

Nouvel outil de prototypage d'équations différentielles, application aux modèles de films

N. Cellier^a, C. Ruyer-Quil^b

a. Université Grenoble Alpes contact@nicolas-cellier.net

b. Université Grenoble Alpes christian.ruyer-quil@univ-smb.fr

Résumé :

Scikit-fdiff (anciennement Triflow) a été développé en interne afin de faciliter le prototypage de modèles mathématiques. Il a été créé afin d'essayer rapidement les différents modèles de films ruisselants avec divers couplages (thermique, transfert de masse) obtenus par approche asymptotique.

La méthode des lignes est utilisée : les dérivées spatiales sont discrétisées pour mener à un système d'équations ordinaires. Il est alors possible d'utiliser un solveur temporel pour résoudre ce système.

Scikit-fdiff automatise ces étapes en effectuant symboliquement la discrétisation par différences finies. Cette discrétisation inclut la prise en compte de conditions limites complexes, et est capable de gérer un nombre arbitraire de dimensions. La matrice jacobienne associée est elle aussi obtenue par calcul formel, évitant son approximation numérique. Ces formes symboliques sont ensuite traduites en routine numérique accessible par les solveurs temporels implémentés. En particulier, des schémas temporels implicites d'ordres élevés (type Rosenbrock-Wanner) ont été inclus dans la librairie. Tous les solveurs possèdent un adaptateur de pas de temps. Ces deux dernières fonctionnalités (solveurs implicites et adaptateur de pas de temps) rendent le logiciel adapté à la résolution de problèmes fortement non-linéaires.

Les différences finies ont été choisies pour leur côté versatile : cette méthode est capable de discrétiser n'importe quelle équation aux dérivées partielles indépendamment de leur structure. Étant une méthode intrinsèquement locale, elle mène à un système numérique très creux que les algorithmes actuels sont capables de résoudre efficacement.

Le logiciel a été validé sur le problème de la rupture de barrage ainsi que sur celui du lac au repos, tous deux modélisés par une équation de Saint-Venant. Il a également été appliqué avec succès à la résolution de films ruisselants couplés aux transferts de chaleur modélisés par une approche intégrale, ainsi qu'à des problématiques d'écoulement de gouttes modélisé par l'équation de la lubrification.

Abstract :

Scikit-fdiff (formally Triflow) has been developed in order to facilitate mathematic models building. It has been made to quickly build and try many asymptotic falling film modelling with different phenomena coupling (energy and mass transfer).

It uses the method of line : temporal derivatives are discretized first, which leads to a system of ordinary differential equations. A proper temporal solver can then be used to solve this system.

Scikit-fdiff makes these steps easier, using symbolic computation to operate the finite difference discretization, taking arbitrary boundary conditions into account. The derivative matrix (also known as Jacobian matrix) is also obtained via symbolic computation, avoiding a costly and potentially inaccurate numerical approximation. Both evolution equations and its derivatives are translated into numerical routines which can be used by the available temporal solvers. High order implicit schemes have been implemented. All the available solvers come with a time step controller. These two features (implicit solvers and adaptative time-stepping) make the software able to solve stiff and highly non-linear problems.

Finite differences have been chosen for their versatility : this method can discretize any partial differential equation independently from their structure. Due to its local nature, the method leads to intrinsically sparse numerical system, easily solved by the proper algorithm.

Scikit-fdiff have been validated on the dam-break and the steady lake study case. It has also been successfully applied on heated falling film and on droplet flow models.

Mots clefs : Modélisation, équations aux dérivés partielles, python, open-source, différences finies, méthode des lignes

1 Introduction

Afin de modéliser des phénomènes complexes, les physiciens ont besoin de travailler sur de nombreux modèles décrits par des systèmes d'équations aux dérivées partielles (EDP), dont les propriétés et les structures mathématiques peuvent grandement différer les unes des autres. Il arrive régulièrement que ces modèles ne soient pas compatibles avec les codes industriels, ou alors au prix de nombreuses opérations algébriques. Dans ces cas, des "codes maisons" sont souvent développés. Ces codes sont alors réutilisés au fur et à mesure du temps, modifiés pour s'adapter à de nouvelles problématiques. Ils sont cependant rarement disponibles hors d'une sphère d'utilisation restreinte, ne sont pas toujours correctement testés, et il peut être de plus en plus complexe de les adapter à de nouveaux problèmes.

Ces codes utilisent souvent les différences finies pour discrétiser le modèle mathématique primitif. Cela s'explique par la simplicité et la versatilité de cette méthode. En contrepartie, elle est difficilement applicable à des domaines complexes, ce qui explique sa relative absence au sein des codes industriels.

