



MASTER SCIENCE DE LA MATIÈRE
École Normale Supérieure de Lyon
Université Claude Bernard Lyon I

Stage 2015–2016
Lasserre Marvin
M2 Physique

Méthode de volumes finis pour l'équation de Burgers dans la métrique de Schwarzschild

Résumé : Le modèle de Burgers est un modèle simplifié décrivant la dynamique de fluides sans viscosité et peut donc servir comme première approche dans l'étude d'un fluide. Nous nous intéressons plus particulièrement à la version relativiste de ce modèle dans un espace-temps de Schwarzschild. Après avoir introduit des généralités sur les systèmes de lois de conservation hyperboliques et notamment le problème de Riemann, nous dérivons l'équation de Burgers relativiste à partir des équations d'Euler. Nous cherchons alors ses solutions statiques pour définir le problème de Riemann généralisé pour ce modèle. Enfin, nous introduisons les méthodes de volumes finis de Lax-Friedrichs et de Glimm pour résoudre ce problème puis discutons de la cohérence des résultats obtenus et des points forts et points faibles des deux méthodes. Il s'avère que la méthode de Lax-Friedrichs bien que simple à mettre en œuvre est diffusive au contraire la méthode de Glimm qui ne contient pas de diffusion numérique mais est lourde à mettre en œuvre notamment par le besoin préalable de résoudre le problème de Riemann généralisé.

Mots clefs : *Équation de Burgers relativiste, métrique de Schwarzschild, loi de conservation hyperbolique, méthode de volumes finis*

Stage encadré par :

Philippe LeFloch, directeur de recherche au CNRS

Laboratoire Jacques Louis Lions

Centre National de la Recherche Scientifique (CNRS)

4 place Jussieu

75005 Paris



Remerciements

Je tiens à remercier chaleureusement Philippe LeFloch pour m’avoir permis de découvrir son domaine de recherche en m’acceptant en stage, ainsi que pour m’avoir aidé et orienté dans mon travail. Je le remercie enfin de m’avoir permis de participer à un séminaire Oberwolfach, ce qui fût une très bonne expérience.

Je tiens également à remercier Shuyang Xiang qui m’a aidé dans mon travail en répondant aux nombreuses questions que j’ai pu me poser.

Pour finir, je remercie le Laboratoire Jacques-Louis Lions de m’avoir accueilli durant ces quatre mois de stage.

Table des matières

1	Introduction	1
2	Généralités sur les systèmes de lois de conservation hyperboliques	1
2.1	La méthode des caractéristiques	2
2.2	Domaine de dépendance	2
2.3	Forme intégrale et solutions faibles	3
2.4	Le problème de Riemann	3
2.4.1	Cas d’un choc : $v_l > v_r$	3
2.4.2	Cas d’une raréfaction : $v_l < v_r$	4
3	L’équation de Burgers relativiste	5
3.1	Dérivation de l’équation standard	6
3.2	L’équation relativiste	7
3.3	Solutions statiques	7
3.3.1	Résolution de l’équation statique	8
3.3.2	Domaine de définition de la solution	8
3.3.3	Propriétés des solutions statiques	9
3.4	Le problème de Riemann généralisé	9
3.4.1	Cas d’un choc	10
3.4.2	Cas d’une raréfaction	11
4	Méthodes numériques	12
4.1	La Méthode de Lax-Friedrichs	12
4.1.1	Dérivation de la méthode	13
4.1.2	La propriété de consistance	14
4.1.3	La condition CFL	14
4.1.4	Résultats numériques	16
4.2	La méthode de Glimm	16
4.2.1	La version standard	16
4.2.2	La version généralisée	17
4.2.3	Résultats numériques	17
5	Conclusion	18
A	Codes sources des différentes méthodes utilisées	19
A.1	Headers et fonctions auxiliaires	19
A.2	Lax-Friedrichs	21
A.3	Glimm	26

1 Introduction

L'équation de Burgers, introduite dans les années 1920 comme modèle de turbulence en dynamique des fluides, est dans sa version sans viscosité et à une dimension d'espace, l'exemple le plus simple d'une loi de conservation hyperbolique non-linéaire. Aujourd'hui, cette équation est utilisée comme modèle dans d'autres branches de la physique comme en dynamique des interfaces [1], ou bien en cosmologie [2]. Elle a pour expression

$$\partial_t v(x, t) + \partial_x \frac{v^2(x, t)}{2} = 0, \quad (1)$$

où $v : \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}$ est le champ de vitesse du fluide d'intérêt. Du fait de sa non linéarité, l'existence d'une solution de classe \mathcal{C}^1 sur $\mathbb{R} \times \mathbb{R}^+$ n'est pas en général vérifiée et ce même si la donnée initiale $v(x, 0) = v_0(x)$ est continue. Cela se traduit donc par l'existence non plus d'une unique solution classique mais une infinité de solutions dites faibles. Ces solutions faibles sont solutions de l'équation sous sa forme intégrale, plus générale, et l'apparition de chocs et/ou de raréfactions peut avoir lieu. Malgré tout, cette équation est soluble analytiquement, ce qui la rend intéressante comme point de départ dans l'établissement de méthodes numériques. Nous nous intéressons ici à son extension à un espace courbe qui amène de nouvelles subtilités autant dans sa résolution analytique que numérique et permet la généralisation des méthodes appliquées dans le cas d'un espace plat. Plus particulièrement, nous nous placerons dans un espace-temps de Schwarzschild dont la métrique est une solution non-triviale des équations d'Einstein et qui pour des coordonnées adéquates $(x^0, x^1, x^2, x^3) = (ct, r, \theta, \varphi)$ s'écrit

$$g = g_{\alpha\beta} dx^\alpha dx^\beta = -\left(1 - \frac{2M}{r}\right) c^2 dt^2 + \left(1 - \frac{2M}{r}\right)^{-1} dr^2 + r^2(d\theta^2 + \sin^2 \theta d\varphi^2). \quad (2)$$

Nous avons posé ici la vitesse de la lumière c et la constante de gravitation G égales à l'unité. La métrique de Schwarzschild décrit l'espace temps autour d'un objet sphérique de masse M et de rayon R . Si ce dernier est inférieur au rayon de Schwarzschild $R_S = 2M$, c'est alors un trou noir séparant l'espace-temps en deux régions $]0, 2M[$ et $]2M, +\infty[$ qui sont respectivement les régions internes et externes de celui-ci. Nous nous intéressons à la solution du modèle de Burgers relativiste sur le domaine extérieur $r \in]2M, +\infty[$ et pour des temps $t \in [0, +\infty[$. Notons cependant que l'équation peut bien sûr être généralisée à d'autres métriques que celle de Schwarzschild [3] et il existe des travaux portant sur sa généralisation à la métrique de Friedman-Lemaître-Walker-Robertson (FLWR) [4] ou encore à la métrique de De Sitter [5]. À noter également des travaux plus généraux sur l'étude de lois de conservation hyperboliques sur des variétés [6], [7]. Le travail effectué durant ce stage s'est concentré sur l'élaboration et l'implémentation de méthodes numériques pour la résolution de l'équation de Burgers relativiste avec une donnée initiale discontinue. Les méthodes qui ont été utilisées sont des méthodes aux volumes finis permettant de représenter et évaluer les équations aux dérivées partielles (EDP) sous une forme algébrique et donc d'approximer les solutions faibles de l'équation.

