

**H2D2 : Développement de stratégies  
de résolutions non-stationnaires et  
non-linéaires**

Rapport de recherche R-1147    Mars 2010



**H2D2 : DÉVELOPPEMENT DE STRATÉGIES DE RÉOLUTIONS  
NON-STATIONNAIRES ET NON-LINÉAIRES**

par

Yves SECRETAN

Pascal MATTE

Institut national de la recherche scientifique, INRS-ETE

Québec (Québec), Canada

G1K 9A9

Rapport de recherche N° R-1147

Mars 2010

## Équipe de réalisation

Institut National de la Recherche Scientifique  
Centre – Eau Terre Environnement

Yves Secretan  
Pascal Matte

Ph.D., professeur  
Étudiant au doctorat

© INRS-ETE 2010

Pour fins de citations :

Secretan, Y. et Matte, P. (2010). *H2D2 : Développement de stratégies de résolutions non-stationnaires et non-linéaires*. Rapport de recherche R-1147, INRS-ETE, 31 pp.

## Table des matières

---

Liste des figures .....	v
Liste des tableaux.....	vii
1. Introduction.....	1
2. Éléments d’algorithmique.....	3
2.1. Algorithmes de Picard et de Newton .....	3
2.2. Backtracking .....	4
2.3. Limiteurs .....	5
2.4. Relaxation de la matrice tangente .....	5
2.5. Caractéristiques des solveurs .....	6
2.6. Stratégies de résolution et d’intégration temporelle .....	6
2.7. Formulation du couvrant-découvrant.....	8
3. Outils de visualisation.....	11
3.1. Traceurs de convergence.....	11
4. Protocole d’analyse.....	13
5. Résultats.....	15
5.1. Identification de stratégies de résolution .....	15
5.1.1. Impact local du limiteur .....	17
5.1.2. Impact local du backtracking.....	17
5.1.3. Impact local de la relaxation de la matrice tangente.....	21
5.2. Identification de stratégies d’intégration temporelle .....	21
5.2.1. Impact global de la formulation du couvrant-découvrant.....	22
5.2.2. Impact global de la bisection du pas de temps .....	24
6. Problèmes stationnaires .....	27
7. Conclusion .....	29
8. Références.....	31



## Liste des figures

---

Figure 1 – Définition du backtracking à l’intérieur du fichier de commande dans H2D2 .....	5
Figure 2 – Définition du limiteur à l’intérieur du fichier de commande dans H2D2 .....	5
Figure 3 – Diagramme représentant la séquence de nœuds formant une stratégie .....	7
Figure 4 – Écoulement à surface libre avec zones couvertes et découvertes.....	8
Figure 5 – Variation du coefficient de Manning $n$ en fonction de la profondeur $H$ pour la formulation CDMH .....	9
Figure 6 – Modèle de couvrant-découvrant dans H2D2, formulation CDYS .....	9
Figure 7 – Modèle de couvrant-découvrant dans H2D2, formulation CDY2.....	10
Figure 8 – Définition de la formulation à l’intérieur du fichier de commande dans H2D2 .....	10
Figure 9 – Définition du traceur à l’intérieur du fichier de commande dans H2D2.....	11
Figure 10 – Arborescence des deux stratégies de résolution développées .....	16
Figure 11 – Définition des stratégies de résolution à l’intérieur du fichier de commande dans H2D2 .....	16
Figure 12 – Traceur de convergence pour un limiteur à 10 m <sup>2</sup> /s en débits spécifiques et à 1 m en niveaux d’eau (haut) et pour un limiteur à 4 m <sup>2</sup> /s en débits spécifiques et à 0,3 m en niveaux d’eau (bas) .....	18
Figure 13 – Traceur de convergence pour un backtracking quadratique sur Newton avec $tmin$ égal à 0,001 (haut) et égal à 0,1 (bas); résultats obtenus avec la Stratégie #1 .....	19
Figure 14 – Traceur de convergence pour une simulation sans backtracking sur Picard (haut) et avec backtracking quadratique sur le premier bloc de Picard avec $tmin$ égal à 0,001 (bas); résultats obtenus avec la Stratégie #2.....	20
Figure 15 – Arborescence des deux stratégies d’intégration temporelle développées .....	22
Figure 16 – Définition des stratégies d’intégration temporelle à l’intérieur du fichier de commande dans H2D2 .....	22
Figure 17 – Traceur de convergence pour une simulation avec la formulation du couvrant- découvrant CDY2 (haut) et CDYS (bas).....	23
Figure 18 – Exemple d’une séquence de résolution en stationnaire avec changement de propriétés globales à l’intérieur du fichier de commande dans H2D2 .....	27





## Liste des tableaux

---

Tableau 1 – Statistiques globales représentées par classe de nombre d'itérations nécessaires à la convergence.....	25
--	----



# 1. Introduction

---

Ce rapport rassemble les résultats de tests effectués dans H2D2 visant à évaluer la performance de différentes stratégies de résolution en non-stationnaire et leur impact sur la convergence de la solution et les temps de calcul globaux. Différentes stratégies de résolution et d'intégration temporelle ont été identifiées lors des tests et sont proposées comme point de départ d'une simulation en non-stationnaire, pouvant être raffinée en fonction du type de problème traité.

Le rapport présente d'abord quelques éléments d'algorithmique et outils de visualisation à disposition dans H2D2. Il introduit ensuite le protocole d'analyse suivi lors des tests, puis présente les résultats obtenus. Une brève discussion sur les problèmes stationnaires, suivie d'une conclusion, viennent clore le rapport.

Pour une documentation plus détaillée des outils et algorithmes à disposition dans H2D2, se référer au site du Groupe de Recherche et d'Étude en Éco-Hydraulique Numérique de l'INRS (<http://www.gre-ehn.inrs-ete.quebec.ca/>).



## 2. Éléments d'algorithmique

---

### 2.1. Algorithmes de Picard et de Newton

Le système non-stationnaire du premier ordre à résoudre peut s'écrire sous la forme matricielle suivante (Dhatt et al. 2005) :

$$[C]\{\dot{U}\} + [K]\{U\} = \{F\} \quad (1)$$

Pour un schéma implicite, on exprime  $\{\dot{U}\}$  par différences finies centrées à droite :

$$\{\dot{U}_{t+\Delta t}\} \approx \frac{1}{\Delta t} (\{U_{t+\Delta t}\} - \{U_t\}) \quad (2)$$

Le système à résoudre peut être réécrit comme suit :

$$([C] + \Delta t[K])\{U_{t+\Delta t}\} = \Delta t\{F_{t+\Delta t}\} + [C]\{U_t\} = \{R_{t+\Delta t}\} \quad (3)$$

Sous forme incrémentale, on a :

$$([C] + \Delta t[K])\{\Delta U\} = \Delta t(\{F_{t+\Delta t}\} - [K]\{U_t\}) \quad (4)$$

$$\{U_{t+\Delta t}\} = \{U_t\} + \{\Delta U\} \quad (5)$$

Pour des problèmes non-linéaires, la résolution du système matriciel ce fait de façon itérative, c'est-à-dire par une résolution successive du système suivant :

$$[K_{nl}]\{\Delta U^i\} = \{R_{nl}\} \quad (6)$$

$$\{U_{t+\Delta t}^i\} = \{U_{t+\Delta t}^{i-1}\} + \{\Delta U^i\} \quad (7)$$

où :

$$\{R_{nl}\} = \Delta t (\{F_{t+\Delta t}\} - [K(U_{t+\Delta t}^{i-1})]\{U_{t+\Delta t}^{i-1}\}) + [C](\{U_t\} - \{U_{t+\Delta t}^{i-1}\}) \quad (8)$$

La forme de la matrice  $[K_{nl}]$  (équation (6)) est fonction de l'algorithme de résolution employé. Pour l'algorithme de Picard (aussi appelé méthode de substitution), sa forme est la suivante :

$$[K_{nl}] = [C] + \Delta t [K(U_{t+\Delta t}^{i-1})] \quad (9)$$

Pour l'algorithme de Newton-Raphson, sa forme est la suivante :

$$[K_{nl}] = [K_t] = [C] + \Delta t [K(U_{t+\Delta t}^{i-1})] + \Delta t \left[ \frac{\partial K}{\partial U} \cdot U \right]_{t+\Delta t}^{i-1} \quad (10)$$

## 2.2. Backtracking

Le backtracking est un algorithme de globalisation implanté dans H2D2 sous trois types : linéaire, quadratique et cubique. Il effectue une recherche du résidu minimum le long de la direction de descente représentée par l'incrément de solution. La mise à jour de la solution à chaque itération (équation (7)) est remplacée par :

$$\{U_{t+\Delta t}^i\} = \{U_{t+\Delta t}^{i-1}\} + a \{\Delta U^i\} \quad (11)$$

où le facteur  $a$  est choisi de façon à minimiser le résidu  $R(u + t \Delta u)$ .

