

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

INTÉGRATEURS EXPONENTIELS POUR LA PRÉVISION DU TEMPS :
ANALYSE ET APPLICATIONS

MÉMOIRE PRÉSENTÉ
COMME EXIGENCE PARTIELLE DE LA MAITRISE
EN SCIENCES DE L'ATMOSPHÈRE

PAR
KHADIJA KHORCHANI

MAI 2018

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

Je tiens à dire merci particulièrement à mon directeur de mémoire, monsieur Jean Côté, pour son encadrement et soutien durant toute ma période de recherche.

Je remercie aussi tous mes professeurs du département de Sciences de la Terre et de l'atmosphère pour leur participation à ma réussite.

Mes remerciements vont aussi à toute ma grande famille pour leur support très apprécié qui m'a toujours encouragée de bien finir mon projet.

Enfin j'adresse mes remerciements aux personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire.

DÉDICACE

À mes parents, mon mari et mes enfants.

TABLES DES MATIÈRES

LISTE DES FIGURES.....	vi
LISTE DES TABLEAUX.....	ix
LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES	x
RÉSUMÉ	xi
INTRODUCTION	1
CHAPITRE I LE PROBLÈME À RÉSOUDRE.....	4
1.1 Les équations de Saint-Venant ou Shallow-Water (SWE).....	4
1.2 Les conditions initiales	9
1.3 L'équation non-linéaire générale en 1D et sa solution en série.....	10
1.4 Discrétisation spatiale des équations	12
1.4.1 Méthode spectrale	12
1.4.2 Traitement des termes non-linéaires	13
CHAPITRE II LES MÉTHODES D'INTÉGRATION TEMPORELLE	14
2.1 Méthodes usuelles en météorologie.....	14
2.1.1 Les méthodes d'Adams-Bashforth d'ordre r (AB_r).....	15
2.1.2 Les méthodes de Runge-Kutta (RK).....	17
2.1.3 Approximation Crank-Nicolson (CN)	19
2.2 Méthodes d'intégration exponentielle	20
2.2.1 Les fonctions φ	22
2.2.2 Méthodes d'Adams-Bashforth exponentielles (ETD_r)	22
2.2.3 Méthodes exponentielles Runge-Kutta (ETD_xRK)	23
2.2.4 Ordre de précision avec rigidité	25
2.3 Étude de stabilité des intégrateurs exponentiels	30
2.4 Synthèse partielle.....	35

CHAPITRE III RÉSULTATS ET DISCUSSION	36
3.1 La programmation EXPINT	36
3.2 Tests préliminaires	38
3.2.1 Tests avec l'équation de Burgers	39
3.2.2 Tests avec l'équation KdV 1D	43
3.2.3 Tests avec l'équation Hochbruck-Ostermann	47
3.3 Résultats sur les équations Shallow-Water (SWE).....	51
3.3.1 SWE dans EXPINT	51
3.3.2 Tests sur SWE.....	52
3.3.3 Test sur l'implémentation de la méthode Crank-Nicolson	55
3.3.4 Super-convergence de ETD3RK.....	58
3.4 Synthèse partielle.....	60
CONCLUSION GÉNÉRALE.....	62
ANNEXE A LA SOLUTION DE L'ÉQUATION DE BURGERS PÉRIODIQUE	64
ANNEXE B LES TABLEAUX DE BUTCHER DES MÉTHODES ETD3RK	66
ANNEXE C CODE DES ÉQUATIONS DE SAINT-VENANT EN 1D	71
RÉFÉRENCES.....	77

LISTE DES FIGURES

Figure 1.1 : SWE dans un plan (x, z)	6
Figure 2.1 : Les régions de stabilité pour les méthodes <i>etd2</i> (bleu), <i>etd3</i> (vert) et <i>etd4</i> (rouge) pour $c = -20$	32
Figure 2.2 : Les régions de stabilité pour les méthodes <i>etd2</i> (bleu), <i>etd3</i> (vert) et <i>etd4</i> (rouge) pour $c = I$	32
Figure 2.3 : Les régions de stabilité pour les méthodes <i>etd3rk</i> (vert), <i>etd4rk</i> (magenta), <i>etd4rk7o12</i> (bleu), <i>strehmelweiner</i> (jaune), <i>strehmelweinerMod</i> (rouge), <i>hochost4</i> (noir) pour $c = -20$	33
Figure 2.4 : Les régions de stabilité pour les méthodes <i>etd3rk</i> (vert), <i>etd4rk</i> (magenta), <i>etd4rk7o12</i> (bleu), <i>strehmelweiner</i> (jaune), <i>strehmelweinerMod</i> (rouge), <i>hochost4</i> (noir) pour $c = I$	34
Figure 3.1 : Structure de la programmathèque EXPINT (on a ajouté dans EXPINT les codes qui sont en caractère gras).....	37
Figure 3.2 : L'erreur sur un pas de temps en fonction du pas de temps h pour l'équation de Burgers	40
Figure 3.3 : L'erreur globale en fonction du pas de temps h pour l'équation de Burgers	41

Figure 3.4 : L'erreur globale en fonction du temps de calcul utilisé pour l'équation de Burgers	42
Figure 3.5 : L'erreur sur un pas de temps en fonction du pas de temps h pour l'équation KdV	44
Figure 3.6 : L'erreur globale en fonction du pas de temps h pour l'équation KdV	45
Figure 3.7 : L'erreur globale en fonction du temps de calcul utilisé pour l'équation KdV	46
Figure 3.8 : L'erreur sur un pas de temps en fonction du pas de temps h pour l'équation Hochbruck-Ostermann.....	48
Figure 3.9 : L'erreur globale en fonction du pas du temps h pour l'équation Hochbruck-Ostermann	49
Figure 3.10 : L'erreur globale en fonction du temps de calcul utilisé pour l'équation Hochbruck-Ostermann	50
Figure 3.11 : L'erreur sur un pas de temps en fonction du pas de temps h pour SWE	52
Figure 3.12 : L'erreur globale en fonction du pas de temps h pour SWE	53
Figure 3.13 : L'erreur globale en fonction du temps de calcul utilisé pour SWE.....	54
Figure 3.14 : L'erreur globale en fonction de h pour SWE avec $\varepsilon = 0.01$	56
Figure 3.15 : L'erreur globale en fonction de h pour SWE avec $\varepsilon = 0.1$	57

Figure 3.16 : L'erreur globale en fonction de h pour SWE avec $\varepsilon=1$ 58

LISTE DES TABLEAUX

Tableau 2.1 : Les différents coefficients β_k pour les méthodes AB en fonction de r , l'ordre de la méthode (Gander <i>et al.</i> , 2010 ; Ascher et Greif, 2011).	17
Tableau 2.2 : Tableau de Butcher pour les méthodes RK explicites.	18
Tableau 2.3 : Tableau de Butcher pour la méthode RK4.....	19
Tableau 2.4 : Les différents coefficients $\beta_k(z)$ des méthodes ETD r pour $r = 1..4$ (Minchev et Wright, 2005).....	23
Tableau 2.5 : Le tableau de Butcher étendu pour ETD r K	25
Tableau 2.6 : Les conditions d'ordre de rigidité (Hochbruck et Ostermann, 2010) ...	26
Tableau 2.7 : Tableau de Butcher de la méthode <i>etd4rk7o12</i>	29
Tableau 2.8 : Tableau de Butcher de la méthode <i>strehmelweinerMod</i>	30

LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES

1D	:	Une dimension
IE	:	Intégrateurs Exponentiels
IF	:	Integrating Factor (Facteur intégrant)
SWE	:	Shallow-Water equations (Équations de Saint-Venant)
ETD	:	Exponential Time Differencing
EDP	:	Équations aux dérivées partielles
EDO	:	Équations aux dérivées ordinaires
FFT	:	Transformation de Fourier Rapide
AB	:	Adams-Bashforth
RK	:	Runge-Kutta
CN	:	Crank-Nicolson
ETD r	:	Adams-Bashforth Exponentielle d'ordre r
ETDRK	:	ETD de type Runge-Kutta
KdV	:	Korteweg-de Vries
GEM	:	Global Environmental Multiéchelle

RÉSUMÉ

On étudie une nouvelle approche d'intégration temporelle appelée « intégration exponentielle ». Cette approche a été introduite depuis assez longtemps déjà pour résoudre les systèmes d'équations différentielles rigides (stiff systems), mais son utilisation est restée très limitée à cause de la complexité du calcul de l'exponentielle d'un opérateur. Récemment, avec l'évolution des systèmes de calcul, ces intégrateurs sont devenus le champ d'études de plusieurs chercheurs car ils semblent être plus avantageux que les méthodes classiques utilisées présentement. Ici on s'intéresse aux intégrateurs exponentiels pour la prévision du temps. On explore les intégrateurs exponentiels explicites existants et on fait des tests numériques sur des problèmes simples afin de pouvoir comparer les différentes méthodes. Les méthodes sélectionnées, soit les ETD de type Runge-Kutta d'ordre de précision trois et quatre, sont ensuite appliquées à un modèle de Saint-Venant à une dimension (Shallow-Water 1D). Les résultats obtenus montrent que ces intégrateurs donnent une meilleure précision qu'une méthode implicite classique utilisée maintenant dans les modèles atmosphériques avec potentiellement un coût de calcul beaucoup plus petit. Ce qui montre l'avantage de continuer d'explorer ces méthodes.

Mots clés : intégrateurs temporels – système rigide – prévision du temps – Shallow-Water 1D – ETD Runge-Kutta.

INTRODUCTION

En prévision numérique du temps, on cherche toujours à augmenter la résolution des modèles. Ceci est rendu possible grâce à l'évolution technologique des systèmes de calcul. Ces systèmes étant devenus massivement parallèles, on doit adapter nos méthodes de calcul à cette avancée pour les rendre plus performants sur ce type de système; entre autres on voudrait que le temps d'exécution diminue linéairement lorsque que le nombre de processeurs mis en jeu augmente.

La rigidité inhérente des équations météorologiques découle notamment de la présence des ondes de gravité rapides (Sato, 2014). Or cette partie de l'écoulement n'en est pas la principale composante et une certaine latitude est permise pour la représentation des ondes les plus rapides. Alors que les modèles numériques explicites des équations météorologiques ne peuvent produire des intégrations stables qu'au prix d'une contrainte excessive sur le pas de temps, les modèles de prévision actuels utilisent souvent une approche implicite ou semi-implicite (Lin *et al.*, 1997 ; Lauritzen *et al.*, 2011). Avec cette approche, on obtient la stabilité au prix d'une distorsion des ondes de gravité. L'architecture massivement parallèle des ordinateurs du futur semble cependant privilégier une approche explicite. On voudrait donc trouver des méthodes explicites qui ne souffrent pas de la contrainte des petits pas de temps.

Dans ce contexte plusieurs centres de prévision dont le Centre Météorologique Canadien ont commencé à explorer les intégrateurs exponentiels explicites (IE) (Clancy *et al.*, 2013 ; Gaudreault et Pudykiewicz, 2016). Ces intégrateurs explicites ont moins de contrainte sur le pas du temps tout en ne distordant pas les ondes de

gravité rapides. Évidemment pour les centres opérationnels se pose la question sur les coûts de la mise en œuvre de ces méthodes et de l'amélioration des prévisions à haute résolution. Ils ont limité leur travail aux méthodes de premier et deuxième ordre, mais avec une linéarisation dépendante du temps. Ce qui nécessite à refaire la linéarisation à chaque pas de temps.

Comme ces intégrateurs sont relativement nouveaux en prévision numérique du temps, nous nous proposons dans ce travail d'étudier la formulation des intégrateurs exponentiels explicites et de les appliquer aux équations de Saint-Venant en 1D (Haut et Wingate, 2014 ; Majda, 2003 ; Pedlosky, 1979 ; Vallis, 2017). Nous utiliserons toujours un état de base constant (une linéarisation indépendante du temps), mais nous explorerons des intégrateurs d'ordre plus élevé.

Cette classe de méthodes a été développée pour la première fois au début des années soixante par Certaine (1960) qui a mis au point deux méthodes d'ETD implicites. Puis, quelques années plus tard Norsett (1969) a développé les méthodes ETD explicites, qui sont basées sur le schéma d'Adams-Bashforth. En 1978 Friedli a mis au point une nouvelle famille de ces IE basée sur le schéma de Runge-Kutta explicite, ETDRK. Dans les années qui suivirent, plusieurs de ces méthodes ont été retrouvées et modifiées par d'autres chercheurs, mais leur utilisation est restée limitée à cause de la complexité du calcul de l'exponentielle d'un opérateur. Récemment les chercheurs ont commencé à explorer ces méthodes dont :

Cox et Matthews (2000) qui ont fait une comparaison entre cinq méthodes d'IE d'ordre deux en les testant sur plusieurs types d'équations différentielles.

Hochbruck et Ostermann (2005) ont utilisé les méthodes ETDRK d'ordre 2, 3 et 4 pour des problèmes parabolique semi-linéaire et ils ont découvert qu'ils y avaient une réduction d'ordre de précision pour certaines de ces méthodes. Dans cette même

année Berland *et al.* (2005) ont mis au point une programmathèque, appelée EXPINT. Celle-ci contient un grand nombre de ces IE appliqués à des équations différentielles qui servent de test standard.

En 2010 Hochbruck et Ostermann ont publié un article de revue sur ces IE. Clancy et Pudykiewicz (2013) ont étudié l'utilisation de ces IE dans les modèles atmosphériques. Eichwald (2013) a utilisé les IE pour la simulation de vagues non-linéaires.

Malgré que l'usage de ces méthodes d'IE soit resté très limité au début de leur découverte, aujourd'hui avec les avancées technologiques, on est de nouveau intéressé par ces IE. On pense que ces IE ont le potentiel de devenir plus avantageux que les méthodes traditionnelles qui utilisent l'approche semi-implicite ou implicite en modélisation atmosphérique. D'où notre motivation à étudier cette classe de méthodes d'intégration.

Ce document se divise en trois chapitres. Dans le premier, on introduit le problème qu'on va utiliser pour tester ces intégrateurs ainsi que sa méthode de discrétisation spatiale (pseudo-spectrale). Dans le deuxième, on présente les différentes méthodes d'intégration temporelles classiques en usage en météorologie et on explore quelques méthodes d'intégration exponentielles; on fait une comparaison entre les régions de stabilité de ces IE. Dans le dernier chapitre, on présente et on discute les résultats de l'application de ces intégrateurs sur des problèmes classiques et finalement sur les équations de Saint-Venant en 1D.

CHAPITRE I

LE PROBLÈME À RÉSOUDRE

Pour explorer notre nouvelle approche de résolution des systèmes d'équations aux dérivées partielles (EDP) non-linéaires, on a besoin d'un problème connu pour tester ces méthodes. On va utiliser les équations de Saint-Venant, plus habituellement nommées Shallow-Water équations (SWE), qui décrivent le mouvement des eaux peu profondes. Ces équations sont utilisées généralement comme un test préliminaire dans la modélisation atmosphérique. Pour cela dans ce présent chapitre, on va commencer par établir ces équations. Ensuite on introduit les conditions initiales utilisées et l'équation non-linéaire générale en 1D ainsi que sa solution en série. Finalement, on présente la méthode utilisée pour la discrétisation spatiale de nos équations.

1.1 Les équations de Saint-Venant ou Shallow-Water (SWE)

Les équations de Saint-Venant (SWE) décrivent l'écoulement d'un fluide incompressible, hydrostatique et en rotation, en contact avec une surface inférieure, et avec une surface libre supérieure, en présence d'un champ de gravité. Ces équations sont déduites des équations de conservation de la quantité de mouvement et de la masse. Pour établir ces équations, on considère l'écoulement d'un fluide homogène incompressible et de densité constante. La direction de propagation horizontale est suivant l'axe des x . La hauteur du fluide est par rapport à la hauteur de référence $z = 0$.

On note que l'échelle de longueur horizontale doit être beaucoup plus grande que l'échelle de longueur verticale pour respecter la condition hydrostatique. Cette condition implique que la vitesse verticale du fluide est faible. Nous pouvons simplifier en mettant la topographie égale à zéro, puisque ce n'est pas un aspect primordial de ce travail.

On note :

- La densité ρ du fluide (l'eau) est constante.
- u et v sont les vitesses du fluide dans le plan horizontal,
- H est la hauteur du fluide,
- H_0 est la hauteur d'équilibre du fluide,
- η est la perturbation de la hauteur du fluide par rapport à la hauteur d'équilibre,
- f est le paramètre de Coriolis,
- g est l'accélération de gravité.

La figure suivante montre le système étudié dans le plan (x, z) où la ligne en tiret représente la hauteur d'équilibre du fluide.

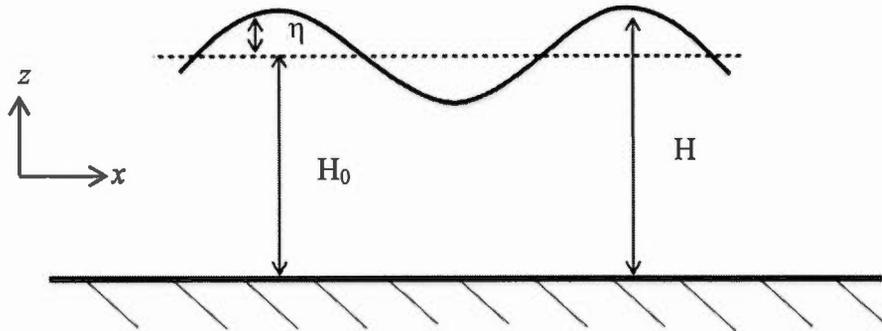


Figure 1.1 : SWE dans un plan (x, z)

Les équations du mouvement horizontal sont :

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} - fv + g \frac{\partial \eta}{\partial x} &= 0, \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + fu + g \frac{\partial \eta}{\partial y} &= 0. \end{aligned} \quad (1.1)$$

L'équation de conservation de la masse est :

$$\frac{\partial \eta}{\partial t} + \frac{\partial [(H_0 + \eta)u]}{\partial x} + \frac{\partial [(H_0 + \eta)v]}{\partial y} = 0. \quad (1.2)$$

L'analyse pour passer aux équations non dimensionnelles (ou sans dimension) s'inspire du chapitre 5 de Vallis (2017). D'abord on fixe les échelles suivantes :

L : l'échelle d'espace,

T : l'échelle de temps,

U : l'échelle de vitesse,

N_0 : l'échelle de la perturbation de la hauteur du fluide par rapport à sa position d'équilibre H_0 .

Donc on exprime nos variables comme suit :

$$\begin{aligned}(x, y) &= L(x', y'), \\ t &= Tt', \\ (u, v) &= U(u', v'), \\ \eta &= N_0\eta'.\end{aligned}\tag{1.3}$$

On obtient :

$$\begin{aligned}\frac{U}{T} \frac{\partial u'}{\partial t'} + \frac{U^2}{L} \left(u' \frac{\partial u'}{\partial x'} + v' \frac{\partial u'}{\partial y'} \right) - fUv' + \frac{gN_0}{L} \frac{\partial \eta'}{\partial x'} &= 0, \\ \frac{U}{T} \frac{\partial v'}{\partial t'} + \frac{U^2}{L} \left(u' \frac{\partial v'}{\partial x'} + v' \frac{\partial v'}{\partial y'} \right) + fUu' + \frac{gN_0}{L} \frac{\partial \eta'}{\partial y'} &= 0, \\ \frac{N_0}{T} \frac{\partial \eta'}{\partial t'} + \frac{UN_0}{L} \left(\frac{\partial (u'\eta')}{\partial x'} + \frac{\partial (v'\eta')}{\partial y'} \right) + \frac{UH_0}{L} \left(\frac{\partial u'}{\partial x'} + \frac{\partial v'}{\partial y'} \right) &= 0.\end{aligned}\tag{1.4}$$

L'échelle de tendance de l'accélération due à Coriolis est fU .

L'échelle de tendance d'advection divisée par l'échelle de tendance de l'accélération

due à Coriolis est le nombre de Rossby $Ro = \frac{U^2}{LfU} = \frac{U}{fL}$, et si nous considérons des

échelles de temps de l'ordre du temps d'advection, alors $T = \frac{L}{U}$ et l'échelle de

tendance locale divisée par l'échelle de Coriolis est aussi égale au nombre de Rossby Ro . Ce nombre est petit dans l'atmosphère et encore plus petit dans l'océan ; alors soit le terme de Coriolis domine et nous avons un état au repos, ou bien la force de Coriolis est balancée par la force de gravité et nous avons l'équilibre géostrophique

où les fluctuations de hauteur sont de l'ordre $N_0 = \frac{fUL}{g}$. Les équations peuvent

s'écrire :

$$\begin{aligned} \varepsilon \frac{\partial u'}{\partial t'} + \varepsilon \left(u' \frac{\partial u'}{\partial x'} + v' \frac{\partial u'}{\partial y'} \right) - v' + \frac{\partial \eta'}{\partial x'} &= 0, \\ \varepsilon \frac{\partial v'}{\partial t'} + \varepsilon \left(u' \frac{\partial v'}{\partial x'} + v' \frac{\partial v'}{\partial y'} \right) + u' + \frac{\partial \eta'}{\partial y'} &= 0, \\ \varepsilon F \left[\frac{\partial \eta'}{\partial t'} + \left(\frac{\partial (u' \eta')}{\partial x'} + \frac{\partial (v' \eta')}{\partial y'} \right) \right] + \left(\frac{\partial u'}{\partial x'} + \frac{\partial v'}{\partial y'} \right) &= 0, \end{aligned} \quad (1.5)$$

où $\varepsilon = Ro$, pour faire ressortir que c'est une petite quantité, et $F = \frac{f^2 L^2}{gH}$, le carré du

rapport entre l'échelle de longueur et le rayon de déformation de Rossby $L_d = \frac{\sqrt{gH}}{f}$.

En laissant tomber les variables primées et la dépendance en y , et avec le changement

de la variable de hauteur $\eta(x, t) = \frac{h(x, t)}{\sqrt{F}}$ on obtient finalement les équations (Majda,

2003) que l'on augmente d'une diffusion empirique:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} - \frac{v}{\varepsilon} + \frac{1}{\varepsilon \sqrt{F}} \frac{\partial h}{\partial x} &= \mu (-1)^{\frac{n-2}{2}} \frac{\partial^n u}{\partial x^n}, \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial t} + \frac{u}{\varepsilon} &= \mu (-1)^{\frac{n-2}{2}} \frac{\partial^n v}{\partial x^n}, \\ \frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{1}{\varepsilon \sqrt{F}} \frac{\partial u}{\partial x} &= \mu (-1)^{\frac{n-2}{2}} \frac{\partial^n h}{\partial x^n}, \end{aligned} \quad (1.6)$$

où $\mu \geq 0$ est le coefficient de diffusion. On a fait la plupart de nos expériences avec les paramètres : $n = 4$, $\mu = 10^{-4}$, $\varepsilon = 10^{-1}$ et $F = 1$. Alors l'équation 1.6 devient:

$$\begin{aligned}
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} - \frac{v}{\varepsilon} + \frac{1}{\varepsilon} \frac{\partial h}{\partial x} &= -\mu \frac{\partial^4 u}{\partial x^4}, \\
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + \frac{u}{\varepsilon} &= -\mu \frac{\partial^4 v}{\partial x^4}, \\
\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{1}{\varepsilon} \frac{\partial u}{\partial x} &= -\mu \frac{\partial^4 h}{\partial x^4}.
\end{aligned} \tag{1.7}$$

Avec ces paramètres si on prend L correspondant à l'échelle synoptique. On obtient un mouvement quasi-géostrophique qui a les caractéristiques suivantes (Vallis, 2017) :

$$U = \varepsilon f L = 10 \text{ m/s}$$

$$T = L/U = 10^5 \text{ s} \approx 28 \text{ heures}$$

$$H_0 = (fL)^2 / (gF) = 10^3 \text{ m}$$

$$h = \sqrt{F} \eta = N_0 \sqrt{F} = fUL\sqrt{F} / g = 100 \text{ m}$$

1.2 Les conditions initiales

On a choisi de travailler avec les conditions initiales utilisées par Haut et Wingate (2014) qui sont initialement très agéostrophiques et qui vont générer des ondes de gravité que les méthodes d'IE devraient réussir à intégrer d'une manière stable et précise. Elles s'écrivent :

$$\begin{cases} u(x,0) = 0, \\ v(x,0) = 0, \\ h(x,0) = c_1 \left[e^{-4\left(x-\frac{\pi}{2}\right)^2} \sin\left(3\left(x-\frac{\pi}{2}\right)\right) + e^{-2(x-\pi)^2} \sin(8(x-\pi)) \right] + c_0 + c_2x + c_3x^2 + c_4x^3. \end{cases} \quad (1.8)$$

où les constantes c_0, c_1, c_2, c_3 et c_4 sont choisies pour satisfaire les contraintes suivantes:

$$\int_0^{2\pi} h(x,0)dx = 0 \text{ et } \max_x |h(x,0)| = 1. \quad (1.9)$$

On note ici que Haut et Wingate ont pris seulement les constantes c_0 et c_1 (c_2, c_3 et c_4 sont nulles). Pour déterminer ces constantes, on a utilisé les deux contraintes citées dans l'équation 1.9 et les conditions de continuité de $h(x,0)$, de $\frac{\partial h(x,0)}{\partial x}$ et de

$$\frac{\partial^2 h(x,0)}{\partial x^2}.$$

1.3 L'équation non-linéaire générale en 1D et sa solution en série

Le modèle d'équation différentielle ordinaire (EDO) en une dimension s'écrit comme suit :

$$\frac{du}{dt} = cu + F(u(t),t) \quad (1.10)$$

où c est une constante et $F(u(t), t)$ est un terme non-linéaire général. On intègre cette équation de t_0 à t avec la condition initiale à t_0 $u(t_0) = u_0$. Dans ce qui suit, on peut prendre $t_0=0$.

Une approche qui nous permet de calculer formellement, la solution exacte est de procéder par série de Taylor. Le développement de $u(t)$ en séries de t est :

$$u(t) = \sum_{i=0}^{\infty} \frac{1}{i!} U_i t^i \quad (1.11)$$

Le développement de F en séries de t incluant la dépendance à travers $u(t)$ est :

$$F(u(t), t) = \sum_{i=0}^{\infty} \frac{1}{i!} F_i(u(t)) t^i, \quad (1.12)$$

$$F_i(u(t)) = \sum_{j=0}^{\infty} \frac{1}{j!} F_{ij} (u(t))^j.$$

Il suffit de remplacer $u(t)$ et $F(u, t)$ par leurs développements en séries respectifs dans l'équation (1.10) et de déterminer les coefficients U_i itérativement :

$$U_i = \left. \frac{d^{i-1} (cu(t) + F(u(t), t))}{dt^{i-1}} \right|_{t=0}, \quad (i = 1 \dots N) \quad (1.13)$$

On a utilisé le système de langage symbolique Maple™ pour obtenir les coefficients. Ceux-ci deviennent rapidement compliqués; on ne donne ici les résultats que jusqu'à l'ordre 4 :

$$\begin{aligned}
U_0 &= u_0, \\
U_1 &= cu_0 + [F_{00}], \\
U_2 &= c^2u_0 + [F_{00} + F_{01}u_0]c + [F_{10} + F_{01}F_{00}], \\
U_3 &= c^3u_0 + [F_{00} + F_{02}u_0^2 + 2F_{01}u_0]c^2 + [(F_{01}^2 + 2F_{02}F_{00} + 2F_{11})u_0 + F_{10} + 2F_{01}F_{00}]c... \\
&\quad + [2F_{11}F_{00} + F_{01}^2F_{00} + F_{02}F_{00}^2 + F_{20} + F_{01}F_{10}], \\
U_4 &= c^4u_0 + [F_{03}u_0^3 + 4F_{02}u_0^2 + F_{00} + 3F_{01}u_0]c^3... \\
&\quad + [(3F_{12} + 3F_{03}F_{00} + 4F_{01}F_{02})u_0^2 + (3F_{01}^2 + 5F_{11} + 8F_{02}F_{00})u_0 + F_{10} + 3F_{01}F_{00}]c^2... \\
&\quad + \left[(8F_{01}F_{02}F_{00} + 5F_{01}F_{11} + 3F_{02}F_{10} + F_{01}^3 + 6F_{12}F_{00} + 3F_{03}F_{00}^2 + 3F_{21})u_0 + 4F_{02}F_{00}^2... \right] c... \\
&\quad + \left[F_{20} + 3F_{01}^2F_{00} + 5F_{11}F_{00} + 2F_{01}F_{10} \right. \\
&\quad \left. + \left[F_{01}^2F_{10} + F_{03}F_{00}^3 + F_{01}F_{20} + F_{01}^3F_{00} + F_{30} + 3F_{11}F_{10} + 3F_{12}F_{00}^2 + 5F_{01}F_{11}F_{00} + 3F_{21}F_{00}... \right] \right. \\
&\quad \left. + 4F_{01}F_{02}F_{00}^2 + 3F_{02}F_{00}F_{10} \right].
\end{aligned} \tag{1.14}$$

Ces expressions servent à déterminer l'ordre de précision des différentes méthodes d'intégration. Si le terme non-linéaire est nul, on retrouve le développement en série de l'exponentielle.

1.4 Discrétisation spatiale des équations

1.4.1 Méthode spectrale

Cette méthode sert à la discrétisation spatiale et elle est fréquemment utilisée dans des modèles de prévisions météorologiques mondiales. L'idée de la méthode spectrale est de décomposer un signal en une somme de fonctions de base bien connues que multiplient des coefficients à déterminer. Nous utilisons une décomposition en séries trigonométriques, ce qui permet d'utiliser la transformée de Fourier pour déterminer ces coefficients et ainsi passer de l'espace physique à l'espace de Fourier. Elle permet aussi le passage inverse, c'est-à-dire de l'espace de Fourier à l'espace physique. L'utilisation de la transformation de Fourier rapide (FFT) permet de diminuer énormément le temps de calcul.

1.4.2 Traitement des termes non-linéaires

Pour résoudre les SWE, on utilise la méthode pseudo-spectrale. Le calcul des termes non-linéaires se fait alors dans l'espace physique.

Lorsqu'on utilise la méthode pseudo-spectrale pour calculer les termes non-linéaires, un type d'erreur numérique dite « aliasing » peut se produire en reliant les points de l'espace de Fourier à ceux de l'espace physique avec le nombre minimal de points d'espace requis pour une troncature spectrale R donnée, soit $2R+1$. On contourne ce problème en augmentant le nombre de points dans l'espace physique à $3R+1$ (Machenhauer, 1979). On choisit $R = 64$ ce qui correspond à un espacement de points de grille $dx \approx 5208$ m.

CHAPITRE II

LES MÉTHODES D'INTÉGRATION TEMPORELLE

Dans ce chapitre on va étudier les principales méthodes d'intégration temporelle qui sont utilisées pour la résolution d'EDP non-linéaires telle que les SWE. La première partie sera consacrée aux méthodes usuelles en météorologie, la deuxième partie sera réservée aux méthodes d'Intégrateurs Exponentiels (IE), alors que dans la troisième partie on fera une étude sur les régions de stabilité des méthodes IE. En partant de l'équation non-linéaire générale suivante :

$$\frac{du}{dt} = F(u(t), t) \quad (2.1)$$

En intégrant l'équation (2.1) entre t_n et t_{n+1} avec la valeur de $u(t_n)$ connue, on obtient la solution exacte $u(t_{n+1})$ au temps t_{n+1} :

$$u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} F(u(t), t) dt \quad (2.2)$$

2.1 Méthodes usuelles en météorologie

La résolution d'EDP non-linéaires se fait généralement de façon numérique. Pour obtenir une approximation numérique de la solution $u(t)$, il faut estimer sa valeur en un nombre fini de points t_i pour $i = 0, \dots, N$. L'écart entre deux points est appelé le pas

de temps, noté par h . Il existe plusieurs méthodes que l'on peut utiliser pour obtenir la solution numérique de ce type d'équation. Le choix des méthodes à utiliser est basé sur deux critères : la stabilité et la précision.

La stabilité indique que la différence entre la solution numérique et la solution exacte des équations discrétisées diminue lorsqu'on diminue le pas du temps.

L'ordre de précision d'une méthode est donné par l'entier p de l'expression suivante :

$$\varepsilon_{n+1} = \left| \tilde{u}(t_{n+1}) - u_{n+1} \right| = O(h^p) \quad (2.3)$$

où $\tilde{u}(t_{n+1})$ est la solution exacte et u_{n+1} est la solution numérique au temps t_{n+1} .

Les méthodes numériques sont classées en deux types. On trouve des méthodes à pas multiples et d'autres à un pas de temps.

Dans les méthodes à pas multiples, pour le calcul de la valeur de u_{n+1} , on utilise plusieurs valeurs aux temps précédents $t_n, t_{n-1}, t_{n-2}, \dots$. Parmi ces méthodes il y a les méthodes d'Adams-Bashforth (AB).

Dans les méthodes à un pas de temps, pour calculer la valeur discrète au point t_{n+1} , on n'utilise que la valeur au temps précédent t_n . Parmi ces méthodes il y a les méthodes de Runge-Kutta.

2.1.1 Les méthodes d'Adams-Bashforth d'ordre r (AB r)

Ces méthodes de résolution numérique des équations différentielles utilisent plusieurs pas de temps dans le calcul de la solution. Elles sont obtenues par l'approximation de la fonction F avec le polynôme d'interpolation de Lagrange P basé sur les valeurs de la solution aux pas de temps précédents. On considère une méthode AB d'ordre r .

Notons $F_i = F(u_i, t_i)$ et soit le polynôme d'interpolation de Lagrange sur une grille uniforme de pas h qui passe par les valeurs de F_n à F_{n+1-r} :

$$\begin{aligned}
 P(t) &= \sum_{k=0}^{r-1} w_k(\tau) F_{n-k} \text{ avec } \tau = \frac{t-t_n}{h}, \\
 w_k(\tau) &= \prod_{m=0, m \neq k}^{r-1} \frac{\tau + m}{-k + m} = \sum_{j=0}^{r-1} \gamma_{kj} \tau^j, \\
 \gamma_{kj} &= \frac{1}{j!} \left. \frac{d^j w_k(\tau)}{d\tau^j} \right|_{\tau=0}.
 \end{aligned} \tag{2.4}$$

En intégrant on obtient la solution la suivante :

$$\begin{aligned}
 u_{n+1} &= u_n + \int_{t_n}^{t_{n+1}} P(t) dt = u_n + h \sum_{k=0}^{r-1} \beta_k F_{n-k}, \\
 \beta_k &= \int_0^1 w_k(\tau) d\tau = \sum_{j=0}^{r-1} \frac{\gamma_{kj}}{j+1}.
 \end{aligned} \tag{2.5}$$

Un inconvénient des méthodes Adams-Bashforth est qu'elles requièrent des procédures spéciales pour démarrer l'intégration.

Par exemple, la méthode d'AB2 s'écrit comme suit (Gander *et al.*, 2010):

$$\begin{cases} u_0 \\ u_1 \text{ calculé avec une méthode à un pas} \\ u_{n+1} = u_n + \frac{h}{2}(3F_n - F_{n-1}) \end{cases} \tag{2.6}$$

Les différents coefficients β_k sont aisément tabulés :

Tableau 2.1 : Les différents coefficients β_k pour les méthodes AB en fonction de r , l'ordre de la méthode (Gander *et al.*, 2010 ; Ascher et Greif, 2011).

r	β_0	β_1	β_2	β_3
2	3/2	-1/2		
3	23/12	-16/12	5/12	
4	55/24	-59/24	37/24	-9/24

2.1.2 Les méthodes de Runge-Kutta (RK)

Les méthodes RK utilisent un seul pas de temps pour démarrer et elles divisent l'intervalle en plusieurs sous-intervalles pour augmenter l'ordre de précision. On commence par introduire s points intermédiaires, où des approximations intermédiaires de u seront calculées. La solution pour le pas de temps est une pondération de F calculée à ces valeurs intermédiaires.

La forme générale des méthodes RK explicites à s étages est (Gander *et al.*, 2010) :

$$\left\{ \begin{array}{l} \forall i = 1, 2, \dots, s, \left\{ \begin{array}{l} t_{n,i} = t_n + c_i h, \\ u_{n,i} = u_n + h \sum_{j=1}^{i-1} a_{ij} F(u_{n,j}, t_{n,j}), \end{array} \right. \\ u_{n+1} = u_n + h \sum_{i=1}^s b_i F(u_{n,i}, t_{n,i}), \end{array} \right. \quad (2.7)$$

où les coefficients a_{ij} , b_i et c_i sont des constantes qui satisfont aux contraintes :

$$\left\{ \begin{array}{l} \sum_{j=1}^s b_j = 1 \\ \sum_{j=1}^s a_{ij} = c_i, \quad \forall 1 \leq i \leq s \end{array} \right. \quad (2.8)$$

Le tableau de Butcher est souvent utilisé comme notation qui résume ces méthodes. Il rassemble tous ces coefficients.

Tableau 2.2 : Tableau de Butcher pour les méthodes RK explicites.

c_1					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
\vdots	\vdots	\vdots	\ddots		
c_s	a_{s1}	a_{s2}	\dots	$a_{s,s-1}$	
	b_1	b_2	\dots	b_{s-1}	b_s

La méthode Runge-Kutta d'ordre 4 (RK4), dont le tableau de Butcher est donné plus bas, est la plus célèbre de cette famille. Elle peut s'écrire comme :

$$u_{n+1} = u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$\text{où } \begin{cases} k_1 = hF(u_{n,1}, t_n), u_{n,1} = u_n, \\ k_2 = hF(u_{n,2}, t_n + \frac{h}{2}), u_{n,2} = u_n + \frac{k_1}{2}, \\ k_3 = hF(u_{n,3}, t_n + \frac{h}{2}), u_{n,3} = u_n + \frac{k_2}{2}, \\ k_4 = hF(u_{n,4}, t_n + h), u_{n,4} = u_n + k_3. \end{cases} \quad (2.9)$$

Tableau 2.3 : Tableau de Butcher pour la méthode RK4

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

2.1.3 Approximation Crank-Nicolson (CN)

Cette méthode est considérée la méthode implicite à deux pas de temps la plus précise. On l'utilise actuellement dans le modèle global canadien (GEM) (Côté *et al.*, 1998). L'idée de cette méthode est d'utiliser le développement en série de Taylor de u pour approximer la dérivée de u à un point donné $t_{n+1/2}$. La méthode est dite centrée parce qu'on a besoin de la valeur de u au pas t_{n+1} et au pas t_n . Par exemple pour un schéma CN avec un pas de temps constant h on a :

$$\begin{aligned}
 u_{n+1} &= u(t_{n+1/2}) + \frac{h}{2} \frac{du}{dt} \Big|_{t_{n+1/2}} + \frac{h^2}{8} \frac{d^2u}{dt^2} \Big|_{t_{n+1/2}} + \frac{h^3}{48} \frac{d^3u}{dt^3} \Big|_{t_{n+1/2}} + \dots \\
 u_n &= u(t_{n+1/2}) - \frac{h}{2} \frac{du}{dt} \Big|_{t_{n+1/2}} + \frac{h^2}{8} \frac{d^2u}{dt^2} \Big|_{t_{n+1/2}} - \frac{h^3}{48} \frac{d^3u}{dt^3} \Big|_{t_{n+1/2}} + \dots
 \end{aligned} \tag{2.11}$$

En soustrayant on obtient l'approximation d'ordre 2 pour la dérivée temporelle à $t_{n+1/2}$:

$$\left. \frac{du}{dt} \right|_{t_{n+1/2}} = \frac{u_{n+1} - u_n}{h} + o(h^2) \quad (2.12)$$

De même on discrétise le terme non-linéaire au temps $t_{n+1/2}$:

$$F(u, t_{n+1/2}) = \frac{F_{n+1} + F_n}{2} + o(h^2) \quad (2.13)$$

En combinant les deux équations précédentes on obtient la solution :

$$u_{n+1} = u_n + \frac{h}{2} (F_{n+1} + F_n) \quad (2.14)$$

F_{n+1} dépend de la solution $u(t_{n+1})$, d'où la nécessité de résoudre un problème implicite que l'on résout par itération après avoir soustrait de F une partie linéaire :

$$F(u(t), t) = Lu(t) + N(u(t), t). \quad (2.15)$$

2.2 Méthodes d'intégration exponentielle

Les intégrateurs exponentiels (IE) ont été mis au point pour la résolution numérique des équations différentielles non-linéaires qui ont une partie linéaire importante. On écrit en notant $u \equiv u(t)$ où il n'y a pas d'ambiguïté possible pour alléger le texte :

$$\frac{du}{dt} = Lu + N(u, t) \text{ avec la condition initiale } u_0 \text{ à } t_0. \quad (2.16)$$

Ces méthodes avaient pour objectif de surmonter le problème de rigidité (stiffness) dans des équations à croissance ou décroissance rapide causée par le terme linéaire. L'idée de ces méthodes est de résoudre de manière exacte la partie linéaire et d'approximer le terme non-linéaire.

On peut classer les IE en deux familles : les Exponential Time Differencing notées ETD et les Integrating Factor (Facteurs intégrants) notée IF. Dans notre travail, on va utiliser les ETD en se basant sur les résultats des travaux de Cox et Matthews (2002) qui ont montré que les ETD sont plus consistantes et performantes que les méthodes IF.

Pour obtenir facilement les intégrateurs exponentiels, on multiplie notre équation (2.16) par le facteur intégrant e^{-tL} , puis on intègre entre t_n et t_n+h avec la condition initiale u_n (Cox et Matthews, 2002) :

$$\int_{t_n}^{t_n+h} \left[\frac{d(e^{-tL}u)}{dt} - e^{-tL}N(u(t),t) \right] dt = 0, \quad (2.17)$$

ce qui peut s'écrire :

$$u_{n+1} = e^{hL} \left[u_n + \int_{t_n}^{t_n+h} e^{-(t-t_n)L} N(u(t),t) dt \right], \quad (2.18)$$

et après un changement de la variable d'intégration :

$$u_{n+1} = e^{hL} u_n + e^{hL} \int_0^h e^{-\theta L} N(u(t_n + \theta), t_n + \theta) d\theta \text{ avec } \theta = t - t_n. \quad (2.19)$$

Il reste donc à calculer l'intégrale de la partie non-linéaire. Vu qu'il existe plusieurs méthodes qui peuvent approximer cette intégrale, on trouve plusieurs variétés d'ETD.

Dans la famille de méthodes ETD, on distingue deux types de méthodes :

- Les ETD à pas multiples qui utilisent plusieurs valeurs au temps passé ($u_n, u_{n-1}, u_{n-2}, \dots$) dans le calcul de la valeur de u_{n+1} au point t_{n+1} . Elles

se basent sur la méthode d'Adams-Bashforth et on la note $ETDr$ où r indique l'ordre de la méthode.

- Les ETD à un pas de temps et à plusieurs étages/étapes qui font intervenir une seule valeur passée pour calculer u_{n+1} au temps t_{n+1} . Elles sont de type Runge-Kutta et on les note par $ETDxRK$ où x indique l'ordre de la méthode.

2.2.1 Les fonctions φ

Les fonctions φ appartiennent à une famille particulière de fonctions qui surviennent dans le développement des intégrateurs exponentiels. Elles sont définies pour un argument z , qui peut être un opérateur, comme suit :

$$\varphi_0(z) = e^z, \quad \varphi_1(z) = \frac{e^z - 1}{z}, \quad \dots \quad \varphi_l(z) = \frac{1}{(l-1)!} \int_0^1 e^{(1-\tau)z} \tau^{l-1} d\tau, \quad (2.20)$$

Les fonctions φ satisfont la relation de récurrence suivante :

$$\varphi_l(z) = z\varphi_{l+1}(z) + \frac{1}{l!}, \quad l = 0, 1, \dots, \quad (2.21)$$

avec $\varphi_l(0) = \frac{1}{l!}$.

2.2.2 Méthodes d'Adams-Bashforth exponentielles ($ETDr$)

Comme pour les méthodes d'Adams-Bashforth classiques, on approxime le terme non-linéaire de (2.18) par un polynôme d'interpolation de Lagrange déjà développé en (2.4) et qui utilise les valeurs aux temps précédents. On obtient les $ETDr$ à l'aide des coefficients γ_{kj} et des fonctions $\varphi_l(z)$:

$$u_{n+1} = \varphi_0(z)u_n + h \sum_{k=0}^{r-1} \beta_k(z)N_{n-k}, \text{ avec } z = Lh, \text{ et} \quad (2.22)$$

$$\beta_k(z) = \sum_{j=0}^{r-1} j! \gamma_{kj} \varphi_{j+1}(z).$$

Ces méthodes ont été décrites par Cox et Matthews (2002), mais ils ont utilisé des fonctions $g_i(z)$ qui sont linéairement reliés aux fonctions $\varphi_i(z)$. Dans la limite où z tend vers 0, on retrouve les méthodes d'Adams-Bashforth classiques. Notant $\varphi_i \equiv \varphi_i(z)$, les différents coefficients $\beta_k(z)$ sont aisément tabulés :

Tableau 2.4 : Les différents coefficients $\beta_k(z)$ des méthodes ETD r pour $r = 1..4$
(Minchev et Wright, 2005)

r	$\beta_0(z)$	$\beta_1(z)$	$\beta_2(z)$	$\beta_3(z)$
1	φ_1			
2	$\varphi_1 + \varphi_2$	$-\varphi_2$		
3	$\varphi_1 + \frac{3}{2}\varphi_2 + \varphi_3$	$-2\varphi_2 - 2\varphi_3$	$\frac{1}{2}\varphi_2 + \varphi_3$	
4	$\varphi_1 + \frac{11}{6}\varphi_2 + 2\varphi_3 + \varphi_4$	$-3\varphi_2 - 5\varphi_3 - 3\varphi_4$	$\frac{3}{2}\varphi_2 + 4\varphi_3 + 3\varphi_4$	$-\frac{1}{3}\varphi_2 - \varphi_3 - \varphi_4$

On va utiliser les méthodes avec $r = 2, 3$ et 4 , c'est-à-dire ETD2 (*abnorsett2*), ETD3 (*abnorsett3*) et ETD4 (*abnorsett4*).

2.2.3 Méthodes exponentielles Runge-Kutta (ETDxRK)

La formulation générale des méthodes ETD r K à s étages et un seul pas de temps h utilisée pour la résolution de l'équation 2.16 est adaptée de Hochbruck et Ostermann (2005) :

$$\begin{aligned}
u_{n+1} &= \varphi_0(z)u_n + h \sum_{i=1}^s b_i(z)G_{ni}, \\
u_{n,i} &= \varphi_0(c_i z)u_n + h \sum_{j=1}^s a_{ij}(z)G_{nj}, \\
G_{nj} &= N(u_{n,j}, t_n + c_j h).
\end{aligned} \tag{2.23}$$

où $z = hL$ et les $u_{n,i}$ sont les étapes internes d'approximation de la solution exacte pour i allant de 1 à s , $a_{ij}(z)$ et $b_j(z)$ sont les coefficients de ces méthodes. Ils doivent satisfaire :

$$\sum_{j=1}^s b_j(z) = \varphi_1(z), \quad \sum_{j=1}^s a_{ij}(z) = c_i \varphi_1(c_i z), \quad 1 \leq i \leq s \tag{2.24}$$

On note que pour les ETDRK explicites $c_1 = 0$, $\varphi_0(c_1 z) = 1$ et $a_{ij}(z) = 0$ pour $1 \leq i \leq j \leq s$. Ces coefficients a_{ij} , b_j et c_i sont regroupés dans le tableau de Butcher étendu suivant (Hochbruck et Ostermann, 2010) :

Tableau 2.5 : Le tableau de Butcher étendu pour ETD RK

c_1					1
c_2	$a_{21}(z)$				$\varphi_0(c_2 z)$
\vdots	\vdots	\ddots			\vdots
c_s	$a_{s1}(z)$	\dots	$a_{s,s-1}(z)$		$\varphi_0(c_s z)$
	$b_1(z)$	\dots	$b_{s-1}(z)$	$b_s(z)$	$\varphi_0(z)$

Dans ce qui suit pour alléger l'écriture on va noter $\varphi_i \equiv \varphi_i(z)$ et $\varphi_{i,j} \equiv \varphi_i(c_j z)$.

Parmi ces méthodes on utilisera les méthodes *etd2rk*, *etd3rk*, *etd4rk* de Cox et Matthews (2002), *Strehmel-Weiner*, ainsi que *krogstad* de Krogstad (2005) et *hochost4* de Hochbruck et Ostermann (2005). Les tableaux de Butcher de ces méthodes sont donnés dans Berland *et al.* (2005) et ils sont reproduits dans l'annexe B.

2.2.4 Ordre de précision avec rigidité

Les méthodes ETD Runge-Kutta ont besoin de satisfaire, en plus des conditions de précision ordinaire obtenues en comparant la solution exacte (1.14) à la série de Taylor de la solution discrétisée sur un pas de temps, des conditions d'ordre de rigidité (connues par stiff-order conditions) pour s'assurer qu'elles gardent leur ordre de précision en fonction de la rigidité ou du pas de temps. Ces conditions ont été énoncées par Hochbruck et Ostermann (2005). Les conditions d'ordre de rigidité jusqu'au 4^e ordre des méthodes ETD RK sont illustrées dans le Tableau 2.5 où J et K sont des opérateurs arbitraires qui dépendent du problème. On s'assure que les

résultats sont valables quelque soit leurs formes exactes. On ne va pas les préciser d'avantage puisque cette analyse est générique, d'ailleurs même dans l'article de revue Hochbruck et Ostermann (2010) ceux-ci ne sont pas explicités.

Tableau 2.6 : Les conditions d'ordre de rigidité (Hochbruck et Ostermann, 2010)

No.	Ordre	Condition
1	1	$\psi_1(z) = 0 \Rightarrow \varphi_1 - \sum_{k=1}^s b_k(z) = 0$
2	2	$\psi_2(z) = 0 \Rightarrow \varphi_2 - \sum_{k=1}^s b_k(z)c_k = 0$
3	2	$\psi_{1,i}(z) = 0 \Rightarrow \varphi_{1,i}c_i - \sum_{j=1}^{i-1} a_{ij}(z) = 0, i = 2, s$
4	3	$\psi_3(z) = 0 \Rightarrow 2\varphi_3 - \sum_{i=1}^s b_i(z)c_i^2 = 0$
5	3	$\sum_{i=1}^s b_i(z)J\psi_{2,i}(z) = 0 \Rightarrow \sum_{i=1}^s b_i(z)J \left(\varphi_{2,i}c_i^2 - \sum_{j=1}^{i-1} a_{ij}(z)c_j \right) = 0$
6	4	$\psi_4(z) = 0 \Rightarrow 6\varphi_4 - \sum_{i=1}^s b_i(z)c_i^3 = 0$
7	4	$\sum_{i=1}^s b_i(z)J\psi_{3,i}(z) = 0 \Rightarrow \sum_{i=1}^s b_i(z)J \left(2\varphi_{3,i}c_i^3 - \sum_{j=2}^{i-1} a_{ij}(z)c_j^2 \right) = 0$
8	4	$\sum_{i=1}^s b_i(z)J \sum_{j=2}^{i-1} a_{ij}(z)J\psi_{2,j}(z) = 0 \Rightarrow \sum_{i=1}^s b_i(z)J \sum_{j=2}^{i-1} a_{ij}J \left(\varphi_{2,j}c_j^2 - \sum_{k=2}^{j-1} a_{jk}(z)c_k \right) = 0$
9	4	$\sum_{i=1}^s b_i(z)c_i K\psi_{2,i}(z) = 0 \Rightarrow \sum_{i=1}^s b_i(z)c_i K \left(\varphi_{2,i}c_i^2 - \sum_{j=2}^{i-1} a_{ij}(z)c_j \right) = 0$

Avec :

$$\begin{aligned}\psi_j(z) &= \varphi_j - \sum_{k=1}^s b_k(z) \frac{c_k^{j-1}}{(j-1)!}, \\ \psi_{j,i}(z) &= \varphi_{j,i} c_i^j - \sum_{k=1}^{i-1} a_{ik}(z) \frac{c_k^{j-1}}{(j-1)!}, \quad i = 2, s.\end{aligned}\tag{2.25}$$

Minchev et Wright (2005) énoncent le lemme suivant pour qu'une méthode soit d'ordre de rigidité p :

$$\begin{aligned}\text{Cette méthode doit satisfaire les conditions d'ordre } p-1 \text{ avec } b_i(z) \\ \text{et les conditions d'ordre } p \text{ plus faiblement avec au moins } b_i(0).\end{aligned}\tag{2.26}$$

Hochbruck et Ostermann ont fait en 2005 une étude de plusieurs méthodes d'ETDRK et ils ont conclu que ces méthodes pouvaient subir une réduction de l'ordre de précision pour certains problèmes lorsqu'on augmentait le pas de temps et ils ont construit une méthode à 5 étages qui satisfait ces conditions, la méthode *hochost4*. Ils ont aussi conclu qu'il n'existait pas de méthode explicite à 4 étages qui serait d'ordre de rigidité 4. Ils ont toutefois fait cette étude avec une restriction sur le choix des c_i . On a donc cherché dans un espace de paramètres plus grand (i.e., c_2, c_3, c_4 variant plus librement) une solution avec la forme suggérée par Friedli (Minchev et Wright, 2005) pour le vecteur b :

$$\begin{aligned}b_i(z) &= \sum_{k=1}^s B_i^{[k]} \varphi_k, \quad i = 2, s, \\ b_1(z) &= \varphi_1 - \sum_{i=2}^s b_i(z).\end{aligned}\tag{2.27}$$

et pour la matrice A :

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ c_2 \varphi_{1,2} & 0 & 0 & 0 \\ c_3 \varphi_{1,3} - a_{32} & a_{32} & 0 & 0 \\ c_4 \varphi_{1,4} - a_{42} - a_{43} & a_{42} & a_{43} & 0 \end{bmatrix} \quad (2.28)$$

avec les a_{32} , a_{42} et a_{43} dépendants de z . On doit déterminer en plus des c_i , les $B_i^{[k]}$ ainsi que a_{32} , a_{42} et a_{43} . Ici aussi on a obtenu la solution avec MAPLE™. Notons que par construction les conditions 1 et 3 sont satisfaites.

On ne trouve effectivement pas de solutions du 4^e ordre à 4 étages qui satisfont le lemme (2.26), mais on trouve quand même une famille de solutions qui satisfait aux conditions 1, 2, 3, 4, 5 et 9 avec $b_i(z)$, et aux conditions 6 et 7 avec $b_i(0)$ et la condition 8 sous une forme affaiblie quand z tend vers 0.

Notons que pour cette famille de solutions $b_2(z) = 0$ ainsi que $\psi_{2,3}, \psi_{2,4} = 0$, ce qui assure que les conditions 5 et 9 sont satisfaites quels que soient les opérateurs J et K . Puisque la condition 7 doit être seulement satisfaite avec $b_2(0)$ qui commute avec l'opérateur J , on a aussi que cette condition est satisfaite pour tout J . Cette famille (*etd4rk₂*) est caractérisée par le vecteur c :

$$c = \left[0, c_2, \frac{1}{2}, 1 \right] \quad (2.29)$$

où c_2 prend une valeur plus grande que 0 jusqu'à 1, excluant 1/2. La valeur $c_2 = 7/12$ est intéressante puisque si L commute avec J , la condition 8 tend quadratiquement vers 0 quand z tend vers 0. D'autres valeurs intéressantes pour les fonctions φ en jeu sont $c_2 = 1/4$ et 1. Toutes ces solutions satisfont mieux les contraintes que les méthodes à 4 étages considérées par Hochbruck et Ostermann (2005). Nous allons donc considérer *etd4rk7o12* dans les résultats qui suivent.

Tableau 2.7 : Tableau de Butcher de la méthode *etd4rk7o12*

0					1
$\frac{7}{12}$	$\frac{7}{12}\varphi_{1,2}$				$\varphi_{0,2}$
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,3} - \frac{3}{7}\varphi_{3,2}$	$\frac{3}{7}\varphi_{3,2}$			$\varphi_{0,3}$
1	$\varphi_1 - a_{42} - a_{43}$	a_{42}	a_{43}		φ_0
	$\varphi_1 - 3\varphi_2 + 4\varphi_3$	0	$4\varphi_2 - 8\varphi_3$	$-\varphi_2 + 4\varphi_3$	φ_0

où

$$\begin{aligned}
 a_{42} &= -\frac{72}{7}\varphi_2 + \frac{288}{7}\varphi_3 - 12\varphi_{2,3} + \frac{144}{7}\varphi_{3,3}, \\
 a_{43} &= 14\varphi_2 - 48\varphi_3 + 14\varphi_{2,3} - 24\varphi_{3,3}.
 \end{aligned}
 \tag{2.30}$$

En examinant la méthode de Strehmel-Weiner, qui est une méthode de type ETDRK de 4 étapes mais d'ordre de rigidité 3 seulement, on a remarqué qu'en faisant une petite modification dans les coefficients a_{42} et a_{43} , cette méthode satisfait mieux les contraintes de rigidité. Elle satisfait maintenant les conditions de 1 à 5 et 9 avec $b_i(z)$, la condition 6 avec $b_i(0)$ et les conditions 7 et 8 avec $b_i(0)$ et $a_{ij}(0)$. Ce qui nous donne une nouvelle famille de méthodes qui est caractérisée par le vecteur c donné par l'équation 2.29, mais où la valeur $c_2 = 1/2$ est maintenant permise. La nouvelle méthode avec $c_2 = 1/2$ sera nommée Strehmel-Weiner modifiée (*strehmelweinerMod*) et elle a le tableau de Butcher suivant :

Tableau 2.8 : Tableau de Butcher de la méthode *strehmelweinerMod*

0				1
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2}$			$\varphi_{0,2}$
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2} - \frac{1}{2}\varphi_{2,2}$	$\frac{1}{2}\varphi_{2,2}$		$\varphi_{0,2}$
1	$\varphi_1 - 2\varphi_2$	$2\varphi_2 - 2\varphi_1$	$2\varphi_1$	φ_0
	$\varphi_1 - 3\varphi_2 + 4\varphi_3$	0	$4\varphi_2 - 8\varphi_3$	$-\varphi_2 + 4\varphi_3$
				φ_0

2.3 Étude de stabilité des intégrateurs exponentiels

Pour examiner les régions de stabilité des intégrateurs, on s'appuie sur les résultats des travaux réalisés par Cox et Matthews (2002) qui ont fait une comparaison entre les régions de stabilité des méthodes *etd2* et *etd2rk* avec deux autres méthodes implicites ordinaires. Ils ont conclu que la méthode *etd2rk* a une région de stabilité plus large que celle de la méthode *etd2*. Dans notre étude on va examiner les régions de stabilité des méthodes d'Adams-Bashforth et de Runge-Kutta exponentielles des 2^e, 3^e et 4^e ordres. Pour l'analyse de stabilité on considère la même approche que celle de Cox et Matthews (2002) et de Krogstad (2005) et qui ont utilisé l'équation suivante :

$$\frac{du}{dt} = cu + \lambda u \quad (2.7)$$

où c et λ sont des constantes complexes. Dans ce qui suit, le terme cu représente la principale composante et il est considéré comme la partie linéaire, alors que le terme λu représente une petite perturbation de type oscillatoire ou amortie et elle sera considérée comme la partie non-linéaire de l'équation. Pour un pas de temps h , on définit $y = ch$ et $x = \lambda h$ et on trace les régions de stabilité dans le plan complexe x ($x_{\text{réel}}$, $x_{\text{imaginaire}}$) pour une valeur fixe de c purement réelle négative ou purement imaginaire. Notons que les courbes pour une valeur de c purement imaginaire négative sont obtenues par symétrie en changeant le signe de $x_{\text{imaginaire}}$. (i.e. réflexion dans l'axe horizontal). On note aussi que la région de stabilité est la zone intérieure qui inclut l'origine des courbes fermées représentées dans les figures suivantes pour différentes valeurs c .

Les figures 2.1 et 2.2 montrent les régions de stabilité des méthodes d'Adams-Bashforth exponentielles d'ordre 2, 3 et 4.

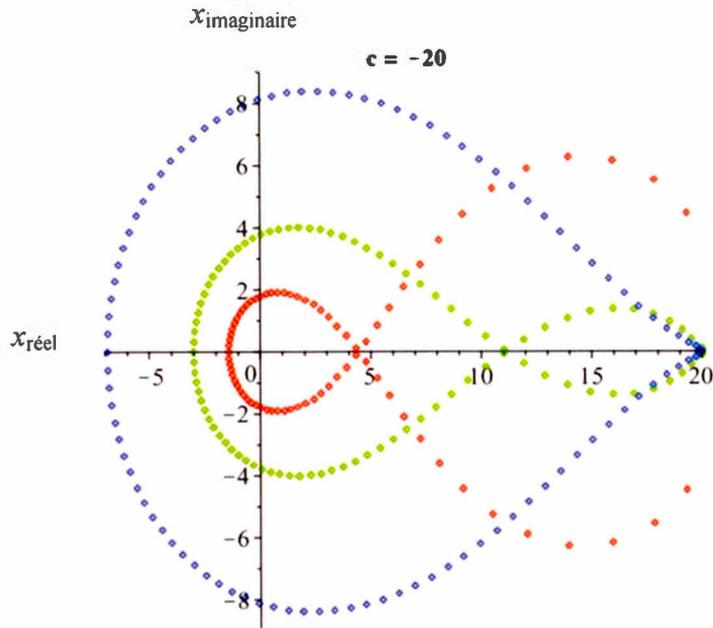


Figure 2.1 : Les régions de stabilité pour les méthodes *etd2* (bleu), *etd3* (vert) et *etd4* (rouge) pour $c = -20$

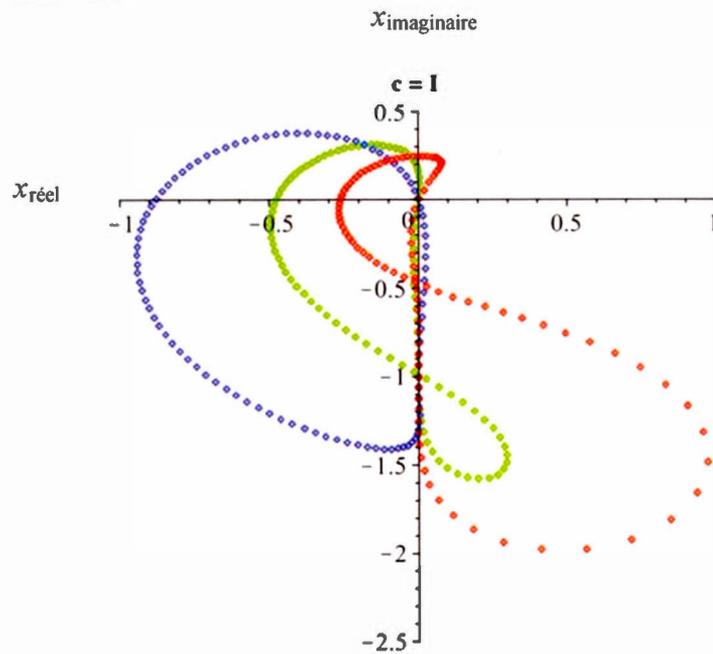


Figure 2.2 : Les régions de stabilité pour les méthodes *etd2* (bleu), *etd3* (vert) et *etd4* (rouge) pour $c = 1$

Les régions de stabilité pour les méthodes de Runge-Kutta exponentielles d'ordre 2, 3 et 4 sont représentées par les zones intérieures des courbes dans les figures 2.3 et 2.4 ci-dessous.

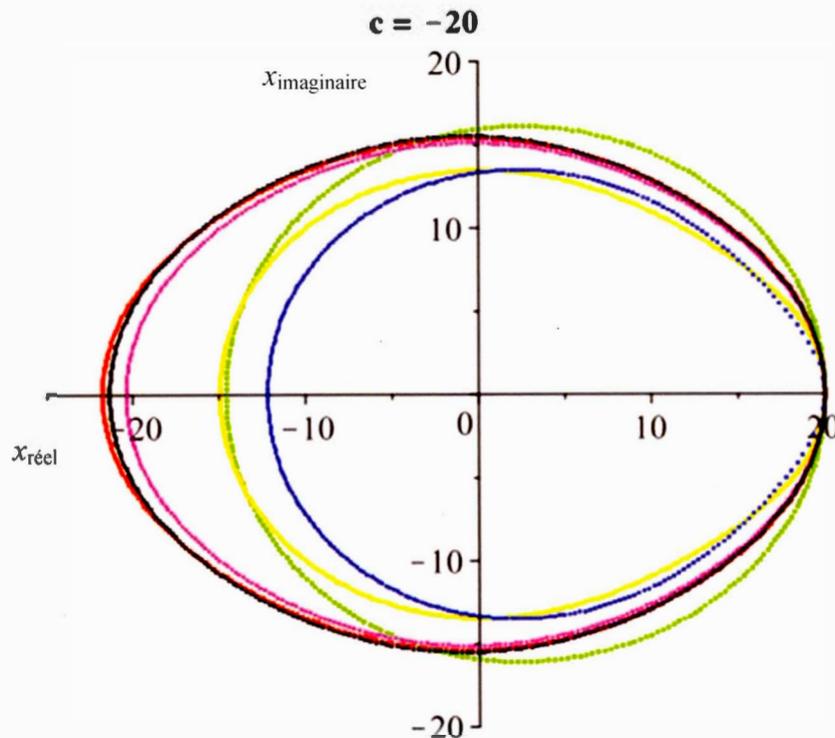


Figure 2.3 : Les régions de stabilité pour les méthodes *etd3rk* (vert), *etd4rk* (magenta), *etd4rk7o12* (bleu), *strehmelweiner* (jaune), *strehmelweinerMod* (rouge), *hochost4* (noir) pour $c = -20$

On cherche la méthode qui a la région de stabilité la plus large et qui contient le plus de valeurs imaginaires (l'intersection avec l'axe des x imaginaire) parce que les valeurs de λ risquent de se trouver sur cet axe.

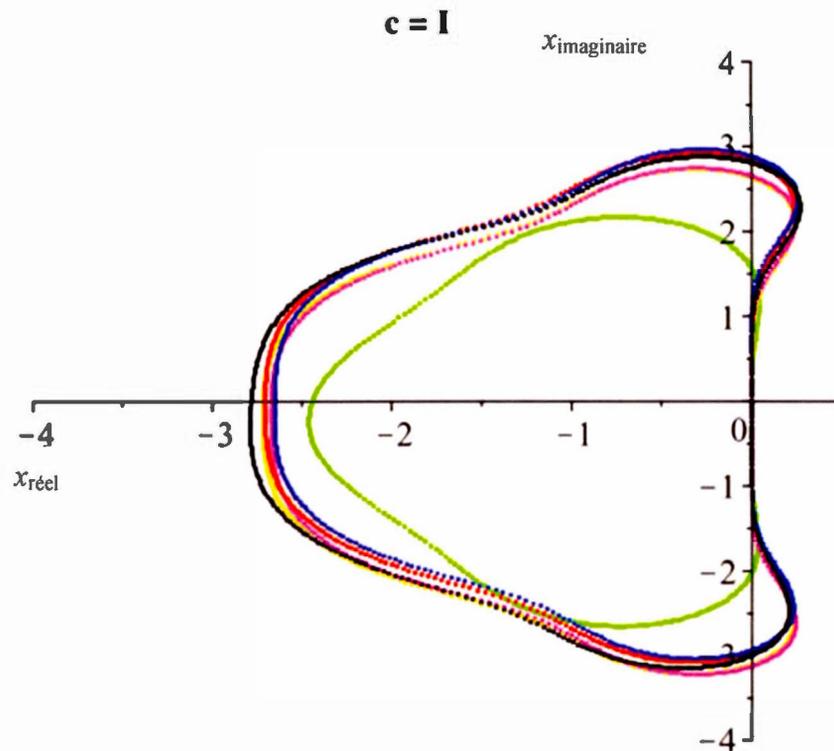


Figure 2.4 : Les régions de stabilité pour les méthodes *etd3rk* (vert), *etd4rk* (magenta), *etd4rk7o12* (bleu), *strehmelweiner* (jaune), *strehmelweinerMod* (rouge), *hochost4* (noir) pour $c = I$

L'analyse des figures 2.3 et 2.4 montre que les méthodes ETDRK permettent plus de valeurs imaginaires de λ . Notons aussi que pour les méthodes ETD, plus on augmente l'ordre de la méthode, moins la zone de stabilité contient de valeurs imaginaires, alors que pour les ETDRK la région de stabilité de la méthode du 4^e ordre contient plus de valeurs imaginaires de λ que celle du 3^e ordre pour $c = I$.

À partir de la figure 2.3 on peut déduire que les méthodes ETDRK restent toujours meilleures que celles des ETD parce que leurs régions de stabilité couvrent mieux l'axe des y . On a fait d'autres tests avec d'autres valeurs réelles négatives de c et on a constaté que la région de stabilité de la méthode *Strehmel-Weiner modifiée* devient

plus large après notre modification. Pour c imaginaire, toutes les méthodes ETD4RK ont des régions de stabilité très semblables et plus grande que celle d'*etd3rk*.

2.4 Synthèse partielle

Nous avons vu dans ce chapitre les deux principaux types de méthodes d'intégration exponentielle, ETD et ETDRK. On a présenté aussi des méthodes sélectionnées du type ETD qu'on va l'utiliser dans notre étude: *abnorsett2* (ETD2) et *abnorsett4* (ETD4) et celles de type ETDRK : *etd3rk*, *etd4rk*, *hochost4*, *strehmelweiner*, *strehmelweinerMod* et *krogstad*. En plus de ces méthodes, en examinant les conditions d'ordre de rigidité, on a trouvé une autre méthode de type ETD4RK qui satisfait mieux ces contraintes. Cette méthode est appelée *etd4rk7o12*. On a conclu ce chapitre par une étude de comparaison entre les zones de stabilité des méthodes. On a reproduit les résultats de Krogstad (2005), qui a étudié la stabilité des méthodes : ETD1, ETD2, ETD3 et ETD4RK, et Cox et Matthews (2002), qui ont étudié la stabilité de ETD2 et ETD2RK. De plus on en a ajouté d'autres : *abnorsett4* (ETD4), *etd3rk*, *etd4rk7o12*, *hochost4*, *strehmelweiner*, *strehmelweinerMod* et *krogstad*. Ceci nous amène à déduire que les méthodes ETDRK des 3^e et 4^e ordres seraient les meilleures pour les problèmes tels que les SWE (EDO avec des valeurs propres imaginaires).

CHAPITRE III

RÉSULTATS ET DISCUSSION

Après l'étude des différentes méthodes d'IE existantes, et la comparaison des zones de stabilité des méthodes sélectionnées, dans ce chapitre on fait des tests numériques pour étudier leurs précisions relatives. On présente aussi les résultats de l'application de ces IE sur nos SWE. Au préalable, on introduit la programmhèque EXPINT qu'on a utilisée pour notre travail et on montre ce qu'on a ajouté à cette programmhèque.

3.1 La programmhèque EXPINT

EXPINT est une programmhèque ou "package" pour les intégrateurs exponentiels écrite en langage de programmation MATLAB™ par Berland *et al.*, en 2005. Ils ont utilisé la programmation orientée objet ce qui facilite grandement l'ajout de schémas et de problèmes. Cette programmhèque est accessible gratuitement sur le Web via le lien suivant : <https://www.math.ntnu.no/num/expint/>

Les objectifs d'EXPINT sont (Berland *et al.*, 2005) :

- Création d'une interface qui permet de comparer différents IE sur un ensemble de problèmes standards
- Fournir des outils pour une visualisation facile du comportement numérique des IE

- Facilité de modification pour que les utilisateurs puissent ajouter leurs propres problèmes et IE

On a utilisé EXPINT pour l'élaboration de nos objectifs et on a commencé par la compréhension de sa structure et les différents scripts MATLAB™ qu'elle contient. La structure d'EXPINT est expliquée dans la figure suivante :

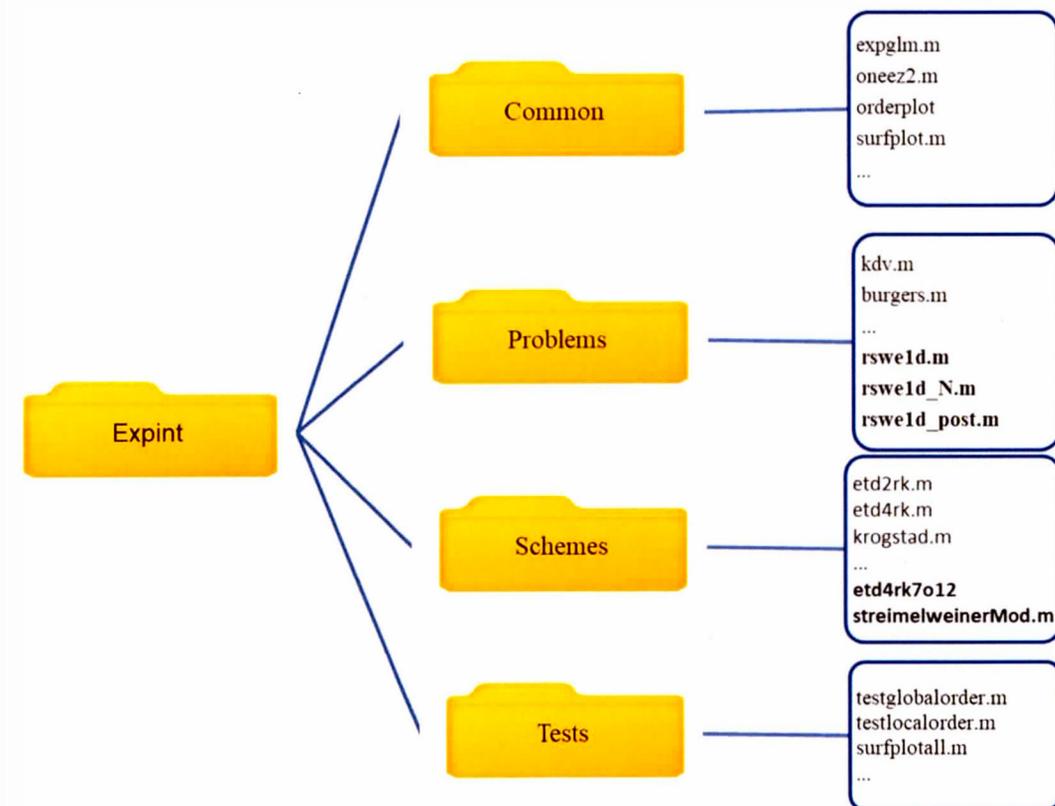


Figure 3.1 : Structure de la programmation EXPINT (on a ajouté dans EXPINT les codes qui sont en caractère gras).

À noter ici que l'existence de plusieurs problèmes déjà programmés nous a beaucoup aidé pour faire des tests préliminaires. Parmi ces problèmes, on a utilisé l'équation de Burgers (*burgers.m*), Kortweg-de Vries notée KdV (*kdv.m*) et l'équation de Hochbruck et Ostermann (*hochost.m*).

On a ajouté notre problème, les SWE à 1D, à la programmathèque. Cela a nécessité l'ajout de quelques sous-programmes : *rswe1d.m*, *rswe1d_N.m* et *rswe1d_post.m*.

Dans la programmathèque plusieurs schémas d'intégration sont inclus, dont la majorité sont des IE. Parmi ces méthodes on a utilisé :

- CN (*cranknicolson.m*) qui est une méthode implicite classique
- ETD : ETD2 (*abnorsett2.m*), ETD4 (*abnorsett4.m*)
- ETD RK : *etd2rk.m*, *etd3rk.m*, *etd4rk.m*, *strehmelweiner.m*, *krogstad.m*, *hochost4.m*

On a aussi ajouté à EXPINT les nouvelles méthodes qu'on a mises au point *etd4rk7o12.m* et *Strehmel-Weiner modifiée (strehmelweinerMod.m)*.

3.2 Tests préliminaires

Pour étudier la précision des méthodes d'intégration, on considère trois exemples numériques pour la comparaison des méthodes discutées dans le chapitre précédent. Les résultats présentés dans cette section sont tous obtenus en utilisant les codes qui existent déjà dans EXPINT. On a utilisé le script *testglobalorder.m* pour tracer l'erreur à la fin de l'intégration ce qui correspond à l'erreur globale et le script *testlocalorder.m* pour tracer l'erreur sur un pas de temps ce qui correspond à l'erreur locale. On n'a pas tracé l'erreur local de deux méthodes *abnorsett2* et *abnorsett4* parce qu'il faut spécifier un point de départ.

3.2.1 Tests avec l'équation de Burgers

La forme générale de l'équation de Burgers périodique (Bateman, 1915) est :

$$\frac{\partial u(x,t)}{\partial t} + u(x,t) \frac{\partial u(x,t)}{\partial x} = \mu \frac{\partial^2 u(x,t)}{\partial x^2} \text{ dans le domaine } x \in [-\pi, \pi]. \quad (3.1)$$

où μ est le coefficient de viscosité cinématique pris égal à 0,03 et où on utilise une discrétisation spectrale avec une résolution spatiale $ND = 128$. Les conditions initiales périodiques $u(x, 0)$ utilisées sont :

$$u(x, 0) = \sin(x). \quad (3.2)$$

L'équation 3.1 modèle la partie adjectice des équations de SWE 1D auxquelles on a ajouté un terme de diffusion. C'est une équation non-linéaire qui a la particularité de posséder une solution analytique et, même si c'est impossible d'écrire cette dernière sous une forme finie, on peut obtenir assez facilement une solution qui peut être aussi précise que l'on veut. L'établissement de la solution analytique est en annexe A. Notons que sans le terme de diffusion, nous obtiendrions une solution qui développe une discontinuité (un choc) (Strang, 2007) dans un temps fini.

Cette équation est déjà programmée dans la programmathèque (*burgers.m*). Les résultats sont montrés dans les figures ci-dessous :

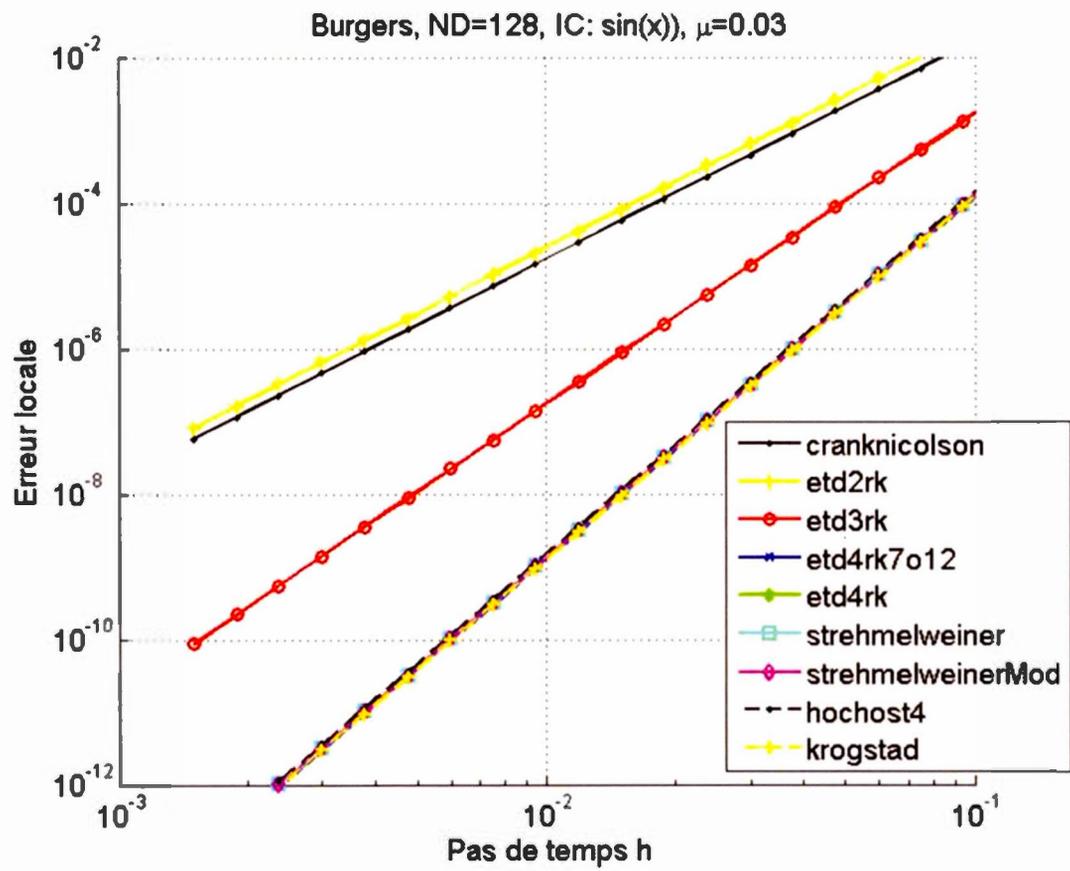


Figure 3.2 : L'erreur sur un pas de temps en fonction du pas de temps h pour l'équation de Burgers

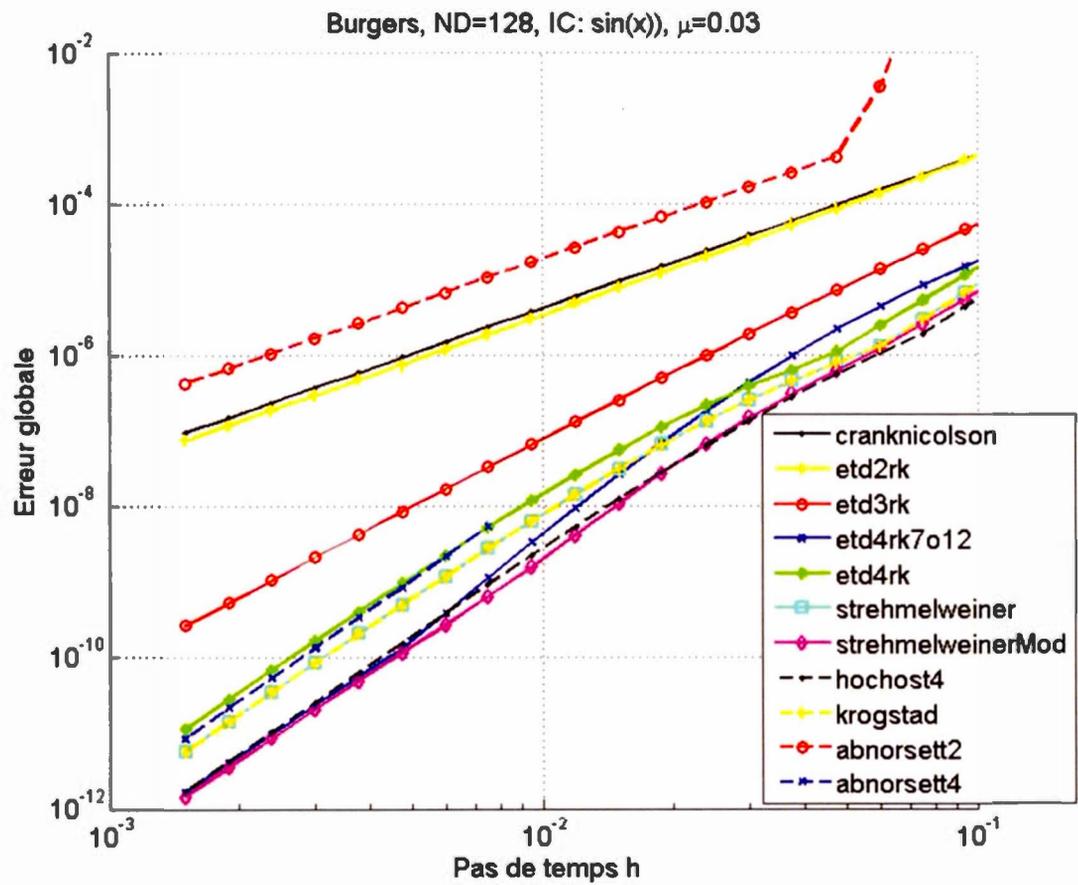


Figure 3.3 : L'erreur globale en fonction du pas de temps h pour l'équation de Burgers

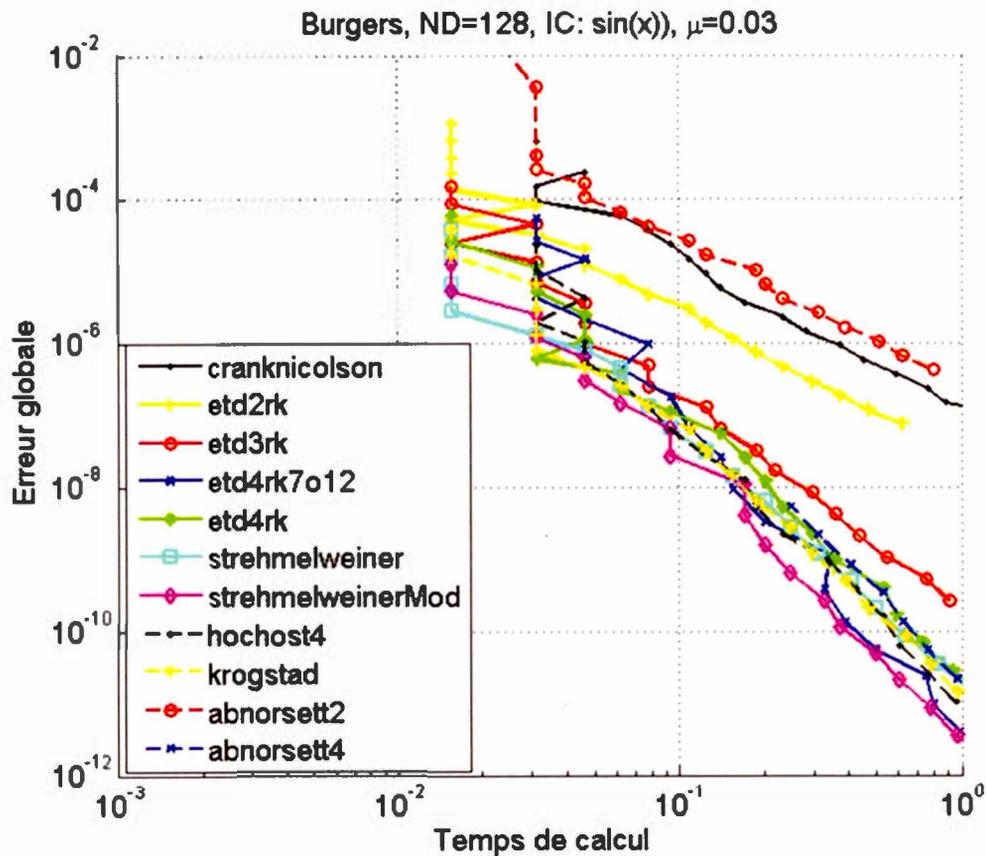


Figure 3.4 : L'erreur globale en fonction du temps de calcul utilisé pour l'équation de Burgers

La figure 3.2 présente l'erreur locale (sur un pas de temps) en fonction du pas de temps pour le problème de Burgers. La figure montre trois droites seulement parce que toutes les méthodes ETD4RK (*etdrk47o12*, *etd4rk*, *strehmelweiner*, *strehmelweinerMod*, *hochost4* et *krogstad*) sont superposées et elles sont plus précises que *etd3rk* et *etd2rk*. Les méthodes *etd2rk* et *cranknicolson* ont la même précision (les droites sont confondues). On remarque aussi que plus l'erreur locale augmente plus les pentes tendent vers l'horizontale. Par contre plus on augmente l'ordre des méthodes, plus la précision augmente.

Dans les figures 3.3 et 3.4 on trace l'erreur globale en fonction respectivement du pas de temps et du temps de calcul utilisé. Quoique les problèmes et les intégrateurs ne sont pas programmés de la manière la plus efficace possible, le temps de calcul utilisé nous donne une indication sur l'efficacité de ces méthodes puisqu'elles sont tous programmées de la même manière. En examinant ces deux figures on peut constater que :

- La méthode *etd2rk* est aussi précise que la méthode implicite de *cranknicolson* (traits jaune et noir),
- La méthode *etd3rk* et toutes les méthodes ETD4RK sont plus précises que la méthode de *cranknicolson*,
- Il y a un changement de pente pour toutes les méthodes ETD4RK,
- La méthode *abnorsett4* est instable pour des pas de temps supérieur à $8 \cdot 10^{-3}$
- La méthode la plus efficace pour ce problème est *strehmelweinerMod*. Elle donne la meilleure précision avec le moins de temps de calcul.

3.2.2 Tests avec l'équation KdV 1D

L'équation de KdV fait partie des équations aux dérivées partielles dont on connaît la solution exacte. On l'obtient de l'équation de Burgers en fixant $\mu = 1$ et en modifiant le terme diffusif. L'équation KdV à 1D s'écrit comme suit (Berland *et al.*, 2005) :

$$\frac{\partial u(x,t)}{\partial t} + u(x,t) \frac{\partial u(x,t)}{\partial x} = -\frac{\partial^3 u(x,t)}{\partial x^3} \quad \text{dans le domaine } x \in [-\pi, \pi], \quad (3.3)$$

On considère une solution exacte utilisée par Krogstad (2005) :

$$u(x,t) = 3\lambda \operatorname{sech}^2\left(\frac{\sqrt{\lambda}}{2}(x - \lambda t)\right) \quad \text{avec } \lambda = 625. \quad (3.4)$$

On utilise 256 points pour la discrétisation spatiale de Fourier et on intègre de $t = 0$ jusqu'à $t = \pi/\lambda$ avec les conditions initiales suivantes :

$$u(x, 0) = 3\lambda \operatorname{sech}^2\left(\frac{x\sqrt{\lambda}}{2}\right), \quad (3.5)$$

Notons que la solution dépendante du temps est obtenue des conditions initiales par $u(x, t) = u(x - \lambda t, 0)$.

Dans les figures suivantes, on trace les résultats en utilisant les 11 méthodes définies dans la première section.

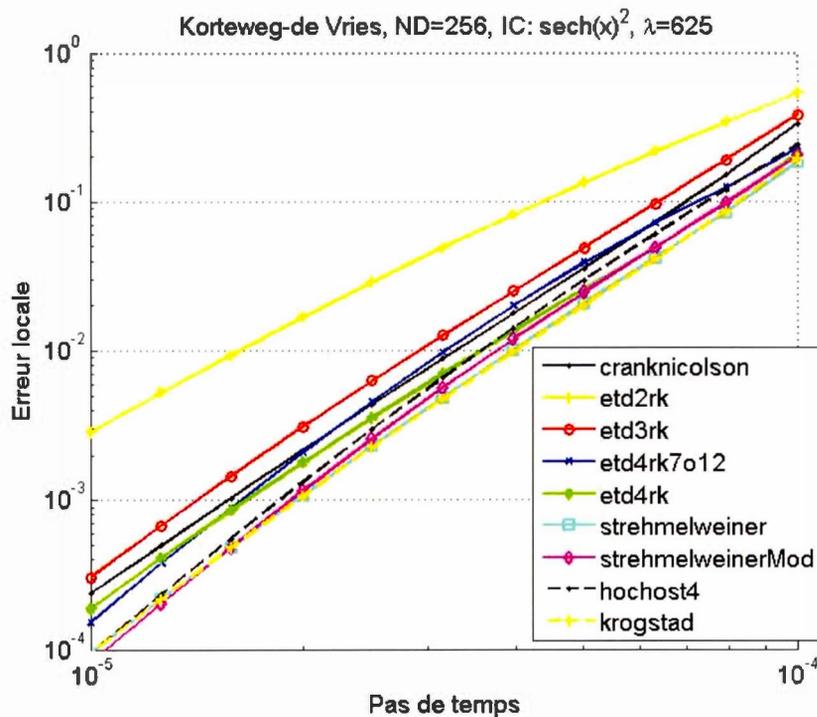


Figure 3.5 : L'erreur sur un pas de temps en fonction du pas de temps h pour l'équation KdV

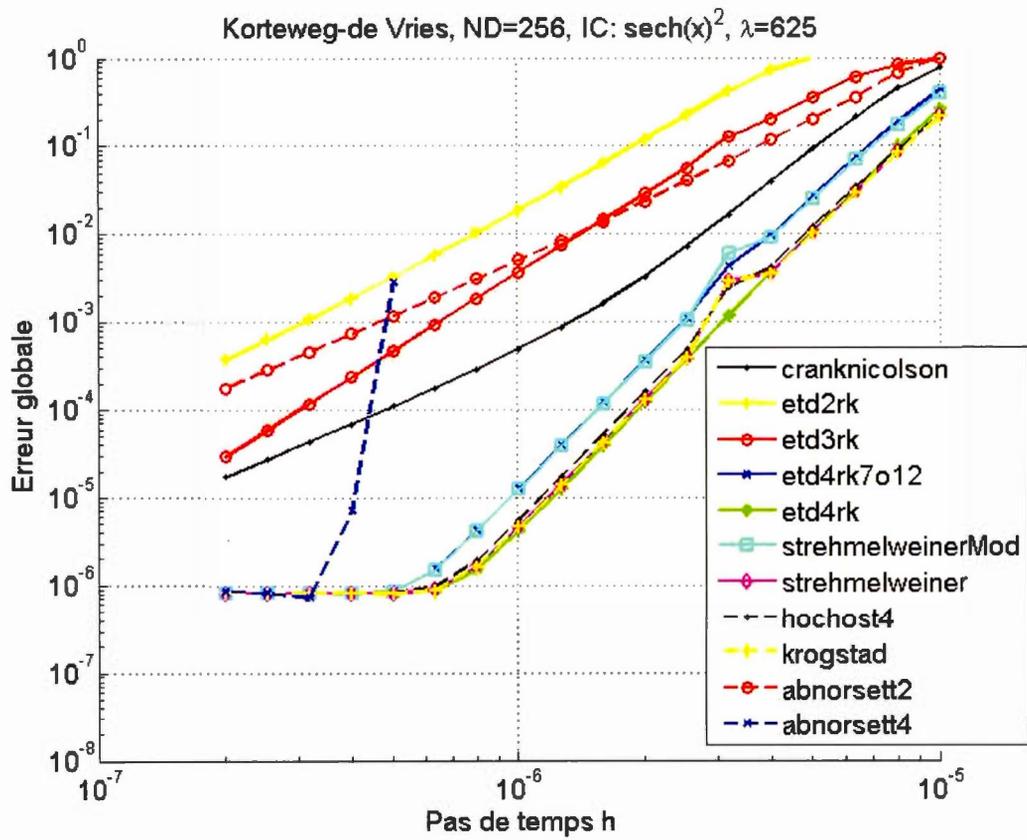


Figure 3.6 : L'erreur globale en fonction du pas de temps h pour l'équation KdV

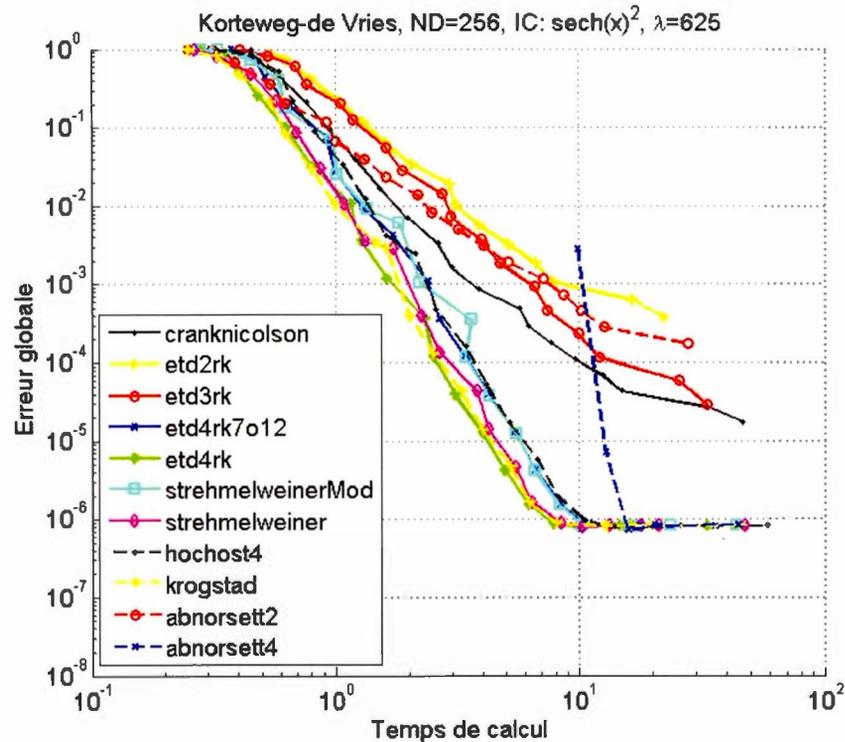


Figure 3.7 : L'erreur globale en fonction du temps de calcul utilisé pour l'équation KdV

La caractéristique la plus remarquable dans les figures 3.5 et 3.6 est la mauvaise performance des méthodes *etd2rk* et *etd3rk*. Elles sont moins précises que la méthode implicite ordinaire *cranknicolson*. Les méthodes EDRK deviennent plus avantageuses que la méthode *cranknicolson* si on considère une méthode du 4^e ordre. La méthode *abnorsett2* est moins précise que la méthode implicite, alors que celle d'*abnorsett4* est très précise, mais elle devient rapidement instable. On remarque aussi que toutes les méthodes EDRK du 4^e ordre donnent une bonne précision avec de faibles différences entre elles. Les méthodes *etd4rk* et *krogstad* sont les plus efficaces pour ce problème.

3.2.3 Tests avec l'équation Hochbruck-Ostermann

L'équation utilisée par d'Hochbruck et Ostermann est un problème parabolique semi-linéaire avec des conditions homogènes de Dirichlet (Hochbruck et Ostermann, 2005) :

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{1}{1+u^2} + \phi \quad x \in [0,1], \quad (3.6)$$

où ϕ est choisi tel que la solution exacte soit $u(x,t) = x(1-x)e^t$. Ce problème est discrétisé dans l'espace à l'aide d'un schéma de différences finies à point de grille standard ($ND = 200$), puis intégré entre $t = 0$ et $t = 1$. Les résultats sont tracés dans les figures ci-dessous :

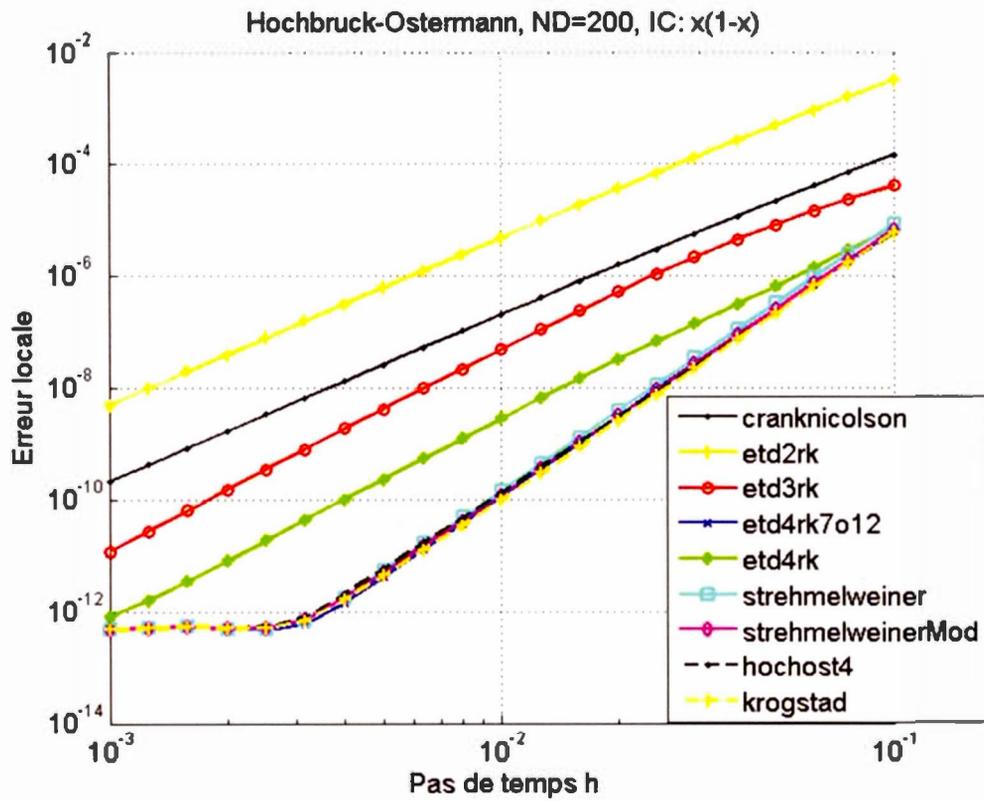


Figure 3.8 : L'erreur sur un pas de temps en fonction du pas de temps h pour l'équation Hochbruck-Ostermann

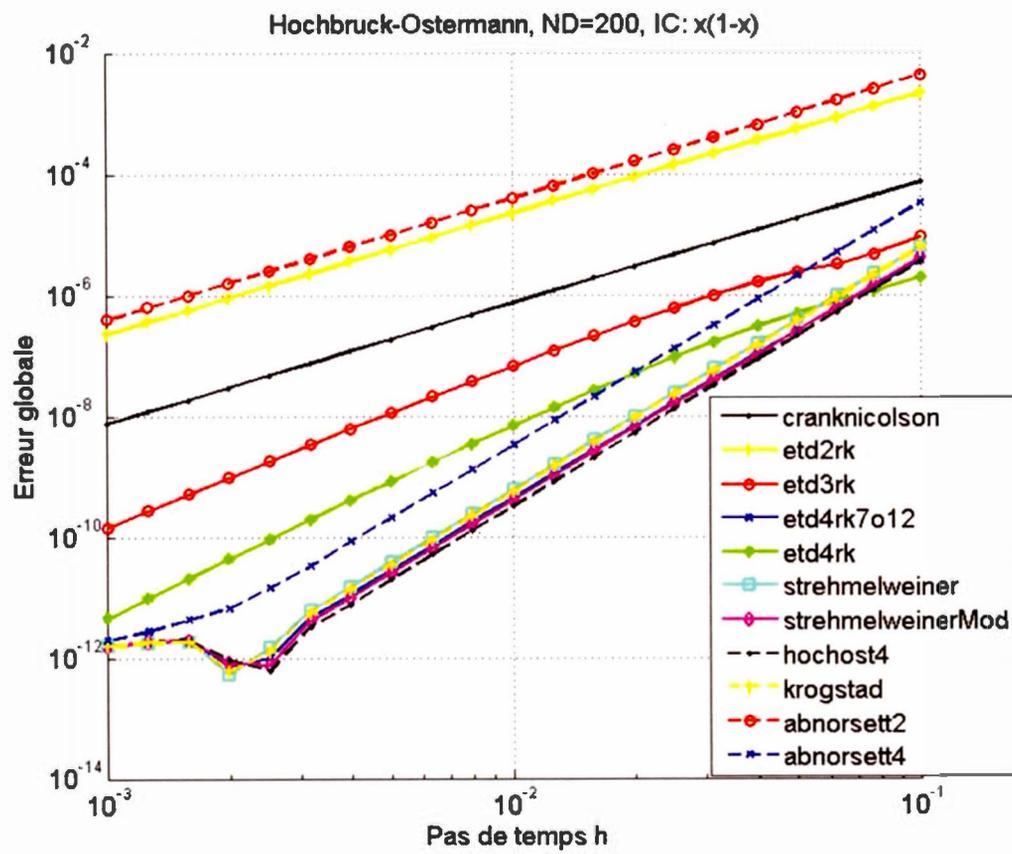


Figure 3.9: L'erreur globale en fonction du pas du temps h pour l'équation Hochbruck-Osternann

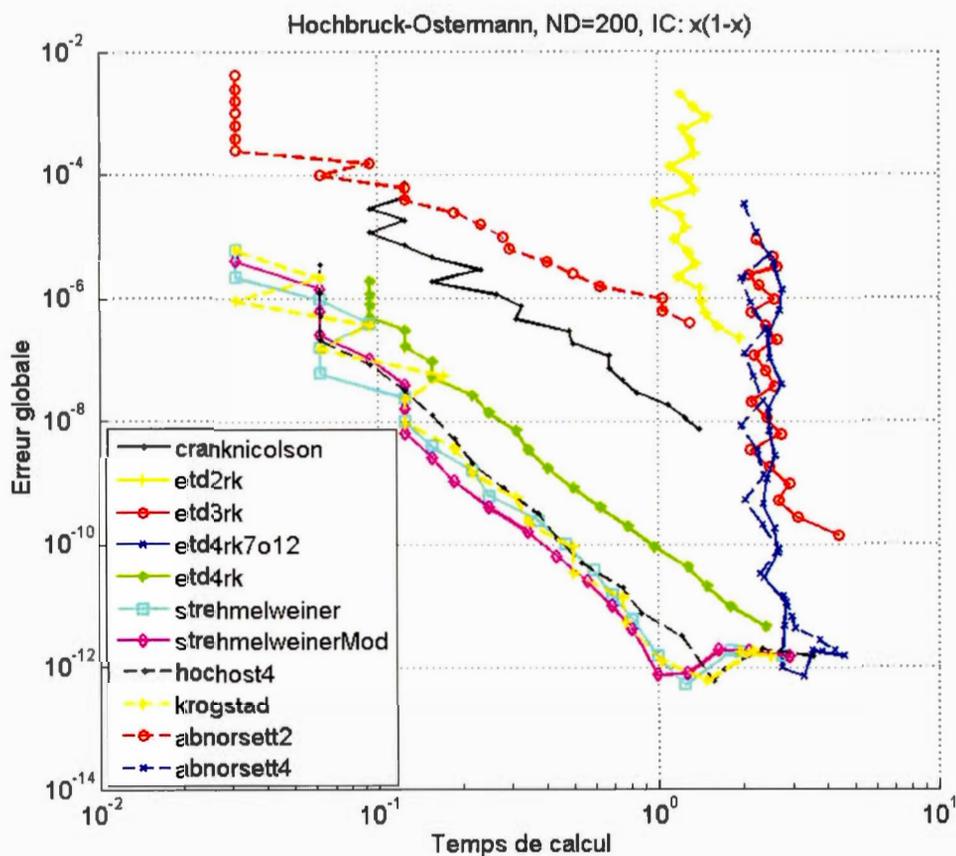


Figure 3.10 : L'erreur globale en fonction du temps de calcul utilisé pour l'équation Hochbruck-Osternann

Comme on voit dans les figures 3.8 et 3.9, les méthodes ETD4RK, spécifiquement *etd4rk7o12*, *strehmelweiner*, *strehmelweinerMod*, *hochost4* et *krogstad* dont les traits se confondent, se comportent très bien pour ce problème. En même temps, on remarque que la méthode *etd4rk7o12* est aussi précise que les autres méthodes du 4^e ordre, mais elle est plus coûteuse en temps de calcul. C'est la méthode de *strehmelweinerMod* qui est la plus efficace pour l'équation Hochbruck-Osternann.

3.3 Résultats sur les équations Shallow-Water (SWE)

Dans cette section on applique les méthodes sélectionnées après les tests préliminaires faits dans la section précédente sur nos SWE. Principalement les méthodes ETDRK des 2^e, 3^e et 4^e ordres.

3.3.1 SWE dans EXPINT

Dans cette section on revient enfin au problème principal décrit dans le premier chapitre par l'équation (1.7). Il fallait tout d'abord réécrire ces systèmes d'équations en séparant la partie linéaire et la partie non-linéaire. On définit les matrices suivantes :

$$U = \begin{pmatrix} u \\ v \\ h \end{pmatrix}, L = \frac{1}{\varepsilon} \begin{pmatrix} 0 & 1 & -\frac{\partial}{\partial x} \\ -1 & 0 & 0 \\ -\frac{\partial}{\partial x} & 0 & 0 \end{pmatrix}, N(U) = \begin{pmatrix} -u \frac{\partial u}{\partial x} \\ -v \frac{\partial v}{\partial x} \\ -\frac{\partial(hu)}{\partial x} \end{pmatrix} \text{ et } D = -\mu \begin{pmatrix} \frac{\partial^4}{\partial x^4} & 0 & 0 \\ 0 & \frac{\partial^4}{\partial x^4} & 0 \\ 0 & 0 & \frac{\partial^4}{\partial x^4} \end{pmatrix}, \quad (3.8)$$

où L représente l'opérateur linéaire et N l'opérateur non-linéaire, et D représente le terme de diffusion qui est aussi un terme linéaire et qui s'ajoute au terme L dans l'algorithme IE. Le système (1.7) s'écrit donc sous la forme :

$$\frac{dU}{dt} = (L + D)U + N(U), \quad (3.9)$$

On remarque que l'opérateur L est non-diagonal, ce qui a nécessité un traitement spécial. L'utilisation de la méthode pseudo-spectrale pour la discrétisation spatiale

nous a permis de faire la diagonalisation par bloc de cette matrice. Les détails du calcul sont expliqués dans Majda (2003).

3.3.2 Tests sur SWE

Dans cette section, on va présenter les résultats de l'application des méthodes ETD4RK qui ont été sélectionnées après l'analyse de stabilité (section 2.4) et les tests préliminaires (section 3.2). Afin d'avoir une vue plus complète, on a décidé de tester aussi les autres méthodes ETD de type RK qui étaient moins efficaces, telles que *etd2rk* et *etd3rk*. On utilise la méthode Crank-Nicolson comme étalon. Les résultats obtenus sont présentés dans les figures ci-dessous où R représente la troncature spectrale, ε le nombre de Rossby et $alias = 2$ indique l'aliasing quadratique pour les termes non-linéaires.

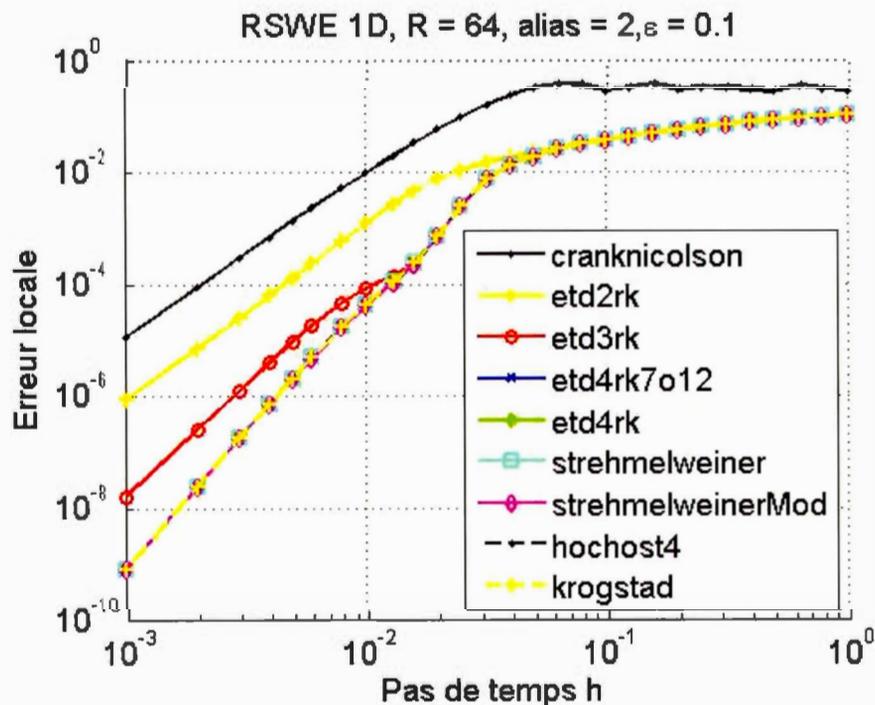


Figure 3.11 : L'erreur sur un pas de temps en fonction du pas de temps h pour SWE

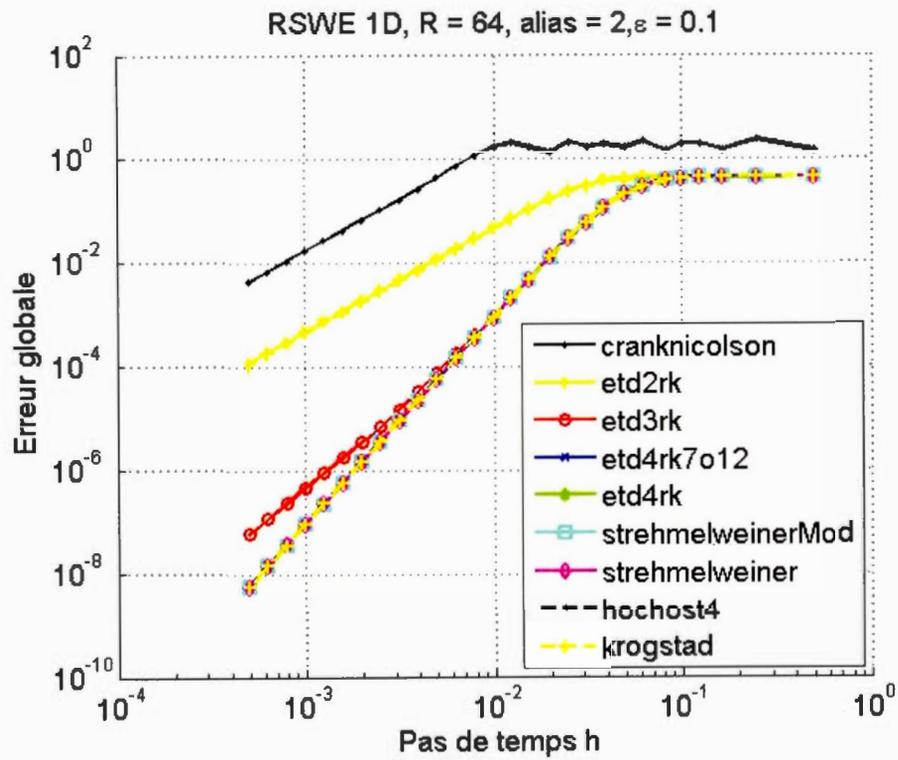


Figure 3.12 : L'erreur globale en fonction du pas de temps h pour SWE

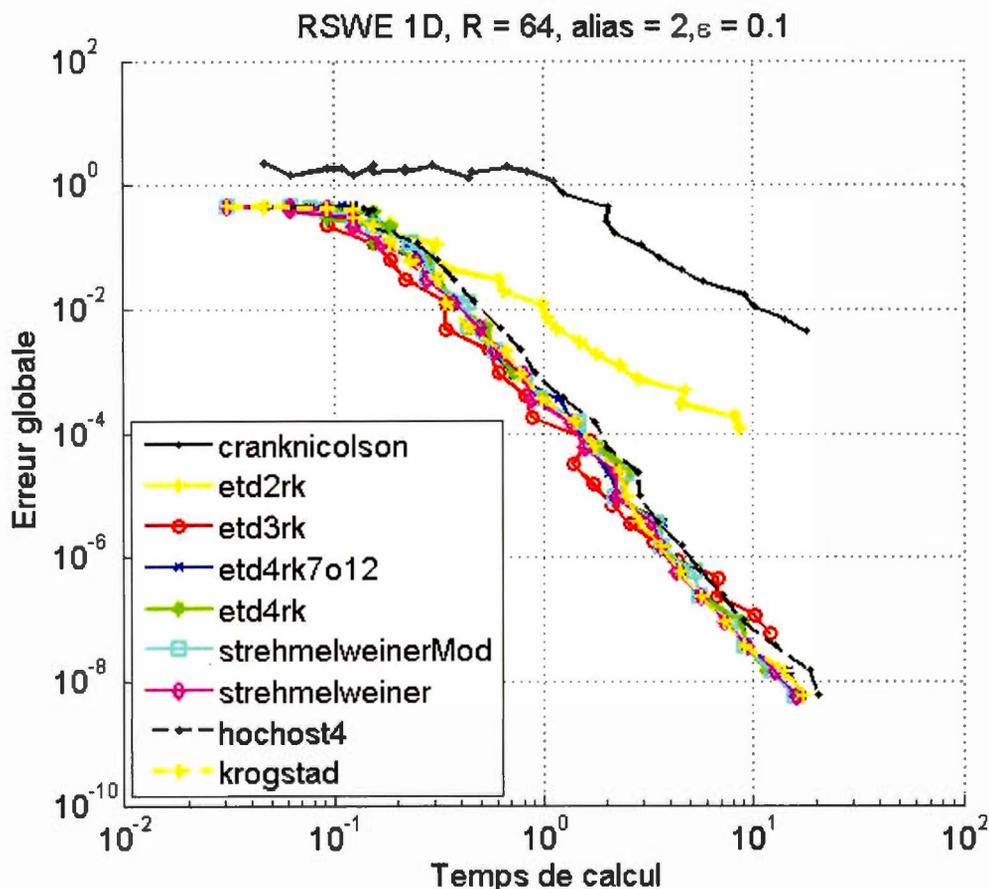


Figure 3.13 : L'erreur globale en fonction du temps de calcul utilisé pour SWE

L'analyse des figures 3.11 et 3.12 montre que toutes les méthodes d'IE sont plus précises que la méthode implicite ordinaire *cranknicolson* pour le système SWE et que toutes les méthodes ETDRK du 4^e ordre (*etd4rk*, *etd4rk7o12*, *strehmelweiner*, *strehmelweinerMod*, *hochost4*, *krogstad*) ont des courbes superposées. On remarque aussi que la méthode *etd2rk* a la même pente que celle de *cranknicolson* pour les petits pas de temps et elle finit par rejoindre la pente des méthodes des 3^e et 4^e ordres pour les grands pas de temps. Dans la figure 3.12, on constate que la méthode *etd3rk* rejoint la courbe des méthodes du 4^e ordre à partir de $h = 3 \times 10^{-3}$. Les méthodes du 4^e ordre à 4 étapes donnent la même précision que la méthode à 5 étapes (*hochost4*).

On peut déduire de la figure 3.13 que la méthode *etd3rk* est la plus efficace parce qu'elle donne autant de précision avec un moindre temps de calcul que les autres méthodes du 4^e ordre.

3.3.3 Test sur l'implémentation de la méthode Crank-Nicolson

Une question qui nous est venue lors de l'analyse des résultats est : pourquoi toutes les méthodes d'IE sont plus précises que la méthode implicite *cranknicolson* pour les SWE, alors qu'elle est utilisée avec succès dans plusieurs modèles météorologiques, dont GEM?

Pour répondre à cette question, on a examiné la manière dont elle était programmée dans la programmathèque EXPINT. Premièrement, elle utilise l'approximation eulérienne pour résoudre le terme non-linéaire alors que le modèle GEM utilise l'approximation semi-lagrangienne. De plus elle est programmée de manière itérative pour le terme non-linéaire avec quatre itérations. On a testé cette méthode avec six et huit itérations et on trouve que la précision ne change pas même si on augmente le nombre d'itérations (les trois courbes de *cranknicolson4*, *cranknicolson6* et *cranknicolson8* sont superposées dans les figures ci-dessous). Par contre, elle change si on change le paramètre ε qui représente le nombre de Rossby. Quand on prend les valeurs 0.01, 0.1 et 1, on obtient les figures suivantes :

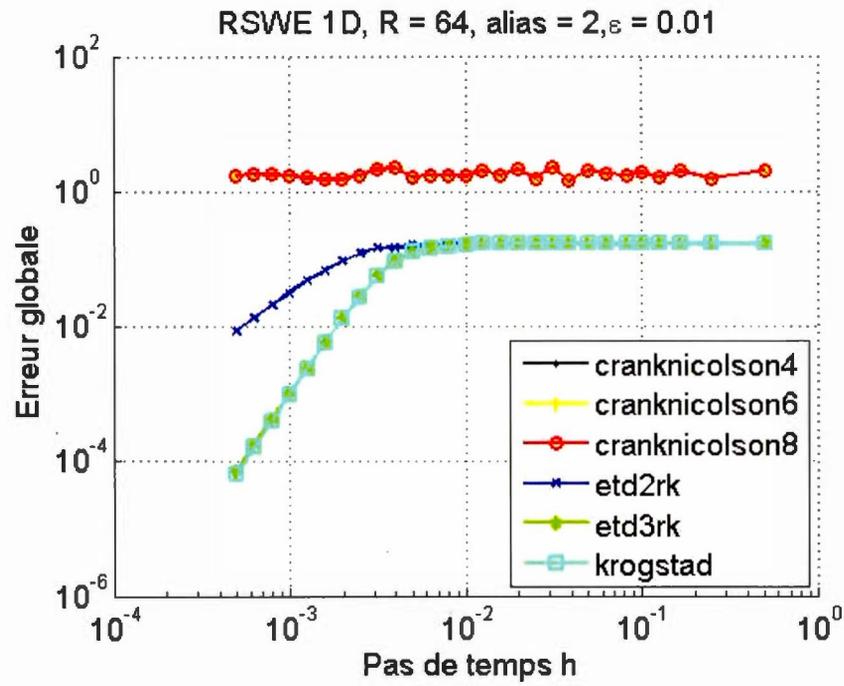


Figure 3.14 : L'erreur globale en fonction de h pour SWE avec $\varepsilon = 0.01$

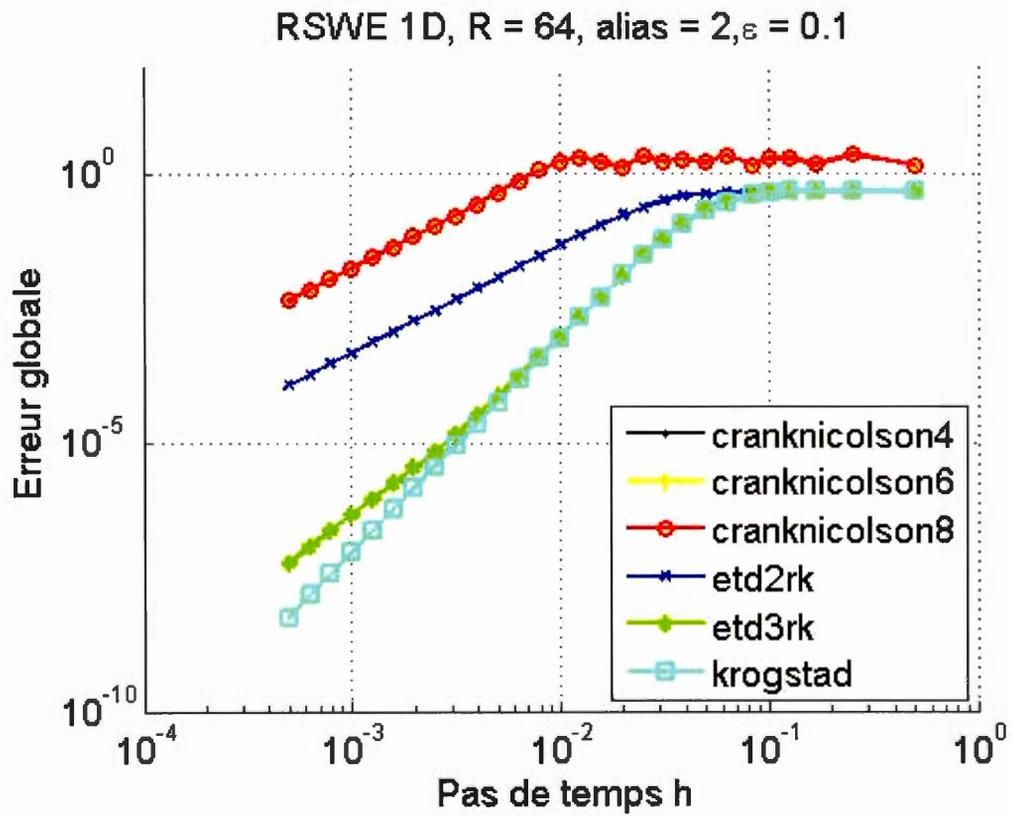


Figure 3.15 : L'erreur globale en fonction de h pour SWE avec $\varepsilon = 0.1$

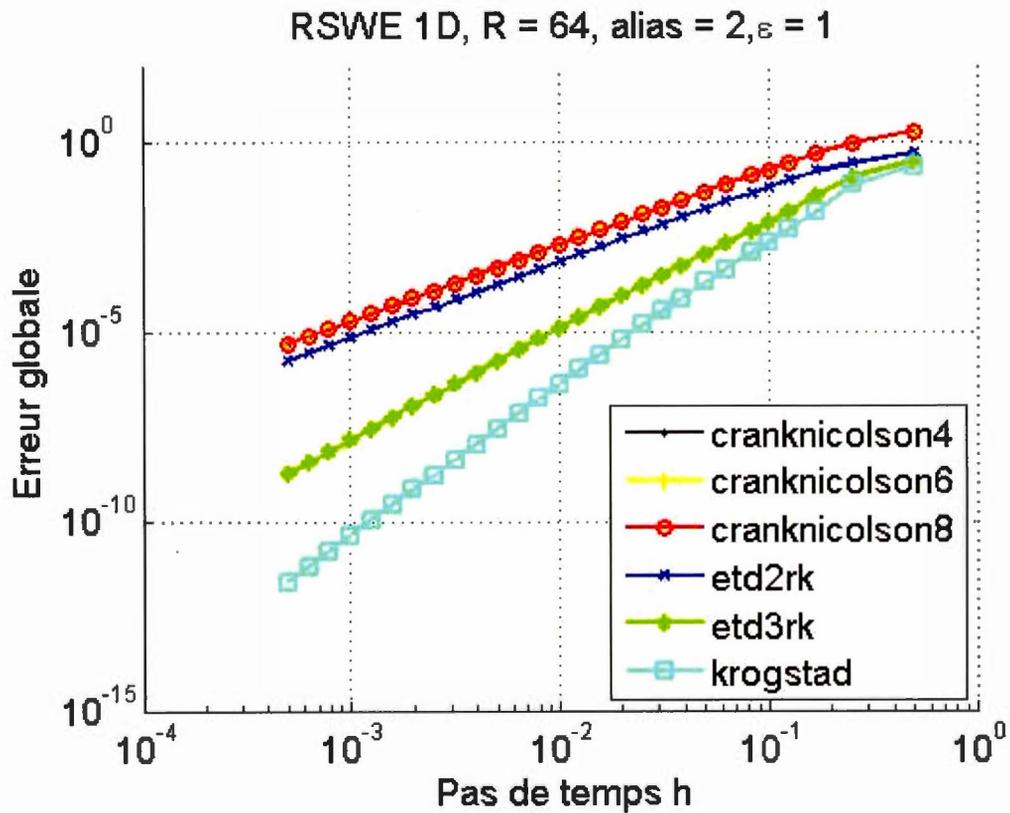


Figure 3.16 : L'erreur globale en fonction de h pour SWE avec $\varepsilon=1$

L'analyse des figures ci-dessus montre que pour $\varepsilon = 0.01$ les méthodes implicites (*cranknicolson4*, *cranknicolson6* et *cranknicolson8*) ne sont plus valables parce que l'erreur reste supérieure à 1, alors que plus on augmente ε plus les méthodes implicites (traits superposés) deviennent précises et tendent à rejoindre la méthode *etd2rk* (trait en bleu). On remarque aussi que la méthode *etd3rk* s'éloigne progressivement de la méthode *etd4rk* plus ε augmente.

3.3.4 Super-convergence de ETD3RK

Pour mieux comprendre les résultats obtenus dans la section 3.3.2, on examine dans cette section la super-convergence de la méthode *etd3rk*. On commence par le

développement en série de Taylor de $u(t)$ et $F(u)$ de notre équation en supposant que $F(t)$ dépend seulement de t , ce qui est une simplification de l'équation modèle 1.10 :

$$\frac{du}{dt} = cu + F(t), \quad (3.10)$$

Le développement de u en série de t jusqu'à l'ordre 5 est comme suit :

$$u(t) = u_0 + u_1 t + \frac{1}{2} u_2 t^2 + \frac{1}{6} u_3 t^3 + \frac{1}{24} u_4 t^4 + \frac{1}{120} u_5 t^5. \quad (3.11)$$

Le développement de F en série de t jusqu'à l'ordre 4 est :

$$F(t) = F_0 + F_1 t + \frac{1}{2} F_2 t^2 + \frac{1}{6} F_3 t^3 + \frac{1}{24} F_4 t^4. \quad (3.12)$$

En remplaçant le développement de $F(t)$ et $u(t)$ dans l'équation 3.10 on obtient :

$$\begin{aligned} & u_1 + u_2 t + \frac{1}{2} u_3 t^2 + \frac{1}{6} u_4 t^3 + \frac{1}{24} u_5 t^4 - c(u_0 + u_1 t + \frac{1}{2} u_2 t^2 + \frac{1}{6} u_3 t^3 + \frac{1}{24} u_4 t^4 + \frac{1}{120} u_5 t^5) \\ & + F_0 + F_1 t + \frac{1}{2} F_2 t^2 + \frac{1}{6} F_3 t^3 + \frac{1}{24} F_4 t^4 = 0 \end{aligned} \quad (3.13)$$

En regroupant les puissances semblables de t on obtient les coefficients de la solution :

$$\begin{aligned} u_1 &= cu_0 + F_0 \\ u_2 &= c^2 u_0 + cF_0 + F_1 \\ u_3 &= c^3 u_0 + c^2 F_0 + cF_1 + F_2 \\ u_4 &= c^4 u_0 + c^3 F_0 + c^2 F_1 + cF_2 + F_3 \\ u_5 &= c^5 u_0 + c^4 F_0 + c^3 F_1 + c^2 F_2 + cF_3 + F_4 \end{aligned} \quad (3.14)$$

On va maintenant comparer ces coefficients avec les coefficients de la solution de notre équation différentielle obtenue en utilisant les méthodes *etd3rk* et *etd4rk*. On convertit en série de Taylor la solution u_{etd3rk} , puis on calcule l'erreur. En

comparant l'erreur des méthodes *etd3rk* et *etd4rk*, on constate que les erreurs sont les mêmes. La différence entre la solution u_{etd3rk} et la solution u_{etd4rk} est nulle.

Les coefficients des erreurs jusqu'à l'ordre 4 sont nuls et le coefficient de l'erreur d'ordre 5 est :

$$err_{-5} = \frac{u_5}{120}. \quad (3.15)$$

Ceci explique en partie les résultats obtenus dans la section 3.3.2.

3.4 Synthèse partielle

Les tests préliminaires qu'on a faits montrent que les IE du 4^e ordre sont plus précis que la méthode implicite classique *cranknicolson* pour tous les problèmes étudiés. La méthode *strehmelweinerMod* est la plus efficace pour l'équation de Burgers et celle d'*Hochbruck-Ostermann*, tandis que c'est la méthode *krogstad* qui la plus efficace pour l'équation *kdv*. On a constaté aussi que la méthode ETD2 (*abnorsett2*) est moins précise que la méthode implicite pour les trois problèmes, alors que celle d'ETD4 (*abnorsett4*) présente une instabilité avec les grands pas de temps. En s'appuyant sur ces résultats, on a décidé de tester les méthodes ETDRK des 2^e, 3^e et 4^e ordres sur notre problème principal (SWE), tout en excluant les méthodes ETD r (*abnorsett2* et *abnorsett4*). Les résultats obtenus confirment bien nos attentes. Toutes les méthodes ETDRK donnent une précision meilleure que celle de *cranknicolson* et toutes les méthodes ETDRK du 4^e ordre donnent la même précision, mais c'est la méthode *etd3rk* qui est la plus efficace. On a aussi remarqué que la méthode *etd4rk7o12* donne des résultats aussi précis que les autres méthodes ETD4RK, même celle avec 5 étapes (*hochost4*). Ces résultats nous encouragent à faire d'autres tests plus approfondis avec ces méthodes. On a fait un dernier test pour essayer d'expliquer la bonne convergence

de *etd3rk*. On a trouvé que de l'erreur de la méthode *etd3rk* et celle de *etd4rk* sont égales. Ceci indique une super-convergence de la méthode *etd3rk* dans les conditions de notre étude.

CONCLUSION GÉNÉRALE

Dans la prévision numérique du temps, on est toujours à la recherche d'une meilleure méthode d'intégration temporelle pour résoudre les équations qui gouvernent l'évolution de l'atmosphère. Le choix d'une telle méthode est limité par deux contraintes : la précision et l'efficacité. Plusieurs approches ont été utilisées dans l'histoire de la modélisation atmosphérique pour contourner ces contraintes. Récemment, plusieurs chercheurs ont commencé à étudier la nouvelle approche des intégrateurs exponentiels explicites. Aujourd'hui on pense que ces intégrateurs, malgré qu'ils nécessitent un calcul complexe de l'exponentiel d'un opérateur, peuvent être avantageux. C'est dans ce contexte qu'on a étudié les méthodes d'intégration exponentielle pour la prévision du temps.

Dans un premier temps, nous avons présenté le problème des équations de Saint-Venant (SWE) en 1D avec des valeurs des paramètres réalistes, la méthode de discrétisation spatiale utilisée et les conditions initiales choisies. Ensuite, on a étudié les méthodes d'intégration temporelles classiques utilisées en météorologie et les méthodes d'intégration exponentielle existantes. En examinant les conditions d'ordre de rigidité on a trouvé une autre méthode du type ETD4RK qui satisfait mieux ces contraintes. Cette méthode est appelée *etd4rk7o12*. On a conclu ce chapitre par une étude de comparaison entre les zones de stabilité des méthodes. On a reproduit les résultats de Krogstad (2005), qui a étudié de stabilité des méthodes : ETD1, ETD2, ETD3 et ETD4RK, ainsi que de de Cox et Matthews (2002), qui ont étudié la stabilité des ETD2 et ETD2RK. De plus on a étudié d'autres méthodes : *abnorsett4* de type ETD, ainsi que *etd3rk*, *etd4rk7o12*, *hochost4*, *strehmelweiner*, *strehmelweinerMod* et

krogstad de type ETDRK. Ceci nous amène à conclure que les méthodes ETDRK des 3^e et 4^e ordres seraient les meilleures pour les problèmes tels que les SWE.

Dans le dernier chapitre on commence par des tests préliminaires avec ces intégrateurs sur des problèmes classiques (l'équation de Burgers , l'équation kdv et celle d'Hochbruck-Ostermann). On a trouvé que les méthodes ETDRK du 4^e ordre donnent de très bons résultats, ce qui nous a encouragés à passer à la résolution des SWE 1D avec ces méthodes. Les résultats obtenus confirment bien notre attente. Les méthodes ETDRK des 3^e et 4^e ordres sont beaucoup plus précises que la méthode implicite classique utilisée actuellement et la méthode *etd3rk* est la plus efficace avec les paramètres choisis qui sont représentatifs des écoulements atmosphériques. La très bonne performance de la méthode *etd3rk* nous amener à comparer cette méthode à *etd4rk* et on a trouvé que dans certaines conditions elle donne les mêmes résultats.

Ces résultats nous incitent à investiguer ces intégrateurs exponentiels en 2D et 3D.

ANNEXE A

LA SOLUTION DE L'ÉQUATION DE BURGERS PÉRIODIQUE

La solution de l'équation de Burgers est obtenue à l'aide de la transformation de Cole-Hopf. En effet on montre facilement que si on définit $u(x,t)$ comme étant

$$u(x,t) = -2\mu \frac{\partial \ln(\phi(x,t))}{\partial x}, \quad (\text{A.1})$$

et que $\phi(x,t)$ satisfait l'équation de diffusion linéaire suivante :

$$\frac{\partial \phi(x,t)}{\partial t} = \mu \frac{\partial^2 \phi(x,t)}{\partial x^2} \text{ dans le domaine } x \in [-\pi, \pi], \quad (\text{A.2})$$

avec la condition initiale

$$\frac{d \ln(\phi(x,0))}{dx} = \frac{u(x,0)}{-2\mu}, \quad (\text{A.3})$$

alors $u(x,t)$ est bien la solution recherchée.

Une solution de (A.2) est simplement :

$$\phi(x,0) = \exp\left(-\int^x \frac{u(x',0)}{2\mu} dx'\right). \quad (\text{A.5})$$

Si $u(x,0)$ est simplement $\sin(x)$, alors à une constante près :

$$\phi(x,0) = cte \exp\left(\frac{\cos(x)}{2\mu}\right), \quad (\text{A.6})$$

et comme (A.5) admet le développement de Fourier suivant (Abramowitz et Stegun, 1968, éq. 9.6.34) :

$$\phi(x,0) = cte \sum_{m=0}^{\infty} (2 - \delta_{m0}) I_m\left(\frac{1}{2\mu}\right) \cos(mx), \quad (\text{A.7})$$

où I_m est la fonction de Bessel modifiée d'ordre m . La solution du problème linéaire (A.2) est ensuite obtenue terme à terme:

$$\phi(x,t) = cte \sum_{m=0}^{\infty} (2 - \delta_{m0}) I_m\left(\frac{1}{2\mu}\right) \exp(-m^2 \mu t) \cos(mx), \quad (\text{A.8})$$

d'où on tire la solution $u(x,t)$. Pour obtenir des résultats numériques, on doit tronquer la somme de (A.7) après un nombre $N+1$ de termes où N est suffisamment grand pour garantir la précision souhaitée. On peut choisir la constante arbitraire cte de telle sorte que le premier coefficient de la série soit 1.

ANNEXE B

LES TABLEAUX DE BUTCHER DES MÉTHODES ETRK

Voici les tableaux de Butcher des méthodes citées dans ce mémoire (Berland *et al.*, 2005) :

etd2rk :

0			1
1	φ_1		φ_0
	$\varphi_1 - \varphi_2$	φ_2	φ_0

etd3rk :

$$\begin{array}{ccc|ccc}
 0 & & & & & & 1 \\
 \frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & & \varphi_{0,2} \\
 1 & -\varphi_1 & & 2\varphi_1 & & & \varphi_0 \\
 \hline
 & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 4\varphi_2 - 8\varphi_3 & -\varphi_2 + 4\varphi_3 & & & \varphi_0
 \end{array}$$

etd4rk :

$$\begin{array}{ccc|ccc}
 0 & & & & & & 1 \\
 \frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & & \varphi_{0,2} \\
 \frac{1}{2} & & & \frac{1}{2}\varphi_{1,2} & & & \varphi_{0,2} \\
 1 & \frac{1}{2}\varphi_{1,2}(\varphi_{0,2} - 1) & & \varphi_{1,2} & & & \varphi_0 \\
 \hline
 & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & -\varphi_2 + 4\varphi_3 & & \varphi_0
 \end{array}$$

krogstad :

0				1
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2}$			$\varphi_{0,2}$
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2} - \varphi_{2,2}$	$\varphi_{2,2}$		$\varphi_{0,2}$
1	$\varphi_1 - 2\varphi_2$		$2\varphi_2$	φ_0
	$\varphi_1 - 3\varphi_2 + 4\varphi_3$	$2\varphi_2 - 4\varphi_3$	$2\varphi_2 - 4\varphi_3$	$-\varphi_2 + 4\varphi_3$
				φ_0

Strehmel-Weiner :

0		1
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2}$	$\varphi_{0,2}$
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2} - \frac{1}{2}\varphi_{2,2}$	$\varphi_{0,2}$
1	$\varphi_1 - 2\varphi_2$	φ_0
	$\varphi_1 - 3\varphi_2 + 4\varphi_3$	φ_0
	0	
	$4\varphi_2 - 8\varphi_3$	
	$-\varphi_2 + 4\varphi_3$	

hochost4 :

0						1
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2}$					$\varphi_{0,2}$
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2} - \varphi_{2,2}$	$\varphi_{2,2}$				$\varphi_{0,2}$
1	$\varphi_1 - 2\varphi_2$	φ_2	φ_2			φ_0
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2} - 2a_{52} - a_{54}$	a_{52}	a_{52}	a_{54}		$\varphi_{0,2}$
	$\varphi_1 - 3\varphi_2 + 4\varphi_3$	0	0	$-\varphi_2 + 4\varphi_3$	$4\varphi_2 - 8\varphi_3$	φ_0

Avec $a_{52} = \frac{1}{2}\varphi_{2,2} - \varphi_3 + \frac{1}{4}\varphi_2 - \frac{1}{2}\varphi_{3,2},$ $a_{54} = \frac{1}{2}\varphi_{2,2} - a_{52}$

ANNEXE C

CODE DES ÉQUATIONS DE SAINT-VENANT EN 1D

Définition du problème SWE-1D

```
function problem = rsweld(varargin)
% RSWE1D - Generates a problem structure for the 1D Rotating Shallow
Water Equations
% domain 0 < x < 2pi with periodic BCs using spectral method
% d/dt,d/dx = partial derivatives
% du/dt = - (-v + (F)^(-1/2)*dh/dx)/epsilon - u * du/dx - mu *
d4u/dx4
% dv/dt = - u/epsilon - u * dv/dx - mu *
d4v/dx4
% dh/dt = - ((F)^(-1/2)*du/dx)/epsilon - d(h*u)/dx - mu *
d4h/dx4
% in spectral space for each component k from 0 to R
% SYNOPSIS:
%   problem = rsweld;
%   problem = rsweld(problemdata);
%   problem = rsweld('f1', v1, ..., 'fn', vn);
%% PARAMETERS:
%   problemdata - OPTIONAL STRUCT specifying problem parameters.
%   The following fields are accepted:
%%       R - Spectral truncation
%           Default value: R = 64;
%   alias - Control of aliasing
%           Default value: alias = 2;
%   IC - Initial condition.
%         See source code for applicable names.
%           Default value: IC = 'smooth';
% continuity - Continuity of initial conditions
%           Default value: continuity = -1;
%   mu - Diffusion coefficient
%           Default value: mu = 1.0e-4;
%   exp_dif - exponent of diffusion.
%           Default value: exp_dif = 4;
%   F - Froude number
%           Default value: F = 1
```

```

%      epsilon - Rossby parameter
%                  Default value: epsilon = 1.0e-2;
% 'f1', ..., vn -
%                  OPTIONAL field/value pairs similar to STRUCT
%                  constructor. Valid field names are the same as for
%                  `problemdata'.
% RETURNS:
% problem - Structure containing problem specific parameters of the
%           equation. `problem' has at least the following fields:
%           x - Vector of all points in physical space.
%           y0 - Initial condition as a vector
%           L - ODE linear operator.
%           N - Name of function evaluating ODE non-linear operator.
% problemname -
%           String with relevant details of specific problem.
%           May be used in plot titles and filenames.
% This file is part of the 'Expint'-package,
% see http://www.math.ntnu.no/num/expint/
% $Revision: 1.0 $ $Date: 2016/03/03 00:00:00 $
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1) Handle default and user choices:
user_val = struct([]);
default_val = struct('R', 64, ...
                    'alias', 2, ...
                    'IC', 'smooth', ...
                    'continuity', -1, ...
                    'mu', 1.0e-4, ...
                    'exp_dif', 4, ...
                    'F', 1, ...
                    'epsilon', 1.0e-2, ...
                    'length', 2*pi ); % Spatial domain size (1D).

if nargin > 0,
    user_val = makestruct(varargin{:});
end
s = mergestructs(default_val, user_val);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2) Discretisation and equation specific parameters
problem.R = s.R;
problem.alias = s.alias;
problem.mu = s.mu;
problem.exp_dif = s.exp_dif;
problem.F = s.F;
problem.epsilon = s.epsilon;
problem.length = s.length;
problem.M = 2 * s.R + 2; % linear grid (display grid)
problem.n = ( 1 + s.alias ) * s.R + 1; % non-linear grid
if rem( problem.n, 2 ) ~= 0
    problem.n = problem.n+1;
end
problem.ND = 3 * ( s.R + 1 );
% Wave numbers in Fourier space (positive values):
I = sqrt( -1 );

```

```

problem.der_vec = I * (0:1:s.R);
% Spatial discretisation points for display in x-direction.
problem.x = 0 : s.length/problem.M : s.length;
% Matrix array of Fourier modes (linear part including diffusion):
[ problem.my_eigen, my_val ] = create_eig_big_L_D( s.F, s.R,
                                                s.epsilon, s.mu, s.exp_dif );
problem.L = reshape( my_val, problem.ND, 1 );
my_eig_inv = zeros( 3, 3, s.R+1);
for k = 0 : s.R
    for i = 1 : 3
        for j = 1 : 3
            my_eig_inv(i,j,k+1)= conj( problem.my_eigen(j,i,k+1) );
        end
    end
end
end
problem.my_eig_inv = my_eig_inv;
% display( problem.my_eig_inv )
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3) Initial condition
switch lower(s.IC);
    case { 'smooth' }
        % Initial Conditions (filter high-end of spectrum)
        n_continuity = s.continuity;
        if n_continuity == -1
            [ h_0, x, n_0, n_1, n_2, n_3, n_4 ] =
initial_conditions( problem.M, -1 );
            disp( [ 'n_0, n_1, n_2, n_3, n_4 = ' num2str(n_0) ' '
num2str(n_1) ' ' num2str(n_2) ' ' num2str(n_3) ' ' num2str(n_4) ] )
        elseif n_continuity == 0
            [ h_0, x, n_0, n_1, n_2, n_3, n_4 ] =
initial_conditions( problem.M, 0 );
            disp( [ 'n_0, n_1, n_2, n_3, n_4 = ' num2str(n_0) ' '
num2str(n_1) ' ' num2str(n_2) ' ' num2str(n_3) ' ' num2str(n_4) ] )
        elseif n_continuity == 1
            [ h_0, x, n_0, n_1, n_2, n_3, n_4 ] =
initial_conditions( problem.M, 1 );
            disp( [ 'n_0, n_1, n_2, n_3, n_4 = ' num2str(n_0) ' '
num2str(n_1) ' ' num2str(n_2) ' ' num2str(n_3) ' ' num2str(n_4) ] )
        elseif n_continuity == 2
            [ h_0, x, n_0, n_1, n_2, n_3, n_4 ] =
initial_conditions( problem.M, 2 );
            disp( [ 'n_0, n_1, n_2, n_3, n_4 = ' num2str(n_0) ' '
num2str(n_1) ' ' num2str(n_2) ' ' num2str(n_3) ' ' num2str(n_4) ] )
        end
        figure( 1 )
        plot(x, h_0, 'k-');
        un = zeros(1,problem.M);
        vn = zeros(1,problem.M);
        hn(1,1:problem.M) = h_0(1:problem.M);
        y = zeros(3,problem.M);
        for j = 1 : problem.M
            y(1,j) = un(j);

```


Terme non-linéaire

```

function N = rsweld_N(y, t, problem)
% rsweld_N - Non-linear term of Rotating Shallow-Water Equations in
1D.
% SYNOPSIS:
%   N = rsweld_N(y, t, problem);
% DESCRIPTION:
%   Evaluates the non-linear term of the semi-discretised
%   RSWE 1d problem.
% PARAMETERS:
%   y           - Evaluation point in Fourier space
%                 (rolled out to vector)
%   t           - Evaluation point in time. Not used in this function.
%   problem     - Problem dependent parameters.
%                 Should be defined by rsweld.
% RETURNS:
%   N           - Value of non-linear term at 'U'.
% SEE ALSO:
%   RSWE1D
% This file is part of the 'Expint'-package,
% see http://www.math.ntnu.no/num/expint/
% $Revision: 1.0 $ $Date: 2016/04/05
% reshape into matrix form, in fourier space:
y_mode = reshape( y, 3, problem.R+1 );
yf = zeros( 3, problem.R+1 );
for k = 0 : problem.R
    yf( :, k+1 ) = problem.my_eigen( :, :, k+1 ) * y_mode( :, k+1 );
end
% compute derivative in fourier space
yd = yf;
for i_c = 1 : 3
    yd(i_c,:) = problem.der_vec .* yd(i_c,:);
end
% transform to physical space on non-aliasing grid
yp = Fourier2real( yf, problem.n, 3 );
dy = Fourier2real( yd, problem.n, 3 );
% u = yp(:,1); v = yp(:,2) ; h = yp(:,3);
% ux = yd(:,1); vx = yd(:,2) ; hx = yd(:,3);
yp( 2, : ) = - yp( 1, : ) .* dy( 2, : );
yp( 3, : ) = - yp( 1, : ) .* dy( 3, : ) - dy( 1, : ) .* yp( 3, : );
yp( 1, : ) = - yp( 1, : ) .* dy( 1, : );
% reshape into vector form in fourier space:
yf = real2Fourier( yp, problem.n, 3 );
for k = 0 : problem.R
    y_mode(:, k+1) = problem.my_eig_inv(:, :, k+1) * yf(:, k+1);
end
% y_mode = zeros( 3, problem.R+1 );
N = reshape( y_mode, problem.ND, 1 )

```

Post-traitement

```

function yprocessed = rsweld_post(y, problem)
% rsweld_post- Postprocessing of numerical solution to the
%               RSWE 1D equation.
% SYNOPSIS:
%   yprocessed = rsweld_post(y, problem);
% PARAMETERS:
%       y - Numerical solution at some point in time, row-vector of
%           dimension ND (rolled out).
%   problem - Problem specific parameters.
% RETURNS:
%   yprocessed - Processed numerical solution 'un, vn, hn', in
%               matrix form.
% SEE ALSO:
%   rsweld, EXPGLM
% This file is part of the 'Expint'-package,
% see http://www.math.ntnu.no/num/expint/
% $Revision: 1.4 $ $Date: 2016/04/05 02:50:13 $
% reshape into matrix form, still in mode + fourier space:
y = reshape( y, 3, 1+problem.R);
yf = zeros( 3, 1+problem.R );
for k = 0 : problem.R
    yf(:,k+1 ) = problem.my_eigen( :, :, k+1 ) * y( :, k+1 );
end
% Into physical space (periodicity is enforced):
yprocessed = Fourier2real( yf, problem.M, 3 );
yprocessed = [ yprocessed yprocessed(:,1) ];
return
end

```

RÉFÉRENCES

Abramowitz M. et Stegun I. eds, (1968). *Handbook of Mathematical Functions*. Dover Publication INC., New York, 1045 pp.

Ascher M., et Greif C. (2011). *A First Course in Numerical Methods*. SIAM Computational Science and Engineering. The Society for Industrial and Applied Mathematics, 551 pp.

Bateman H., (1915). *Some recent researches on the motion of fluids*. Mon. Weather Review, **43**, 163-170.

Berland H., Skaflestad, B. et Wright, W.M. (2007). *EXPINT: A MATLAB package for exponential integrators*. Journal ACM Transactions on Mathematical Software (TOMS). **33**, 4:1- 4:17.

Clancy C. et Pudykiewicz, J.A., (2013). *On the use of exponential time integration methods in atmospheric models*. Tellus A. **65**, 1-16.

Certaine J., (1960). *The solution of ordinary differential equations with large time constant*. Mathematical methods for digital computers, 128–132.

Côté J., Gravel, S. Méthot A., Patoine, A. Roch, M. et Staniforth, A., (1998). *The Operational CMC–MRB Global Environmental Multiscale (GEM) Model. Part I: Design Considerations and Formulation*. Mon. Weather Review, **126**, 1373-1395.

Cox S.M. et Matthews P.C., (2002). *Exponential time differencing for stiff systems*. J. Comput. Phys. **176**, 430–455.

Durrán D.R., (2010). *Numerical Methods for Fluid Dynamics with applications to geophysics* (2nd ed.). Texts in applied mathematics **32**. Springer, New York, 516 pp.

Eichwald B., (2013). *Intégrateurs exponentiels modifiés pour la simulation des vagues non linéaires*. Thèse de doctorat, Université Nice, Sophia Antipolis, 130 pp.

Gander, W., Gander, M.J. et Kwok F., (2010). *Scientific computing: An Introduction using Maple and MATLAB*. Texts in computational science and engineering **11**. Springer, 905 pp.

Gaudreault S., et Pudykiewicz J., (2016). *An efficient exponential time integration method for the numerical solution of the shallow water equations on the sphere*. Journal of Computational Physics. **322**, 827–848.

Haut T. et Wingate B., (2014). *An asymptotic parallel-in-time method for highly oscillatory PDEs*. SIAM Journal on Scientific Computing. **36**, A693-A713.

Hochbruck M. et Ostermann A., (2005). *Explicit exponential runge-kutta methods for semilinear parabolic problems*. SIAM J. Numer. Anal. **43**, 1069-1090.

Hochbruck M. et Ostermann A., (2010). *Exponential integrators*. Acta Numerica. **19**, 209-286.

Krogstad S., (2005). *Generalized integrating factor methods for stiff PDEs*. J. Comput. Phys. **203**, 72–88.

Lauritzen P.H., Jablonowski C., Taylor M.A. et Nair R.D. eds, (2011). *Numerical Techniques for Global Atmospheric Models*. LNCSE **80**. Springer, 556 pp.

Lin C., Laprise, R. et Ritchie H. eds, (1997). *Numerical Methods in Atmospheric and Oceanic Modelling: the André J. Robert Memorial Volume*. SCMO/CNRC, 581 pp+52 pages suppl.

Machenhauer B., (1979). *The spectral method*, in A. Kasahara (ed.), *Numerical Methods Used in Atmospheric Models*, Vol. 2, GARP Publications Series No 17, WMO and ICSU, Geneva, 121-275.

Majda A., (2003). *Introduction to PDEs and waves for the atmosphere and ocean*. Courant Lecture Notes in Mathematics. American Mathematical Society, 234 pp.

Minchev, B.V et Wright, W.M, (2005). *A review of exponential integrators for first order semi-linear problems*. Rapport Technique 2/2005, Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway, 44 pp.

Pedlosky J., (1979). *Geophysical Fluid Dynamics*. Springer-Verlag, New York Heidelberg Berlin, 626 pp.

Satoh M., (2014). *Atmospheric Circulation Dynamics and General Circulation Models*. (2). Springer-Verlag Berlin Heidelberg, 730 pp.

Strang G., (2007). *Computational Science and Engineering*. Wellesley-Cambridge Press, 716 pp.

Tokman M., (2005). *Efficient integration of large stiff system of ODEs with exponential programming iterative (EPI) methods*. J. Comput. Phys. **213**, 748-776.

Vallis G. K., (2017). *Climate and the Oceans*. Cambridge University Press, 964 pp.