2 Généralités sur les systèmes de lois de conservation hyperboliques

De nombreux phénomènes physiques sont gouvernés par des **systèmes de lois de conservation** qui sont des systèmes d'EDP d'ordre 1. En une dimension d'espace, ils s'écrivent

$$\partial_t v(x, t) + \partial_x f(v(x, t)) = 0, \quad (3)$$

où $v : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^N$ est un vecteur de quantités conservées à N dimensions et $f : \mathbb{R} \rightarrow \mathbb{R}^N$ une fonction vectorielle (généralement non linéaire) qualifiée de **fonction de flux**. L'équation de Burgers standard, ainsi que sa version relativiste, sont toutes deux des lois de conservation scalaire ($N = 1$) qualifiées en plus d'**hyperboliques**, c'est-à-dire que la matrice jacobienne $f'(v)$ est diagonalisable et que pour toute valeur de v , les valeurs propres sont réelles. Il est possible d'étendre ces définitions à plusieurs dimensions, mais seul le cas monodimensionnel nous intéressera pour la suite.

2.1 La méthode des caractéristiques

Les lois de conservations hyperboliques, pour une condition initiale régulière et pour des temps courts, sont solubles par la **méthode des caractéristiques**. Cette méthode consiste à trouver les courbes du plan $x - t$ appelées caractéristiques, telles que l'EDP devienne une équation différentielle ordinaire (EDO) sur celles-ci.

Plus précisément, nous cherchons à résoudre le **problème de Cauchy** suivant

$$\begin{cases} \partial_t v(x, t) + f'(v) \partial_x v(x, t) = 0 \\ v(x, 0) = v_0(x), \end{cases} \quad (4)$$

avec $v_0(x)$ une fonction régulière. Nous avons ici écrit l'équation (3) sous sa forme dite quasi-linéaire. Soit alors γ une application de classe \mathcal{C}^1 telle que

$$\frac{d\gamma}{dt}(t) = f'(v(\gamma(t), t)). \quad (5)$$

On appelle caractéristique le graphe $\{(t, \gamma(t)) | t \in \mathbb{R}\}$ de la fonction γ définie ci-dessus. La fonction v est alors constante le long de chacune de ces caractéristiques, puisqu'on a

$$\begin{aligned} \frac{d}{dt} v(\gamma(t), t) &= \partial_t v(\gamma(t), t) + \gamma'(t) \partial_x v(\gamma(t), t) \\ &= \partial_t v(\gamma(t), t) + f'(v(\gamma(t), t)) \partial_x v(\gamma(t), t) \\ &= 0. \end{aligned} \quad (6)$$

De plus, d'après (5), la pente $\gamma'(t)$ est constante et les caractéristiques sont donc des lignes droites parallèles ayant chacune une origine x_0 différente. En conclusion, sous les hypothèses que l'on s'est fixées, on peut résoudre le problème de Cauchy (4) sur chaque droite $(\gamma(t), t)$ dont l'expression est

$$\gamma(t) = x_0 + v(\gamma(0), 0)t. \quad (7)$$

Le long de celles-ci, la solution vaut alors

$$v(\gamma(t), t) = v(x_0, 0). \quad (8)$$

2.2 Domaine de dépendance

Nous introduisons à présent le concept de **domaine de dépendance** qui sera important pour obtenir la **condition de Courant-Friedrichs-Lewy** (CFL), nécessaire à la stabilité de la solution numérique.

À nouveau, nous illustrons cette nouvelle notion en prenant pour exemple le cas du modèle de Burgers standard avec une donnée initiale régulière et en se restreignant à des temps courts. Dans ces conditions, la solution $v(x, t)$ possède la propriété que pour n'importe quel point (\bar{x}, \bar{t}) que l'on se fixe, sa valeur en ce point ne dépendra que de la valeur de la donnée initiale au point \bar{x}_0 . Graphiquement, cela se traduit par le fait que (\bar{x}, \bar{t}) est sur la caractéristique ayant pour origine \bar{x}_0 . Ainsi, si nous changeons la valeur de la donnée initiale sauf en \bar{x}_0 , la valeur de la solution au point (\bar{x}, \bar{t}) n'en sera pas affectée. On définit alors le domaine de dépendance du point (\bar{x}, \bar{t}) , comme étant l'ensemble $\mathcal{D}(\bar{x}, \bar{t}) = \{\bar{x}_0\}$. Dans notre cas scalaire, le domaine ne contient qu'un seul point mais pour le cas vectoriel, le domaine sera en général un intervalle. Les lois de conservation hyperboliques possèdent de plus la propriété que cet intervalle est borné ce qui implique que l'information ne peut se propager qu'à une vitesse maximale finie que l'on notera v_{max} . En d'autres termes, la solution en (\bar{x}, \bar{t}) ne peut être influencée par la donnée initiale que pour des valeurs de x étant à une distance finie de \bar{x} . La taille de cet ensemble va donc dépendre de \bar{t} , mais sera bornée : $\mathcal{D} \subset \{x : |x - \bar{x}| \leq v_{max} \bar{t}\}$. Inversement, la valeur de la donnée initiale à n'importe quel point x_0 , ne pourra influencer la solution que dans le cône de lumière $\{x : |x - x_0| \leq v_{max} t\}$.

2.3 Forme intégrale et solutions faibles

Comme nous allons nous intéresser à la résolution de l'équation de Burgers relativiste avec une donnée initiale contenant des discontinuités, l'équation différentielle (3) ne sera pas correctement définie sur tout le domaine d'intérêt. De plus, comme nous l'avons fait remarquer en introduction, il est possible que même avec une condition initiale régulière, la solution devienne discontinue à partir d'un certain temps. Ce phénomène est lié au fait que l'équation de Burgers est **non-linéaire**. En terme des caractéristiques, cela se traduit par le fait que deux caractéristiques peuvent se croiser et la solution être ainsi multivaluée. Pour palier à ce problème, il nous faut reformuler cette équation sous une **forme intégrale** permettant des solutions moins régulières appelées **solutions faibles**. En contrepartie, ces solutions faibles ne sont pas la plupart du temps uniques et il nous faut donc une condition pour discriminer entre celles-ci laquelle correspond au cas physique. On appelle cette condition la **condition d'entropie**¹.

Pour écrire cette formulation intégrale, on multiplie l'équation (3) par une fonction test $\phi \in \mathcal{C}_0^1(\mathbb{R} \times \mathbb{R})$, où $\mathcal{C}_0^1(\mathbb{R} \times \mathbb{R})$ correspond à l'ensemble des fonctions qui sont continûment différentiables et à support compact, puis on intègre par rapport à x et à t pour obtenir

$$\int_0^\infty \int_{-\infty}^{+\infty} (\phi(x, t) \partial_t v(x, t) + \phi(x, t) \partial_x (f(v))) dx dt = 0. \quad (9)$$

En intégrant par partie, on a finalement

$$\int_0^\infty \int_{-\infty}^{+\infty} (\partial_t \phi(x, t) v(x, t) + \partial_x \phi(x, t) f(v)) dx dt = - \int_{-\infty}^{+\infty} \phi(x, 0) v(x, 0) dx, \quad (10)$$

où seul un terme de bord subsiste grâce à la condition que la fonction ϕ soit à support compact. La fonction v est alors une solution faible, si l'équation (10) est valable quelque soit $\phi \in \mathcal{C}_0^1(\mathbb{R} \times \mathbb{R}^+)$.

2.4 Le problème de Riemann

Nous introduisons à présent le **problème de Riemann**, dont la résolution est indispensable pour la mise en œuvre de la **méthode de Glimm**. Le problème de Riemann consiste à résoudre l'équation de conservation (3) avec pour condition initiale une fonction **constante par morceaux** possédant une seule discontinuité

$$v(x, 0) = \begin{cases} v_l & \text{si } x < 0, \\ v_r & \text{si } x > 0. \end{cases} \quad (11)$$

La solution de ce problème dépend de la relation entre v_l et v_r .

