

# Réduction du temps de calcul et du volume de stockage du modèle HYDROTEL

Rapport présenté à:

Direction de l'expertise hydrique du Ministère du Développement durable, de l'Environnement et de la Lutte contre les changements climatiques (MDDELCC)

Alain N. Rousseau, ing., Ph.D.  
Sébastien Tremblay

Centre Eau Terre Environnement  
Institut national de la recherche scientifique (INRS-ETE)  
490, rue de la Couronne, Québec (QC), G1K 9A9

Rapport d'étape No R-1833

Janvier 2019



© Alain N. Rousseau, 2019  
Tous droits réservés

ISBN : 978-2-89146-920-3 (version électronique)

Dépôt légal - Bibliothèque et Archives nationales du Québec, 2019  
Dépôt légal - Bibliothèque et Archives Canada, 2019



## Table des matières

---

1	Contexte .....	1
2	Approbation d'une version de départ.....	3
3	Sélection d'un bassin et d'une simulation de référence .....	5
4	Détermination des temps de calcul actuel .....	7
5	Analyse des pistes potentielles d'optimisation.....	11
5.1	Processus à prioriser (surlignés en vert) .....	11
5.2	Processus avec potentiel et/ou contrainte (surlignés en bleu) .....	12
6	Autres aspects à considérer : technologie multi-cœur .....	15
7	Conclusion.....	17



# 1 Contexte

---

Dans le contexte de la mise à jour 2020 de l'Atlas hydroclimatique du Québec méridional, le MDDELCC souhaite réduire les temps de calcul ainsi que le volume de stockage liés aux simulations produites par le modèle hydrologique HYDROTEL.

La version finale du code source doit permettre d'utiliser le format NetCDF pour la lecture des données météorologiques et pour la sauvegarde des résultats de simulation.

Tout au long du projet, après approbation d'une version de départ, les résultats de simulation devront autant que possible être identiques à ceux produits par la version de départ.





## 2 Approbation d'une version de départ

---

À la demande du *MDDELCC*, la version actuelle (4.1.4) d'*HYDROTEL* bonifiée d'une correction au module de calcul du bilan vertical a été sélectionnée pour ce projet.

La correction effectuée au module de calcul du bilan vertical des 3 couches (*BV3C*) concerne la variation maximale du contenu en eau relatif des sols. Auparavant, *HYDROTEL* tentait de satisfaire un critère de variation en réduisant le pas de temps interne de *BV3C*. Cependant, il pouvait arriver dans certains cas que ce pas de temps ne soit pas assez petit pour satisfaire le critère et les résultats obtenus étaient alors incohérents et pouvaient engendrer de l'instabilité numérique. Afin de limiter l'impact de la modification sur les résultats et ainsi ne pas invalider les bassins déjà calés par le *MDDELCC*, la solution suivante a été retenue : (i) dans un premier temps *HYDROTEL* détermine le pas de temps interne nécessaire pour satisfaire le critère de variation maximale du contenu en eau relatif; (ii) si le pas de temps satisfaisant le critère est inférieur au pas de temps déterminé selon l'ancienne méthode, on utilise le pas de temps satisfaisant le critère (version 4.1.4 bonifiée); (iii) sinon on utilise le pas de temps déterminé selon la méthode classique (version 4.1.4).



### 3 Sélection d'un bassin et d'une simulation de référence

---

Afin d'analyser l'impact des modifications sur les résultats produits par la version 4.1.4 bonifiée, on devait sélectionner un bassin de référence. Les résultats produits par la version de départ d'*HYDROTEL* avec le bassin de référence seront donc comparés avec les résultats obtenus durant la réalisation de ce projet afin d'assurer un minimum de cohérence dans les résultats.

Le bassin de référence retenu par le *MDDELCC* est le projet *HYDROTEL GASPESIE* (non-calé) qui sera utilisé dans *l'Atlas 2020*. Ce projet a 10 079 *UHRH*. Le pas de temps utilisé est de 3 heures. La résolution de la grille météo utilisée est de 0.1°.