En fonction du type de backtracking, la recherche du résidu minimum sera linéaire avec  $n$  pas réguliers entre  $tmin$  et  $tmax$ , ou basée sur une approximation polynomiale du second ou du

troisième ordre. L'algorithme cherche une valeur de  $t$ , comprise entre  $tmin$  et  $tmax$ , qui minimise le résidu. La Figure 1 présente la syntaxe de H2D2 pour la définition du backtracking.

```

npas = 20
tmin = 0.001
tmax = 1
h_btrk1 = backtracking (npas, tmin, tmax) #backtracking linéaire
h_btrk2 = backtracking_2(tmin, tmax)      #backtracking quadratique
h_btrk3 = backtracking_3(tmin, tmax)      #backtracking cubique

```

**Figure 1** – Définition du backtracking à l'intérieur du fichier de commande dans H2D2

Le backtracking minimise le résidu localement, dans la direction de descente, mais n'a pas une vision globale de l'espace solution. Il s'agit d'un outil à la fois très puissant d'aide à la convergence, mais peut aussi diriger le système vers un minimum local et l'y maintenir.

### 2.3. Limiteurs

À chaque itération, les incréments sur la solution calculés pour chaque degré de liberté peuvent être limités, afin d'empêcher la solution d'effectuer de trop grands sauts et possiblement de diverger :

$$u^{i+1} = u^i + \text{sign} [\Delta u, \min (\Delta u, \Delta u_{\max})] \quad (12)$$

La Figure 2 présente la syntaxe de H2D2 pour la définition du limiteur.

```

h_lim = limiter(h_sol, 1.0e+01, 1.0e+01, 1.0e-00)

```

**Figure 2** – Définition du limiteur à l'intérieur du fichier de commande dans H2D2

### 2.4. Relaxation de la matrice tangente

La matrice tangente  $[K_t]$  peut être relaxée au moyen de la propriété globale de relaxation. Cette propriété consiste en un facteur multiplicatif placé devant le dernier membre de droite de l'équation (10). Sa valeur est comprise entre 0 et 1, 0 correspondant à l'algorithme de Picard et 1 correspondant à l'algorithme de Newton-Raphson.

## 2.5. Caractéristiques des solveurs

L'algorithme de Newton converge de façon quadratique lorsque le système est assez proche de la solution mais peut s'avérer très inefficace si la solution courante est loin de la solution recherchée. L'algorithme de Picard, quant à lui, résout le système de façon linéaire. Il est moins coûteux mais ne peut à lui seul fournir une solution adéquate au problème non-linéaire à résoudre. Son rôle est plutôt d'abaisser le résidu de façon à faciliter la convergence avec Newton. Par exemple, lorsque la solution initiale est relativement loin de la solution recherchée, un certain nombre d'itérations de Picard peut permettre d'abaisser le résidu au tout début de la résolution. Dans le cas où l'algorithme de Newton, combiné au backtracking, maintiennent la solution autour d'un minimum local, des itérations de Picard peuvent également s'avérer utiles pour rediriger les corrections sur la solution vers le minimum global et ainsi empêcher une stagnation de la convergence.

Une combinaison des algorithmes de Picard, de Newton et de backtracking à l'intérieur de stratégies de résolution peut donc permettre de limiter les incréments sur la solution risquant de faire diverger le système et ainsi assurer une convergence pour un maximum de pas de temps, en fournissant un meilleur contrôle sur la descente du résidu.

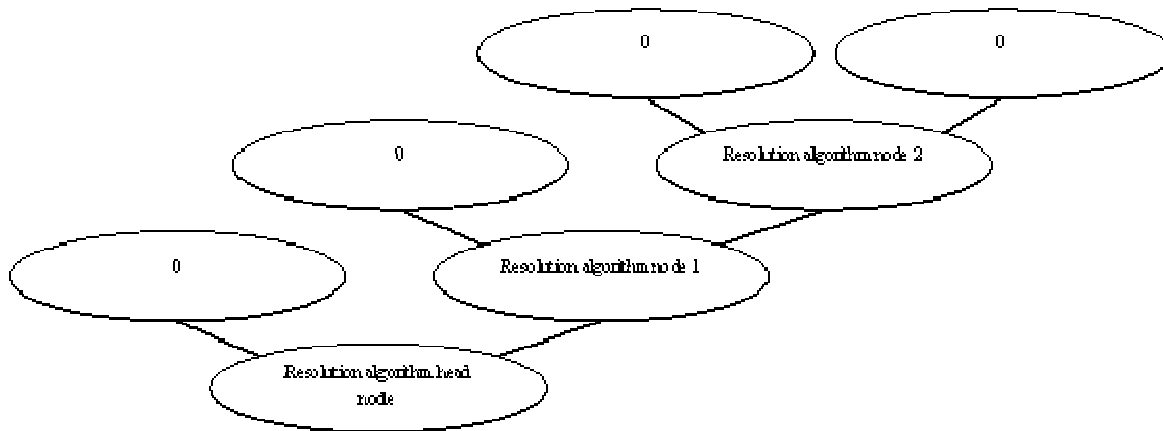
## 2.6. Stratégies de résolution et d'intégration temporelle

Différentes stratégies de résolution et d'intégration temporelle peuvent être développées dans H2D2 permettant un contrôle optimisé de la convergence en cours de résolution. Les stratégies permettent de combiner les algorithmes de résolution et d'intégration temporelle à l'intérieur d'un arbre décisionnel. À chaque nœud de l'arbre sont associés un algorithme et une expression à évaluer, laquelle détermine la prochaine branche à emprunter. Certains nœuds ont des fonctions particulières, comme le nœud *for*, permettant l'exécution d'algorithmes en boucle, et le nœud de bisection du pas de temps, permettant de diviser le pas de temps de façon à réduire les écarts entre la solution du pas précédent et la solution recherchée.



La Figure 3 présente sous forme schématisée la manière dont est définie une stratégie dans H2D2. Ce diagramme doit se lire de bas en haut; sa structure est très similaire à une structure « if-else-endif ». Deux principaux points caractérisent cette structure :

- Les algorithmes et les stratégies sont interchangeables : le nœud d'une stratégie peut avoir comme algorithme de résolution une autre stratégie;
- Un algorithme peut être vu comme une stratégie avec un seul nœud, sans expression ni nœud fille.



**Figure 3** – Diagramme représentant la séquence de nœuds formant une stratégie

La stratégie de résolution est définie par son nœud de tête. L'algorithme contenu dans le nœud de tête est exécuté. Si le résultat est convergé, le processus s'arrête à ce niveau (branche de gauche); sinon, le second algorithme contenu dans le nœud de droite (branche de droite) est exécuté en utilisant la stratégie ou la méthode définie sur ce nœud; et ainsi de suite, jusqu'à ce que le processus s'arrête, au dernier nœud, indépendamment du résultat de convergence.