Un nouvel outil permettant de formaliser et de résoudre rapidement des modèles mathématiques a été développé, et permet aujourd'hui de résoudre des systèmes d'équations aux dérivés partielles avec un nombre arbitraire de dimensions. Cet outil se concentre sur la facilité d'écriture et de résolution plus que sur la performance de la routine obtenue, tout en assurant que la résolution puisse se faire en un temps raisonnable sur des machines "standard" (PC portables, petites stations de calcul). Il sera possible dans un future proche d'adapter son utilisation à des machines haute performances.

L'outil a été développé sous licence open-source, et est donc disponible à l'utilisation et à la modification par toute personne ou institution le souhaitant. Il suit une série de bonnes pratiques incluant l'utilisation de tests unitaires, l'écriture d'une documentation complète et l'utilisation d'outils d'intégration continue pour assurer son bon fonctionnement au fil des versions sur les plateformes standards (linux, windows).

2 Mécanismes internes

Le fonctionnement de scikit-fdiff repose sur la méthode des lignes. Elle consiste à opérer la discrétisation des systèmes d'équations aux dérivées partielles en deux temps. Tout d'abord, les dérivées spatiales sont remplacées par une approximation discrète selon une méthode choisie, les différences finies dans le cas de scikit-fdiff. Cette étape transforme un système d'équations aux dérivées partielles en un système d'équations différentielles ordinaire de taille beaucoup plus élevée. Ce système est alors résolu avec des solveurs adaptés dont font partie les solveurs multi-étapes de type Runge-Kutta (implicites ou explicites). Scikit-fdiff utilise par défaut des méthodes dites Rosenbrock-Wanner [3], particulièrement adaptées à des problèmes raides.

La discrétisation des dérivées spatiales est faite par calcul formel. Associée à un moteur de résolution des conditions limites, elle permet d'obtenir une description du système discret par morceaux. Cette description mathématique discrète est alors transformée en routine de résolution numérique à travers un *backend*. Ce fonctionnement modulaire donne au logiciel la possibilité d'être étendu à des résolutions hautes performances. Le backend par défaut utilise *numpy*, la librairie python dédiée à la manipulation de structures de données numériques. Il est cependant possible d'écrire un nouveau backend utilisant un langage bas niveau comme du Fortran, du C ou même d'écrire un kernel spécialisé dans la résolution numérique sur GPU type cuda.

La même opération est effectuée sur la matrice Jacobienne $J = \frac{\partial F}{\partial U}$. Dans le cas d'un système obtenu par différence finies, elle possède une structure très creuse. L'obtention de sa description symbolique permet de la calculer de façon très performante en limitant les erreurs d'approximation. De plus, un système intelligent de vectorisation des données d'entrées permet de réduire la taille de la bande de la matrice Jacobienne obtenue. La figure 1 montre la différence de structure obtenue dans le cas d'un problème 2D couplé décrit par l'équation (1) et un domaine périodique.

$$\begin{cases} \frac{\partial u}{\partial t} &= \partial_{xx}v \times \partial_{xx}u + \partial_{yy}v \times \partial_{yy}u \\ \frac{\partial v}{\partial t} &= \partial_{xx}u + \partial_{xx}v + \partial_{yy}u + \partial_{yy}v \end{cases} \quad (1)$$

3 Application à des modèles de mécanique des fluides

Chaque exemple donné dans cette section comportera une description succincte de la physique et le modèle mathématique associé, ainsi que les résultats de simulation. Le code source permettant de reproduire les résultats avec le logiciel est disponible à l'adresse : <https://nbviewer.jupyter.org/gist/celliern/c9f408c905b2cd533dcb7c77ffbe9c851>

3.1 Vague acoustique 2D

Une vague acoustique, ou onde sonore, est une onde mécanique qui se propage par compression et décompression successive de son milieu de propagation. Elle se propage à la vitesse du son, celui-ci dépendant du milieu.

Elle est décrite par l'équation aux dérivées partielles vectorielles suivantes :

