARALLEL THROUGH:

```

if(a.eq.180.)then
jn = jo - 1
area = dx*(x2o-x1o)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

- 2.3 IF FLOW IS AT AN ANGLE:

```

tanda = tand(a)
p1 = x1o + dx * tanda
p2 = x2o + dx * tanda
p3 = x3o + dx * tanda

```

- 2.3.1 IF SOME FLOW GOES NORTH:

```

if(p1.lt.0.)then
in = io - 1
jn = jo
idirn = 3
x1n = - x1o / tanda
f1n = f1o

```

- 2.3.1.1 IF ALL FLOW GOES NORTH:

```

if(p2.le.0.)then
x2n = - x2o / tanda
x3n = - x3o / tanda
area = 0.5*(x2o-x1o)*(x1n+x2n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

- 2.3.1.2 IF FLOW GOES NORTH AND WEST:

```

x2n = dx
ic = ic + 1
iw(ic) = io
jw(ic) = jo - 1
idirw(ic) = 2
x1w(ic) = 0.
x2w(ic) = p2

if(p3.le.0.) then
x3n = - x3o / tanda
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = - p3 / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(-p3)
  
```

```

        g2 = 0.5*(f2n+f2o)*p2
        else
        f2n = f2o
        g1 = f2o*(-p3) + 0.5*(f1o+f2o)*(p3-p1)
        g2 = f2o*p2
        endif
    else
    x3n = x1n
    i3n = 1
    x3w(ic) = p3
    i3w(ic) = i3o
    if(i3o.eq.1) then
        f2n = f1o
        g1 = f1o*(-p1)
        g2 = f1o*p3 + 0.5*(f1o+f2o)*(p2-p3)
    else
        f = p3 / (p3-p1)
        f2n = f*f1o + (1.-f)*f2o
        g1 = 0.5*(f1o+f2n)*(-p1)
        g2 = 0.5*(f2n+f2o)*p3 + f2o*(p2-p3)
    endif
    endif

    f1w(ic) = f2n
    f2w(ic) = f2o
    g = g1 / (g1+g2)
    fn = g * fo
    fw(ic) = fo - fn

    area = dx*(x2o-x1o) + 0.5*p1*(x2n-x1n)
    tda(is, js) = tda(is, js) + area
    RETURN
    endif

```

* 2.3.2 IF SOME FLOW GOES WEST:

```

    if(p1.lt.dy)then
    in = io
    jn = jo - 1
    idirn = 2
    x1n = p1
    f1n = f1o

```

* 2.3.2.1 IF ALL FLOW GOES WEST:

```

    if(p2.le.dy)then
    x2n = p2
    x3n = p3
    area = dx*(x2o-x1o)
    tda(is, js) = tda(is, js) + area
    RETURN
    endif

```

* 2.3.2.2 IF FLOW GOES WEST AND SOUTH:

```

x2n = dy
ic = ic + 1
iw(ic) = io + 1
jw(ic) = jo
idirw(ic) = 1
x1w(ic) = 0.
x2w(ic) = (p2-dy) / tanda

if(p3.le.dy) then
x3n = p3
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = (dy-p3) / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(dy-p3)
    g2 = 0.5*(f2n+f2o)*(p2-dy)
  else
    f2n = f2o
    g1 = 0.5*(f1o+f2o)*(p3-p1) + f2o*(dy-p3)
    g2 = f2o*(p2-dy)
  endif
else
x3n = x1n
i3n = 1
x3w(ic) = (p3-dy) / tanda
i3w(ic) = i3o
  if(i3o.eq.1) then
    f2n = f1o
    g1 = f1o*(dy-p1)
    g2 = f1o*(p3-dy) + 0.5*(f1o+f2o)**(p2-p3)
  else
    f = (p3-dy) / (p3-p1)
    f2n = f*f1o + (1.-f)*f2o
    g1 = 0.5*(f1o+f2n)**(dy-p1)
    g2 = 0.5*(f2n+f2o)**(p3-dy) + f2o*(p2-p3)
  endif
endif

f1w(ic) = f2n
f2w(ic) = f2o
g = g1 / (g1+g2)
fn = g * fo
fw(ic) = fo - fn

area = dx*(x2o-x1o) - 0.5*(x2w(ic)**(p2-dy))
tda(is, js) = tda(is, js) + area
RETURN
endif

```

** 2.3.3 IF ALL FLOW GOES SOUTH:

```

in = io + 1
idirn = 1

```

```

x1n = (p1-dy) / tanda
x2n = (p2-dy) / tanda
x3n = (p3-dy) / tanda
area = (x2o-x1o)*(dx-0.5*(x1n+x2n))
tda(is, js) = tda(is, js) + area
RETURN

* 3. IF FLOW ENTERED THROUGH THE SOUTHERN FACE (idiro = 3):
*-----

30   continue

** 3.1 IF CONVERGENCE IS FOUND:

      if(a.ge.180..or.a.eq.0.) then
      tca(io,jo) = tca(io,jo) - fo
      call CONVERGENCE
      RETURN
      endif

* 3.2 IF DIRECTION IS PARALLEL THROUGH:

      if(a.eq.90.)then
      in = io - 1
      area = dy*(x2o-x1o)
      tda(is, js) = tda(is, js) + area
      RETURN
      endif

** 3.3 IF FLOW IS AT AN ANGLE:

      tanda = tand(a)
      p1 = x1o - dy / tanda
      p2 = x2o - dy / tanda
      p3 = x3o - dy / tanda

** 3.3.1 IF SOME FLOW GOES EAST:

      if(p1.lt.0.)then
      in = io
      jn = jo + 1
      idirn = 4
      x1n = x1o **tanda
      f1n = f1o

** 3.3.1.1 IF ALL FLOW GOES EAST:

      if(p2.le.0.)then
      x2n = x2o **tanda
      x3n = x3o **tanda
      area = 0.5*(x2o-x1o)*(x1n+x2n)
      tda(is, js) = tda(is, js) + area
      RETURN
      endif

```

* 3.3.1.2 IF FLOW GOES EAST AND NORTH:

```

x2n = dy
ic = ic + 1
iw(ic) = ic - 1
jw(ic) = jo
idirw(ic) = 3
x1w(ic) = 0.
x2w(ic) = p2

if(p3.le.0.) then
x3n = x3o * tanda
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = - p3 / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(-p3)
    g2 = 0.5*(f2n+f2o)*p2
  else
    f2n = f2o
    g1 = f2o*(-p3) + 0.5*(f2o+f1o)*(p3-p1)
    g2 = f2o ** p2
  endif
else
x3n = x1n
i3n = 1
x3w(ic) = p3
i3w(ic) = i3o
  if(i3o.eq.1) then
    f2n = f1o
    g1 = f1o*(-p1)
    g2 = f1o*p3 + 0.5*(f1o+f2o)*(p2-p3)
  else
    f = p3 / (p3-p1)
    f2n = f*f1o + (1.-f)*f2o
    g1 = 0.5*(f1o+f2n)*(-p1)
    g2 = 0.5*(f2n+f2o)*p3 + f2o*(p2-p3)
  endif
endif

f1w(ic) = f2n
f2w(ic) = f2o
g = g1 / (g1+g2)
fn = g * fo
fw(ic) = fo - fn

area = dy*(x2o-x1o) + 0.5*p1*(x2n-x1n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

** 3.3.2 IF SOME FLOW GOES NORTH:

```

if(p1.lt.dx)then
in = io - 1
jn = jo
idirn = 3
x1n = p1
f1n = f1o

```

* 3.3.2.1 IF ALL FLOW GOES NORTH:

```

if(p2.le.dx)then
x2n = p2
x3n = p3
area = dy*(x2o-x1o)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

• 3.3.2.2 IF FLOW GOES NORTH AND WEST:

```

x2n = dx
ic = ic + 1
iw(ic) = io
jw(ic) = jo - 1
idirw(ic) = 2
x1w(ic) = 0.
x2w(ic) = (dx-p2) * tanda

if(p3.le.dx) then
x3n = p3
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = (dx-p3) / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(dx-p3)
    g2 = 0.5*(f2n+f2o)*(p2-dx)
  else
    f2n = f2o
    g1 = f2o*(dx-p3) + 0.5*(f2o+f1o)*(p3-p1)
    g2 = f2o*(p2-dx)
  endif
else
x3n = x1n
i3n = 1
x3w(ic) = (dx-p3) * tanda
i3w(ic) = i3o
  if(i3o.eq.1) then
    f2n = f1o
    g1 = f1o*(dx-p1)
    g2 = f1o*(p3-dx) + 0.5*(f1o+f2o)*(p2-p3)
  else
    f = (p3-dx) / (p3-p1)
    f2n = f*f1o + (1.-f)*f2o
    g1 = 0.5*(f1o+f2n)*(dx-p1)

```

```

        g2 = 0.5*(f2n+f2o)*(p3-dx) + f2o*(p2-p3)
    endif
endif

f1w(ic) = f2n
f2w(ic) = f2o
g = g1 / (g1+g2)
fn = g * fo
fw(ic) = fo - fn

area = dy*(x2o-x1o) - 0.5*x2w(ic)**(p2-dx)
tda(is, js) = tda(is, js) + area
RETURN
endif

** 3.3.3 IF ALL FLOW GOES WEST:

jn = jo - 1
idirn = 2
x1n = (dx-p1) **tanda
x2n = (dx-p2) **tanda
x3n = (dx-p3) **tanda
area = (x2o-x1o)**(dy-0.5*(x1n+x2n))
tda(is, js) = tda(is, js) + area
RETURN

** 4. IF FLOW ENTERED THROUGH THE WESTERN FACE (idiro = 4):
*-----

40    continue

* 4.1 IF CONVERGENCE IS FOUND:

if(a.ge.90..and.a.le.270.) then
tca(io,jo) = tca(io,jo) - fo
call CONVERGENCE
RETURN
endif

** 4.2 IF DIRECTION IS PARALLEL THROUGH:

if(a.eq.0.)then
jn = jo + 1
area = dx*(x2o-x1o)
tda(is, js) = tda(is, js) + area
RETURN
endif

** 4.3 IF FLOW IS AT AN ANGLE:

tanda = tand(a)
p1 = x1o + dx **tanda
p2 = x2o + dx * tanda
p3 = x3o + dx * tanda

```

* 4.3.1 IF SOME FLOW GOES SOUTH:

```

if(p1.lt.0.)then
in = io + 1
jn = jo
idirn = 1
x1n = - x1o / tanda
f1n = f1o

```

* 4.3.1.1 IF ALL FLOW GOES SOUTH:

```

if(p2.le.0.)then
x2n = - x2o / tanda
x3n = - x3o / tanda
area = 0.5*(x2o-x1o)*(x1n+x2n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 4.3.1.2 IF FLOW GOES SOUTH AND EAST:

```

x2n = dx
ic = ic + 1
iw(ic) = io
jw(ic) = jo + 1
idirw(ic) = 4
x1w(ic) = 0.
x2w(ic) = p2

if(p3.le.0.) then
x3n = - x3o / tanda
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = - p3 / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(-p3)
    g2 = 0.5*(f2n+f2o)*p2
  else
    f2n = f2o
    g1 = 0.5*(f1o+f2o)*(p3-p1) + f2o*(-p3)
    g2 = f2o*p2
  endif
else
x3n = x1n
i3n = 1
x3w(ic) = p3
i3w(ic) = i3o
  if(i3o.eq.1) then
    f2n = f1o
    g1 = f1o ** (-p1)
    g2 = f1o*p3 + 0.5*(f1o+f2o)**(p2-p3)
  else
    f = - p1 / (p3-p1)

```

```

        f2n = f*f2o + (1.-f)*f1o
        g1 = 0.5*(f1o+f2n)*(-p1)
        g2 = 0.5*(f2n+f2o)*p3 + f2o*(p2-p3)
    endif
endif

f1w(ic) = f2n
f2w(ic) = f2o
g = g1 / (g1+g2)
fn = g * fo
fw(ic) = fo - fn

area = dx*(x2o-x1o) + 0.5*p1*(x2n-x1n)
tda(is, js) = tda(is, js) + area
RETURN
endif

* 4.3.2 IF SOME FLOW GOES EAST:

    if(p1.lt.dy)then
        in = io
        jn = jo + 1
        idirn = 4
        x1n = p1
        f1n = f1o

* 4.3.2.1 IF ALL FLOW GOES EAST:

        if(p2.le.dy)then
            x2n = p2
            x3n = p3
            area = dx*(x2o-x1o)
            tda(is, js) = tda(is, js) + area
            RETURN
        endif

* 4.3.2.2 IF FLOW GOES EAST AND NORTH:

            x2n = dy
            ic = ic + 1
            iw(ic) = io - 1
            jw(ic) = jo
            idirw(ic) = 3
            x1w(ic) = 0.
            x2w(ic) = (p2-dy) / tanda

            if(p3.le.dy) then
                x3n = p3
                i3n = i3o
                x3w(ic) = 0.
                i3w(ic) = 1
                if(i3o.eq.1) then
                    f = (dy-p3) / (p2-p3)
                    f2n = f*f2o + (1.-f)*f1o
                    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(dy-p3)
                endif
            endif
        endif
    endif

```

```

        g2 = 0.5*(f2n+f2o)*(p2-dy)
    else
        f2n = f2o
        g1 = f2o*(dy-p3) + 0.5*(f2o+f1o)*(p3-p1)
        g2 = f2o*(p2-dy)
    endif

    else
        x3n = x1n
        i3n = 1
        x3w(ic) = (p3-dy) / tanda
        i3w(ic) = i3o
        if(i3o.eq.1) then
            f2n = f1o
            g1 = f1o*(dy-p1)
            g2 = f1o*(p3-dy) + 0.5*(f2o+f1o)*(p2-p3)
        else
            f = (p3-dy) / (p3-p1)
            f2n = f*f1o + (1.-f)*f2o
            g1 = 0.5*(f1o+f2n)*(dy-p1)
            g2 = f2o*(p2-p3) + 0.5*(f2o+f2n)*(p3-dy)
        endif
    endif

    endif

    f1w(ic) = f2n
    f2w(ic) = f2o
    g = g1 / (g1+g2)
    fn = g * fo
    fw(ic) = fo - fn

    area = dx*(x2o-x1o) - 0.5*x2w(ic)*(p2-dy)
    tda(is, js) = tda(is, js) + area
    RETURN
endif

```

• 4.3.3 IF ALL FLOW GOES NORTH:

```

    in = io - 1
    idirn = 3
    x1n = (p1-dy) / tanda
    x2n = (p2-dy) / tanda
    x3n = (p3-dy) / tanda
    area = (x2o-x1o)*(dx-0.5*(x1n+x2n))
    tda(is, js) = tda(is, js) + area
    RETURN

end

```

• SUBROUTINE FLOW_1D

• This subroutine projects a line of concentrated flow through a pixel. It takes the variables describing the flow entering the

```

* | pixel and determines which pixel ('receiving pixel') the flow
* | will move to; and computes the variables describing the flow
* | entering the receiving pixel.
* |
* | INPUT:
* |   in, jn, fn, idirn, xln, angle(,), tca(,)
* | OUTPUT:
* |   tca(,), in, jn, fn, idirn, xln
* |

```

```

include 'demon_downslope.var'

io = in
jo = jn
fo = fn
idiro = idirn
xlo = xln

tca(io,jo) = tca(io,jo) + fo
a = angle(io,jo)

goto(10, 20, 30, 40) idiro

* 1. IF FLOW ENTERED THROUGH THE NORTHERN FACE (idiro = 1):
*-----

10 continue

* 1.0 IF WE'RE AT A CORNER:

if(xlo.eq.0.and.idirn.eq.1) then
call CORNERS
RETURN
endif

* 1.1 IF CONVERGENCE IS FOUND:

if(a.le.180.) then
tca(io,jo) = tca(io,jo) - fo
call CONVERGENCE
RETURN
endif

* 1.2 IF DIRECTION IS PARALLEL THROUGH:

if(a.eq.270.)then
in = io + 1
jn = jo
xln = xlo
idirn = idiro
RETURN
endif

* 1.3 IF FLOW IS AT AN ANGLE:

```

```

        p1 = x1o - dy * tand(270.-a)
* 1.3.1 IF FLOW GOES WEST:
        if(p1.lt.0.)then
            in = io
            jn = jo - 1
            idirn = 2
            xln = x1o * tand(a)
            RETURN
        endif
* 1.3.2 IF FLOW GOES SOUTH:
        if(p1.lt.dx)then
            in = io + 1
            jn = jo
            idirn = 1
            xln = p1
            RETURN
        endif
* 1.3.3 IF FLOW GOES EAST:
        in = io
        jn = jo + 1
        idirn = 4
        xln = dy + (dx-x1o) * tand(a)
        RETURN
* 2. IF FLOW ENTERED THROUGH THE EASTERN FACE (idiro = 2):
*-----
20      continue
* 2.1 IF CONVERGENCE IS FOUND:
        if(a.le.90..or.a.ge.270.) then
            tca(io,jo) = tca(io,jo) - fo
            call CONVERGENCE
            RETURN
        endif
* 2.2 IF DIRECTION IS PARALLEL THROUGH:
        if(a.eq.180.)then
            in = io
            jn = jo - 1
            xln = x1o
            idirn = idiro
            RETURN
        endif
* 2.3 IF FLOW IS AT AN ANGLE:
        p1 = x1o + dx * tand(a)

```

```

* 2.3.1 IF FLOW GOES NORTH:

      if(p1.lt.0.)then
        in = io - 1
        jn = jo
        idirn = 3
        xln = xlo * tand(a-90.)
        RETURN
      endif

* 2.3.2 IF FLOW GOES WEST:

      if(p1.lt.dy)then
        in = io
        jn = jo - 1
        idirn = 2
        xln = pl
        RETURN
      endif

* 2.3.3 IF FLOW GOES SOUTH:

      in = io + 1
      jn = jo
      idirn = 1
      xln = dx - (dy-xlo) * tand(270.-a)
      RETURN

* 3. IF FLOW ENTERED THROUGH THE SOUTHERN FACE (idiro = 3):
*-----

30   continue

* 3.1 IF CONVERGENCE IS FOUND:

      if(a.ge.180..or.a.eq.0.) then
        tca(io,jo) = tca(io,jo) - fo
        call CONVERGENCE
        RETURN
      endif

* 3.2 IF DIRECTION IS PARALLEL THROUGH:

      if(a.eq.90.)then
        in = io - 1
        jn = jo
        xln = xlo
        idirn = idiro
        RETURN
      endif

* 3.3 IF FLOW IS AT AN ANGLE:

      pl = xlo - dy * tand(90.-a)

```

- 3.3.1 IF FLOW GOES EAST:

```

    if(p1.lt.0.)then
      in = io
      jn = jo + 1
      idirn = 4
      xln = xio * tand(a)
      RETURN
    endif

```

- * 3.3.2 IF FLOW GOES NORTH:

```

    if(p1.lt.dx)then
      in = io - 1
      jn = jo
      idirn = 3
      xln = p1
      RETURN
    endif

```

- * 3.3.3 IF FLOW GOES WEST:

```

    in = io
    jn = jo - 1
    idirn = 2
    xln = dy + (dx-xio) * tand(a)
    RETURN

```

- * 4. IF FLOW ENTERED THROUGH THE WESTERN FACE (idiro = 4):

```
40    continue
```

- * 4.1 IF CONVERGENCE IS FOUND:

```

    if(a.ge.90..and.a.le.270.) then
      tca(io,jo) = tca(io,jo) - fo
      call CONVERGENCE
      RETURN
    endif

```

- 4.2 IF DIRECTION IS PARALLEL THROUGH:

```

    if(a.eq.0.)then
      in = io
      jn = jo + 1
      xln = xio
      idirn = idiro
      RETURN
    endif

```

- * 4.3 IF FLOW IS AT AN ANGLE:

```

    pi = xio + dx * tand(a)

```

• 4.3.1 IF FLOW GOES SOUTH:

```

if(pl.lt.0.)then
in = io + 1
jn = jo
idirn = 1
xln = xlo * tand(a-270.)
RETURN
endif

```

* 4.3.2 IF FLOW GOES EAST:

```

if(pl.lt.dy)then
in = io
jn = jo + 1
idirn = 4
xln = pl
RETURN
endif

```

• 4.3.3 IF FLOW GOES NORTH:

```

in = io - 1
jn = jo
idirn = 3
xln = dx - (dy-xlo) * tand(90.-a)
RETURN
end

```

*

SUBROUTINE CONVERGENCE

*

*

*

*

*

*

*

*

*

*

This subroutine determines the next pixel to step to in the case of flow angles in two neighbor pixels converging to the shared boundary.

INPUT: io, jo, fo, idiro

OUTPUT: in, jn, fn, idirn, xln, x2n

include 'demon_downslope.var'

xln = 0.

x2n = 0.

idirn = 1

fn = fo

goto(10, 20, 30, 40) idiro

* 1. IF CONVERGENCE IS AT THE NORTHERN BOUNDARY (idiro = 1):

*

```

10  z1 = z(io, jo)
    z2 = z(io, jo+1)
    if(z1.le.z2) then
      in = io
      jn = jo
    else
      in = io
      jn = jo + 1
    endif
    RETURN

```

```

* 2. IF CONVERGENCE IS AT THE EASTERN BOUNDARY (idiro = 2):
*-----

```

```

20  z2 = z(io, jo+1)
    z4 = z(io+1, jo+1)
    if(z2.le.z4) then
      in = io
      jn = jo + 1
    else
      in = io + 1
      jn = jo + 1
    endif
    RETURN

```

```

* 3. IF CONVERGENCE IS AT THE SOUTHERN BOUNDARY (idiro = 3):
*-----

```

```

30  z3 = z(io+1, jo)
    z4 = z(io+1, jo+1)
    if(z3.le.z4) then
      in = io + 1
      jn = jo
    else
      in = io + 1
      jn = jo + 1
    endif
    RETURN

```

```

* 4. IF CONVERGENCE IS AT THE WESTERN BOUNDARY (idiro = 4):
*-----

```

```

40  z1 = z(io, jo)
    z3 = z(io+1, jo)
    if(z1.le.z3) then
      in = io
      jn = jo
    else
      in = io + 1
      jn = jo
    endif
    RETURN

```

```

end

```

```

*-----*
* SUBROUTINE CORNERS
*-----*
*
* This subroutine routes flow from a pixel corner to another
* pixel corner, in the direction of maximal slope.
* INPUT: io, jo, z(,), dx, dy
* OUTPUT: in, jn
*
include 'demon_downslope.var'

z1 = z(io, jo)
zN = z(io-1, jo)
zE = z(io, jo+1)
zS = z(io+1, jo)
zW = z(io, jo-1)
zNE= z(io-1, jo+1)
zSE= z(io+1, jo+1)
zSW= z(io+1, jo-1)
zNW= z(io-1, jo-1)

sN = (z1 - zN) / dy
sE = (z1 - zE) / dx
sS = (z1 - zS) / dy
sW = (z1 - zW) / dx
diag = sqrt(dx**2 + dy**2)
sNE = (z1 - zNE) / diag
sSE = (z1 - zSE) / diag
sSW = (z1 - zSW) / diag
sNW = (z1 - zNW) / diag

smax = 0.
if(sN.gt.smax) then
smax = sN
in = io - 1
jn = jo
endif
if(sE.gt.smax) then
smax = sE
in = io
jn = jo + 1
endif
if(sS.gt.smax) then
smax = sS
in = io + 1
jn = jo
endif
if(sW.gt.smax) then
smax = sW
in = io
jn = jo - 1
endif

```

```

if(sNE.gt.smax) then
smax = sNE
in = io - 1
jn = jo + 1
endif
if(sSE.gt.smax) then
smax = sSE
in = io + 1
jn = jo + 1
endif
if(sSW.gt.smax) then
smax = sSW
in = io + 1
jn = jo - 1
endif
if(sNW.gt.smax) then
smax = sNW
in = io - 1
jn = jo - 1
endif

end

```

*

SUBROUTINE WAITLIST

*

*

<pre> This subroutine takes an element from the waiting list and makes it the 'next' element to step to. INPUT: ic2, ic, iw(ic2), jw(ic2), fw(ic2), idirw(ic2), x3w(ic2), i3w(ic2), f1w(ic2), f2w(ic2) OUTPUT: ic2, ic, in, jn, fn, idirn, x3n, i3n, f1n, f2n </pre>
--

*

```
include 'demon_downslope.var'
```

```
ic2 = ic2 + 1
```

```

in = iw(ic2)
jn = jw(ic2)
fn = fw(ic2)
idirn = idirw(ic2)
x1n = x1w(ic2)
x2n = x2w(ic2)
x3n = x3w(ic2)
i3n = i3w(ic2)
f1n = f1w(ic2)
f2n = f2w(ic2)

```

```

if(ic2.eq.ic) then
ic = 0
ic2 = 0
endif

```

```

end

*
-----
SUBROUTINE SCA_AND_SDA
*
-----
*
* This subroutine computes the flow width and obtains the
* specific contributing areas by dividing the total contributing
* areas by the flow width in the pixel.
* INPUT: angle(,), tca(,), tda(,), dx, dy
* OUTPUT: sca(,), sda(,)
*
-----

include 'demon_downslope.var'

do 10 i = 2, nr
do 10 j = 2, nc

a = angle(i,j)

if(a.lt.90.) alpha = a
if(a.ge.90..and.a.lt.180.) alpha = 180. - a
if(a.ge.180..and.a.lt.270.) alpha = 270. - a
if(a.ge.270.) alpha = 360. - a

beta = abs( alpha - anglecut )

diagonal = sqrt( dx**2 + dy**2 )
width = diagonal * cosd(beta)

sca(i,j) = tca(i,j) / width
sda(i,j) = tda(i,j) / width

10 continue

end

*
-----
SUBROUTINE WRITE
*
-----
*
* This subroutine writes the specific contributing areas in file
* SCAfile, which name is to be specified by the user in the file
* 'demon_downslope.var'
*
-----

include 'demon_downslope.var'

open(2, file=SCAfile, status='new')
do 10 i = 2, nr
20 write(2, *) ( sca(i,j), j = 2, nc )

```

```
close(2)

open(3, file=SDAfile, status='new')
do 20 i = 2, nr
20 write(3, *) ( sda(i,j), j = 2, nc )
close(3)
end
```

ANNEXE III : PARAMÈTRES DE L'UNIVERS DE LA RIVIÈRE SAMSON

Projection	MTM (<i>Modified Transverse Mercator</i>)
Fuseau	#7
Datum	NAD 27
Rayon terrestre	Clark, 1866
Méridien central	-70,5°
Latitude (vraie origine)	0°
Facteur d'échelle	0,9999
Coordonnée X de la fausse origine	304 800
Coordonnée Y de la fausse origine	0

ANNEXE IV : LISTAGE DU PROGRAMME INITFICH

A. Programme qui crée les fichiers nécessaires pour les matrices du plan incliné et du biplan à double inclinaison

```

*
-----
PROGRAM CONSTRUCTION_FICHER_INITIAL
-----
*
*
-----
*
* Définition des paramètres
*
parameter ( nr = 15,
&          nc = 15,
&          nr1 = 16,
&          nc1 = 16,
&          ne = 225,
&          dx = 100,
&          dy = 100,
&          anglecut = 45,
&          pi = 3.14159265359 )

* Définition des variables
common      z(nr1, nc1), s(nr1, nc1),
&          angle(nr1, nc1), cn(nr1,nc1),
&          istop(nr1, nc1), iflat(nr1, nc1),
&          sinangle, cosangle, n

character*20 DEMfile, SLOfile, ASPfile, FICHTest, CNfile,
&          ARROWfile

DEMfile = '1matrice.txt'
SLOfile = '1slofile.txt'
ASPfile = '1aspfile.txt'
FICHTest = '1fichtes.txt'
CNfile = '1cnfile.txt'
ARROWfile = '1arrow.dat'