À la fin de chaque nœud d'une stratégie, une réinitialisation de la solution est possible. Cette option est particulièrement utile pour des résolutions non-stationnaires lorsqu'utilisée à l'intérieur d'une stratégie d'intégration temporelle. Elle donne ainsi la possibilité de repartir de la solution initiale dans le cas où une stratégie de résolution s'avère inefficace à un pas de temps donné.

## 2.7. Formulation du couvrant-découvrant

La Figure 4 présente le modèle de couvrant-découvrant implanté dans H2D2, pour lequel aucune limite n'est imposée sur la position de la surface libre, qui peut plonger sous le lit de la rivière et générer des profondeurs autant négatives que positives.

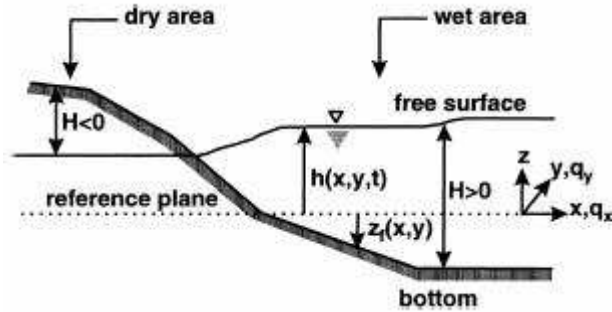
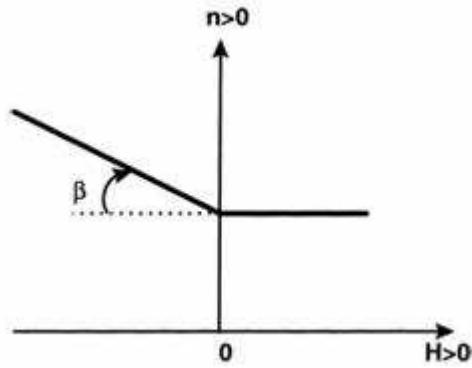


Figure 4 – Écoulement à surface libre avec zones couvertes et découvertes

Trois formulations du couvrant-découvrant sont disponibles dans H2D2. La première, CDMH, est celle employée dans Hydrosim (Heniche et al. 2000) où le coefficient de friction de Manning  $n$  est augmenté linéairement dans la zone découverte proportionnellement à la profondeur d'eau (négative), dans le but de geler l'écoulement dans cette région (voir Figure 5). Le frottement de Manning s'exprime comme suit, en zones couvertes et découvertes, respectivement :

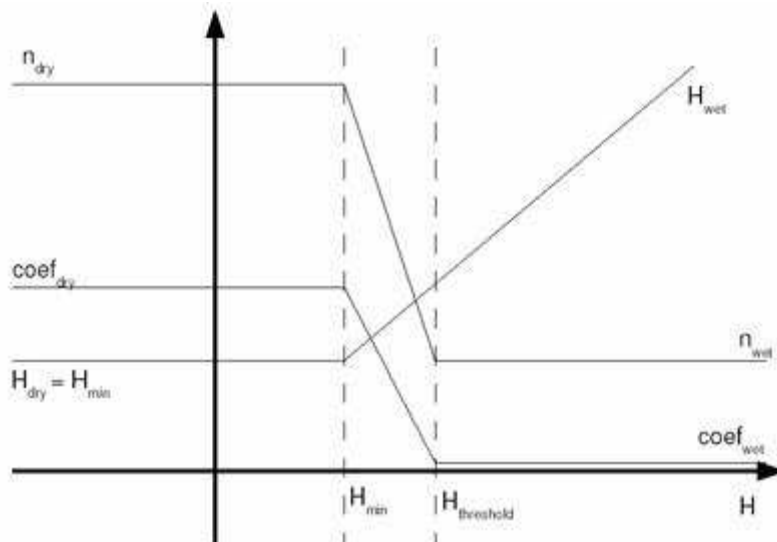
$$n_{H \geq 0} = n \quad (13)$$

$$n_{H < 0} = n(1 + \beta|H|) \quad (14)$$



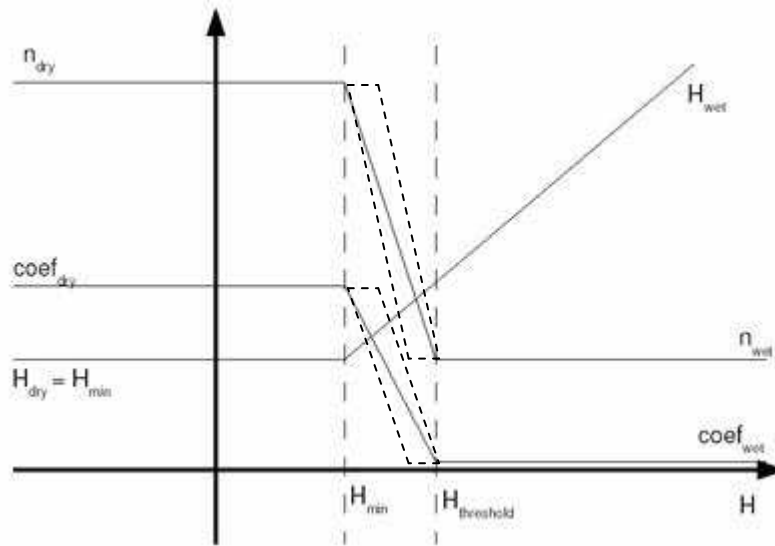
**Figure 5** – Variation du coefficient de Manning  $n$  en fonction de la profondeur  $H$  pour la formulation CDMH

La seconde formulation, CDYS, pose une limite à la profondeur minimale admissible et introduit une zone de profondeurs, délimitée par une profondeur seuil  $H_{threshold}$  et la profondeur minimale  $H_{min}$ , à l'intérieur de laquelle tous les paramètres varient linéairement de leur valeur en zone couverte à leur valeur en zone découverte (voir Figure 6).



**Figure 6** – Modèle de couvrant-découvrant dans H2D2, formulation CDYS

La troisième formulation, CDY2, reprend la formulation CDYS en la représentant sous forme d'une boucle d'hystérèse dans laquelle le passage d'un nœud de l'état couvert à découvert se fait différemment du passage de l'état découvert à couvert. La Figure 7 illustre cette différence.



**Figure 7** – Modèle de couvrant-déouvrant dans H2D2, formulation CDY2

Les trois formulations portent, dans l'ordre, les appellations suivantes dans H2D2; leur syntaxe dans le fichier de commande de H2D2 est présentée à la Figure 8 :

- SV2D\_Conservatif\_CDMH\_NN
- SV2D\_Conservatif\_CDYS\_NN
- SV2D\_Conservatif\_CDY2\_NN

```
h_frm = form('SV2D_Conservatif_CDY2_NN')
```

**Figure 8** – Définition de la formulation à l'intérieur du fichier de commande dans H2D2

### 3. Outils de visualisation

---

#### 3.1. Traceurs de convergence

L'outil de traceur permet à l'utilisateur de suivre la progression et le comportement de la résolution, pendant la simulation aussi bien qu'une fois terminée. Pour ce faire, il extrait l'information synthèse contenu dans le fichier de trace généré par H2D2 au moyen de la commande présentée à la Figure 9. L'outil de visualisation porte en graphique la norme associée au critère d'arrêt en fonction du temps écoulé (se référer à la section 5 pour des exemples de tracés).

```
h_tra = tracer()  
h_tra.set_file(sim_dir+'tracer.log')
```

**Figure 9** – Définition du traceur à l'intérieur du fichier de commande dans H2D2

Deux outils de traces sont inclus dans H2D2 :

- CTTracer : porte en graphique l'historique de convergence d'une simulation avec H2D2. Lorsque la simulation tourne toujours, la figure s'actualise à mesure que progresse le calcul.
- CTCompare : porte en graphique, en parallèle, l'historique de convergence de deux simulations avec H2D2.

Le traceur de convergence permet également de sortir un tableau de statistiques globales pour une simulation donnée, regroupant par classes de 5 itérations le nombre de pas de temps convergés, le nombre total d'itérations effectuées et des statistiques sur la proportion cumulée de pas de temps et d'itérations réalisés et sur le nombre moyen d'itérations par pas de temps (se référer au Tableau 1 pour un exemple de ce type de résumé).