La simulation a été effectuée du 1 janvier 2000 au 1 janvier 2003.



## 4 Détermination des temps de calcul actuel

---

Pour analyser les temps de calcul de la version actuelle (version de départ), l'outil de profilage de performance de *Microsoft Visual Studio 2017* a été utilisé. Cet outil permet de déterminer les temps d'exécution de tous les processus impliqués dans un programme. La simulation de référence a été exécutée pour le profilage.

L'analyse des résultats générés par l'outil a ensuite permis de générer le tableau présenté ici-bas. Ce tableau est une synthèse des temps d'exécution pour chacun des modèles utilisé dans la simulation et met en évidence les processus utilisant le plus de temps d'exécution. Les éléments surlignés en vert démontrent les processus qui seraient à prioriser pour l'optimisation et les éléments surlignés en bleu démontrent les processus ayant le potentiel d'être optimisées, mais sans en avoir la certitude ou connaître a priori l'impact sur les résultats.

## Réduction du temps de calcul et du volume de stockage du modèle HYDROTEL

PROCESSUS	Temps d'exécution			
	Total (%)	Total (MM:SS)	(%)	(MM:SS)
<b>INITIALISATION</b>	<b>7.02</b>	<b>1:10</b>		
Lecture données météo			6.53	1:05
Autres			0.49	0:05
<b>CALCULE</b>	<b>92.28</b>	<b>15:24</b>		
<u>Interpolation (Thiessen)</u>	24.08	4:02		
Répartition données			10.65	1:47
Interpolation données			13.21	2:13
Autres			0.22	0:02
<u>Fonte neige (Degré jour modifié)</u>	23.09	3:52		
Calcule			0.41	0:04
CalculIndiceRadiation			2.32	0:23
CalculeFonte			1.13	0:11
Sauvegarde (output)			18.97	3:11
Autres			0.26	0:03
<u>ETP (Hydro-Québec)</u>	0.74	0:07		
<u>Bilan vertical (BV3C)</u>	20.84	3:28		
TriCoucheOct97				
Calculs (opérations)			5.69	0:56
Fonction de puissance (pow)			7.62	1:15
Autres			1.25	0:13
CalculeEtr				
Fonction exp			1.75	0:18
boost::gregorian::date::day_of_year			1.25	0:13
Autres calculs			1.48	0:15
Autres			0.63	0:06

#### 4. Détermination des temps de calculs actuels

Autres				1.17	0:12
<u>Ruissellement (Onde cinématique)</u>	8.35	1:23			
Calculs				8.28	1:22
Autres				0.07	0:01
<u>Acheminement (Onde cinématique modifié)</u>	15.18	2:32			
TransfertRiviere					
Fonction de puissance (pow)				5.55	0:55
Autres calculs				1.29	0:13
PrendreTronconsAval (accès vecteur)				1.40	0:14
Sauvegarde (output)				4.85	0:49
Autres				2.09	0:21
<b>AUTRES</b>	<b>0.70</b>	<b>0:07</b>			
<b>TOTAL</b>	<b>100</b>	<b>16:41</b>			





## 5 Analyse des pistes potentielles d'optimisation

---

### 5.1 Processus à prioriser (surlignés en vert)

- **Lecture et écriture des données**

Pour la lecture et l'écriture des données, nécessitant les accès au disque dur, on peut considérer ensemble les items suivants : (i) lecture des données météorologiques, (ii) sauvegarde de la variable *Équivalent en eau du couvert nival* (modèle Degré jour modifié), et (iii) sauvegarde de la variable *Débit aval* (modèle Onde cinématique modifié). Ces trois processus ensemble représentent 30.4% du temps de simulation total (16:41), soit 5 minutes et 5 secondes.