*
-----

open(1, file = DEMfile, status = 'old')

do 10 i = 2, nr1
    read(1,*) (z(i,j), j=2,nc1)
10 continue

close(1)

open(5, file = FICHTest, status = 'new')

do 20 i = 2, nr

```

```

do 20 j = 2, nc

n = n+1

z1 = z(i, j)
z2 = z(i, j+1)
z3 = z(i+1, j)
z4 = z(i+1, j+1)
sx = 0.5 * (- z1 + z2 - z3 + z4) / dx
sy = 0.5 * ( z1 + z2 - z3 - z4) / dy
iA = 0
iB = 0
absx = abs(sx)
absy = abs(sy)
if(sx.ne.0) iA = sx / absx
if(sy.ne.0) iB = sy / absy

write (5,*) n, z1, z2, z3, z4, sx, sy, iA, iB

s(i,j) = sqrt(sx*sx + sy*sy)

factor = - iA + iB - iA*iB + 3*iA*iA + 2*iB*iB
&          + iA*iB*iB - 3*iA*iA*iB*iB

dum = 90. * factor
if(iA*iB.ne.0) dum = dum + iA*iB*datan(absy/absx)*180/pi

if(abs(dum-360.) .lt. 1E-3) dum = 0.
angle(i, j) = dum
iflat(i, j) = 0

write (5,1) s(i,j), factor, dum, angle(i,j)
write (5,*) ' '

if(iA.eq.0.and.iB.eq.0) iflat(i, j) = 1

20 continue

close (5)

open(2, file = SLOfile, status = 'new')
open(3, file = ASPfile, status = 'new')
open(4, file = CNfile, status = 'new')

do 30 i = 2, nr1
do 30 j = 2, nc1
  cn(i,j) = 1
  write(2,*) s(i,j)
  write(3,*) angle(i,j)
  write(4,*) cn(i,j)
30 continue

close(2)
close(3)
close(4)

```

```
open(6, file = ARROWfile, status = 'new')

*   entête de fichier pour TECPLOT
write(6,*) 'TITLE = "Orientation des pentes maximales - Matrice #1"'
write(6,*) 'VARIABLES = "X","Y","UORI","VORI"'
write(6,*) 'ZONE T="ZONE1",Z=0.0,I=',nr,',J=',nc,',F=POINT'

do 40 i = 2, nr1
do 50 j = 2, nc1
    sinangle = sin(pi / 180 * angle(i,j))
    cosangle = cos(pi / 180 * angle(i,j))
    if (z(i,j).eq.0.) then
        sinagle = 0
        cosangle = 0
    endif
    write(6,2)(i-1)*100, (j-1)*100, cosangle, sinangle
50 continue

40 continue

close(6)

1   format (F12.6, F12.6, F12.6, F12.6)
2   format (I5, I5, F8.4, F8.4)

end
```

ANNEXE V : LISTAGE DU PROGRAMME PREDEMON

A. Programme PREDEMON qui prépare les fichiers créés dans SPANS pour le programme DEMONII

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define NomFichierDATADEM      "5datadem.dat"
#define NomFichierDEM          "5demfile"
#define NomFichierCN           "5cnfile"
#define NomFichierSLO         "5slofile"
#define NomFichierASP          "5aspfile"
#define NomFichierCOORDXY     "5latlon"

static double minarg1, minarg2;
#define MIN(a,b) (minarg1=(a), minarg2=(b), (minarg1) < (minarg2) ?\
                (minarg1) : (minarg2))

static double maxarg1, maxarg2;
#define MAX(a,b) (maxarg1=(a), maxarg2=(b), (maxarg1) > (maxarg2) ?\
                (maxarg1) : (maxarg2))

struct Point6D
{
    double Easting;
    double Northing;
    float Elevation;
    float Slope;
    float Aspect;
    float CN;
};

typedef struct Point6D Point6D;

/* ----- */

int main ()
{
    Point6D *PointGrilleDEMSpans;
    int NbPointsGrilleDEMSpans;
    float dx, dy, x, y, CalculColonnes, CalculLignes,
          lat, lon, u, v;
    int NumLigne, NumColonne, NumPoint, NombredePoints, NbLignes,
        NbColonnes, compteur, i, j, compt;
    float MatriceElevation[577][324], MatriceSlope[577][324],
        MatriceAspect[577][324], MatriceCN[577][324];
    double xmin, ymin, xmax, ymax, MatriceEasting[577][324],
        MatriceNorthing[577][324];

```

```

FILE *FichierDATAdem;
FILE *FichierDEM;
FILE *FichierCN;
FILE *FichierSLO;
FILE *FichierASP;
FILE *FichierCOORDXY;

/* Le fichier FichierCOORDXY contient la latitude et la longitude */
/* en coordonnees MTM */

printf ("Preprocesseur DEMONII\n");

/*-----*/
/* Definition des parametres devant etre specifiées par l'usager */
/*-----*/

NbPointsGrilleDEMSpans = 186948;
dx = 50;
dy = 50;

xmax = 0.0;
ymax = 0.0;

xmin = 1.e30;
ymin = 1.e30;

/*-----*/
/* Allocation du vecteur contenant les donnees obtenues de Spans */
/*-----*/

free(PointGrilleDEMSpans);
PointGrilleDEMSpans = (Point6D *) malloc ((NbPointsGrilleDEMSpans)
                                           * sizeof (Point6D));
if (PointGrilleDEMSpans == NULL)
{
    printf ("Out of memory!\n");
    exit(1);
}
printf ("memory allocated\n");

/*-----*/
/* Ouverture du fichier ASCII contenant les donnees d'elevation de SPANS */
/*-----*/

if ((FichierDATAdem = fopen (NomFichierDATAdem,"r"))== NULL)
{
    printf ("fopen FichierDATAdem failed.\n");
    exit(0);
}

compt = 0;

```

```

/* Boucle de lecture */

NumPoint=0;
printf ("Lecture du fichier %s ...\n",NomFichierDATAdem);

while (feof(FichierDATAdem) == 0)
{
    fscanf(FichierDATAdem, "%lf %lf %f %f %f %f %f",
           &PointGrilleDEMSpans[NumPoint].Easting,
           &PointGrilleDEMSpans[NumPoint].Northing,
           &lat, &lon,
           &PointGrilleDEMSpans[NumPoint].Elevation,
           &PointGrilleDEMSpans[NumPoint].Slope,
           &PointGrilleDEMSpans[NumPoint].Aspect,
           &PointGrilleDEMSpans[NumPoint].CN);

    NumPoint++;
/*   printf ("%5d\b\b\b\b\b",NumPoint); */
}

/* Fermeture du fichier contenant les donnees de SPANS */

fclose(FichierDATAdem);

/*-----*/
/* Boucle de lecture de la variable PointGrilleDEMSpans pour verification */
/* et modification des unites de paysage en coefficient de ruissellement */
/*-----*/

for (NumPoint=0; NumPoint<NbPointsGrilleDEMSpans; NumPoint++)
{
    if (PointGrilleDEMSpans[NumPoint].CN == 50.0)
        PointGrilleDEMSpans[NumPoint].CN = 0.705;
    else
    if (PointGrilleDEMSpans[NumPoint].CN == 51.0)
        PointGrilleDEMSpans[NumPoint].CN = 0.693;
    else
    if (PointGrilleDEMSpans[NumPoint].CN == 52.0)
        PointGrilleDEMSpans[NumPoint].CN = 0.695;
    else
    if (PointGrilleDEMSpans[NumPoint].CN == 53.0)
        PointGrilleDEMSpans[NumPoint].CN = 0.716;
    else
    if (PointGrilleDEMSpans[NumPoint].CN == 54.0)
        PointGrilleDEMSpans[NumPoint].CN = 0.743;
    else
    if (PointGrilleDEMSpans[NumPoint].CN == 55.0)
        PointGrilleDEMSpans[NumPoint].CN = 0.734;
    else
    if (PointGrilleDEMSpans[NumPoint].CN == 56.0)
        PointGrilleDEMSpans[NumPoint].CN = 0.721;
    else
    if (PointGrilleDEMSpans[NumPoint].CN == 57.0)

```

```
    PointGrilleDEMSpans [NumPoint].CN = 0.678;
else
if (PointGrilleDEMSpans [NumPoint].CN == 58.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.746;
else
if (PointGrilleDEMSpans [NumPoint].CN == 59.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.648;
else
if (PointGrilleDEMSpans [NumPoint].CN == 60.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.699;
else
if (PointGrilleDEMSpans [NumPoint].CN == 61.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.652;
else
if (PointGrilleDEMSpans [NumPoint].CN == 62.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.737;
else
if (PointGrilleDEMSpans [NumPoint].CN == 63.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.747;
else
if (PointGrilleDEMSpans [NumPoint].CN == 64.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.722;
else
if (PointGrilleDEMSpans [NumPoint].CN == 65.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.613;
else
if (PointGrilleDEMSpans [NumPoint].CN == 66.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.712;
else
if (PointGrilleDEMSpans [NumPoint].CN == 67.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.722;
else
if (PointGrilleDEMSpans [NumPoint].CN == 68.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.697;
else
if (PointGrilleDEMSpans [NumPoint].CN == 69.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.792;
else
if (PointGrilleDEMSpans [NumPoint].CN == 70.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.661;
else
if (PointGrilleDEMSpans [NumPoint].CN == 71.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.688;
else
if (PointGrilleDEMSpans [NumPoint].CN == 72.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.713;
else
if (PointGrilleDEMSpans [NumPoint].CN == 73.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.752;
else
if (PointGrilleDEMSpans [NumPoint].CN == 74.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.687;
else
if (PointGrilleDEMSpans [NumPoint].CN == 75.0)
    PointGrilleDEMSpans [NumPoint].CN = 0.679;
```

```

else
if (PointGrilleDEMSpans[NumPoint].CN == 76.0)
    PointGrilleDEMSpans[NumPoint].CN = 0.731;
else
if (PointGrilleDEMSpans[NumPoint].CN == 77.0)
    PointGrilleDEMSpans[NumPoint].CN = 0.766;
else
if (PointGrilleDEMSpans[NumPoint].CN == 78.0)
    PointGrilleDEMSpans[NumPoint].CN = 0.711;
else
if (PointGrilleDEMSpans[NumPoint].CN == 79.0)
    PointGrilleDEMSpans[NumPoint].CN = 0.722;
else
if (PointGrilleDEMSpans[NumPoint].CN == 80.0)
    PointGrilleDEMSpans[NumPoint].CN = 0.689;
else
if (PointGrilleDEMSpans[NumPoint].CN == 81.0)
    PointGrilleDEMSpans[NumPoint].CN = 0.665;
else
if (PointGrilleDEMSpans[NumPoint].CN == 82.0)
    PointGrilleDEMSpans[NumPoint].CN = 0.700;
else
    PointGrilleDEMSpans[NumPoint].CN = 0.0;

/* printf ("\nNumPoint = %d\n", NumPoint);
* printf ("%lf %lf %f %f\n",
*         PointGrilleDEMSpans[NumPoint].Easting,
*         PointGrilleDEMSpans[NumPoint].Northing,
*         PointGrilleDEMSpans[NumPoint].Elevation,
*         PointGrilleDEMSpans[NumPoint].CN); */

if (PointGrilleDEMSpans[NumPoint].Elevation > 0.0)
{
    compt++;
}

/*-----*/
/* Recherche des coordonnees maximales */
/*-----*/

for (NumPoint=0; NumPoint < NbPointsGrilleDEMSpans; NumPoint++)
{
    xmin = MIN (PointGrilleDEMSpans[NumPoint].Easting, xmin);
    ymin = MIN (PointGrilleDEMSpans[NumPoint].Northing, ymin);

    xmax = MAX (PointGrilleDEMSpans[NumPoint].Easting, xmax);
    ymax = MAX (PointGrilleDEMSpans[NumPoint].Northing, ymax);
}

printf ("xmax,ymax,ymin,xmin: %lf %lf %lf %lf\n", xmax, ymax, ymin, xmin);

```

```

/* Calcul de nombre de colonnes et de lignes */

CalculColonnes = (xmax - xmin)/dx;
CalculLignes = (ymax - ymin)/dy;

NbLignes = 577;
NbColonnes = 324;

printf ("Nombre de colonnes: %d %f\n",NbColonnes, CalculColonnes);
printf ("Nombre de lignes: %d %f\n", NbLignes, CalculLignes);

/*-----*/
/* Initialisation des elements de la matrice d'elevation */
/*-----*/

for (NumLigne = 0; NumLigne < NbLignes; NumLigne++)
{
  for (NumColonne = 0; NumColonne < NbColonnes; NumColonne++)
  {
    MatriceElevation[NumLigne][NumColonne] = 0.0;
    MatriceSlope[NumLigne][NumColonne] = 0.0;
    MatriceAspect[NumLigne][NumColonne] = 0.0;
    MatriceCN[NumLigne][NumColonne] = 0.0;
    MatriceEasting[NumLigne][NumColonne] = 0.0;
    MatriceNorthing[NumLigne][NumColonne] = 0.0;
  }
}
printf ("Initialisation de la matrice faite\n");

/*-----*/
/* Constitution des matrices */
/*-----*/

for (NumLigne = 0; NumLigne < NbLignes; NumLigne++)
{
  for (NumColonne = 0; NumColonne < NbColonnes; NumColonne++)
  {
    printf ("%3d\b\b\b", NumLigne);
    x = xmin + NumColonne ** dx;
    y = ymin + NumLigne ** dy;

    for (NumPoint = 0; NumPoint < NbPointsGrilleDEMSpans; NumPoint++)
    {
      if (fabs (x - PointGrilleDEMSpans[NumPoint].Easting) <= (dx/2)
          && fabs (y - PointGrilleDEMSpans[NumPoint].Northing) <= (dy/2))
      {
        MatriceElevation[NumLigne][NumColonne] =
          PointGrilleDEMSpans[NumPoint].Elevation;
        MatriceSlope[NumLigne][NumColonne] =
          PointGrilleDEMSpans[NumPoint].Slope;
        MatriceAspect[NumLigne][NumColonne] =
          PointGrilleDEMSpans[NumPoint].Aspect;
        MatriceCN[NumLigne][NumColonne] =