2.4.1 Cas d'un choc : $v_l > v_r$

Lorsque $v_l > v_r$, la solution faible est unique et contient un choc, c'est-à-dire une discontinuité qui se propage sans déformation au cours du temps comme illustré sur la figure 1a. Elle a pour expression

$$v(x, t) = \begin{cases} v_l & \text{si } x < st, \\ v_r & \text{si } x > st, \end{cases} \quad (12)$$

où s est la vitesse de propagation de la discontinuité. On peut montrer [8] que la vitesse dans le cas général d'un système de lois de conservation, doit vérifier la condition de Hugoniot-Rankine

$$f(v_l) - f(v_r) = s(v_l - v_r). \quad (13)$$

Cependant, pour le cas scalaire, on peut isoler la vitesse de propagation et obtenir simplement

$$s = \frac{f(v_l) - f(v_r)}{v_l - v_r}. \quad (14)$$

1. Ces problèmes ayant été étudiés au départ au travers de la théorie des gaz, le vocabulaire en est resté.

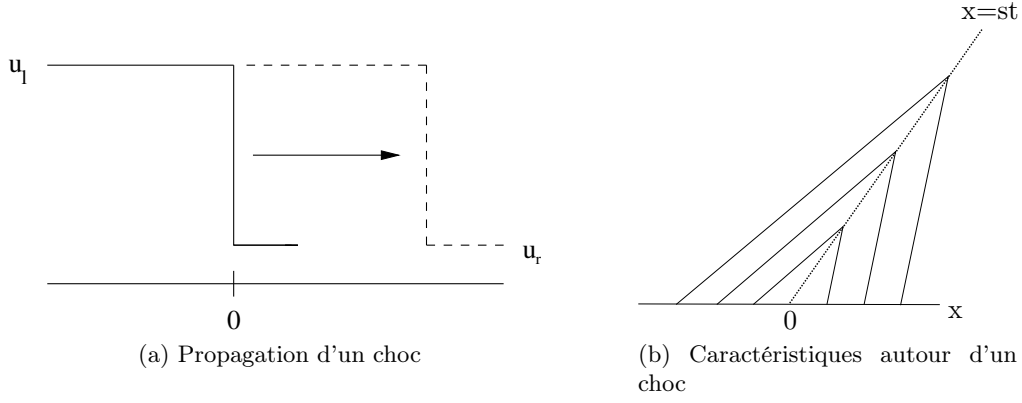


FIGURE 1: Choc - On remarque que les caractéristiques se dirigent vers le choc à mesure que le temps avance dans chaque région où u est constant.

Dans le cas de l'équation de Burgers, $f(v) = \frac{v^2}{2}$ et donc

$$s = \frac{v_l + v_r}{2}. \quad (15)$$

2.4.2 Cas d'une raréfaction : $v_l < v_r$

Dans le cas où $v_l < v_r$, la solution faible n'est plus unique et il nous faut alors utiliser la condition d'entropie pour sélectionner celle qui est physiquement valide. Son énoncé, du à Oleinik [9], est le suivant

Théorème 2.1. *$v(x, t)$ vérifie la condition d'entropie si toutes les discontinuités qu'elle contient vérifient la propriété que*

$$\frac{f(v) - f(v_l)}{v - v_l} \geq s \geq \frac{f(v) - f(v_r)}{v - v_r}, \quad (16)$$

pour tout v compris entre v_l et v_r .

On peut interpréter cette condition graphiquement à l'aide des caractéristiques. En effet, en prenant le cas de l'équation de Burgers pour exemple, si on trace les caractéristiques ainsi que le choc pour le cas précédent, comme sur la figure 1b, on remarque que pour vérifier cette condition les caractéristiques doivent être orientées vers le choc. Maintenant dans notre cas plusieurs solutions sont possibles dont le cas du choc avec $v_l < v_r$ (figure 2a), mais on voit graphiquement que cette dernière ne respecte pas la condition d'entropie puisque les caractéristiques sont orientées cette fois-ci vers l'extérieur du choc (voir figure 2b). La solution physique est donnée par ce qu'on appelle une raréfaction qui vérifie la condition d'entropie et dont l'expression est donnée par

$$v(x, t) = \begin{cases} v_l & x < f'(v_l)t, \\ u(x/t) & f'(v_l)t \leq x \leq f'(v_r)t, \\ v_r & x > f'(v_r)t, \end{cases} \quad (17)$$

avec $u(\xi)$ la solution de l'équation $f'(u(\xi)) = \xi$ appelée onde de raréfaction. Pour l'équation de Burgers, cela nous donne

$$v(x, t) = \begin{cases} v_l & x < v_l t, \\ x/t & v_l t \leq x \leq v_r t, \\ v_r & x > v_r t. \end{cases} \quad (18)$$

Nous avons tracé l'allure de la solution sur la figure 3a ainsi que ses caractéristiques dans le plan $x - t$ sur la figure 3b. Nous voyons sur cette dernière que la condition d'entropie est cette fois bien vérifiée.

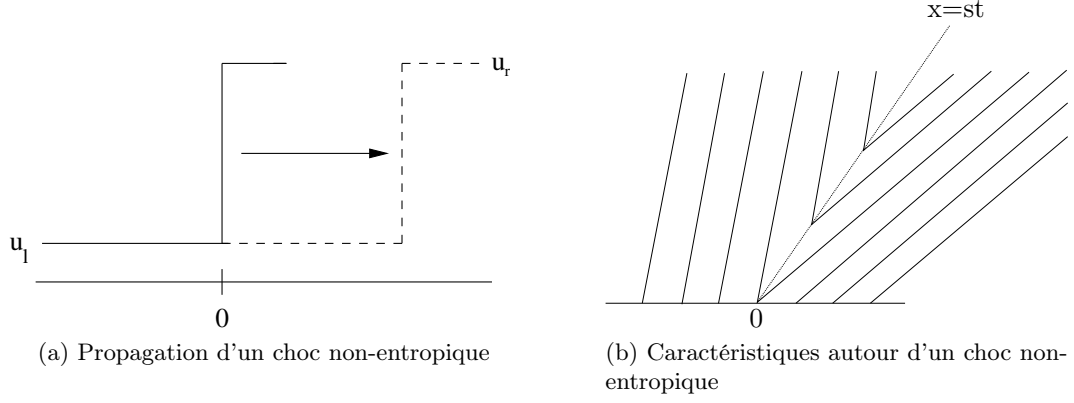


FIGURE 2: Solution faible non-entropique - On remarque que les caractéristiques sont orientées vers l'extérieur du choc : la condition d'entropie n'est pas respectée.

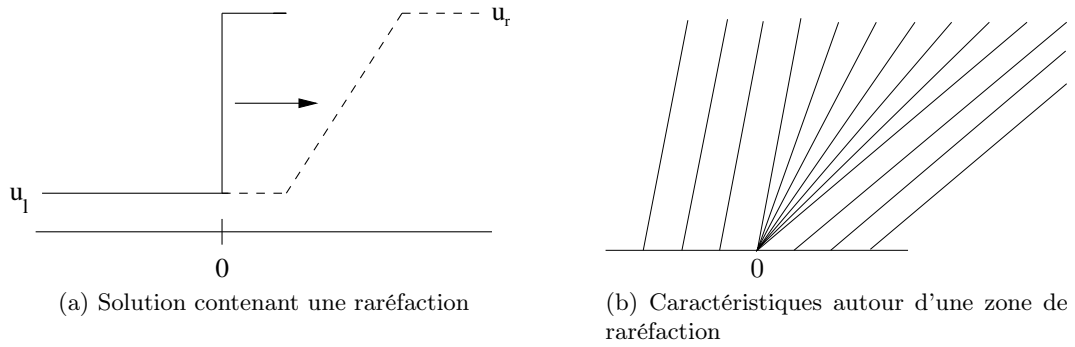


FIGURE 3: Raréfaction

3 L'équation de Burgers relativiste

Comme nous l'avons mentionné en introduction, l'équation de Burgers a été introduite pour modéliser les turbulences au sein d'un fluide. Toutefois, sa version sans viscosité (1) peut-être dérivée des **équations d'Euler**. Ceci est un point important, car c'est de cette manière que la **version relativiste** de l'équation peut être obtenue dans un espace-temps de Schwarzschild.