## 4. Protocole d'analyse

---

Les tests ont été réalisés à partir de deux modèles numériques de terrain de l'estuaire fluvial du Saint-Laurent, le premier couvrant la portion Grondines – Île-aux-Grues et le second couvrant le secteur Trois-Rivières – Île-aux-Grues. Un scénario de crue et de marée de vive-eau correspondant à la période du 22 au 23 mai 1974 a été simulé. Un schéma temporel d'Euler implicite avec un pas de temps global de 5 minutes a été utilisé. Les solveurs matriciels *mkl\_pardiso* et *mumps* ont été employés.

Afin de tester la performance de différentes stratégies de résolution, une série de tests combinant les différents outils et algorithmes à disposition dans H2D2 a été réalisée sur des pas de temps problématiques à partir desquels une divergence de la solution était observée. Les stratégies de résolution s'avérant les plus efficaces pour ces pas de temps ont ensuite été incluses à l'intérieur de stratégies d'intégration temporelle, puis testées sur une période de simulation de 24 heures afin de comparer leur comportement global en termes de convergence et de temps de calcul.



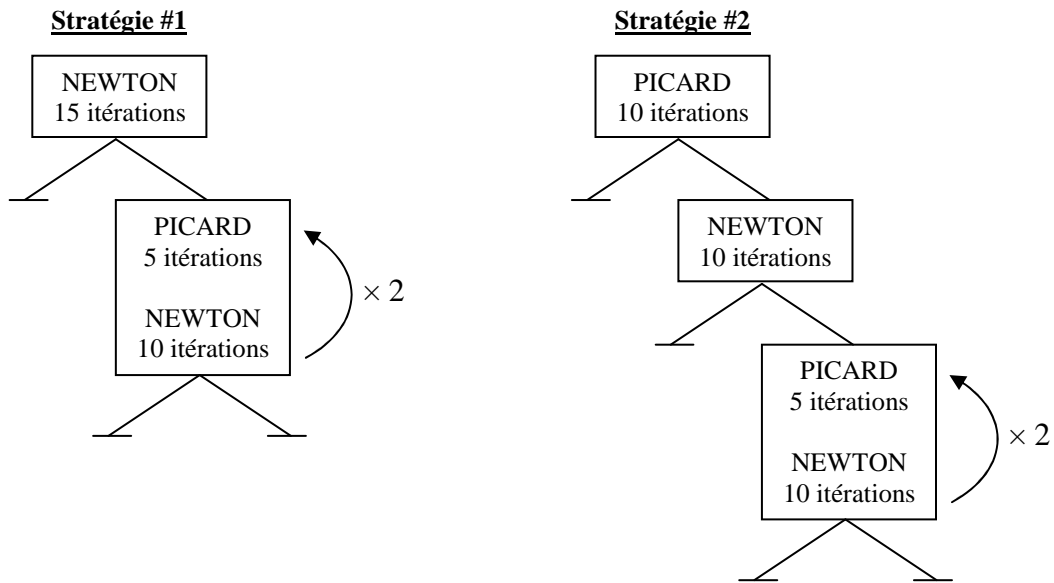


## 5. Résultats

---

### 5.1. Identification de stratégies de résolution

Parmi les différentes combinaisons d'algorithmes testées, deux stratégies de résolution se sont avérées très efficaces pour les pas de temps dont la convergence était plus problématique. Ces stratégies comprennent une combinaison des algorithmes de Picard et de Newton avec pour chacun un nombre d'itérations distinct. Une arborescence des deux stratégies développées est présentée à la Figure 10. À chaque nœud des stratégies, la branche de gauche est empruntée en cas de convergence alors que la branche de droite est empruntée dans le cas non-convergé. Aucune réinitialisation n'est effectuée à la fin de chaque nœud; la solution progresse d'un nœud à l'autre de la stratégie. La première stratégie comprend 15 itérations de Newton suivies d'une boucle de 5 itérations de Picard et de 10 itérations de Newton, répétée à deux reprises. La seconde stratégie débute avec 10 itérations de Picard, enchaîne avec 10 itérations de Newton, puis termine par une boucle de 5 itérations de Picard et de 10 itérations de Newton, répétée à deux reprises. La première stratégie demeure généralement très efficace lorsque la solution varie peu d'un pas de temps à l'autre. La seconde stratégie permet quant à elle d'abaisser le résidu au tout début de la résolution et peut s'avérer efficace lorsque la solution au pas de temps précédent est relativement loin de celle du pas de temps courant. Les itérations de Picard à l'intérieur du bloc Picard-Newton ont pour but d'empêcher la résolution de s'effectuer autour d'un minimum local qui pourrait faire stagner la convergence. Elles permettent de rediriger Newton vers la recherche du résidu minimum global.



**Figure 10** – Arborescence des deux stratégies de résolution développées

La Figure 11 présente en résumé les blocs du fichier de commande dans H2D2 correspondant à la définition des deux stratégies de la Figure 10, incluant la définition du backtracking, du solveur matriciel et des différents algorithmes de résolution employés à chacun des nœuds des stratégies.

```
#Backtracking
h_btrk0 = 0
h_btrk2 = backtracking_2(0.001, 1)

#Solveur matriciel
h_matslv = mumps_linear_solver()

#Algorithmes de résolution
h_pcrd05 = picard(h_btrk0, h_matslv, 5)
h_pcrd10_bktr = picard(h_btrk2, h_matslv, 10)
h_nwtn10 = newton(h_btrk2, h_matslv, 10)
h_nwtn15 = newton(h_btrk2, h_matslv, 15)

#Stratégies de résolution
h_node2 = algo_node (h_pcrd05, 0, 0, h_nwtn10, 0)
h_node1 = algo_node_for(h_node2, 2, 0, 0, 0, 0)
h_node0 = algo_node (h_nwtn15, 0, 0, h_node1, 0)
h_stgy1 = h_node0

h_node4 = algo_node (h_nwtn10, 0, 0, h_node1, 0)
h_node3 = algo_node (h_pcrd10_bktr, 0, 0, h_node4, 0)
h_stgy2 = h_node3
```