```

```

        PointGrilleDEMSpans[NumPoint].CN;
        MatriceEasting[NumLigne][NumColonne]=
            PointGrilleDEMSpans[NumPoint].Easting;
        MatriceNorthing[NumLigne][NumColonne]=
            PointGrilleDEMSpans[NumPoint].Northing;
    }
}
}
NombredePoints = NbColonnes*NbLignes;
printf ("Nombre de points = %d\n", NombredePoints);

/*-----*/
/* Boucle de lecture-ecriture pour la verification des matrices */
/*-----*/

/* for (NumLigne = 0; NumLigne < NbLignes; NumLigne++)
* {
*   for (NumColonne = 0; NumColonne < NbColonnes; NumColonne++)
*   {
*     printf ("%f\n", MatriceEasting[NumLigne][NumColonne]);
*     printf ("%f\n", MatriceNorthing[NumLigne][NumColonne]);
*     printf ("%f\n", MatriceSlope[NumLigne][NumColonne]);
*     printf ("%f\n", MatriceCN[NumLigne][NumColonne]);
*   }
* }*/

/*-----*/
/* Constitution du fichier d'elevation FichierDEM */
/*-----*/

/* Ouverture du fichier qui contient la matrice d'elevation */

if ((FichierDEM = fopen (NomFichierDEM,"w"))== NULL)
{
    printf ("fopen FichierDEM failed.\n");
    exit(0);
}
compteur = 0;

/* Ecriture des donnees de la matrice ELEVATION dans le fichier,
de format ASCII, DEMfile */

for (NumLigne = 0; NumLigne < NbLignes; NumLigne++)
{
    for (NumColonne = 0; NumColonne < NbColonnes; NumColonne++)
    {
        fprintf (FichierDEM, "%f\n", MatriceElevation[NumLigne][NumColonne]);

        if (MatriceElevation[NumLigne][NumColonne] != 0.0)
        {
            compteur++;
        }
    }
}

```

```

    }
  }
}

/* Fermeture du fichier DEMfile */

fclose (FichierDEM);
printf ("compteur = %d\n", compteur);
printf ("compt = %d\n", compt);

/*-----*/
/* Constitution du fichier de pente fichierSLO */
/*-----*/

/* Ouverture du fichier contenant les pentes: SLOfile */

if ((FichierSLO = fopen (NomFichierSLO,"w"))== NULL)
{
  printf("fopen FichierSLO failed. \n");
  exit(0);
}

/* Ecriture des donnees de la matrice SLOPE dans le fichier,
de format ASCII, SLOfile */

for (NumLigne = 0; NumLigne < NbLignes; NumLigne++)
{
  for (NumColonne = 0; NumColonne < NbColonnes; NumColonne++)
  {
    fprintf (FichierSLO, "%f\n", MatriceSlope[NumLigne][NumColonne]/100.);
  }
}

/* Fermeture du fichier SLOfile */

fclose (FichierSLO);

/*-----*/
/* Constitution du fichier d'orientation de la pente max FichierASO */
/*-----*/

/* Ouverture du fichier contenant l'orientation maximale de la pente de chaque
** cellule (aspect): ASPfile */

if ((FichierASP = fopen (NomFichierASP,"w"))== NULL)
{
  printf("fopen FichierASP failed. \n");
  exit(0);
}

```

```

/* Ecriture des donnees de la matrice ASPECT dans le fichier,
de format ASCII, ASPfile */

for (NumLigne = 0; NumLigne < NbLignes; NumLigne++)
{
for (NumColonne = 0; NumColonne < NbColonnes; NumColonne++)
{
fprintf (FichierASP, "%f\n", MatriceAspect[NumLigne][NumColonne]);
}
}

/* Fermeture du fichier ASPfile */

fclose (FichierASP);

/*-----*/
/* Constitution du fichier des coefficient de ruissellement FichierCN */
/*-----*/

/* Ouverture du fichier contenant les curve numbers: CNfile */

if ((FichierCN = fopen (NomFichierCN, "w+"))== NULL)
{
printf("fopen FichierCN failed. \n");
exit(0);
}

/* Ecriture des donnees de la matrice CN dans le fichier,
** de format ASCII, CNfile */

for (NumLigne = 0; NumLigne < NbLignes; NumLigne++)
{
for (NumColonne = 0; NumColonne < NbColonnes; NumColonne++)
{
fprintf (FichierCN, "%f\n", (MatriceCN[NumLigne][NumColonne]));
}
}

/* Fermeture du fichier CNfile */

fclose(FichierCN);

/*-----*/
/* Constitution du fichier des coordonnees X et Y Projection MTM */
/*-----*/

/* Ouverture du fichier contenant les coordonnees MTM : LATLON */

if ((FichierCOORDXY = fopen (NomFichierCOORDXY, "w+"))== NULL)
{
printf("fopen FichierCOORDXY failed. \n");
exit(0);
}

```

```
/* Ecriture des donnees de latitude & longitude en coord MTM dans le fichier
 * en format ASCII: LATLON */

for (NumLigne = 0; NumLigne < NbLignes; NumLigne++)
{
  for (NumColonne = 0; NumColonne < NbColonnes; NumColonne++)
  {
    fprintf (FichierCOORDXY, "%lf %lf\n",
             MatriceEasting(NumLigne) [NumColonne],
             MatriceNorthing(NumLigne) [NumColonne]);
  }
}

/* Fermeture du fichier LATLON */

fclose (FichierCOORDXY);

printf("fin \n");

return 0;
}
```

ANNEXE VI : LISTAGE DES DIFFÉRENTES DÉCLARATIONS DE VARIABLES DE DEMONII UTILISÉES POUR LES ESSAIS

A. Essai avec une discrétisation de 200m x 200m

* DEMON_DOWNSLOPE.VAR

```

parameter ( nr = 144,
&          nc = 81,
&          nr1 = 145,
&          nc1 = 82,
&          ne = 11664,
&          dx = 200.0,   dy = 200.0,
&          rain = 1.0,
&          anglecut = 45,
&          zmanning1 = 0.04,
&          zmanning2 = 0.3 )
real*8 qlat, qlon
common   chemin(500, 500, 5), qlat(nr1,nc1), qlon(nr1,nc1),
&        z(nr1, nc1), s(nr1, nc1),
&        cn(nr1, nc1), angle(nr1, nc1),
&        istop(nr1, nc1), iflat(nr1, nc1),
&        tca(nr1, nc1), sca(nr1, nc1),
&        tda(nr1, nc1), sda(nr1, nc1),
&        iw(ne), jw(ne), fw(ne), fww(ne), idirw(ne),
&        x1w(ne), x2w(ne), x3w(ne), i3w(ne), f1w(ne), f2w(ne),
&        inumchem(ne), inumcell(ne), comptecell(ne), jcell(ne),
&        is, js, i2stop,
&        in, jn, fn, fnw, idirn, x1n, x2n, x3n, i3n, f1n, f2n,
&        io, jo, fo, fow, idiro, x1o, x2o, x3o, i3o, f1o, f2o,
&        ic, ic2, numchem, numcell, n, f3, r,
&        tps_max, tps_cumul, max_chemin, zmanning

```

```

*   Description de la variable chemin:
*   chemin (numchem, numcell, 1) = indice_i
*   chemin (numchem, numcell, 2) = indice_j
*   chemin (numchem, numcell, 3) = segment
*   chemin (numchem, numcell, 4) = vitesse
*   chemin (numchem, numcell, 5) = tps_parcours

```

```

character*20 DEMfile, SCAfile, SDAfile, CNfile, SLOfile, ASPfile,
&            CHEMfile, TCAfile, SPANSfile, LAT_LON
DEMfile = '3demfile'
CNfile = '3cnfile'
SLOfile = '3slofile'
ASPfile = '3aspfile'
CHEMfile = '3chemfil'
SCAfile = '3scafile'
SDAfile = '3sdafile'
TCAfile = '3tcafile'
SPANSfile = '3spans'
LAT_LON = '3latlon'

```

B. Essai avec une discrétisation de 100m x 100m

* DEMON_DOWNSLOPE.VAR

```

parameter ( nr = 289,
&           nc = 162,
&           nr1 = 290,
&           ncl = 163,
&           ne = 46818,
&           dx = 100.0,   dy = 100.0,
&           rain = 1.0,
&           anglecut = 45,
&           zmanning1 = 0.04,
&           zmanning2 = 0.3 )

real*8 qlat,qlon

common      chemin(500, 500, 5), qlat(nr1,ncl), qlon(nr1,ncl),
&           z(nr1, ncl), s(nr1, ncl),
&           cn(nr1, ncl), angle(nr1, ncl),
&           istop(nr1, ncl), iflat(nr1, ncl),
&           tca(nr1, ncl), sca(nr1, ncl),
&           tda(nr1, ncl), sda(nr1, ncl),
&           iw(ne), jw(ne), fw(ne), fww(ne), idirw(ne),
&           x1w(ne), x2w(ne), x3w(ne), i3w(ne), f1w(ne), f2w(ne),
&           inumchem(ne), inumcell(ne), comptecell(ne),jcell(ne),
&           is, js, i2stop,
&           in, jn, fn, fnw, idirn, x1n, x2n, x3n, i3n, f1n, f2n,
&           io, jo, fo, fow, idiro, x1o, x2o, x3o, i3o, f1o, f2o,
&           ic, ic2, numchem, numcell, n, f3, r,
&           tps_max, tps_cumul, max_chemin, zmanning

```

* Description de la variable chemin:

- * chemin (numchem, numcell, 1) = indice_i
- * chemin (numchem, numcell, 2) = indice_j
- * chemin (numchem, numcell, 3) = segment
- * chemin (numchem, numcell, 4) = vitesse
- ** chemin (numchem, numcell, 5) = tps_parcours

```

character*20 DEMfile, SCAfile, SDAfile, CNfile, SLOfile, ASPfile,
&           CHEMfile, TCAfile, SPANSfile, LAT_LON

```

```

DEMfile = '4demfile'
CNfile = '4cnfile'
SLOfile = '4slofile'
ASPfile = '4aspfile'
CHEMfile = '4chemfi'
SCAfile = '4scafile'
SDAfile = '4sdafile'
TCAfile = '4tcafile'
SPANSfile = '4spans'
LAT_LON = '4latlon'

```

C. Essai avec une discrétisation de 50m x 50m

```
* DEMON_DOWNSLOPE.VAR
```

```
parameter ( nr = 577,
&          nc = 324,
&          nr1 = 578,
&          nc1 = 325,
&          ne = 186948,
&          dx = 50.0,  dy = 50.0,
&          rain = 1.0,
&          anglecut = 45,
&          zmanning1 = 0.04,
&          zmanning2 = 0.3 )
real*8 qlat,qlon
common chemin(500, 500, 5), qlat(nr1,nc1), qlon(nr1,nc1),
&       z(nr1, nc1), s(nr1, nc1),
&       cn(nr1, nc1), angle(nr1, nc1),
&       istop(nr1, nc1), iflat(nr1, nc1),
&       tca(nr1, nc1), sca(nr1, nc1),
&       tda(nr1, nc1), sda(nr1, nc1),
&       iw(ne), jw(ne), fw(ne), fww(ne), idirw(ne),
&       x1w(ne), x2w(ne), x3w(ne), i3w(ne), f1w(ne), f2w(ne),
&       inumchem(ne), inumcell(ne), comptecell(ne), jcell(ne),
&       is, js, i2stop,
&       in, jn, fn, fnw, idirn, x1n, x2n, x3n, i3n, f1n, f2n,
&       io, jo, fo, fow, idiro, x1o, x2o, x3o, i3o, flo, f2o,
&       ic, ic2, numchem, numcell, n, f3, r,
&       tps_max, tps_cumul, max_chemin, zmanning
```

• Description de la variable chemin:

```
* chemin (numchem, numcell, 1) = indice_i
* chemin (numchem, numcell, 2) = indice_j
* chemin (numchem, numcell, 3) = segment
* chemin (numchem, numcell, 4) = vitesse
* chemin (numchem, numcell, 5) = tps_parcours
```

```
character*20 DEMfile, SCAfile, SDAfile, CNfile, SLOfile, ASPfile,
&            CHEMfile, TCAfile, SPANSfile, LAT_LON
```

```
DEMfile = '5demfile'
CNfile = '5cnfile'
SLOfile = '5slofile'
ASPfile = '5aspfile'
CHEMfile = '5chemfil'
SCAfile = '5scafile'
SDAfile = '5sdafile'
TCAfile = '5tcafile'
SPANSfile = '5spans'
LAT_LON = '5latlon'
```

ANNEXE VII : LISTAGE DU PROGRAMME DEMONII

Note: La première et la deuxième version du programme sont sensiblement les mêmes. Il suffit de retrancher les passages soulignés à la première version pour obtenir la seconde version

```

*
* PROGRAM DEMON_DOWNSLOPE
*
*
*
* PROGRAM DEMON_DOWNSLOPE, VERSION 1, APRIL 1993.
*
* by Mariza C. Costa-Cabral and Stephen J. Burges
* Dpt. Civil Engineering, Univ. of Washington, Seattle, WA
*
* Reference: Costa-Cabral and Burges, Digital Elevation Model
* Networks (DEMON): a model of flow over hillslopes for
* computation of contributing and dispersal areas,
* Water Resources Research, 20(6), 1994.
*
* This program computes the specific contributing area (SCA) and
* the specific dispersal area (SDA) for each pixel of a
* rectangular DEM.
*
* This program cannot be compiled without the file
* 'demon_downslope.var', which is an integral part of this
* program, containing all parameters, the common variable
* statement, and the name of the DEM file to be read.
* The user should edit the file 'demon_downslope.var' and provide
* the name of the DEM file and the values of all parameters.
* Each time 'demon_downslope.var' is edited this program must be
* compiled again.
* Subroutine READ assumes that the DEM file is in ASCII format
* and that it contains nr rows each with nc values. For a
* different format, modify subroutine READ.
*
* Note: This program treats all points of elevation lesser than
* or equal to those of its neighbors as sinks. Flow entering a
* sink pixel stops. If some of such local minima or flat spots
* in your data are believed not to correspond to enclosed
* depressions, then you should modify your data prior to using
* this code.
*
* Units: If pixel sides (parameters dx and dy) are provided in
* meters (m), then the SCA and SDA are expressed in m**2 / m.
*
* Ce programme a été modifié par Stéphanie Thériault.
* DEUXIÈME VERSION DU PROGRAMME

```

```
call READ
```

```

call PITS
call ANGLES
call TCA_AND_TDA
call SCA_AND_SDA
call WRITE
end

```

```

SUBROUTINE READ

```

```

*
*
* This subroutine assumes that the DEM file is in ASCII format
* and that it contains nr rows each with nc values. For a
* different format, modify this subroutine.
*
* INPUT: DEMfile, CNfile, SLOfile, ASPfile, nr1, nc1
* OUTPUT: z(,), cn(,), s(,), asp(,)
*