Les équations d'Euler décrivent la dynamique d'un fluide parfait, c'est-à-dire sans viscosité ni diffusion thermique. En une dimension, elles ont pour expression

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho v \\ E \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho v \\ \rho v^2 + p \\ v(E + p) \end{pmatrix} = 0, \quad (19)$$

où ρ est la densité, v la vitesse, ρv la quantité de mouvement, E l'énergie et p la pression qui est fonction des autres variables. Elles peuvent s'écrire sous une forme tensorielle plus compacte à l'aide du tenseur énergie-impulsion pour un fluide parfait

$$T^{\alpha\beta} = (\rho + p) u^\alpha u^\beta + p g^{\alpha\beta}, \quad (20)$$

où $u = (u^\alpha)$ est le champ de vecteur décrivant la vitesse au sein du fluide, que l'on prend normé. On le prend de plus comme étant de genre temps et orienté vers le futur. Ces propriétés se traduisent par la relation

$$u^\alpha u_\alpha = g_{\alpha\beta} u^\alpha u^\beta = -1, \quad (21)$$

soit dans le cas d'une métrique diagonale² et à 1 + 1 dimensions

$$g_{00}(u^0)^2 + g_{11}(u^1)^2 = -1. \quad (22)$$

Quant à $g^{\alpha\beta}$, ce sont les composantes de la métrique inverse, définie comme $g^{\alpha\gamma}g_{\gamma\beta} = \delta^\alpha_\beta$. La forme tensorielle des équations d'Euler est alors donnée par

$$\partial_\alpha T^{\alpha\beta} = 0. \quad (23)$$

Cependant, cette équation n'est valable que dans le cas d'un espace plat et il faut pour pouvoir la généraliser à un espace courbe, remplacer les dérivées partielles par des dérivées covariantes

$$\partial_\alpha T^{\alpha\beta} = 0 \rightarrow D_\alpha T^{\alpha\beta} = \partial_\alpha T^{\alpha\beta} + \Gamma_{\alpha\gamma}^\alpha T^{\gamma\beta} + \Gamma_{\alpha\gamma}^\beta T^{\alpha\gamma} = 0. \quad (24)$$

Les coefficients $\Gamma_{\alpha\beta}^\gamma$ sont ce qu'on appelle les symboles de Christoffel. Ceux-ci contiennent l'effet de la courbure et peuvent s'exprimer en fonction de la métrique [10]

$$\Gamma_{\alpha\beta}^\gamma = \frac{1}{2}g^{\gamma\delta} \left(\frac{\partial g_{\delta\beta}}{\partial x^\alpha} + \frac{\partial g_{\delta\alpha}}{\partial x^\beta} - \frac{\partial g_{\alpha\beta}}{\partial x^\delta} \right). \quad (25)$$

Pour pouvoir à présent dériver les deux équations à partir de la relation (24), il faut se placer dans le cas d'un fluide sans pression, soit $p = 0$. Les équations d'Euler se simplifient et deviennent

$$D_\alpha (\rho u^\alpha u^\beta) = 0. \quad (26)$$

On définit ensuite la quantité v comme

$$v := -\frac{1}{g_{00}} \frac{u^1}{u^0} \quad (27)$$

et on réécrit les composantes du champ de vitesse en fonction de cette nouvelle quantité :

$$\begin{cases} (u^0)^2 = -\frac{1}{g_{00}(1 + g_{00}g_{11}v^2)}, \\ (u^1)^2 = -\frac{g_{00}v^2}{1 + g_{00}g_{11}v^2}. \end{cases} \quad (28)$$

Cela nous permet finalement de pouvoir réécrire le tenseur énergie-impulsion en fonction de v en injectant les relations précédentes dans son expression (20). En regroupant ses composantes dans une matrices 2×2 , il s'écrit :

$$\begin{pmatrix} T^{00} & T^{01} \\ T^{10} & T^{11} \end{pmatrix} = \begin{pmatrix} -\frac{\rho}{g_{00}(1 + g_{00}g_{11}v^2)} & -\frac{\rho v}{g_{00}(1 + g_{00}g_{11}v^2)} \\ -\frac{\rho v}{g_{00}(1 + g_{00}g_{11}v^2)} & -\frac{g_{00}\rho v^2}{1 + g_{00}g_{11}v^2} \end{pmatrix}. \quad (29)$$

3.1 Dérivation de l'équation standard

Maintenant que nous avons introduit tout les prérequis nécessaires, nous pouvons obtenir l'équation de Burgers standard en se plaçant dans le cas de la **métrique plate de Minkowski** ($\eta_{\alpha\beta} = \text{diag}(-1, +1, +1, +1)$). Toutes les dérivées covariantes sont alors de simples dérivées partielles ($D_\alpha = \partial_\alpha$), puisque les symboles de Christoffel sont tous nuls. En remplaçant de plus g_{00} et g_{11} par leurs expressions dans (29) et en injectant le résultat dans les équations d'Euler on obtient

$$\beta = 0 : \quad \partial_0 T^{00} + \partial_1 T^{10} = \partial_t \left(\frac{\rho}{1 - v^2} \right) + \partial_r \left(\frac{\rho v}{1 - v^2} \right) = 0, \quad (30)$$

$$\beta = 1 : \quad \partial_0 T^{01} + \partial_1 T^{11} = \partial_t \left(\frac{\rho v}{1 - v^2} \right) + \partial_r \left(\frac{\rho v^2}{1 - v^2} \right) = 0. \quad (31)$$

En réécrivant l'équation pour $\beta = 1$

$$v \partial_t \left(\frac{\rho}{1 - v^2} \right) + \frac{\rho}{1 - v^2} \partial_t v + v \partial_r \left(\frac{\rho v}{1 - v^2} \right) + \frac{\rho v}{1 - v^2} \partial_r v = 0, \quad (32)$$

et en soustrayant l'équation pour $\beta = 0$ multipliée par v , on obtient enfin l'équation de Burgers standard (1).

2. Les deux métriques auxquelles nous nous intéressons (Minkowski et Schwarzschild) étant diagonales, nous nous plaçons dans le cas d'une métrique diagonale quelconque tant qu'il est inutile de préciser laquelle des deux on utilise.