L'utilisation d'un format de fichier binaire pourrait grandement améliorer les temps de lecture et d'écriture. Le format binaire *NetCDF* préconisé par le *MDDELCC* devrait répondre aux attentes. Ce format a l'avantage d'être compatible avec plusieurs logiciels existant afin de visualiser les données ou interagir avec ces logiciels. Un format binaire maison limiterait encore plus la quantité de données à lire/écrire, mais les gains au niveau de la vitesse d'exécution et du stockage ne seraient pas énorme par rapport au format *NetCDF*. Des tests pourraient être envisagés à ce propos si cela était souhaité si un format maison rencontrait les exigences du *MDDELCC*.

- **Répartition des données météorologiques**

Ce processus effectué dans le modèle *Thiessen* nécessite un temps de calcul de 10.7% du temps total ou 1 minute et 47 secondes.

L'algorithme de répartition des données semble inefficace principalement avec les valeurs de pondération, soit au niveau des 2 lignes de code visibles ici-bas :

```
0.00 % 121     size_t nbStation = stations_meteo.PrendreNbStation();
1.10 % 122     for (size_t index_station = 0; index_station < nbStation; ++index_station)
123     {
4.88 % 124         float ponderation = static_cast<float>(_ponderation(index_zone, index_station));
125
2.21 % 126         if (ponderation > 0.0f)
127         {
0.07 % 128             auto station_meteo = static_cast<STATION_METEO*>(stations_meteo[index_station]);
...

```

La ligne numéro 124 peut sans doute être optimisée en évitant de faire une conversion de type *static\_cast*. Il se peut aussi que l'accès au vecteur *\_ponderation* soit responsable de ce ralentissement principal. Il a été observé durant l'analyse que les accès au type de vecteur *Map* soit assez lent. Un autre type de vecteur serait sans doute plus performant.

La ligne 126 qui fait la vérification « *if(pondération > 0.0f)* » pourrait vraisemblablement être évitée en conservant seulement les pondérations supérieures à 0.

- **Interpolation des données météorologiques**

Ce processus non négligeable utilise 13.2% du temps total ou 2 minutes et 13 secondes. Le processus d'interpolation peut aussi être optimisé en modifiant l'algorithme dont le « mode d'accès » au vecteur à la ligne 262 représente presque la totalité du temps utilisé par ce processus.

```
0.01 % 260     for (size_t index = 0; index < zones.PrendreNbZone(); ++index)
261     {
13.13 % 262         if(find(begin(_sim_hyd.PrendreZonesSimules()), end(_sim_hyd.PrendreZonesSimules()),
263         {

```

## 5.2 Processus avec potentiel et/ou contrainte (surlignés en bleu)

- **Fonctions de puissance et exponentielle**

Les calculs des fonctions mathématiques telles les puissances et les exponentielles sont de façon générale reconnues pour être exigeantes en temps de calcul pour les processeurs. Ceux-ci représentent plus de 15% du temps total de la simulation, soit plus de 2 minutes et 30 secondes.

Il existe des algorithmes optimisés d'approximation de ces fonctions. La précision du résultat peut être ajustée à la décimale désirée. Ces algorithmes peuvent améliorer

énormément la rapidité des calculs, soit de l'ordre de 50% et plus. L'impact de la précision de ces calculs sur les variables interne d'*HYDROTEL* reste toutefois inconnu. Cette solution demeure une avenue potentielle en autant que l'impact sur les résultats soit acceptable autant tant du point de vue quantitatif que du point de vue scientifique.

- **Autres processus**

Les autres processus inscrits au tableau des temps d'exécution sont des calculs mathématiques ou des accès à certains vecteurs. C'est le cas pour les processus *TriCoucheOct97* (modèle *BV3C*), *CalculeEtr* (modèle *BV3C*), *TransfertRiviere* (modèle *Onde cinématique modifié*) et *PrendreTronconsAval* (modèle *Onde cinématique modifié*). Si on additionne les temps de calcul de ces processus, ils représentent tout de même 11.1% du temps total de simulation, soit 1 minute et 51 secondes.