```

```

include 'demiivar.f'

```

```

open(1, file = DEMfile, status = 'old')
open(2, file = CNfile, status = 'old')
open(3, file = SLOfile, status = 'old')
open(4, file = ASPfile, status = 'old')

```

```

do 10 i = 2, nr1

```

```

    read(1,*) (z(i,j), j=2,nc1)
    read(2,*) (cn(i,j), j=2,nc1)
    read(3,*) (s(i,j), j=2,nc1)
    read(4,*) (angle(i,j), j=2,nc1)

```

```

10 continue

```

```

close(1)
close(2)
close(3)
close(4)

```

```

end

```

```

SUBROUTINE PITS

```

```

*
*
* This subroutine sets the indicator variable istop(,) to 1 for
* pixels where flow paths end: the DEM boundary and pits.
* Pits are not filled in! They are treated as sinks.
* INPUT: nr1, nc1, z(,)
* OUTPUT: istop(,), npits
*

```

```

*|_____
      include 'demiivar.f'

      do 10 i = 1, nr1
        istop(i,1) = 1
        istop(i,2) = 1
        istop(i,nc) = 1
        istop(i,nc1) = 1
10     continue

      do 20 j = 1, nc1
        istop(1, j) = 1
        istop(2, j) = 1
        istop(nr, j) = 1
        istop(nr1, j) = 1
20     continue

      do 30 i = 3, nr
        do 30 j = 3, nc
          z0 = z(i, j)
          z1 = z(i-1, j-1)
          z2 = z(i-1, j)
          z3 = z(i-1, j+1)
          z4 = z(i, j-1)
          z5 = z(i, j+1)
          z6 = z(i+1, j-1)
          z7 = z(i+1, j)
          z8 = z(i+1, j+1)
          if (z0.le.z1.and.z0.le.z2.and.z0.le.z3.and.z0.le.z4.and.
&z0.le.z5.and.z0.le.z6.and.z0.le.z7.and.z0.le.z8) then
            npits = npits + 1
            istop(i, j) = 1
            *   print *, 'Point (' ,i,',',j,') is a sink'
            endif
30     continue

      do 40 i = 2, nr
        do 40 j = 2, nc
          if (angle(i,j).eq.0) then
            iflat(i,j) = 1
          else
            iflat(i,j) = 0
          endif
40     continue

      end

```

```

*_____
      SUBROUTINE TCA_AND_TDA
*_____

```

```

*_____
*|_____

```

```

*| This subroutine computes the total contributing area, tca(i,j),
*| and the total dispersal area, tda(i,j), of each pixel.
*| Flow is generated at each pixel at a time (the "source" pixel)
*| and is followed downstream until either the edge of the DEM is
*| reached or a "pit". The amount of flow (expressed as a
*| fraction of the area of the source pixel) entering each pixel
*| along the way is added to the tca of that pixel. The area
*| spanned by the flow inside each pixel along the way is added to
*| the tda of the source pixel. After flow has been generated at
*| every pixel (each pixel has been made a source pixel), and its
*| impact on each of the other pixels has been added, the final
*| tca values will have been computed.
*
* INPUT: nr, nc, istop(,)
* OUTPUT: tca(,), tda(,)
*

```

```

*
* _____
include 'demiivar.f'

      tps_max = 0.

      do 10 i = 2, nr
        print *, 'Cellule source ', i, j
        do 10 j = 2, nc

          numchem = 1
          numcell = 1

*       print *, 'Cellule source ', i, j, istop(i,j)

          is = i
          js = j

          in = i
          jn = j

          call SOURCE_PIXEL
          numcell = numcell + 1

          if(istop(in,jn).eq.1) then
            tca(in,jn) = tca(in,jn) + fn*cn(in,jn)
            goto 60
          endif

          goto 40

20      if(x2n.ne.x1n) goto 40

30      call FLOW_1D
*       print *, 'Flow_1d', in, jn
          call CHECK_LOOP
          zmannig = zmannig1
          call CALC_CELL

```

```

        if(istop(in,jn).eq.1.or.i2stop.eq.1) goto 60
        goto 30

40     call FLOW_2D
      *   print*, 'Flow_2d', in, jn
        call CHECK_LOOP
        zmanning = zmanning2
        call CALC_CELL

50     if(istop(in,jn).eq.1.or.i2stop.eq.1) goto 60
        goto 20

60     i2stop = 0
        jcell(numchem) = numcell - 1
        numchem = numchem+1
        numcell = 1
        if(ic.eq.0) goto 10
        call RETRACE_CHEMIN
        call WAITLIST
        if(istop(in,jn).eq.1) goto 60
        goto 40

10     call CHEMIN_MAX
        continue

      end

```

*
SUBROUTINE SOURCE_PIXEL
*

```

*
*
* | This subroutine generates flow over the source pixel and
* | determines the receiving pixel or pixels. If the flow angle in
* | the source pixel is 0, 90, 180 or 270 degrees, then there is a
* | single receiving pixel. Otherwise, there are two receiving
* | pixels, which are cardinal neighbors of the source pixel. If
* | there are two receiving pixels (the most frequent case), flow
* | is split among them.
*
* | INPUT:
* |   in, jn, ic, nr, nc, dx, dy, tca(,), angle(,)
* | OUTPUT:
* |   tca(,)
* |   Regarding a receiving pixel indicated by subscript 'n', for
* |   'next' because it will be the next one to step to:
* |   in, jn, idirn, fn, f1n, f2n, x1n, x2n, x3n, i3n;
* |   If there are two receiving pixels, the output variables
* |   regarding the receiving pixel indicated by 'w', for 'waiting'
* |   because it is put in waiting list (in position no. ic) are:
* |   ic, iw(ic), jw(ic), idirnw(ic), fw(ic), f1w(ic), f2w(ic),
* |   x1w(ic), x2w(ic), x3w(ic), i3w(ic).
*

```

```

include 'demiivar.f'

io = in
jo = jn

chemin(numchem,numcell,1) = io
chemin(numchem,numcell,2) = jo
chemin(numchem,numcell,3) = 0
chemin(numchem,numcell,4) = 0
chemin(numchem,numcell,5) = 0

* 1. GENERATE FLOW OVER THE SOURCE PIXEL :
*-----

fo = dx * dy * rain * cn(io,jo)
fow = dx * dy
tca(io, jo) = tca(io, jo) + fo

* 2. "GOTO" STATEMENTS DEPENDING ON FLOW ANGLE :
*-----

a = angle(io, jo)

if(a.eq.0.) goto 20
if(a.eq.90.) goto 22
if(a.eq.180.) goto 24
if(a.eq.270.) goto 26
goto 30

* 3. ANGLES THAT ARE 90-DEGREE MULTIPLES (1 RECEIVING ELEMENT ONLY):
*-----

* 3.1 ANGLE = 0 DEGREES:

20  jn = jo + 1
    fn = fo
    fnw = fow
    fln = dx
    f2n = dx
    idirn = 4
    x1n = 0.
    x2n = dy
    x3n = 0.
    i3n = 1
    RETURN

* 3.2 ANGLE = 90 DEGREES:

22  in = io - 1
    fn = fo
    fnw = fow
    fln = dy
    f2n = dy

```

```

idirn = 3
x1n = 0.
x2n = dx
x3n = 0.
i3n = 1
RETURN

```

* 3.3 ANGLE = 180 DEGREES:

```

24  jn = jo - 1
     fn = fo
     fnw = fow
     f1n = dx
     f2n = dx
     idirn = 2
     x1n = 0.
     x2n = dy
     x3n = 0.
     i3n = 1
     RETURN

```

* 3.4 ANGLE = 270 DEGREES:

```

26  in = io + 1
     fn = fo
     fnw = fow
     f1n = dy
     f2n = dy
     idirn = 1
     x1n = 0.
     x2n = dx
     x3n = 0.
     i3n = 1
     RETURN

```

* 4. ANGLES THAT ARE *NOT* 90-DEGREE MULTIPLES (2 RECEIVING ELEMENTS):

```

-----
30  ic = ic + 1
     inumchem(ic) = numchem
     inumcell(ic) = numcell
     x1n = 0.
     x2n = dy
     x1w(ic) = 0.
     x2w(ic) = dx
     tanda = tand(a)
     if(a.lt.90.) goto 32
     if(a.lt.180.) goto 34
     if(a.lt.270.) goto 36
     goto 38

```

* 4.1 ANGLE IN 1st QUADRANT:

```

32  jn = jo + 1
     idirn = 4

```

```

iw(ic) = io - 1
jw(ic) = jo
idirw(ic) = 3
f1n = 0.
f2w(ic) = 0.

if(a.le.anglecut) then
f = 1. - 0.5 * dx/dy * tanda
f2n = dx / cosd(a)
x3n = dx * tanda
i3n = 2
x3w(ic) = 0.
i3w(ic) = 1

else
f = 0.5 * dy/dx / tanda
f2n = dy / sind(a)
x3n = 0.
i3n = 1
x3w(ic) = dx - dy / tanda
i3w(ic) = 1
endif

fn = f * fo
fnw = f * fow
fw(ic) = fo - fn
fww(ic) = fow - fnw
flw(ic) = f2n
RETURN

```

* 4.2 ANGLE IN 2nd QUADRANT:

```

34  jn = jo - 1
    idirn = 2
    iw(ic) = io - 1
    jw(ic) = jo
    idirw(ic) = 3
    f2n = 0.
    flw(ic) = 0.

    if(a.le.180.-anglecut) then
f = - 0.5 * dy/dx / tanda
f1n = dy / sind(a)
x3n = 0.
i3n = 1
x3w(ic) = - dy / tanda
i3w(ic) = 2

    else
f = 1. + 0.5 * dx/dy * tanda
f1n = - dx / cosd(a)
x3n = dy + dx * tanda
i3n = 1
x3w(ic) = 0.
i3w(ic) = 1

```

```
endif
```

```
fn = f * fo
fnw = f * fow
fw(ic) = fo - fn
fww(ic) = fow - fnw
f2w(ic) = fln
RETURN
```

• 4.3 ANGLE IN 3rd QUADRANT:

```
36  jn = jo - 1
    idirn = 2
    iw(ic) = io + 1
    jw(ic) = jo
    idirw(ic) = 1
    fln = 0.
    f2w(ic) = 0.

    if(a.le.180.+anglecut) then
      f = 1. - 0.5 * dx/dy * tanda
      f2n = - dx / cosd(a)
      x3n = dx * tanda
      i3n = 2
      x3w(ic) = 0.
      i3w(ic) = 1

    else
      f = 0.5 * dy/dx / tanda
      f2n = - dy / sind(a)
      x3n = 0.
      i3n = 1
      x3w(ic) = dx - dy / tanda
      i3w(ic) = 1
    endif

    fn = f * fo
    fnw = f * fow
    fw(ic) = fo - fn
    fww(ic) = fow - fnw
    flw(ic) = f2n
    RETURN
```

* 4.4 ANGLE IN 4th QUADRANT:

```
38  jn = jo + 1
    idirn = 4
    iw(ic) = io + 1
    jw(ic) = jo
    idirw(ic) = 1
    f2n = 0.
    flw(ic) = 0.

    if(a.le.360.-anglecut) then
      f = - 0.5 * dy/dx / tanda
```

```

fln = - dy / sind(a)
x3n = 0.
i3n = 1
x3w(ic) = - dy / tanda
i3w(ic) = 2

else
f = 1. + 0.5 * dx/dy * tanda
fln = dx / cosd(a)
x3n = dy + dx * tanda
i3n = 1
x3w(ic) = 0.
i3w(ic) = 1
endif

fn = f * fo
fnw = f * fow
fw(ic) = fo - fn
fww(ic) = fow - fnw
f2w(ic) = fln
RETURN

end

```

```

*
SUBROUTINE FLOW_2D
*

```

```

*
*|
*| This subroutine projects a 2D flow "tube" across a pixel. It
*| takes the variables describing the flow entering the pixel and
*| determines which pixel or pixels ('receiving pixels') the flow
*| will move to; and computes the variables describing the flow
*| entering each receiving pixel.
*|
*| INPUT: is, js, in, jn, fn, idirn, x1n, x2n, x3n, i3n, fln,
*|        f2n, angle(,), tca(,), tda(,)
*| OUTPUT:
*|        tca(,), tda(,)
*| Regarding a receiving pixel indicated by subscript 'n', for
*| 'next' because it will be the next one to step to:
*| in, jn, fn, idirn, x1n, x2n, x3n, i3n, fln, f2n,
*| If there are two receiving pixels, the output variables
*| regarding the receiving pixel indicated by 'w', for 'waiting'
*| because it is put in waiting list in position number 'ic' are:
*| ic, iw(ic), jw(ic), idirnw(ic), fw(ic), x1w(ic), x2w(ic),
*| x3w(ic), i3w(ic), flw(ic), f2w(ic).
*|
*|

```

```

include 'demiivar.f'

```

```

io = in
jo = jn
chemin(numchem,numcell,1) = io

```

```

chemin(numchem,numcell,2) = jo
fo = fn * cn(io,jo)
fow = fnw
idiro = idirn
x1o = x1n
x2o = x2n
x3o = x3n
i3o = i3n
f1o = f1n
f2o = f2n

tca(io,jo) = tca(io,jo) + fo
a = angle(io,jo)

goto(10, 20, 30, 40) idiro

*• 1. IF FLOW ENTERED THROUGH THE NORTHERN FACE (idiro = 1):
-----
10  continue

* 1.1 IF CONVERGENCE IS FOUND:

    if(a.le.180.) then
    tca(io,jo) = tca(io,jo) - fo
    call CONVERGENCE
    RETURN
    endif

*• 1.2 IF DIRECTION IS PARALLEL THROUGH:

    if(a.eq.270.)then
    in = io + 1
    area = dy*(x2o-x1o)
    tda(is, js) = tda(is, js) + area
    RETURN
    endif

*• 1.3 IF FLOW IS AT AN ANGLE:

    tanda = tand(a)
    p1 = x1o - dy / tanda
    p2 = x2o - dy / tanda
    p3 = x3o - dy / tanda

*• 1.3.1 IF SOME FLOW GOES WEST:

    if(p1.lt.0.)then
    in = io
    jn = jo - 1
    idirn = 2
    x1n = x1o * tanda
    f1n = f1o

*• 1.3.1.1 IF ALL FLOW GOES WEST:

```

```

if(p2.le.0.)then
x2n = x2o * tanda
x3n = x3o * tanda
area = 0.5*(x2o-x1o)*(x1n+x2n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 1.3.1.2 IF FLOW GOES WEST AND SOUTH:

```

x2n = dy
ic = ic + 1
inumchem(ic) = numchem
inumcell(ic) = numcell
iw(ic) = io + 1
jw(ic) = jo
idirw(ic) = 1
x1w(ic) = 0.
x2w(ic) = p2

if(p3.le.0.) then
x3n = x3o * tanda
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = - p3 / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(-p3)
    g2 = 0.5*(f2n+f2o)*p2
  else
    f2n = f2o
    g1 = 0.5*(f1o+f2o)*(p3-p1) + f2o*(-p3)
    g2 = f2o*p2
  endif
else
x3n = x1n
i3n = 1
x3w(ic) = p3
i3w(ic) = i3o
  if(i3o.eq.1) then
    f2n = f1o
    g1 = f1o*(-p1)
    g2 = f1o*p3 + 0.5*(f1o+f2o)*(p2-p3)
  else
    f = p3 / (p3-p1)
    f2n = f*f1o + (1.-f)*f2o
    g1 = 0.5*(f1o+f2n)*(-p1)
    g2 = 0.5*(f2n+f2o)*p3 + f2o*(p2-p3)
  endif
endif

f1w(ic) = f2n
:2w(ic) = f2o

```

```

g = g1/(g1+g2)
fn = g * fo
fnw = g * fow
fw(ic) = fo - fn
fww(ic) = fow - fnw

area = dy*(x2o-x1o) + 0.5*p1*(x2n-x1n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 1.3.2 IF SOME FLOW GOES SOUTH:

```

if(p1.lt.dx)then
in = io + 1
jn = jo
idirn = 1
x1n = p1
f1n = f1o

```

** 1.3.2.1 IF ALL FLOW GOES SOUTH:

```

if(p2.le.dx)then
x2n = p2
x3n = p3
area = dy*(x2o-x1o)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 1.3.2.2 IF FLOW GOES SOUTH AND EAST:

```

x2n = dx
ic = ic + 1
inumchem(ic) = numchem
inumcell(ic) = numcell
iw(ic) = io
jw(ic) = jo + 1
idirw(ic) = 4
x1w(ic) = 0.
x2w(ic) = (dx-p2) **tanda

if(p3.le.dx) then
x3n = p3
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = (dx - p3) / (p2 - p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(dx-p3)
    g2 = 0.5*(f2n+f2o)*(p2-dx)
  else
    f2n = f2o
    g1 = 0.5*(f1o+f2o)*(p3-p1) + f2o*(dx-p3)
  
```

```

        g2 = f2o*(p2-dx)
    endif
else
x3n = x1n
i3n = 1
x3w(ic) = (dx-p3) * tanda
i3w(ic) = i3o
    if(i3o.eq.1) then
        f2n = f1o
        g1 = f1o*(dx-p1)
        g2 = f1o*(p3-dx) + 0.5*(f1o+f2o)*(p2-p3)
    else
        f = (p3-dx) / (p3-p1)
        f2n = f*f1o + (1.-f)*f2o
        g1 = 0.5*(f1o+f2n)*(dx-p1)
        g2 = 0.5*(f2n+f2o)*(p3-dx) + f2o*(p2-p3)
    endif
endif

f1w(ic) = f2n
f2w(ic) = f2o
g = g1/(g1+g2)
fn = g ** fo
fnw = g * fow
fw(ic) = fo - fn
fww(ic) = fow - fnw

area = dy*(x2o-x1o) - 0.5*x2w(ic)*(p2-dx)
tda(is, js) = tda(is, js) + area
RETURN
endif

* 1.3.3 IF ALL FLOW GOES EAST:

jn = jo + 1
idirn = 4
x1n = (dx-p1) * tanda
x2n = (dx-p2) ** tanda
x3n = (dx-p3) ** tanda
area = (x2o-x1o)*(dy-0.5*(x1n+x2n))
tda(is, js) = tda(is, js) + area
RETURN

* 2. IF FLOW ENTERED THROUGH THE EASTERN FACE (idiro = 2):
*-----

20 continue

*• 2.1 IF CONVERGENCE IS FOUND:

if(a.le.90..or.a.ge.270.) then
tca(io,jo) = tca(io,jo) - fo
call CONVERGENCE
RETURN
endif

```

* 2.2 IF DIRECTION IS PARALLEL THROUGH:

```

if(a.eq.180.)then
jn = jo - 1
area = dx*(x2o-x1o)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 2.3 IF FLOW IS AT AN ANGLE:

```

tanda = tand(a)
p1 = x1o + dx * tanda
p2 = x2o + dx * tanda
p3 = x3o + dx * tanda

```

* 2.3.1 IF SOME FLOW GOES NORTH:

```

if(p1.lt.0.)then
in = io - 1
jn = jo
idirn = 3
x1n = - x1o / tanda
f1n = f1o

```

* 2.3.1.1 IF ALL FLOW GOES NORTH:

```

if(p2.le.0.)then
x2n = - x2o / tanda
x3n = - x3o / tanda
area = 0.5*(x2o-x1o)*(x1n+x2n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 2.3.1.2 IF FLOW GOES NORTH AND WEST:

```

x2n = dx
ic = ic + 1
inumchem(ic) = numchem
inumcell(ic) = numcell
iw(ic) = io
jw(ic) = jo - 1
idirw(ic) = 2
x1w(ic) = 0.
x2w(ic) = p2

if(p3.le.0.) then
x3n = - x3o / tanda
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
      if(i3o.eq.1) then
        f = - p3 / (p2-p3)

```

```

        f2n = f*f2o + (1.-f)*f1o
        g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(-p3)
        g2 = 0.5*(f2n+f2o)*p2
    else
        f2n = f2o
        g1 = f2o*(-p3) + 0.5*(f1o+f2o)*(p3-p1)
        g2 = f2o*p2
    endif
else
x3n = x1n
i3n = 1
x3w(ic) = p3
i3w(ic) = i3o
    if(i3o.eq.1) then
        f2n = f1o
        g1 = f1o*(-p1)
        g2 = f1o*p3 + 0.5*(f1o+f2o)*(p2-p3)
    else
        f = p3 / (p3-p1)
        f2n = f*f1o + (1.-f)*f2o
        g1 = 0.5*(f1o+f2n)*(-p1)
        g2 = 0.5*(f2n+f2o)*p3 + f2o*(p2-p3)
    endif
endif

f1w(ic) = f2n
f2w(ic) = f2o
g = g1 / (g1+g2)
fn = g * fo
fnw = g * fow
fw(ic) = fo - fn
fww(ic) = fow - fnw

area = dx*(x2o-x1o) + 0.5*p1*(x2n-x1n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

•• 2.3.2 IF SOME FLOW GOES WEST:

```

    if(p1.lt.dy)then
        in = io
        jn = jo - 1
        idirn = 2
        x1n = p1
        f1n = f1o
    
```

•• 2.3.2.1 IF ALL FLOW GOES WEST:

```

    if(p2.le.dy)then
        x2n = p2
        x3n = p3
        area = dx*(x2o-x1o)
        tda(is, js) = tda(is, js) + area
    RETURN
    
```

```
endif
```

```
* 2.3.2.2 IF FLOW GOES WEST AND SOUTH:
```

```
x2n = dy
ic = ic + 1
inumchem(ic) = numchem
inumcell(ic) = numcell
iw(ic) = io + 1
jw(ic) = jo
idirw(ic) = 1
x1w(ic) = 0.
x2w(ic) = (p2-dy) / tanda

if(p3.le.dy) then
x3n = p3
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = (dy-p3) / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(dy-p3)
    g2 = 0.5*(f2n+f2o)*(p2-dy)
  else
    f2n = f2o
    g1 = 0.5*(f1o+f2o)*(p3-p1) + f2o*(dy-p3)
    g2 = f2o*(p2-dy)
  endif
else
x3n = x1n
i3n = 1
x3w(ic) = (p3-dy) / tanda
i3w(ic) = i3o
  if(i3o.eq.1) then
    f2n = f1o
    g1 = f1o*(dy-p1)
    g2 = f1o*(p3-dy) + 0.5*(f1o+f2o)*(p2-p3)
  else
    f = (p3-dy) / (p3-p1)
    f2n = f*f1o + (1.-f)*f2o
    g1 = 0.5*(f1o+f2n)*(dy-p1)
    g2 = 0.5*(f2n+f2o)*(p3-dy) + f2o*(p2-p3)
  endif
endif

f1w(ic) = f2n
f2w(ic) = f2o
g = g1 / (g1+g2)
fn = g * fo
fnw = g * fow
fw(ic) = fo - fn
fww(ic) = fow - fnw

area = dx*(x2o-x1o) - 0.5*(x2w(ic)*(p2-dy))
```

```

        tda(is, js) = tda(is, js) + area
        RETURN
    endif

* 2.3.3 IF ALL FLOW GOES SOUTH:

        in = io + 1
        idirn = 1
        x1n = (p1-dy) / tanda
        x2n = (p2-dy) / tanda
        x3n = (p3-dy) / tanda
        area = (x2o-x1o)*(dx-0.5*(x1n+x2n))
        tda(is, js) = tda(is, js) + area
        RETURN

* 3. IF FLOW ENTERED THROUGH THE SOUTHERN FACE (idiro = 3):
*-----

30     continue

* 3.1 IF CONVERGENCE IS FOUND:

        if(a.ge.180..or.a.eq.0.) then
            tca(io,jo) = tca(io,jo) - fo
            call CONVERGENCE
            RETURN
        endif

* 3.2 IF DIRECTION IS PARALLEL THROUGH:

        if(a.eq.90.)then
            in = io - 1
            area = dy*(x2o-x1o)
            tda(is, js) = tda(is, js) + area
            RETURN
        endif

* 3.3 IF FLOW IS AT AN ANGLE:

        tanda = tand(a)
        p1 = x1o - dy / tanda
        p2 = x2o - dy / tanda
        p3 = x3o - dy / tanda

* 3.3.1 IF SOME FLOW GOES EAST:

        if(p1.lt.0.)then
            in = io
            jn = jo + 1
            idirn = 4
            x1n = x1o * tanda
            f1n = f1o

* 3.3.1.1 IF ALL FLOW GOES EAST:

```

```

if(p2.le.0.)then
x2n = x2o * tanda
x3n = x3o * tanda
area = 0.5*(x2o-x1o)*(x1n+x2n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 3.3.1.2 IF FLOW GOES EAST AND NORTH:

```

x2n = dy
ic = ic + 1
inumchem(ic) = numchem
inumcell(ic) = numcell
iw(ic) = io - 1
jw(ic) = jo
idirw(ic) = 3
x1w(ic) = 0.
x2w(ic) = p2

if(p3.le.0.) then
x3n = x3o * tanda
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = - p3 / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(-p3)
    g2 = 0.5*(f2n+f2o)*p2
  else
    f2n = f2o
    g1 = f2o*(-p3) + 0.5*(f2o+f1o)*(p3-p1)
    g2 = f2o * p2
  endif
else
x3n = x1n
i3n = 1
x3w(ic) = p3
i3w(ic) = i3o
  if(i3o.eq.1) then
    f2n = f1o
    g1 = f1o*(-p1)
    g2 = f1o*p3 + 0.5*(f1o+f2o)*(p2-p3)
  else
    f = p3 / (p3-p1)
    f2n = f*f1o + (1.-f)*f2o
    g1 = 0.5*(f1o+f2n)*(-p1)
    g2 = 0.5*(f2n+f2o)*p3 + f2o*(p2-p3)
  endif
endif

f1w(ic) = f2n
f2w(ic) = f2o
g = g1 / (g1+g2)

```

```

fn = g * fo
fnw = g * fow
fw(ic) = fo - fn
fw(ic) = fow - fnw

area = dy*(x2o-x1o) + 0.5*p1*(x2n-x1n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 3.3.2 IF SOME FLOW GOES NORTH:

```

if(p1.lt.dx)then
in = io - 1
jn = jo
idirn = 3
x1n = p1
f1n = f1o

```

* 3.3.2.1 IF ALL FLOW GOES NORTH:

```

if(p2.le.dx)then
x2n = p2
x3n = p3
area = dy*(x2o-x1o)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 3.3.2.2 IF FLOW GOES NORTH AND WEST:

```

x2n = dx
ic = ic + 1
inumchem(ic) = numchem
inumcell(ic) = numcell
iw(ic) = io
jw(ic) = jo - 1
idirw(ic) = 2
x1w(ic) = 0.
x2w(ic) = (dx-p2) * tanda

if(p3.le.dx) then
x3n = p3
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = (dx-p3) / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
    g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(dx-p3)
    g2 = 0.5*(f2n+f2o)*(p2-dx)
  else
    f2n = f2o
    g1 = f2o*(dx-p3) + 0.5*(f2o+f1o)*(p3-p1)
    g2 = f2o*(p2-dx)
  
```

```

        endif
    else
        x3n = x1n
        i3n = 1
        x3w(ic) = (dx-p3) * tanda
        i3w(ic) = i3o
        if(i3o.eq.1) then
            f2n = f1o
            g1 = f1o*(dx-p1)
            g2 = f1o*(p3-dx) + 0.5*(f1o+f2o)*(p2-p3)
        else
            f = (p3-dx) / (p3-p1)
            f2n = f*f1o + (1.-f)*f2o
            g1 = 0.5*(f1o+f2n)*(dx-p1)
            g2 = 0.5*(f2n+f2o)*(p3-dx) + f2o*(p2-p3)
        endif
    endif

    endif

    f1w(ic) = f2n
    f2w(ic) = f2o
    g = g1 / (g1+g2)
    fn = g * fo
    fnw = g * fow
    fw(ic) = fo - fn
    fww(ic) = fow - fnw

    area = dy*(x2o-x1o) - 0.5*x2w(ic)*(p2-dx)
    tda(is, js) = tda(is, js) + area
    RETURN
endif

```

** 3.3.3 IF ALL FLOW GOES WEST:

```

    jn = jo - 1
    idirn = 2
    x1n = (dx-p1) ** tanda
    x2n = (dx-p2) * tanda
    x3n = (dx-p3) ** tanda
    area = (x2o-x1o)*(dy-0.5*(x1n+x2n))
    tda(is, js) = tda(is, js) + area
    RETURN

```

* 4. IF FLOW ENTERED THROUGH THE WESTERN FACE (idiro = 4):

*-----

40 continue

* 4.1 IF CONVERGENCE IS FOUND:

```

    if(a.ge.90..and.a.le.270.) then
        tca(io,jo) = tca(io,jo) - fo
        call CONVERGENCE
        RETURN
    endif

```

* 4.2 IF DIRECTION IS PARALLEL THROUGH:

```

if(a.eq.0.)then
jn = jo + 1
area = dx*(x2o-x1o)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 4.3 IF FLOW IS AT AN ANGLE:

```

tanda = tand(a)
p1 = x1o + dx ** tanda
p2 = x2o + dx ** tanda
p3 = x3o + dx ** tanda

```

* 4.3.1 IF SOME FLOW GOES SOUTH:

```

if(p1.lt.0.)then
in = io + 1
jn = jo
idirn = 1
x1n = - x1o / tanda
f1n = f1o

```

* 4.3.1.1 IF ALL FLOW GOES SOUTH:

```

if(p2.le.0.)then
x2n = - x2o / tanda
x3n = - x3o / tanda
area = 0.5*(x2o-x1o)*(x1n+x2n)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

* 4.3.1.2 IF FLOW GOES SOUTH AND EAST:

```

x2n = dx
ic = ic + 1
inumchem(ic) = numchem
inumcell(ic) = numcell
iw(ic) = io
jw(ic) = jo + 1
idirw(ic) = 4
x1w(ic) = 0.
x2w(ic) = p2

if(p3.le.0.) then
x3n = - x3o / tanda
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
  if(i3o.eq.1) then
    f = - p3 / (p2-p3)
    f2n = f*f2o + (1.-f)*f1o
  
```

```

        g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(-p3)
        g2 = 0.5*(f2n+f2o)*p2
        else
            f2n = f2o
            g1 = 0.5*(f1o+f2o)*(p3-p1) + f2o*(-p3)
            g2 = f2o*p2
        endif
    else
        x3n = x1n
        i3n = 1
        x3w(ic) = p3
        i3w(ic) = i3o
        if(i3o.eq.1) then
            f2n = f1o
            g1 = f1o ** (-p1)
            g2 = f1o*p3 + 0.5*(f1o+f2o)*(p2-p3)
        else
            f = - p1 / (p3-p1)
            f2n = f*f2o + (1.-f)*f1o
            g1 = 0.5*(f1o+f2n)*(-p1)
            g2 = 0.5*(f2n+f2o)*p3 + f2o*(p2-p3)
        endif
    endif

    f1w(ic) = f2n
    f2w(ic) = f2o
    g = g1 / (g1+g2)
    fn = g ** fo
    fnw = g ** fow
    fw(ic) = fo - fn
    fww(ic) = fow - fnw

    area = dx*(x2o-x1o) + 0.5*p1*(x2n-x1n)
    tda(is, js) = tda(is, js) + area
    RETURN.
endif

```

* 4.3.2 IF SOME FLOW GOES EAST:

```

if(p1.lt.dy)then
in = io
jn = jo + 1
idirn = 4
x1n = p1
f1n = f1o

```

* 4.3.2.1 IF ALL FLOW GOES EAST:

```

if(p2.le.dy)then
x2n = p2
x3n = p3
area = dx*(x2o-x1o)
tda(is, js) = tda(is, js) + area
RETURN
endif

```

```

*      4.3.2.2 IF FLOW GOES EAST AND NORTH:

x2n = dy
ic = ic + 1
inumchem(ic) = numchem
inumcell(ic) = numcell
iw(ic) = ic - 1
jw(ic) = jo
idirw(ic) = 3
x1w(ic) = 0.
x2w(ic) = (p2-dy) / tanda

if(p3.le.dy) then
x3n = p3
i3n = i3o
x3w(ic) = 0.
i3w(ic) = 1
      if(i3o.eq.1) then
        f = (dy-p3) / (p2-p3)
        f2n = f*f2o + (1.-f)*f1o
        g1 = f1o*(p3-p1) + 0.5*(f1o+f2n)*(dy-p3)
        g2 = 0.5*(f2n+f2o)*(p2-dy)
      else
        f2n = f2o
        g1 = f2o*(dy-p3) + 0.5*(f2o+f1o)*(p3-p1)
        g2 = f2o*(p2-dy)
      endif
else
x3n = x1n
i3n = 1
x3w(ic) = (p3-dy) / tanda
i3w(ic) = i3o
      if(i3o.eq.1) then
        f2n = f1o
        g1 = f1o*(dy-p1)
        g2 = f1o*(p3-dy) + 0.5*(f2o+f1o)*(p2-p3)
      else
        f = (p3-dy) / (p3-p1)
        f2n = f*f1o + (1.-f)*f2o
        g1 = 0.5*(f1o+f2n)*(dy-p1)
        g2 = f2o*(p2-p3) + 0.5*(f2o+f2n)*(p3-dy)
      endif
endif

f1w(ic) = f2n
f2w(ic) = f2o
g = g1 / (g1+g2)
fn = g * fo
fnw = g * fow
fw(ic) = fo - fn
fww(ic) = fow - fnw

area = dx*(x2o-x1o) - 0.5*x2w(ic)*(p2-dy)
tda(is, js) = tda(is, js) + area

```

```

RETURN
endif

* 4.3.3 IF ALL FLOW GOES NORTH:

in = io - 1
idirn = 3
x1n = (p1-dy) / tanda
x2n = (p2-dy) / tanda
x3n = (p3-dy) / tanda
area = (x2o-x1o)*(dx-0.5*(x1n+x2n))
tda(is, js) = tda(is, js) + area
RETURN

end

*
-----
SUBROUTINE FLOW_1D
*
-----
*
* This subroutine projects a line of concentrated flow through a
* pixel. It takes the variables describing the flow entering the
* pixel and determines which pixel ('receiving pixel') the flow
* will move to; and computes the variables describing the flow
* entering the receiving pixel.
*
* INPUT:
*   in, jn, fn, idirn, x1n, angle(,), tca(,)
* OUTPUT:
*   tca(,), in, jn, fn, idirn, x1n
*
include 'demiivar.f'

io = in
jo = jn
chemin(numchem,numcell,1) = io
chemin(numchem,numcell,2) = jo
fo = fn * cn(io,jo)
fow = fnw
idiro = idirn
xlo = x1n

tca(io,jo) = tca(io,jo) + fo
a = angle(io,jo)

goto(10, 20, 30, 40) idiro

* 1. IF FLOW ENTERED THROUGH THE NORTHERN FACE (idiro = 1):
*-----
10 continue

```

```
* 1.0 IF WE'RE AT A CORNER:

    if(xlo.eq.0..and.idirn.eq.1) then
    call CORNERS
    RETURN
    endif

* 1.1 IF CONVERGENCE IS FOUND:

    if(a.le.180.) then
    tca(io,jo) = tca(io,jo) - fo
    call CONVERGENCE
    RETURN
    endif

* 1.2 IF DIRECTION IS PARALLEL THROUGH:

    if(a.eq.270.)then
    in = io + 1
    jn = jo
    xln = xlo
    idirn = idiro
    RETURN
    endif

* 1.3 IF FLOW IS AT AN ANGLE:

    p1 = xlo - dy * tand(270.-a)

* 1.3.1 IF FLOW GOES WEST:

    if(p1.lt.0.)then
    in = io
    jn = jo - 1
    idirn = 2
    xln = xlo ** tand(a)
    RETURN
    endif

* 1.3.2 IF FLOW GOES SOUTH:

    if(p1.lt.dx)then
    in = io + 1
    jn = jo
    idirn = 1
    xln = p1
    RETURN
    endif

* 1.3.3 IF FLOW GOES EAST:

    in = io
    jn = jo + 1
    idirn = 4
    xln = dy + (dx-xlo) * tand(a)
```

```

RETURN

* 2. IF FLOW ENTERED THROUGH THE EASTERN FACE (idiro = 2):
*-----
20   continue

* 2.1 IF CONVERGENCE IS FOUND:

      if(a.le.90..or.a.ge.270.) then
      tca(io,jo) = tca(io,jo) - fo
      call CONVERGENCE
      RETURN
      endif

* 2.2 IF DIRECTION IS PARALLEL THROUGH:

      if(a.eq.180.)then
      in = io
      jn = jo - 1
      xln = xlo
      idirn = idiro
      RETURN
      endif

* 2.3 IF FLOW IS AT AN ANGLE:

      pl = xlo + dx * tand(a)

* 2.3.1 IF FLOW GOES NORTH:

      if(pl.lt.0.)then
      in = io - 1
      jn = jo
      idirn = 3
      xln = xlo * tand(a-90.)
      RETURN
      endif

* 2.3.2 IF FLOW GOES WEST:

      if(pl.lt.dy)then
      in = io
      jn = jo - 1
      idirn = 2
      xln = pl
      RETURN
      endif

* 2.3.3 IF FLOW GOES SOUTH:

      in = io + 1
      jn = jo
      idirn = 1
      xln = dx - (dy-xlo) * tand(270.-a)

```

```

RETURN

* 3. IF FLOW ENTERED THROUGH THE SOUTHERN FACE (idiro = 3):
*-----

30    continue

* 3.1 IF CONVERGENCE IS FOUND:

      if(a.ge.180..or.a.eq.0.) then
      tca(io,jo) = tca(io,jo) - fo
      call CONVERGENCE
      RETURN
      endif

* 3.2 IF DIRECTION IS PARALLEL THROUGH:

      if(a.eq.90.)then
      in = io - 1
      jn = jo
      xln = xlo
      idirn = idiro
      RETURN
      endif

* 3.3 IF FLOW IS AT AN ANGLE:

      pl = xlo - dy * tand(90.-a)

* 3.3.1 IF FLOW GOES EAST:

      if(pl.lt.0.)then
      in = io
      jn = jo + 1
      idirn = 4
      xln = xlo * tand(a)
      RETURN
      endif

* 3.3.2 IF FLOW GOES NORTH:

      if(pl.lt.dx)then
      in = io - 1
      jn = jo
      idirn = 3
      xln = pl
      RETURN
      endif

* 3.3.3 IF FLOW GOES WEST:

      in = io
      jn = jo - 1
      idirn = 2
      xln = dy + (dx-xlo) * tand(a)

```

```

RETURN

* 4. IF FLOW ENTERED THROUGH THE WESTERN FACE (idiro = 4):
*-----*

40    continue

* 4.1 IF CONVERGENCE IS FOUND:

      if(a.ge.90..and.a.le.270.) then
      tca(io,jo) = tca(io,jo) - fo
      call CONVERGENCE
      RETURN
      endif

* 4.2 IF DIRECTION IS PARALLEL THROUGH:

      if(a.eq.0.)then
      in = io
      jn = jo + 1
      xln = xlo
      idirn = idiro
      RETURN
      endif

* 4.3 IF FLOW IS AT AN ANGLE:

      pl = xlo + dx * tand(a)

* 4.3.1 IF FLOW GOES SOUTH:

      if(pl.lt.0.)then
      in = io + 1
      jn = jo
      idirn = 1
      xln = xlo ** tand(a-270.)
      RETURN
      endif

* 4.3.2 IF FLOW GOES EAST:

      if(pl.lt.dy)then
      in = io
      jn = jo + 1
      idirn = 4
      xln = pl
      RETURN
      endif

* 4.3.3 IF FLOW GOES NORTH:

      in = io - 1
      jn = jo
      idirn = 3
      xln = dx - (dy-xlo) ** tand(90.-a)

```

```
RETURN
```

```
end
```

```
*
```

```
-----  
SUBROUTINE CONVERGENCE  
-----
```

```
*
```

```
*
```

```
*
```

```
*
```

```
*
```

```
*
```

```
*
```

```
*
```

```
*
```

```
This subroutine determines the next pixel to step to in the  
case of flow angles in two neighbor pixels converging to the  
shared boundary.
```

```
INPUT: io, jo, fo, idiro
```

```
OUTPUT: in, jn, fn, idirn, x1n, x2n
```

```
include 'demiivar.f'
```

```
x1n = 0.
```

```
x2n = 0.
```

```
idirn = 1
```

```
fn = fo
```

```
fnw = fow
```

```
goto(10, 20, 30, 40) idiro
```

```
* 1. IF CONVERGENCE IS AT THE NORTHERN BOUNDARY (idiro = 1):
```

```
-----
```

```
10  z1 = z(io, jo)  
    z2 = z(io, jo+1)  
    if(z1.le.z2) then  
      in = io  
      jn = jo  
    else  
      in = io  
      jn = jo + 1  
    endif  
    RETURN
```

```
* 2. IF CONVERGENCE IS AT THE EASTERN BOUNDARY (idiro = 2):
```

```
-----
```

```
20  z2 = z(io, jo+1)  
    z4 = z(io+1, jo+1)  
    if(z2.le.z4) then  
      in = io  
      jn = jo + 1  
    else  
      in = io + 1  
      jn = jo + 1  
    endif  
    RETURN
```

* 3. IF CONVERGENCE IS AT THE SOUTHERN BOUNDARY (idiro = 3):

```
30  z3 = z(io+1, jo)
    z4 = z(io+1, jo+1)
    if(z3.le.z4) then
      in = io + 1
      jn = jo
    else
      in = io + 1
      jn = jo + 1
    endif
    RETURN
```

* 4. IF CONVERGENCE IS AT THE WESTERN BOUNDARY (idiro = 4):

```
40  z1 = z(io, jo)
    z3 = z(io+1, jo)
    if(z1.le.z3) then
      in = io
      jn = jo
    else
      in = io + 1
      jn = jo
    endif
    RETURN
```

end

*

SUBROUTINE CORNERS

*

*

*

```
* | This subroutine routes flow from a pixel corner to another
* | pixel corner, in the direction of maximal slope.
* | INPUT: io, jo, z(,), dx, dy
* | OUTPUT: in, jn
* |
```

include 'demiivar.f'