$$\nabla^2 p - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = 0$$

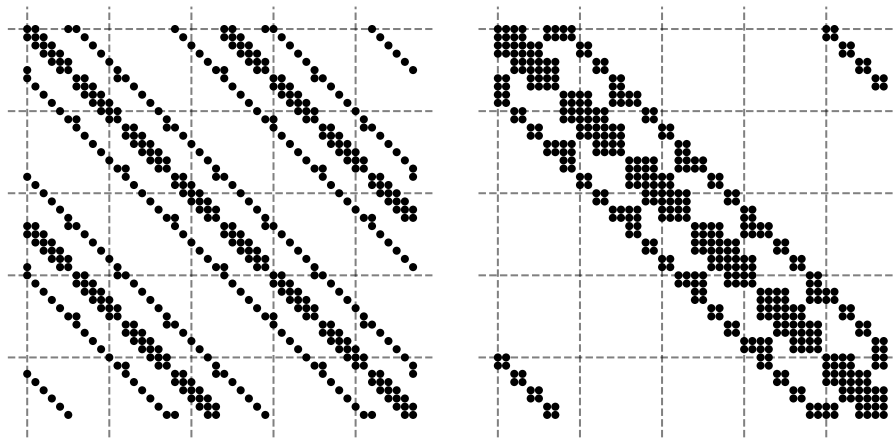


FIGURE 1 – Structure de la Jacobienne obtenue pour une vectorisation naïve (à gauche) et pour une vectorisation optimisée (à droite), chaque point représentant une valeur non-nulle. Après optimisation, les valeurs non-nulles se répartissent autour de la diagonale, structure plus adaptée aux solveurs modernes.

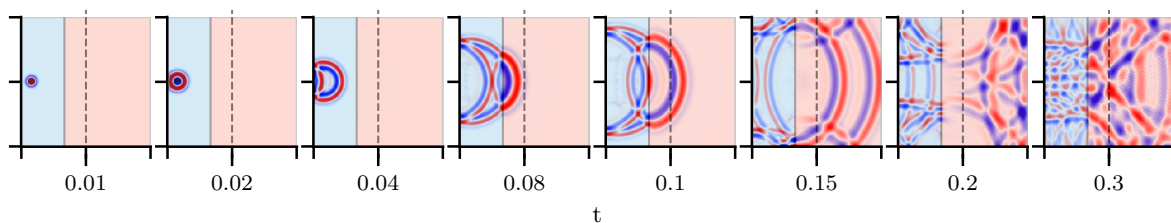


FIGURE 2 – Pression relative d'une onde acoustique se propageant à travers deux milieux différents. La vitesse du son au sein du domaine bleu (à gauche) est supérieure à la vitesse du son dans le milieu rouge (à droite).

La simulation proposée montre un phénomène de réfraction de ces vagues acoustiques lors de leur passage entre deux milieux dans lesquels les vitesses du son diffèrent. Les conditions limites représentent des surfaces réfléchissantes.

Des schémas upwind seront utilisés pour limiter les instabilités numériques occurrentes lors de la résolution d'équations où l'advection domine.

Différents instantanés de la pression sont visibles sur la figure 2.

3.2 Équation Saint Venant 2D, rupture de barrage

Les équations de Saint Venant modélisent des écoulements de liquide lorsque le profil de vitesse est quasiment indépendant de la position verticale. La sous-couche laminaire parabolique au contact avec le fond du support d'écoulement est alors d'épaisseur négligeable vis à vis de la hauteur d'eau, et les vecteurs vitesses sont indépendants de la profondeur.

Ces équations sont très utilisées pour décrire les cours d'eau ou les tsunamis. Dans l'exemple suivant, la position du fond est considérée constante. Le cas étudié est celui de la rupture du barrage : la condition initiale représente une discontinuité d'élévation brusque entre deux domaines (ici, $h_1 = 1m$, $h_2 = 2m$). La zone en surélévation suit un quart de cercle dans le coin inférieur gauche du domaine de résolution.

Ce problème (dans sa version 1D et 2D) représente un cas important de validation. On s'attend à obser-