3.2 L'équation relativiste

De manière analogue, on va obtenir l'équation relativiste en se plaçant dans le cas de la **métrique de Schwarzschild**. Néanmoins, les symboles de Christoffel ne sont pas tous nuls et nous avons

$$\Gamma_{tr}^t = \Gamma_{rt}^t = -\frac{M}{r^2} \left(1 - \frac{2M}{r}\right)^{-1} \quad \Gamma_{tt}^r = \frac{M}{r^2} \left(1 - \frac{2M}{r}\right) \quad \Gamma_{rr}^r = -\frac{M}{r^2} \left(1 - \frac{2M}{r}\right)^{-1} \quad (33)$$

$$\Gamma_{\theta\theta}^r = -r \left(1 - \frac{2M}{r}\right) \quad \Gamma_{\phi\phi}^r = -r \sin^2 \theta \left(1 - \frac{2M}{r}\right) \quad (34)$$

$$\Gamma_{r\theta}^\theta = \Gamma_{\theta r}^\theta = \frac{1}{r} \quad \Gamma_{\phi\phi}^\theta = -\sin \theta \cos \theta \quad \Gamma_{r\phi}^\phi = \Gamma_{\phi r}^\phi = \frac{1}{r} \quad (35)$$

$$\Gamma_{\theta\phi}^\phi = \Gamma_{\phi\theta}^\phi = \cot \theta. \quad (36)$$

En injectant alors leurs expressions dans les équations d'Euler, et en remplaçant également g_{00} et g_{11} par leurs expressions dans (29), on a

$$\beta = 0 : \quad \partial_t \left(\frac{r^2 \rho}{1-v^2} \right) + \partial_r \left(r(r-2M) \frac{\rho v}{1-v^2} \right) = 0, \quad (37)$$

$$\beta = 1 : \quad \partial_t \left(\frac{r^2 \rho v}{1-v^2} \right) + \partial_r \left(r(r-2M) \frac{\rho v^2}{1-v^2} \right) = -M\rho. \quad (38)$$

Comme pour l'équation standard, on réécrit l'équation pour $\beta = 1$

$$v \partial_t \left(\frac{r^2 \rho v}{1-v^2} \right) + \frac{r^2 \rho v}{1-v^2} \partial_t v + v, \partial_r \left(r(r-2M) \frac{\rho v}{1-v^2} \right) + r(r-2M) \frac{\rho v}{1-v^2} \partial_r v = -M\rho, \quad (39)$$

puis en soustrayant l'équation $\beta = 0$ multipliée par v on obtient l'équation de Burgers relativiste dans un espace-temps de Schwarzschild

$$\partial_t v + \left(1 - \frac{2M}{r}\right) \partial_r \left(\frac{v^2}{2} \right) = \frac{M}{r^2} (v^2 - 1). \quad (40)$$

On peut aller plus loin en écrivant cette équation sous une forme conservative (3). Pour la voir apparaître, il suffit de réécrire l'équation selon

$$\partial_t v + \left(1 - \frac{2M}{r}\right) \partial_r \left(\frac{v^2 - 1}{2} \right) - \frac{2M}{r^2} \left(\frac{v^2 - 1}{2} \right) = 0, \quad (41)$$

où l'on voit apparaître la dérivée par rapport à r du rapport de deux fonctions si l'on divise l'équation par $\left(1 - \frac{2M}{r}\right)^2$. On obtient finalement la forme conservative de l'équation :

$$\partial_t \left(\frac{v}{\left(1 - \frac{2M}{r}\right)^2} \right) + \partial_r \left(\frac{v^2 - 1}{2 \left(1 - \frac{2M}{r}\right)} \right) = 0. \quad (42)$$

Cette équation va nous être utile par la suite pour définir des méthodes numériques conservatives et notamment la méthode de Lax-Friedrichs. On finira par remarquer qu'en faisant tendre la masse du trou noir $M \rightarrow 0$ dans (42), on retrouve bien l'équation de Burgers standard

$$\partial_t v + \partial_r \left(\frac{v^2}{2} \right) = 0. \quad (43)$$

3.3 Solutions statiques

Une classe de solutions importantes du modèles de Burgers relativiste sont ses **solutions statiques**. En effet, elles rentrent dans la définition du **problème de Riemann généralisé**, que nous décrirons plus tard et qui est lui-même important dans la définition de la **méthode de Glimm généralisée**.

3.3.1 Résolution de l'équation statique

Pour trouver l'expression de ces solutions statiques, que l'on notera v_* , il nous faut résoudre l'équation de Burgers relativiste indépendante du temps, soit

$$\frac{d}{dr} \left(\frac{v_*^2 - 1}{2 \left(1 - \frac{2M}{r}\right)} \right) = 0. \quad (44)$$

Si pour un rayon donné $r = r_0$, la solution vaut $v(r_0) = v_0$, l'intégration de l'équation précédente nous donne la relation suivante

$$\frac{v_*^2 - 1}{1 - \frac{2M}{r}} = \frac{v_0^2 - 1}{1 - \frac{2M}{r_0}}. \quad (45)$$

On définit alors le paramètre positif $K_*(r_0, v_0) := \sqrt{\frac{1-v_0^2}{1-\frac{2M}{r_0}}}$ et on obtient après réarrangement de l'équation précédente

$$v_*^2 = 1 - \left(1 - \frac{2M}{r}\right) K_*^2(r_0, v_0). \quad (46)$$

En conclusion, pour une vitesse v_0 donnée au rayon $r = r_0 > 2M$, l'équation de Burgers relativiste indépendante du temps admet pour solution unique

$$v_*(r) = \text{sgn}(v_0) \sqrt{1 - K_*^2 \left(1 - \frac{2M}{r}\right)}. \quad (47)$$

Nous avons représenté plusieurs de ces solutions statiques en 4a, avec une valeur de K_* différente à chaque fois.

3.3.2 Domaine de définition de la solution

Comme dit précédemment nous voulons étudier les solutions du modèle de Burgers relativiste sur le **domaine extérieur** du trou noir de Schwarzschild, soit $r \in]2M, +\infty[$. Cependant, d'après l'expression de la solution statique (47), celle-ci n'est pas forcément définie sur tout ce domaine. En effet, si dans cette expression l'argument de la racine devient négatif, la solution va alors cesser d'exister. Le terme $1 - \frac{2M}{r}$ étant compris dans l'intervalle $]0, 1[$, cela va donc dépendre du signe de la quantité $1 - K_*$. Dans le cas où $K_* < 1$, la solution est définie sur tout le domaine extérieur mais dans le cas où $K_* > 1$, l'argument s'annule pour un certain rayon $r = r^\natural$ et devient négatif pour $r > r^\natural$. Ces deux possibilités sont séparées par le cas critique où $K_* = 1$ c'est à dire $|v_0| = \sqrt{\frac{2M}{r_0}}$. Ainsi d'après ce qui précède, c'est la donnée du point (r_0, v_0) que l'on se fixe qui détermine si la solution est définie sur tout le domaine ou bien sur une partie seulement. Cela motive l'introduction des deux solutions stationnaires critiques $v_{**}^\pm = v_{**}^\pm(r)$ ayant pour expression

$$v_{**}^\pm(r) := \pm \sqrt{\frac{2M}{r}}, \quad (48)$$

puisque le graphe de ces courbes sépare l'ensemble possible des valeurs de la vitesse $[-1, 1]$ ³ en trois domaines disjoints : $[-1, v_{**}^-(r)]$, $]v_{**}^-, v_{**}^+]$, $[v_{**}^+, 1]$. Cela nous permet de définir l'ensemble des grandes vitesses par $\mathcal{G} := [-1, v_{**}^-(r)] \cup [v_{**}^+, 1]$ et l'ensemble des petite vitesse par $\mathcal{P} =]v_{**}^-, v_{**}^+]$. Ainsi, si $v_0 \in \mathcal{G}$ (ou autrement dit si $0 \leq K_*(r_0, v_0) \leq 1$), la solution statique peut-être définie sur tout le domaine extérieur tandis que si $v_0 \in \mathcal{P}$ (autrement dit si $K_*(r_0, v_0) > 1$), elle ne peut-être définie que sur la portion $]2M, r^\natural[$. Dans le cas des petites vitesses, le rayon d'annulation de la solution statique $r = r^\natural$ a pour expression⁴