Ces calculs peuvent sans doute être optimisés ainsi que les accès aux vecteurs, mais les gains sur la rapidité d'exécution devraient être assez limités.



## 6 Autres aspects à considérer : technologie multi-cœur

---

Il apparaît important dans le contexte de bien comprendre le fonctionnement de la technologie multi-cœur et l'état actuel d'*HYDROTEL* à ce niveau.

Les processeurs actuels fonctionnent majoritairement avec plusieurs cœurs. Cette gestion est effectuée automatiquement au niveau des processus par les ordinateurs. Par exemple, un ordinateur avec 4 cœurs, peut exécuter simultanément 4 processus (ex.: programmes), ce qui correspond à 100% des ressources processeurs disponibles. Si seulement 1 processus est exécuté, celui-ci utilisera 25% des ressources disponibles sans pouvoir en utiliser plus.

Présentement, *HYDROTEL* ne bénéficie pas de l'avantage fournie par les multiples cœurs. Même si le modèle utilise présentement la technologie *multi-threading* avec le modèle *BV3C*, il faut comprendre que ceci ne s'effectue pas au niveau des cœurs du processeur. Par exemple, si on lance un programme qui exécute 1000 *threads*, ces dernières utilisent seulement 1 cœur du processeur étant donné qu'ils sont « enfants » d'un seul programme. C'est le fonctionnement par défaut des programmes et ordinateurs actuels, c'est-à-dire que si on veut bénéficier de plusieurs cœurs on doit démarrer plusieurs programmes. C'est ce qui permet de lancer un processus exigeant sur un ordinateur tout en continuant à travailler sur ce même ordinateur sans qu'il n'y ait de ralentissement.

Donc son état actuel, si on veut utiliser 100% des processeurs disponibles sur un ordinateur ou un supercalculateur, on doit lancer simultanément autant de simulations qu'il y a de cœurs disponibles. Le pourcentage des ressources processeurs que nous utilisons sur un ordinateur sera toujours égal à l'équation suivante : *Nombre de simulations simultanées / Nombre de cœurs disponibles \* 100*.

Il est bien sûr possible, en utilisant certaines techniques de programmation, de faire en sorte que les *threads* d'un programme utilisent plusieurs cœurs. Pour l'instant, le modèle

*BV3C* pourrait en bénéficier. Le nombre de cœurs à utiliser pourrait par exemple être spécifié dans un fichier de paramètres.

Dans le contexte du présent projet, cette solution serait bénéfique seulement si le nombre de simulations simultanées est inférieur au nombre de cœurs disponibles. Le temps de développement requis afin d'implanter cette solution n'est également pas à négliger dans le cas où le *MDDELCC* voudrait explorer cette avenue.

## 7 Conclusion

---

En conclusion, il apparait clair que les efforts devraient être mis dans un premier temps sur la lecture des données météorologiques et l'écriture des variables de résultat en utilisant un format de fichier binaire. Également, les processus *Répartition* et *Interpolation* du modèle *Thiessen* sont assez gourmands en temps de calcul. À eux seuls, les processus de lecture/écriture et du modèle *Thiessen* représentent plus de la moitié du temps total de la simulation, soit 54% ou 9 minutes et 5 secondes. Conséquemment, l'utilisation d'un format de fichier binaire et l'optimisation des algorithmes du modèle *Thiessen* devraient donner un bon gain sur les temps de simulations actuels.

Dans le cas où l'utilisation de fonctions optimisées pour le calcul des valeurs de puissance et d'exponentielle peut être acceptable, un gain non négligeable pourrait également être obtenu à ce niveau.

Il deviendra plus difficile par la suite d'optimiser les algorithmes actuels afin d'obtenir des gains significatifs de temps de calcul, mais somme toute possible.