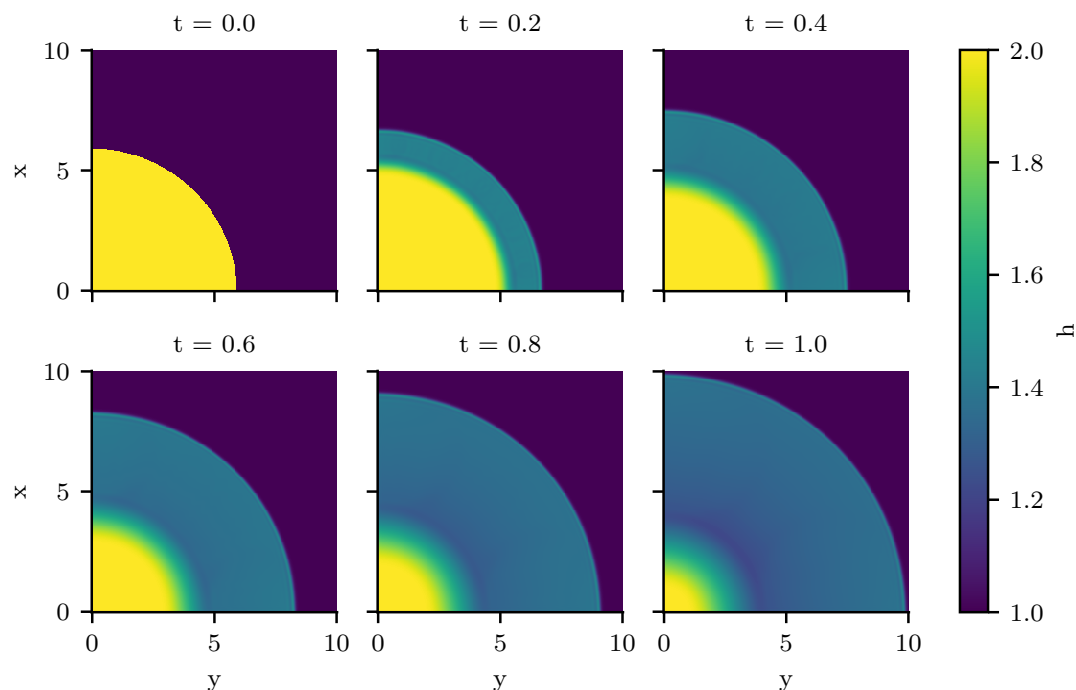


FIGURE 3 – Évolution de la hauteur d'eau au cours du temps. La discontinuité au temps t_0 forme une onde de choc vers l'avant et une onde de raréfaction vers l'arrière.

ver une onde de choc en direction du domaine h_1 et une onde de raréfaction en direction du domaine h_2 . La présence d'une discontinuité importante rend ce cas d'étude difficile à résoudre pour beaucoup de méthodes de résolution, différences finies comprises. Afin de limiter les erreurs d'approximation, deux outils complémentaires seront utilisés : un schéma upwind adapté aux termes convectifs et un filtre numérique passe-bas pour éliminer les erreurs numériques. Ce filtre numérique (filtre gaussien) correctement paramétré sera appliqué après chaque pas de temps afin de supprimer les oscillations à proximité de la discontinuité. Il est important de noter qu'un filtre mal paramétré introduira une forte diffusion numérique et fera disparaître la discontinuité au niveau de l'onde de choc : il faudra alors supprimer les oscillations hautes fréquences représentant les erreurs numériques en gardant la forme correcte de la solution.

4 Conclusion

Un nouvel outil a été développé visant à diminuer le temps et les efforts passés à traduire les modèles mathématiques en routine de résolution numérique. Celui-ci profite des avantages du calcul symbolique pour automatiser les étapes de discrétisation du modèle, et propose une interface utilisateur simple et puissante permettant, en quelques lignes, de passer d'une représentation d'un problème physique complexe sous forme d'équations aux dérivés partielles à sa résolution.

Bien qu'écrit avec un langage haut niveau (Python), son architecture modulaire lui permet de déléguer les calculs numériques à des langages bas niveau haute performance. Il vient également avec une panoplie d'outils tels que la sauvegarde des variables à résoudre (et de grandeurs secondaires) et leur affichage en temps réel.

Il a été validé avec succès sur des cas tests (tels que la rupture de barrage et le cas du lac au repos [2]), et

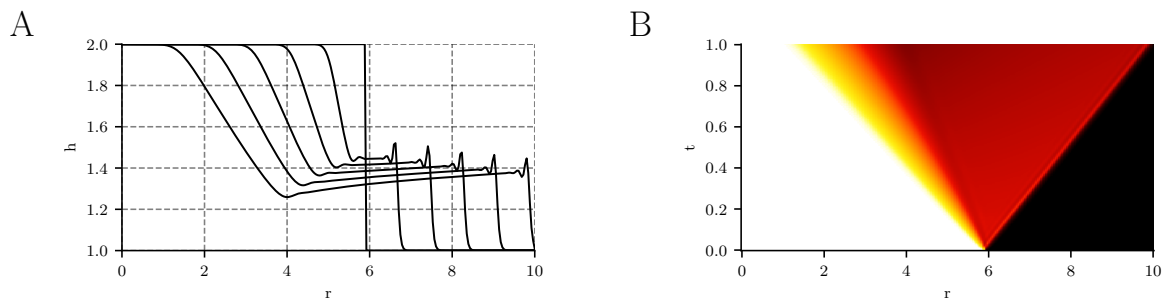


FIGURE 4 – Coupe selon un rayon r de la hauteur d'eau au cours du temps. À gauche, série d'instantanés à différents temps t de la hauteur en fonction du rayon. À droite, évolution spatio-temporelle de la hauteur d'eau le long de cette même coupe. On observe sans ambiguïtés la formation de l'onde de choc et de raréfaction à partir de la discontinuité d'origine.

appliqué à des problématiques de recherche plus complexes telles que l'écoulement de films ruisselants [1].

Références

- [1] Nicolas Cellier. *Optimisation d'échangeurs à films ruisselants*. PhD Thesis, Université de Savoie Mont-Blanc, 2018.
- [2] R J LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.
- [3] Joachim Rang. Improved traditional Rosenbrock-Wanner methods for stiff ODEs and DAEs. *J. Comput. Appl. Math.*, 286 :128–144, 2015.