**Figure 11** – Définition des stratégies de résolution à l'intérieur du fichier de commande dans H2D2

### 5.1.1. Impact local du limiteur

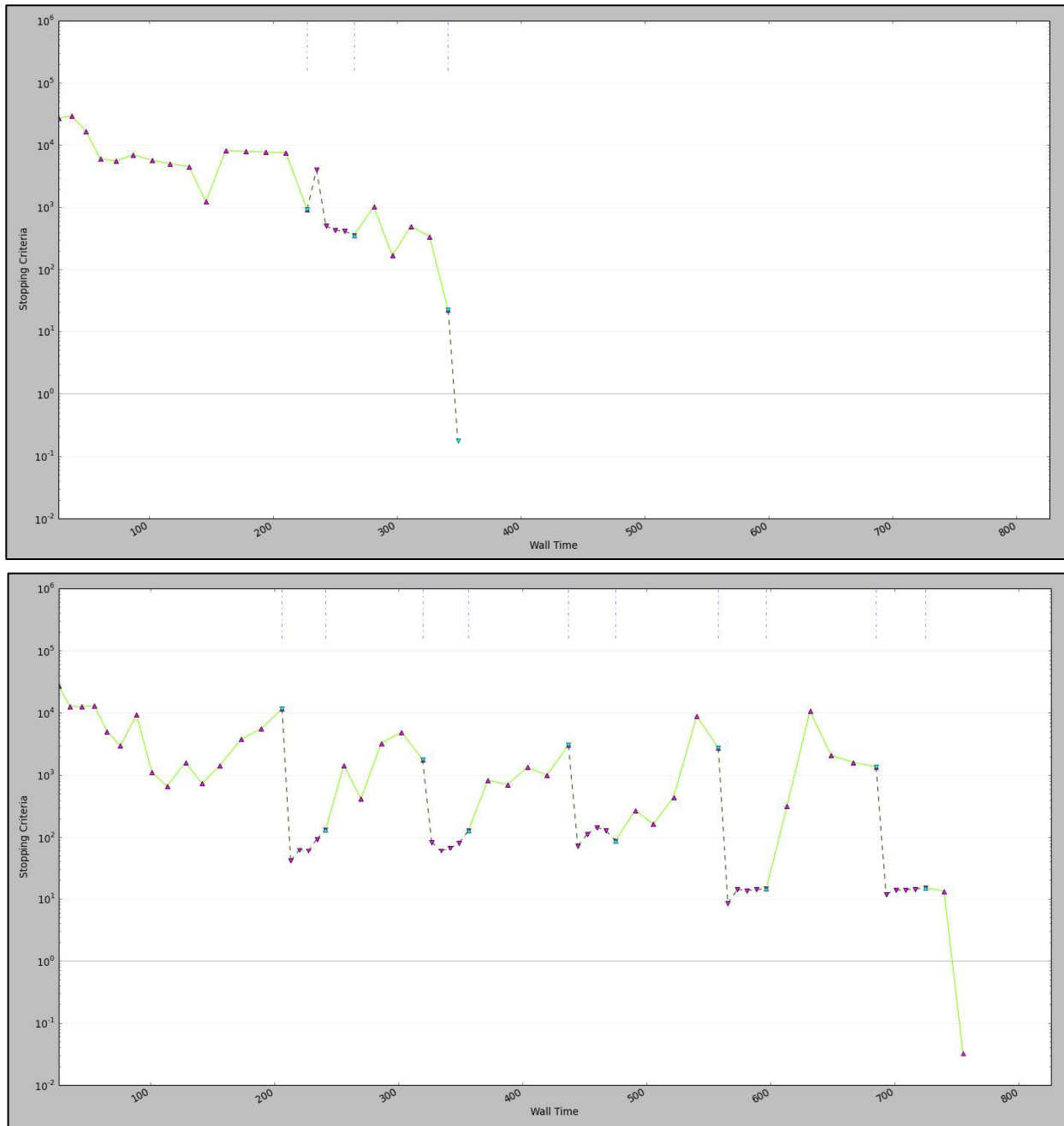
Les tests effectués sur quelques pas de temps ont permis d'observer l'impact du limiteur sur la convergence des pas de temps concernés. Deux cas ont été testés, le premier avec une limite fixée à 10 m<sup>2</sup>/s en débits spécifiques et à 1 m en niveaux d'eau, et le second avec une limite fixée à 4 m<sup>2</sup>/s en débits spécifiques et à 0,3 m en niveaux d'eau. Les résultats montrent dans ce cas précis qu'un limiteur trop restrictif peut ralentir considérablement la convergence. La Figure 12 illustre ce point; en abscisse se trouve le temps écoulé et en ordonnée la norme associée au critère d'arrêt. La solution est jugée convergée lorsque la norme est inférieure à 1.

### 5.1.2. Impact local du backtracking

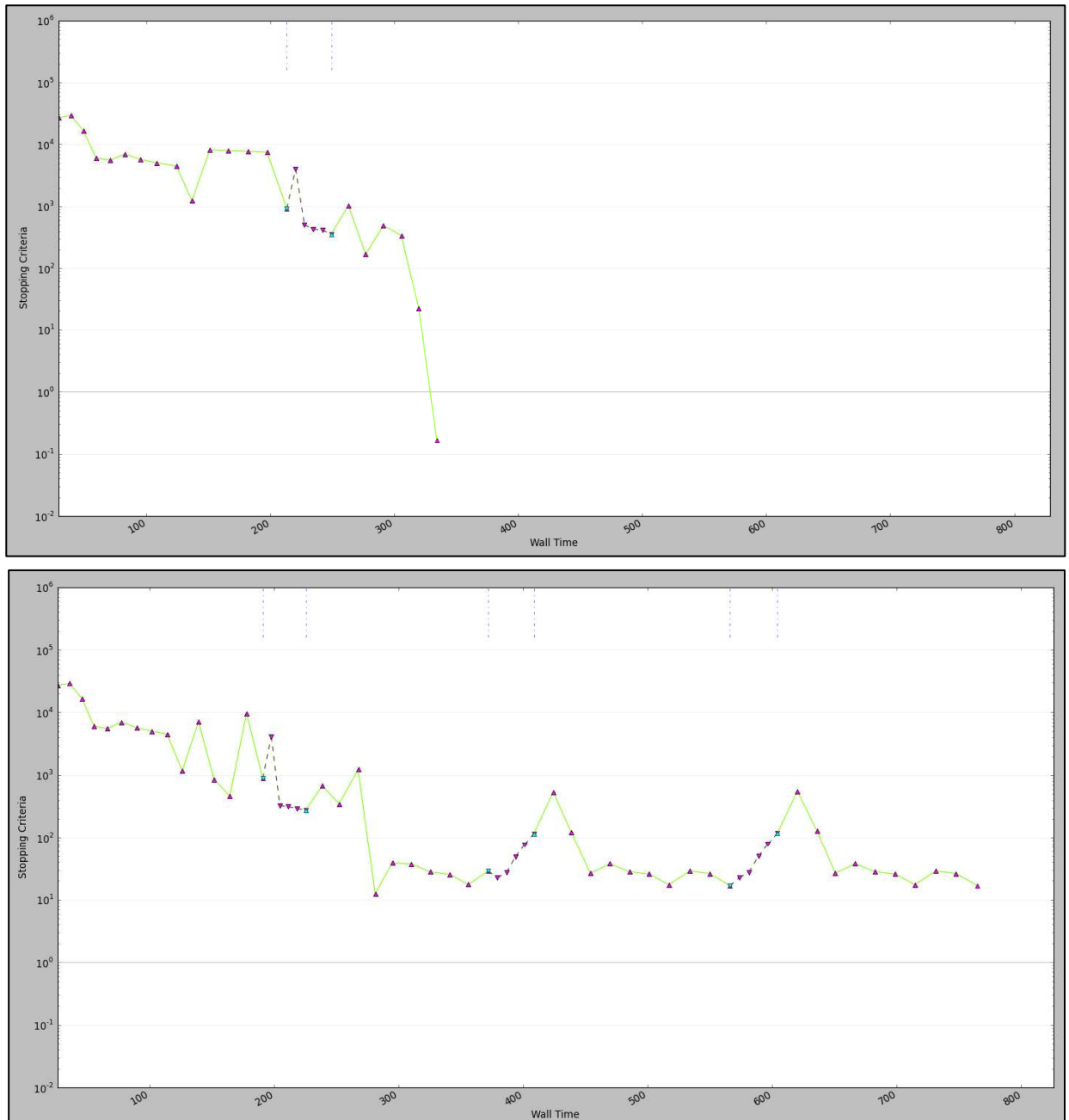
Le backtracking sur Newton est nécessaire puisqu'il empêche la solution de diverger et permet d'avoir un meilleur contrôle sur la descente du résidu. Pour la Stratégie #1, un backtracking quadratique avec *tmin* égal à 0,001 et 0,1 ont été testés (voir Figure 13). Les résultats montrent qu'en forçant un incrément de solution trop élevé la convergence peut s'en trouver affectée. Le meilleur comportement a été obtenu avec un *tmin* égal à 0,001, soit la valeur minimale par défaut dans H2D2.

Des tests similaires ont été réalisés à partir de la Stratégie #2 en imposant un backtracking quadratique sur le premier bloc d'itérations de Picard, avec *tmin* égal à 0,001 (voir Figure 14). Les résultats montrent que la convergence globale du pas de temps testé est beaucoup plus rapide sans backtracking sur Picard. Cependant, en présence de backtracking, une descente plus rapide du résidu est observée sur les 10 premières itérations de Picard, que l'algorithme de Newton n'arrive pas à récupérer au bloc suivant.