$$r^\natural := \frac{2MK_*^2}{K_*^2 - 1}. \quad (49)$$

3. On rappelle qu'on s'est placé dans le cas où $c = 1$.

4. Pour une démonstration, voir [11].

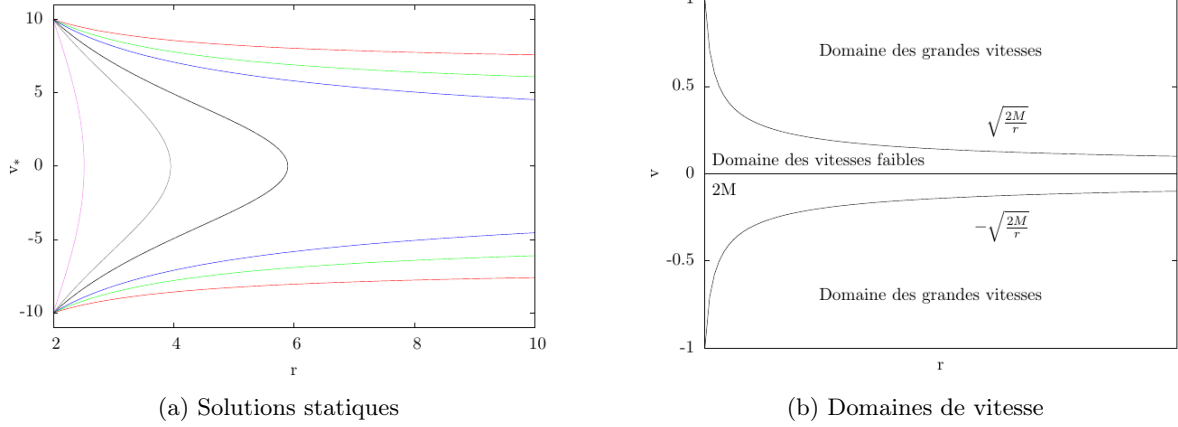


FIGURE 4: Sur la figure 4a, nous avons tracé plusieurs solutions statiques pour une valeur de K_* différente à chaque fois. La figure 4b illustre quant elle les différents domaines de vitesse que l'on a introduit.

3.3.3 Propriétés des solutions statiques

Les **solutions statiques**, dont le domaine de définition est fixé par v_0 , possèdent une première propriété importante. En effet, pour une solution statique $v_* = v_*(r)$, l'intervalle $]2M, r_0[$ sera toujours contenu dans le domaine de définition, indépendamment de la valeur de v_0 , ceci d'après (49). Autrement dit, nous pouvons au moins garantir l'existence d'une solution statique à gauche du point $r = r_0$. Cette propriété sera importante lorsque l'on s'intéressera plus tard au schéma de Glimm généralisé.

Les solutions vérifient de plus, en fonction de la valeur de v_0 , des propriétés de monotonie et de convexité sur leur domaine de définition.

- Si $v_0 \in \mathcal{G}$ et si de plus $v_0 < 0$ (resp. $v_0 > 0$), alors la solution sera croissante (resp. décroissante) et concave (resp. convexe).
- Si $v_0 \in \mathcal{P}$, deux cas se présentent :

Quand $1 < K_* \leq 2$ et que $v_0 > 0$ (resp. $v_0 < 0$), alors la solution v_* est convexe (resp. concave) sur $]2M, r^\sharp[$ et concave (resp. convexe) sur $]r^\sharp, r^\natural[$ si $v_0 > 0$. Ici, le rayon $r = r^\sharp$ est donné par

$$r^\sharp := \frac{3MK_*^2}{2(K_*^2 - 1)}. \quad (50)$$

Quand $K_* > 2$, v_* est concave sur $]2M, r^\natural[$ si $v_0 > 0$ et convexe si $v_0 < 0$.

Enfin, on a les valeurs limites suivantes

- Si $v_0 \in \mathcal{G}$:

$$\lim_{r \rightarrow 2M} v_*(r) = \text{sgn}(v_0), \quad \lim_{r \rightarrow +\infty} v_*(r) = \text{sgn}(v_0)\sqrt{1 - K_*^2}, \quad (51)$$

- Si $v_0 \in \mathcal{P}$:

$$\lim_{r \rightarrow 2M} v_* = \text{sgn}(v_0), \quad \lim_{r \rightarrow r^\natural} v_*(r) = 0, \quad \lim_{r \rightarrow r^\natural} \frac{dv_*}{dr} = -\text{sgn}(v_0)\infty. \quad (52)$$

3.4 Le problème de Riemann généralisé

Le problème de Riemann que nous avons introduit plus tôt, ne peut pas être appliqué tel quel à l'équation de Burgers relativiste. Or, comme nous l'avons déjà mentionné, sa résolution est indispensable pour la définition de la méthode de Glimm.

La condition initiale qui consistait toute à l'heure en deux états constants séparés par une discontinuité en $r = r_0$, doit être modifiée car les états constants ne sont plus des états stationnaires de l'équation relativiste. En d'autres termes, nous allons remplacer les deux états constants u_l et u_r par

deux **solutions statiques** que nous noterons $v_L(r)$ et $v_R(r)$ et dont l'expression est donnée en (47). Le **problème de Riemann généralisé**, va donc consister à résoudre l'équation de Burgers relativiste (42) pour la donnée initiale

$$v(r, t) = \begin{cases} v_L(r) & r < \sigma(t), \\ v_R(r) & r > \sigma(t), \end{cases} \quad (53)$$

où $[\bar{r}, \hat{r}] \subset]2M, +\infty[$ est le domaine sur lequel on restreint notre étude. Les solutions statiques s'écrivent

$$v_L(r) = \text{sgn}(v_L^0) \sqrt{1 - K_L^2 \left(1 - \frac{2M}{r}\right)}, \quad v_R(r) = \text{sgn}(v_R^0) \sqrt{1 - K_R^2 \left(1 - \frac{2M}{r}\right)}, \quad (54)$$

avec $K_L = K(r_0, v_L^0)$ et $K_R = K(r_0, v_R^0)$ où $v_L^0 = v_L(r_0)$ et $v_R^0 = v_R(r_0)$. De manière analogue au cas standard, nous avons deux types de solutions selon la relation entre v_L^0 et v_R^0 : soit un choc si $v_L^0 > v_R^0$, soit une raréfaction si $v_R^0 > v_L^0$. Cependant, comme nous l'avons mentionné plus tôt, le choc ainsi que les caractéristiques ne seront plus des droites dans le plan $r - t$ mais des fonctions plus compliquées. Nous ne faisons ici qu'énoncer les résultats dont la démonstration peut-être trouvée dans [11], [12].

3.4.1 Cas d'un choc

Pour commencer, la vitesse du choc va être modifiée par la géométrie et celle-ci va alors dépendre de r . En utilisant la condition d'Hugoniot-Rankine (13) on trouve

$$s(r) = \left(1 - \frac{2M}{r}\right) \frac{v_L(r) + v_R(r)}{2}, \quad (55)$$

qui nous redonne comme à chaque fois le cas standard en faisant tendre la masse M vers 0. Nous pouvons alors à partir de l'expression de sa vitesse, trouver son expression que l'on note $\sigma(t)$, en intégrant l'EDO

$$\frac{d\sigma}{dt} = \left(1 - \frac{2M}{\sigma}\right) \frac{v_L(\sigma) + v_R(\sigma)}{2}. \quad (56)$$

Après intégration, nous obtenons l'équation non-linéaire suivante à résoudre

$$R_S(\sigma(t); v_L, v_R) - R_S(r_0; v_L, v_R) = t, \quad (57)$$

où la fonction $R_S = R_S(r; v_L, v_R)$ est définie par

$$\begin{aligned} R_S(r, v_L, v_R) = & \text{sgn}(v_L + v_R) \frac{2}{K_*^{R^2} - K_*^{L^2}} \left(\frac{r - 4M}{r - 2M} \left(\sqrt{1 - K_*^{L^2} \left(1 - \frac{2M}{r}\right)} - \sqrt{1 - K_*^{R^2} \left(1 - \frac{2M}{r}\right)} \right) \right. \\ & - M(4 - 3K_*^{R^2}) \left(\ln \left(r \sqrt{1 - K_*^{L^2}} \sqrt{1 - K_*^{R^2} \left(1 - \frac{2M}{r}\right)} + (M - r)K_*^{R^2} + r \right) \right. \\ & \left. + \ln \left(\frac{2r}{(r - 2M)} \sqrt{1 - K_*^{R^2} \left(1 - \frac{2M}{r}\right)} - K_*^{R^2} \right) \right) \\ & + M(4 - 3K_*^{L^2}) \left(\ln \left(r \sqrt{1 - K_*^{L^2}} \sqrt{1 - K_*^{L^2} \left(1 - \frac{2M}{r}\right)} + (M - r)K_*^{L^2} + r \right) \right. \\ & \left. \left. - \ln \left(\frac{2r}{(r - 2M)} \sqrt{1 - K_*^{L^2} \left(1 - \frac{2M}{r}\right)} - K_*^{L^2} \right) \right) \right), \end{aligned} \quad (58)$$

avec $K_*^{L^2} = \frac{1 - v_L^2}{(1 - 2M/r)}$ et $K_*^{R^2} = \frac{1 - v_R^2}{(1 - 2M/r)}$. Dans le cas du choc, la solution a pour expression

$$v(r, t) = \begin{cases} v_L(r) & \bar{r} < r < \sigma(t) \\ v_R(r) & \hat{r} > r > \sigma(t) \end{cases} \quad (59)$$

3.4.2 Cas d'une raréfaction

Pour le cas de la raréfaction, comme nous l'avons dit, les caractéristiques ne sont plus des droites et donc $u_l t$ et $u_r t$ qui délimitaient la zone de raréfaction pour le problème de Riemann standard sont remplacées par les courbes $\sigma_L(t)$ et $\sigma_R(t)$. Ces deux courbes satisfont comme pour $\sigma(t)$, une EDO à intégrer

$$\frac{d\sigma_L}{dt} = \left(1 - \frac{2M}{\sigma_L}\right) v_L(\sigma_L) \quad \text{et} \quad \frac{d\sigma_R}{dt} = \left(1 - \frac{2M}{\sigma_R}\right) v_R(\sigma_R), \quad (60)$$

qui nous donne pour chacune une équation non-linéaire à résoudre. Pour $\sigma_L(t)$, on a

$$R_L(\sigma_L(t), v_L) - R_L(r_0, v_L) = t, \quad (61)$$

où la fonction $R_L = R(r, v_L)$ est donnée par

$$\begin{aligned} R_L(r, v_L) := & \text{sgn}(v_L) \frac{1}{(1 - K_*^{L^2})^{3/2}} \left(2M(1 - K_*^{L^2})^{3/2} \ln(r - 2M) \right. \\ & - 2M(1 - K_*^{L^2})^{3/2} \ln \left(2r \sqrt{1 - K_*^{L^2} \left(1 - \frac{2M}{r}\right)} + (2M - r)K_*^{L^2} \right) \\ & + r \sqrt{1 - K_*^{L^2}} \sqrt{1 - K_*^{L^2} \left(1 - \frac{2M}{r}\right)} \\ & \left. + M(2 - 3K_*^{L^2}) \ln \left(r \sqrt{1 - K_*^{L^2}} \sqrt{1 - K_*^{L^2} \left(1 - \frac{2M}{r}\right)} + (M - r)K_*^{L^2} + r \right) \right), \end{aligned} \quad (62)$$

avec $K_*^{L^2}$ ayant la même expression que dans le cas du choc. Quant à $\sigma_R(t)$, elle vérifie l'équation non-linéaire

$$R_R(\sigma_R(t), v_R) - R_R(r_0, v_R) = t, \quad (63)$$

où $R_R = R(r, v_R)$ est donnée par

$$\begin{aligned} R_R(r, v_R) := & \text{sgn}(v_R) \frac{1}{(1 - K_*^{R^2})^{3/2}} \left(2M(1 - K_*^{R^2})^{3/2} \ln(r - 2M) \right. \\ & - 2M(1 - K_*^{R^2})^{3/2} \ln \left(2r \sqrt{1 - K_*^{R^2} \left(1 - \frac{2M}{r}\right)} + (2M - r)K_*^{R^2} \right) \\ & + r \sqrt{1 - K_*^{R^2}} \sqrt{1 - K_*^{R^2} \left(1 - \frac{2M}{r}\right)} \\ & \left. + M(2 - 3K_*^{R^2}) \ln \left(r \sqrt{1 - K_*^{R^2}} \sqrt{1 - K_*^{R^2} \left(1 - \frac{2M}{r}\right)} + (M - r)K_*^{R^2} + r \right) \right), \end{aligned} \quad (64)$$

avec $K_*^{R^2}$ ayant la même expression que dans le cas du choc. Pour finir, l'expression de l'onde de raréfaction $w(t, r)$ va être aussi modifiée et avoir pour expression

$$w(t, r) = \text{sgn}(r - r_0) \sqrt{1 - K^2(t, r) \left(1 - \frac{2M}{r}\right)}, \quad (65)$$

où $K = K(t, r) > 0$ est une fonction de $t > 0$ et de $r > 2M$ satisfaisant l'égalité

$$\text{sgn}(r - r_0) \frac{R(r, K) - R(r_0, K)}{t} = 1, \quad (66)$$

avec la fonction $R = R(r, K)$ donnée par

$$\begin{aligned} R(r, K) := & \frac{1}{(1 - K^2)^{3/2}} \left(2M(1 - K^2)^{3/2} \ln(r - 2M) \right. \\ & - 2M(1 - K^2)^{3/2} \ln \left(2r \sqrt{1 - K^2} \left(1 - \frac{2M}{r} \right) + (2M - r)K^2 \right) \\ & + \left(r \sqrt{1 - K^2} \sqrt{1 - K^2} \left(1 - \frac{2M}{r} \right) \right. \\ & \left. \left. + M(2 - 3K^2) \ln \left(r \sqrt{1 - K^2} \sqrt{1 - K^2} \left(1 - \frac{2M}{r} \right) + (M - r)K^2 + r \right) \right) \right). \end{aligned} \quad (67)$$

Dans le cas du choc, la solution a pour expression

$$v(r, t) = \begin{cases} v_L(r) & \bar{r} < r < \sigma_L(t), \\ w(t, r) & \sigma_L(t) \leq r \leq \sigma_R(t), \\ v_R(r) & \hat{r} > r > \sigma_R(t). \end{cases} \quad (68)$$

Nous pouvons réécrire les deux solutions (59) et (68) sous la forme d'une équation générale qui rassemble ces deux cas :

$$v(t, r) = \begin{cases} v_L(r) & \bar{r} < r < \sigma_-(t), \\ w(t, r) & \sigma_-(t) < r < \sigma_+(t), \\ v_R(r) & \sigma_+(t) < r < \hat{r}, \end{cases} \quad (69)$$

où

$$\sigma_-(t) = \begin{cases} \sigma(t), & v_L^0 > v_R^0, \\ \sigma_L(t), & v_R^0 > v_L^0, \end{cases} \quad \sigma_+(t) = \begin{cases} \sigma(t), & v_L^0 > v_R^0, \\ \sigma_R(t), & v_R^0 > v_L^0. \end{cases} \quad (70)$$

À noter que l'on retrouve tous les résultats du cas standard 2.4 en faisant tendre la masse du trou noir $M \rightarrow 0$.