```
z1 = z(io, jo)
zN = z(io-1, jo)
zE = z(io, jo+1)
zS = z(io+1, jo)
zW = z(io, jo-1)
zNE= z(io-1, jo+1)
zSE= z(io+1, jo+1)
zSW= z(io+1, jo-1)
zNW= z(io-1, jo-1)
```

sN = (z1 - zN) / dy

```
sE = (z1 - zE) / dx
sS = (z1 - zS) / dy
sW = (z1 - zW) / dx
diag = sqrt(dx**2 + dy**2)
sNE = (z1 - zNE) / diag
sSE = (z1 - zSE) / diag
sSW = (z1 - zSW) / diag
sNW = (z1 - zNW) / diag
```

```
smax = 0.
if(sN.gt.smax) then
smax = sN
in = io - 1
jn = jo
endif
if(sE.gt.smax) then
smax = sE
in = io
jn = jo + 1
endif
if(sS.gt.smax) then
smax = sS
in = io + 1
jn = jo
endif
if(sW.gt.smax) then
smax = sW
in = io
jn = jo - 1
endif
if(sNE.gt.smax) then
smax = sNE
in = io - 1
jn = jo + 1
endif
if(sSE.gt.smax) then
smax = sSE
in = io + 1
jn = jo + 1
endif
if(sSW.gt.smax) then
smax = sSW
in = io + 1
jn = jo - 1
endif
if(sNW.gt.smax) then
smax = sNW
in = io - 1
jn = jo - 1
endif
```

```
end
```

* SUBROUTINE WAITLIST

```

*
*
* | This subroutine takes an element from the waiting list and
* | makes it the 'next' element to step to.
* | INPUT: ic2, ic, iw(ic2), jw(ic2), fw(ic2), idirw(ic2),
* | x3w(ic2), i3w(ic2), f1w(ic2), f2w(ic2)
* | OUTPUT: ic2, ic, in, jn, fn, idirn, x3n, i3n, f1n, f2n
* |
*

```

```

include 'demiivar.f'

```

```

ic2 = ic2 + 1

```

```

in = iw(ic2)
jn = jw(ic2)
fn = fw(ic2)
fnw = fww(ic2)
idirn = idirw(ic2)
x1n = x1w(ic2)
x2n = x2w(ic2)
x3n = x3w(ic2)
i3n = i3w(ic2)
f1n = f1w(ic2)
f2n = f2w(ic2)

```

```

if(ic2.eq.ic) then
ic = 0
ic2 = 0
endif

```

```

end

```

```

SUBROUTINE RETRACE_CHEMIN

```

```

include 'demiivar.f'

```

```

do 10 i=1, inumcell(ic)

```

```

chemin(numchem,i,1) = chemin(inumchem(ic),i,1)
chemin(numchem,i,2) = chemin(inumchem(ic),i,2)
chemin(numchem,i,3) = chemin(inumchem(ic),i,3)
chemin(numchem,i,4) = chemin(inumchem(ic),i,4)
chemin(numchem,i,5) = chemin(inumchem(ic),i,5)

```

```

10 continue

```

```

numcell = inumcell(ic) + 1

```

```

RETURN

```

```

end

```

```

*
SUBROUTINE CHECK_LOOP
*

include 'demiivar.f'

fin = numcell-5
do 10 i=1, fin

if (in.eq.chemin(numchem,i,1).and.jn.eq.chemin(numchem,i,2)) then
i2stop = 1
RETURN
endif

10 continue

RETURN

end

*
SUBROUTINE CALC_CELL
*

include 'demiivar.f'

• Calcul de la largeur en fonction de l'angle (width)

a= angle (io,jo)

if(a.lt.90.) alpha = a
if(a.ge.90..and.a.lt.180.) alpha = 180. - a
if(a.ge.180..and.a.lt.270.) alpha = 270. - a
if(a.ge.270.) alpha = 360. - a

beta = abs( alpha - anglecut )
diagonal = sqrt( dx**2 + dy**2 )
width = diagonal * cosd(beta)

• Calcul de la vitesse

segment = abs (fow / width)
r = (fo / (dx*dy))
r = abs(r)
vitesse = (r**(2./3.) * (s(io,jo))**(0.5)) / zmannig

chemin(numchem,numcell,3) = segment
chemin(numchem,numcell,4) = vitesse
if (vitesse.eq.0) then
chemin(numchem, numcell,5) =0
else
chemin(numchem,numcell,5) = segment/vitesse
endif
numcell = numcell + 1

```

```
RETURN
```

```
end
```

```
SUBROUTINE CHEMIN_MAX
```

```
include 'demiivar.f'
```

```
do 10 i= 1, numchem
  tps_cumul = 0
  numcell = jcell(i)
  do 10 j= 1, numcell
    if (tps_cumul.gt.tps_max) then
      tps_max = tps_cumul
      max_chemin = i
      max_i = is
      max_j = js
    endif
  tps_cumul = 0
```

```
10 continue
```

```
if (is.eq.nr.and.js.eq.nc) then
write (*,*) 'CHEMFILE'
open(9, file = CHEMfile, status = 'new')
```

```
write (9,*) ' '
write (9,*) '-----'
write (9,8) 'Temps de parcours maximal: ',tps_max,' sec'
write (9,9) 'sur le chemin #',max_chemin, max_i, max_j
```

```
close (9)
endif
```

```
8 format (A, F25.4, A)
9 format (A, I4, I4, I4)
```

```
end
```

```
SUBROUTINE SCA_AND_SDA
```

```
*|
*| This subroutine computes the flow width and obtains the
*| specific contributing areas by dividing the total contributing
*| areas by the flow width in the pixel.
*| INPUT: angle(), tca(), tda(), dx, dy
*| OUTPUT: sca(), sda()
*|
```

```
include 'demiivar.f'
```



```
write(26,*)'1 1 16.6000 XX Coordx MTM'  
write(26,*)'2 1 16.6000 YY Coordy MTM'  
write(26,*)'3 1 16.6000 SCA SCA'  
write(26,*)'4 1 16.6000 TCA TCA'  
write(26,*)'5 1 16.6000 SDA SDA'  
write(26,*)'DATA'  
  
do 40 i = 2, nrl  
do 40 j = 2, ncl  
40 write(26,88) qlat(i,j),qlon(i,j),sca(i,j),tca(i,j),sda(i,j)  
continue  
close(26)  
  
66 format (161F12.4)  
88 format (F14.6, F15.6, F12.4, F12.4, F12.4)  
  
end
```

ANNEXE VIII : ESTIMATION DU TEMPS DE CONCENTRATION DU BASSIN VERSANT DE LA RIVIÈRE SAMSON

Parmi les méthodes d'estimation du temps de concentration d'un bassin versant, la méthode empirique de Johnstone et Cross (Singh 1988) est retenue pour sa simplicité de mise en oeuvre et parce qu'elle convient au bassin versant de la rivière Samson qui fait un peu moins de trois cents kilomètres carrés de superficie. Elle provient de l'analyse de dix-neuf bassins versants entre 64 et 4202 km² (25 et 1624 miles²) dans la région des rivières Scoties et Sandusky en Ohio. La relation s'exprime par l'équation suivante :

$$T_c = \frac{4,7}{r^2} \left(\frac{L}{S} \right)^{0,5}$$

où T_c est le temps de concentration en heures, L est la longueur du cours d'eau principal en miles, S est la pente moyenne du cours d'eau principal en pieds par mile et r est le facteur d'embranchement du cours d'eau selon le schéma de rivière.

Il existe une bonne approximation de cette équation qui permet de négliger le facteur d'embranchement (r).

$$T_c = 5 \left(\frac{L}{S} \right)^{0,5}$$

Pour les bassin versant de la rivière Samson, les paramètres sont les suivants:

$L = 57.5$ miles (92.0 km)

$S = 55$ pieds par miles (environ 984 pieds sur 18 miles)

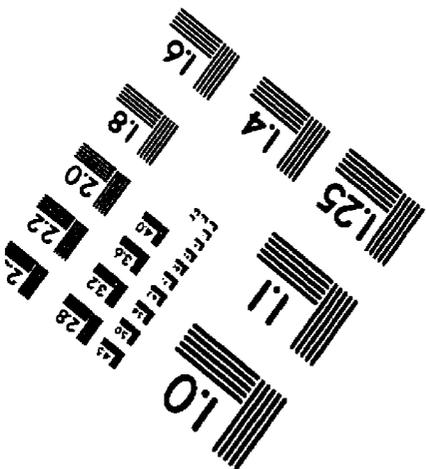
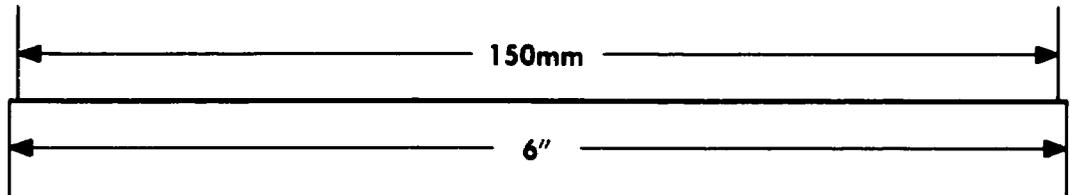
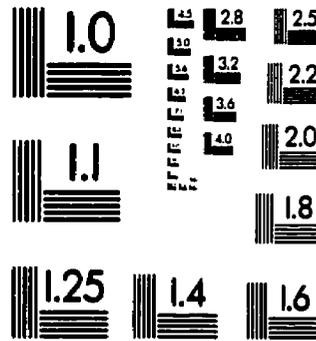
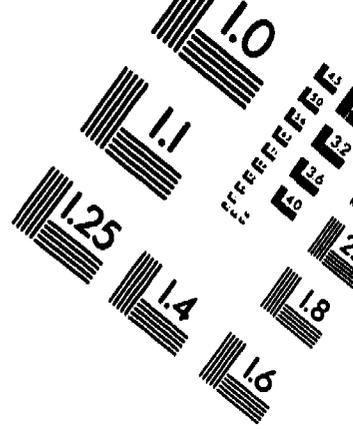
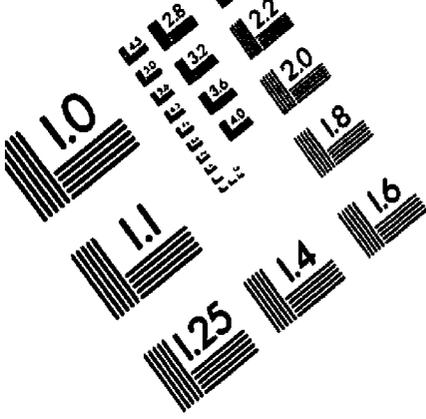
donc

$$T_c \approx 5,1 \text{ heures}$$

ANNEXE IX : EXEMPLE DE CALCUL MANUEL DE TEMPS DE CONCENTRATION

numéro	Coord X	Coord. Y	TCA	Pentes	R	V(Manning)	Temps	
							$TCA/100 \cdot 100 \ (R^{2/3} \cdot S^{1/2}) / 0,04$	$100/V$
1	291260.368	5048985	30	0	0.003	0	0	
2	291354.228	5049084.82	91	0	0.0091	0	0	
3	291260.584	5049085.02	135	0.05	0.0135	0.316942775	315.51437	
4	291159.136	5049085.24	116	0	0.0116	0	0	
5	291354.443	5049184.84	22	0.08	0.0022	0.119609807	836.051847	
6	291260.8	5049185.04	98	0.12	0.0098	0.396595713	252.145943	
7	291159.354	5049185.26	102	0.14	0.0102	0.43995083	227.298128	
8	291354.658	5049284.86	99	0.24	0.0099	0.564680026	177.091442	
9	291261.017	5049285.06	112	0.26	0.0112	0.63812442	156.709251	
10	291159.572	5049285.28	116	0.27	0.0116	0.665672399	150.224044	
11	291354.873	5049384.87	110	0.24	0.011	0.605769523	165.079285	
12	291261.233	5049385.08	158	0.24	0.0158	0.771170405	129.673026	
13	291159.79	5049385.3	158	0.26	0.0158	0.802659605	124.585814	
14	291355.088	5049484.89	107	0.07	0.0107	0.321177328	311.35448	
15	291261.449	5049485.09	125	0.07	0.0125	0.35625615	280.696908	
16	291160.008	5049485.31	24	0.01	0.0024	0.044814047	2231.44317	
17	291355.303	5049584.91	119	0.03	0.0119	0.2257002	443.065625	
18	291261.666	5049585.11	154	0.06	0.0154	0.379049663	263.817673	
19	291160.226	5049585.33	161	0.14	0.0161	0.596423313	167.666149	
20	291355.517	5049684.93	210	0.1	0.021	0.601754766	166.180653	
21	291261.882	5049685.13	109	0.15	0.0109	0.475996	210.085799	
22	291160.444	5049685.35	119	0.19	0.0119	0.567999714	176.056427	
23	291355.732	5049784.95	193	0.16	0.0193	0.719512273	138.983036	
24	291254.296	5049785.17	117	0.19	0.0117	0.56161761	178.057095	
25	291160.662	5049785.37	125	0.17	0.0125	0.555185112	180.120104	
26	291355.947	5049884.97	117	0.16	0.0117	0.515375665	194.033221	
27	291254.512	5049885.18	123	0.15	0.0123	0.515927938	193.825518	
28	291160.88	5049885.39	111	0.11	0.0111	0.412589657	242.371563	
29	291356.162	5049984.98	122	0.12	0.0122	0.458955437	217.886078	
30	291254.729	5049985.2	135	0.14	0.0135	0.530346702	188.555901	
31	291161.098	5049985.41	131	0.11	0.0131	0.460769667	217.028175	
32	291356.377	5050085	104	0.12	0.0104	0.412622454	242.352298	
33	291254.945	5050085.22	120	0.13	0.012	0.472460874	211.657738	
34	291161.316	5050085.42	122	0.09	0.0122	0.397467067	251.593171	
35	291356.592	5050185.02	130	0.11	0.013	0.45842179	218.139718	

Temps total	373785.1552	sec.
Temps total	103.8292098	heures



APPLIED IMAGE, Inc
 1653 East Main Street
 Rochester, NY 14609 USA
 Phone: 716/482-0300
 Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved