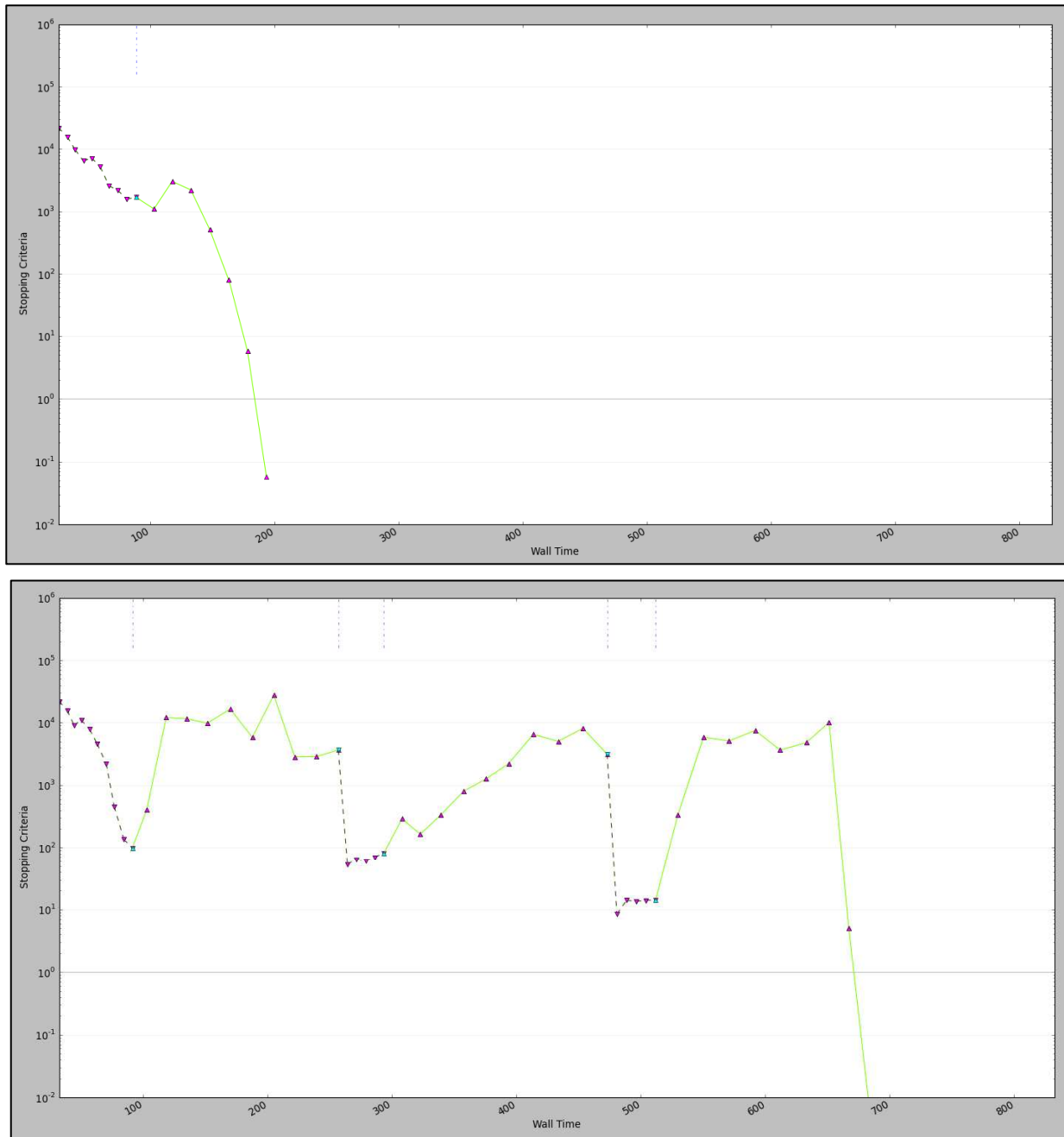
Le backtracking sur Picard à l'intérieur des blocs Picard-Newton s'avère inefficace puisqu'il limite les corrections sur la solution à une fraction de l'incrément total calculé. Le rôle de l'algorithme de Picard à l'intérieur de ces blocs est précisément de sortir le système d'un éventuel minimum local en effectuant des corrections suffisamment grandes sur la solution.



**Figure 12** – Traceur de convergence pour un limiteur à  $10 \text{ m}^2/\text{s}$  en débits spécifiques et à  $1 \text{ m}$  en niveaux d'eau (haut) et pour un limiteur à  $4 \text{ m}^2/\text{s}$  en débits spécifiques et à  $0,3 \text{ m}$  en niveaux d'eau (bas)



**Figure 13** – Traceur de convergence pour un backtracking quadratique sur Newton avec  $tmin$  égal à 0,001 (haut) et égal à 0,1 (bas); résultats obtenus avec la Stratégie #1



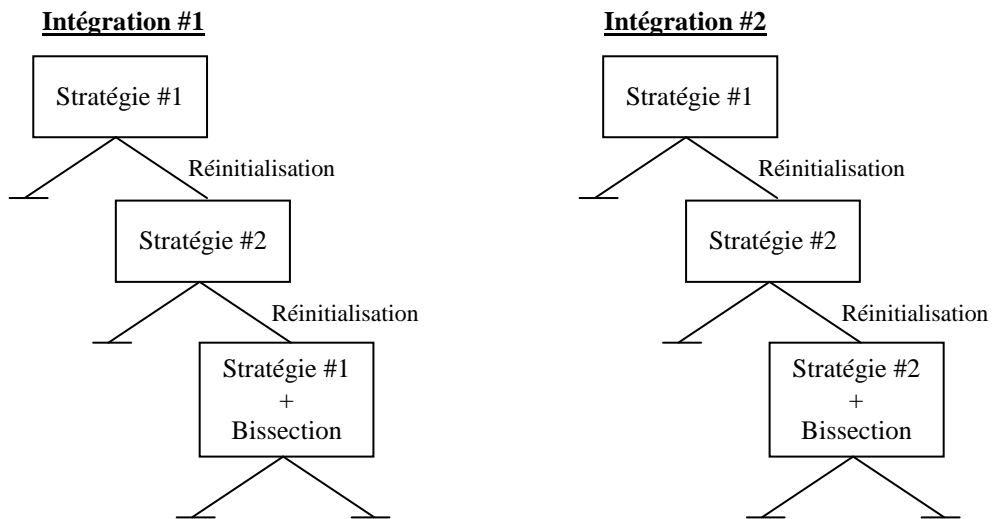
**Figure 14** – Traceur de convergence pour une simulation sans backtracking sur Picard (haut) et avec backtracking quadratique sur le premier bloc de Picard avec  $t_{min}$  égal à 0,001 (bas); résultats obtenus avec la Stratégie #2

### 5.1.3. Impact local de la relaxation de la matrice tangente

Les tests réalisés pour différentes valeurs (entre 0 et 1) de la propriété de relaxation de la matrice tangente n'ont mené à aucune amélioration de la convergence pour les pas de temps problématiques testés.

## 5.2. Identification de stratégies d'intégration temporelle

Deux stratégies d'intégration temporelle ont été élaborées en combinant les deux stratégies de résolution développées à la section 5.1. La Figure 15 en présente l'arborescence. Pour chaque stratégie, on retrouve trois nœuds faisant appel à l'une ou l'autre des stratégies de résolution de la section 5.1. À la fin de chaque nœud, une réinitialisation de la solution est effectuée si cette dernière n'est pas convergée. Le dernier nœud des stratégies d'intégration temporelle consiste en une bisection du pas de temps qui comprend deux niveaux : une première division par deux du pas de temps est réalisée, suivie d'une division par quatre si le dernier sous-pas de temps n'arrive pas à convergence. Le nœud de bisection est basé sur la Stratégie #1 dans le cas de la première stratégie d'intégration temporelle, alors qu'il est basé sur la Stratégie #2 dans le cas de la seconde. La bisection du pas de temps est la solution de dernier recours lorsqu'aucune autre stratégie n'a mené le système à convergence. Elle permet d'effectuer de plus petits sauts dans le temps et ainsi de réduire les écarts entre la solution du pas précédent et la solution recherchée. Toutefois, ce processus augmente considérablement la charge de calcul.



**Figure 15** – Arborecence des deux stratégies d'intégration temporelle développées

La Figure 16 présente la portion du fichier de commande dans H2D2 qui permet de définir les stratégies d'intégration temporelle décrites à la Figure 15.

```
#Algorithmes d'intégration temporelle
delt = 5*60                                # 5 min
h_euler2 = euler(h_stgy1, delt/2, 1.0)     # Implicite
h_euler1 = euler(h_stgy2, delt, 1.0)      # Implicite
h_euler0 = euler(h_stgy1, delt, 1.0)      # Implicite

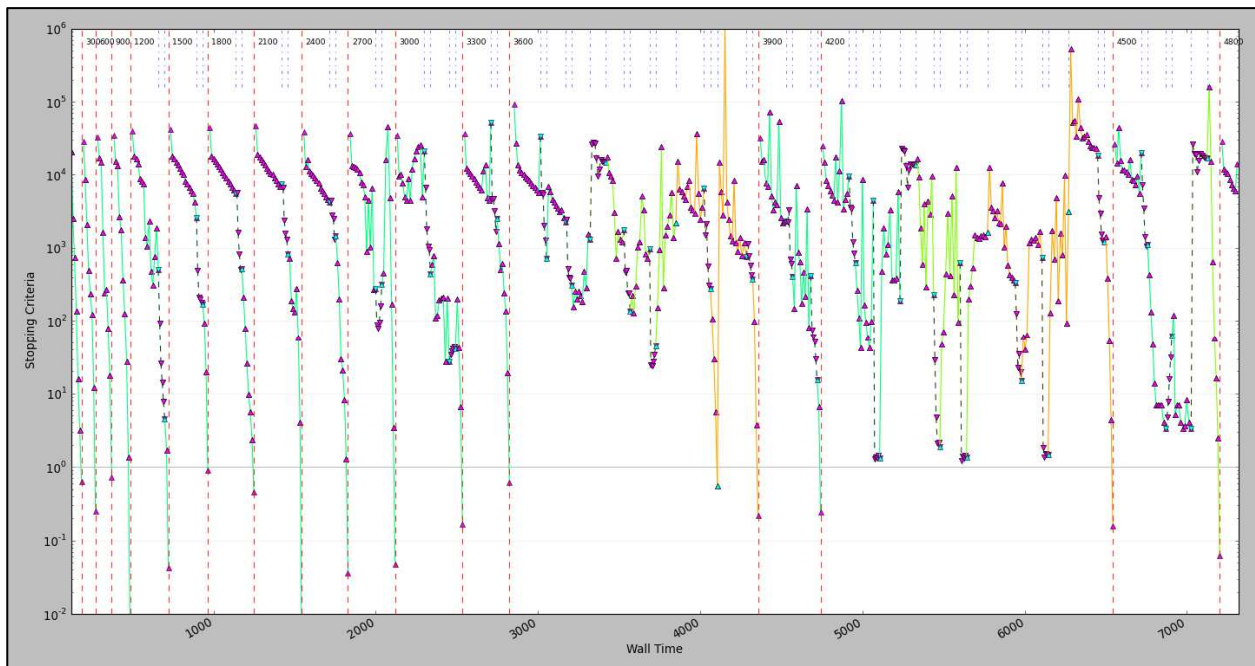
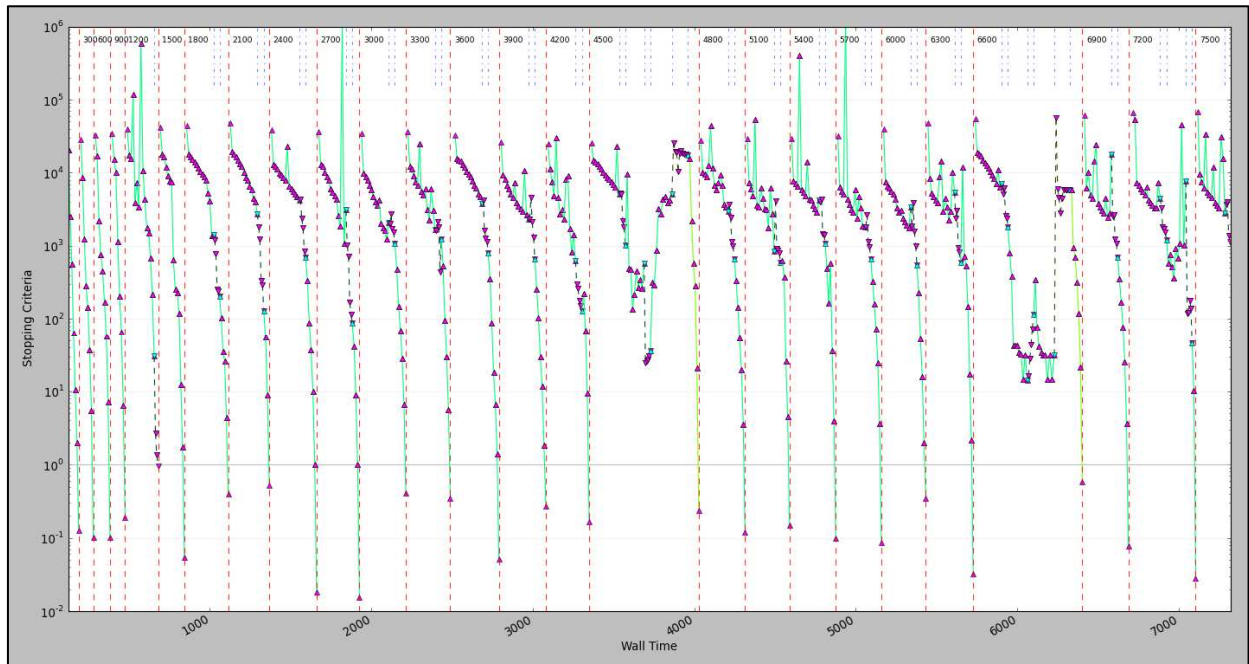
#Stratégie d'intégration temporelle (Intégration #1)
h_node_eulr2 = algo_node_bi(h_euler2, 2, 0, 0, 0, 0)
h_node_eulr1 = algo_node (h_euler1, 0, 0, h_node_eulr2, 1)
h_node_eulr0 = algo_node (h_euler0, 0, 0, h_node_eulr1, 1)
h_solver = h_node_eulr0
```

**Figure 16** – Définition des stratégies d'intégration temporelle à l'intérieur du fichier de commande dans H2D2

### 5.2.1. Impact global de la formulation du couvrant-découvrant

Deux formulations du couvrant-découvrant ont été testées, soit CDYS et CDY2, pour des simulations en non-stationnaire totalisant 24 heures de marées. L'impact de la formulation s'est avéré considérable puisqu'avec la formulation CDY2 les temps de calcul totaux ont été réduits de moitié par rapport à la formulation CDYS. La Figure 17 illustre le comportement de convergence pour les deux formulations testées sur une fenêtre de 2 heures.





**Figure 17** – Traceur de convergence pour une simulation avec la formulation du couvrant-découvant CDY2 (haut) et CDYS (bas)

### 5.2.2. Impact global de la bisection du pas de temps

Une comparaison des simulations réalisées pour l'une ou l'autre des stratégies d'intégration temporelle montre qu'une bisection basée sur la Stratégie #1 est légèrement plus efficace que lorsque basée sur la Stratégie #2. Globalement, on passe d'une moyenne de 37,12 itérations par pas de temps à une moyenne de 35,79, ce qui représente une économie de près d'une heure en temps de calcul sur une trentaine d'heures de simulation.

L'impact de la bisection peut se lire à travers le fichier de statistiques globales présenté au Tableau 1; ce tableau est généré à partir de l'outil CTTracer de H2D2 (voir section 3.1). On y lit que 91 % des pas de temps ont été convergés à l'intérieur des deux premiers blocs de stratégie, soit avant bisection (95 premières itérations) : 86 % des pas de temps convergent à l'intérieur du premier bloc de 45 itérations (Stratégie #1); 5 % convergent à l'intérieur du second bloc de 50 itérations (Stratégie #2). La bisection permet donc d'amener à convergence les 9 % restants. Toutefois, ce pourcentage représente un coût de calcul non négligeable puisqu'il fait passer le nombre moyen d'itérations par pas de temps de 21,58 à 35,79.

**Tableau 1** – Statistiques globales représentées par classe de nombre d’itérations nécessaires à la convergence

class	Classes		Cumulated		
	nseq	niter	nseq	iter	it/sq
0- 5	0	0			
5- 10	16	140	16 ( 6%)	140 ( 1%)	8.75
10- 15	116	1357	132 ( 46%)	1497 ( 15%)	11.34
15- 20	10	160	142 ( 49%)	1657 ( 16%)	11.67
20- 25	33	752	175 ( 61%)	2409 ( 23%)	13.77
25- 30	52	1365	227 ( 79%)	3774 ( 37%)	16.63
30- 35	8	245	235 ( 82%)	4019 ( 39%)	17.10
35- 40	9	342	244 ( 85%)	4361 ( 42%)	17.87
40- 45	3	126	247 ( 86%)	4487 ( 44%)	18.17
45- 50	0	0			
50- 55	0	0			
55- 60	0	0			
60- 65	4	249	251 ( 87%)	4736 ( 46%)	18.87
65- 70	2	131	253 ( 88%)	4867 ( 47%)	19.24
70- 75	3	218	256 ( 89%)	5085 ( 49%)	19.86
75- 80	3	230	259 ( 90%)	5315 ( 52%)	20.52
80- 85	0	0			
85- 90	3	266	262 ( 91%)	5581 ( 54%)	21.30
90- 95	1	94	263 ( 91%)	5675 ( 55%)	21.58
95-100	0	0			
100-105	0	0			
105-110	1	108	264 ( 92%)	5783 ( 56%)	21.91
110-115	2	224	266 ( 92%)	6007 ( 58%)	22.58
115-120	0	0			
120-125	2	241	268 ( 93%)	6248 ( 61%)	23.31
125-130	1	129	269 ( 93%)	6377 ( 62%)	23.71
130-135	3	396	272 ( 94%)	6773 ( 66%)	24.90
135-140	0	0			
140-145	1	142	273 ( 95%)	6915 ( 67%)	25.33
145-150	1	146	274 ( 95%)	7061 ( 69%)	25.77
150-155	0	0			
155-160	2	314	276 ( 96%)	7375 ( 72%)	26.72
160-165	1	160	277 ( 96%)	7535 ( 73%)	27.20
165-170	0	0			
170-175	0	0			
175-180	0	0			
180-185	0	0			
185-190	1	188	278 ( 97%)	7723 ( 75%)	27.78
190-195	0	0			
195-200	0	0			
200-205	0	0			
205-210	0	0			
210-215	0	0			
215-220	1	215	279 ( 97%)	7938 ( 77%)	28.45
220-225	0	0			
225-230	1	229	280 ( 97%)	8167 ( 79%)	29.17
230-235	1	233	281 ( 98%)	8400 ( 81%)	29.89
235-240	1	236	282 ( 98%)	8636 ( 84%)	30.62
240-245	0	0			
245-250	1	248	283 ( 98%)	8884 ( 86%)	31.39
250-255	0	0			
255-260	1	255	284 ( 99%)	9139 ( 89%)	32.18
260-265	0	0			
265-270	0	0			
270-275	0	0			
275-280	1	277	285 ( 99%)	9416 ( 91%)	33.04
280-285	0	0			
285-290	1	289	286 ( 99%)	9705 ( 94%)	33.93
290-295	0	0			
295-300	1	299	287 (100%)	10004 ( 97%)	34.86
300-305	1	303	288 (100%)	10307 (100%)	35.79



## 6. Problèmes stationnaires

---

Il est possible de s'inspirer des stratégies de résolution présentées à la section 5.1 et de transposer l'analyse à la résolution de problèmes en régime stationnaire. Toutefois, quelques adaptations devront être faites, qui seront notamment fonction de la solution initiale utilisée. Pour arriver à faire converger un système à partir d'une solution initiale relativement loin de la solution recherchée, il peut être nécessaire de modifier les stratégies en augmentant le nombre d'itérations du premier bloc de Picard, en jouant avec les paramètres du backtracking, ou en se faisant succéder plusieurs blocs de résolution pour lesquels on relaxe graduellement les propriétés globales contrôlant la viscosité (damping, Péclet, longueur de mélange) (voir Figure 18).

```
pgl1 = 13    #amortissement
pgl2 = 19    #amortissement en découverture
h_pgl[pgl1] = 1.0e-004
h_pgl[pgl2] = 1.0e-004
h_solver.solve(h_sol, h_cria, h_lim, h_pst, t_ini)
h_pgl[pgl1] = 0.0
h_pgl[pgl2] = 0.0
h_solver.solve(h_sol, h_cria, h_lim, h_pst, t_ini)
```

**Figure 18** – Exemple d'une séquence de résolution en stationnaire avec changement de propriétés globales à l'intérieur du fichier de commande dans H2D2



## 7. Conclusion

---

Les différents outils à disposition dans H2D2 ont permis d'évaluer la performance de plusieurs stratégies de résolution et d'intégration temporelle et de déterminer les conditions optimales de simulation. Deux stratégies de résolution ont été retenues et combinées à l'intérieur des stratégies d'intégration temporelle.

Les résultats des tests effectués permettent de tirer les conclusions suivantes :

- Un limiteur sur l'incrément de solution trop restrictif peut ralentir la convergence de façon significative; ses limites sont fixées à 10 m<sup>2</sup>/s en débits spécifiques et à 1 m en niveaux d'eau;
- Un backtracking sur Newton est nécessaire afin d'empêcher l'algorithme de diverger, mais ne doit pas forcer de trop grands incréments minimum; *tmin* et *tmax* sont fixées à 0,001 et 1 respectivement pour un backtracking quadratique;
- Un backtracking sur Picard peut dans certains cas être avantageux s'il s'agit du premier bloc de la stratégie, mais ne garantit pas la convergence des blocs de Newton qui le suivent;
- La relaxation de la matrice tangente ne présente aucun avantage;
- La formulation du couvrant-découvrant CDY2 mène à des temps de calcul inférieurs par un facteur 2 aux temps obtenus de la formulation CDYS;
- La bisection du pas de temps permet de récupérer les pas de temps non-convergés mais ajoute une charge de calcul considérable;
- Une stratégie d'intégration temporelle dont le nœud de bisection est basé sur la Stratégie de résolution #1 est légèrement plus efficace qu'avec la Stratégie #2.

Les travaux futurs pourront être orientés de façon à :

- Tester l'impact global du limiteur sur la convergence;
- Évaluer le comportement de convergence global pour différentes combinaisons de backtracking sur le premier bloc d'itérations de Picard;

- Raffiner les stratégies de résolution pour potentiellement réduire le nombre de pas de temps nécessitant la bisection;
- Élaborer des stratégies de résolution adaptées aux spécificités des problèmes en conditions stationnaires.



## 8. Références

---

- Dhatt, G., Touzot, G. et Lefrançois, E. 2005. Méthode des éléments finis – Une présentation. Hermes – Lavoisier, 601 p.
- Dubé, P. et Secretan, Y. 2009. Manuel d'utilisation : Logiciel H2D2. Rapport de recherche R-1051, INRS-ETE, 13 p.
- Heniche, M., Secretan, Y., Boudreau, P. et Leclerc, M. 2000. A two-dimensional finite element drying-wetting shallow water model for rivers and estuaries. *Advances in Water Resources* **23**: 359-372.