4 Méthodes numériques

À présent que nous avons introduit l'équation de Burgers dans sa version **standard** (1) et **relativiste** (42), nous allons nous intéresser à la résolution numérique de cette deuxième avec une **méthode de Lax-Friedrichs et de Glimm**. Dans les deux cas, nous discrétisons le plan $r - t$ avec Δt , le pas en temps, et Δr , le pas en espace, tous deux constants. Les points (r_j, t_n) du maillage sont alors définis par

$$r_j = 2M + j\Delta r, \quad j \in \mathbb{N}^+ \quad (71)$$

$$t_n = n\Delta t, \quad n \in \mathbb{N}^+. \quad (72)$$

4.1 La Méthode de Lax-Friedrichs

La méthode de Lax-Friedrichs est une méthode numérique dite conservative, c'est-à-dire qu'elle s'écrit sous la forme

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta r} [F(U_j^n, U_{j+1}^n) - F(U_{j-1}^n, U_j^n)], \quad (73)$$

où U_j^n est l'approximation de la solution de l'EDP au point r_j et au temps t_n . Quant à F , elle est ce qu'on appelle la fonction de flux numérique. Dans le cas de Lax-Friedrichs appliqué au modèle de Burgers standard, elle a pour expression [8]

$$F(U_j^n, U_{j+1}^n) = \frac{\Delta r}{2\Delta t} (U_j^n - U_{j+1}^n) + \frac{1}{2} (f(U_j^n) + f(U_{j+1}^n)), \quad (74)$$

avec $f(u) = \frac{u^2}{2}$.

On veut maintenant dériver une **méthode de Lax-Friedrichs** pour l'équation de Burgers relativiste (42).

4.1.1 Dérivation de la méthode

La relation algébrique (73) étant valable pour n'importe quelle méthode conservative, c'est donc le flux numérique qui va différer de celui du cas standard. Dans le but de trouver son expression, nous modifions l'équation relativiste dans sa forme conservative (42) en posant

$$w = \frac{v}{(1 - \frac{2M}{r})^2}, \quad (75)$$

ce qui nous donne une fois l'équation réécrite en terme de cette nouvelle fonction w :

$$\partial_t w + \partial_r \left(\frac{(1 - 2M/r)^4 w^2 - 1}{2(1 - 2M/r)} \right) = 0. \quad (76)$$

En intégrant (76) de $r_{j-1/2}$ à $r_{j+1/2}$ en espace et de t_n à t_{n+1} en temps, on a :

$$\int_{r_{j-1/2}}^{r_{j+1/2}} \int_{t_n}^{t_{n+1}} \partial_t w \, dr dt + \int_{r_{j-1/2}}^{r_{j+1/2}} \int_{t_n}^{t_{n+1}} \partial_r \left(\frac{(1 - 2M/r)^4 w^2 - 1}{2(1 - 2M/r)} \right) \, dr dt = 0,$$

ce qui nous donne

$$\begin{aligned} \int_{r_{j-1/2}}^{r_{j+1/2}} (w(t_{n+1}, r) - w(t_n, r)) \, dr + \int_{t_n}^{t_{n+1}} \left[\left(1 - \frac{2M}{r_{j+1/2}}\right)^3 \frac{w^2(t, r_{j+1/2})}{2} - \frac{1}{2\left(1 - \frac{2M}{r_{j+1/2}}\right)} \right. \\ \left. - \left(1 - \frac{2M}{r_{j-1/2}}\right)^3 \frac{w^2(t, r_{j-1/2})}{2} + \frac{1}{2\left(1 - \frac{2M}{r_{j-1/2}}\right)} \right] dt = 0. \end{aligned}$$

Soit W_j^n la valeur moyenne en espace de la solution dans le sens où

$$W_j^n := \frac{1}{h} \int_{r_{j-1/2}}^{r_{j+1/2}} w(t_{n+1}, r) \, dr, \quad (77)$$

on peut alors en déduire que la méthode de Lax-Friedrichs pour le modèle de Burgers relativiste :

$$W_j^{n+1} = W_j^n - \frac{k}{h} \left(F_{j+1/2}^n - F_{j-1/2}^n \right),$$

où le flux numérique $F_{j+1/2}^n$ est donné par

$$F_{j+1/2}^n = \left(1 - \frac{2M}{r_{j+1/2}}\right)^3 F_s(W_j^n, W_{j+1}^n) - \frac{1}{2\left(1 - \frac{2M}{r_{j+1/2}}\right)}, \quad (78)$$

avec F_s défini comme

$$F_s(W_j^n, W_{j+1}^n) = \frac{f_s(W_j^n) + f_s(W_{j+1}^n)}{2} - \frac{h}{2k} (W_{j+1}^n - W_j^n). \quad (79)$$

On remarque que F_s représente le flux numérique dans le modèle de Burgers standard (43).

4.1.2 La propriété de consistance

On veut à présent montrer que la méthode de Lax-Friedrichs que l'on vient de dériver pour le modèle de Burgers relativiste (78) est **consistante**, c'est-à-dire, qu'elle **préserve la solution statique** tel que, quelque soit l'entier $n \geq 0$, la relation suivante est vérifiée :

$$W_j^{n+1} = W_j^n. \quad (80)$$

Proposition 4.1. *Si la solution donnée par la méthode de Lax-Friedrichs pour le modèle de Burgers relativiste est consistante, c'est-à-dire, si on a (80) quelque soit l'entier n , alors l'équation statique pour le modèle de Burgers (47) est préservée lorsque l'on prend la limite $\Delta r \rightarrow 0$ pour le pas en espace.*

Démonstration. D'après (78), (80), on a

$$\begin{aligned} & \left(1 - \frac{2M}{r_{j+1/2}}\right)^3 \left[f_s(W_j^n) + f_s(W_{j+1}^n) - \frac{h}{k}(W_{j+1}^n - W_j^n) \right] - \frac{1}{1 - \frac{2M}{r_{j+1/2}}} \\ & - \left(1 - \frac{2M}{r_{j-1/2}}\right)^3 \left[f_s(W_{j-1}^n) + f_s(W_j^n) - \frac{h}{k}(W_j^n - W_{j-1}^n) \right] + \frac{1}{1 - \frac{2M}{r_{j-1/2}}} = 0. \end{aligned} \quad (81)$$

Un développement de Taylor par rapport à Δr donne

$$\begin{aligned} \left(1 - \frac{2M}{r_{j\pm 1/2}}\right)^3 &= \left(1 - \frac{2M}{r_j}\right)^3 \pm \frac{3M}{r_j^2} \left(1 - \frac{2M}{r_j}\right)^2 h + O(h^2) \\ \frac{1}{1 - \frac{2M}{r_{j\pm 1/2}}} &= \frac{1}{1 - \frac{2M}{r_j}} \mp \frac{M}{r_j^2} \frac{1}{(1 - 2M/r)^2} h + O(h^2), \\ f_s(W_j^n) + f_s(W_{j\pm 1}^n) \mp \frac{h}{k}(W_{j\pm 1}^n - W_j^n) &= 2f_s(W_j^n) \pm (f_s(W_j^n))' h + O(h^2), \end{aligned} \quad (82)$$

où les valeurs primées sont dérivées par rapport à r .

En utilisant (82) et (81), on obtient

$$\left(1 - \frac{2M}{r_j}\right)^3 (f_s(W_j^n))' + \frac{6M}{r_j^2} \left(1 - \frac{2M}{r_j}\right)^2 f(W_j^n) + \frac{M}{r_j^2} \frac{1}{\left(1 - \frac{2M}{r_j}\right)^2} + O(h) = 0,$$

ce qui nous donne

$$\left(1 - \frac{2M}{r}\right)^3 w'(t, r)w(t, r) + \frac{3M}{r^2} \left(1 - \frac{2M}{r}\right)^2 w^2(t, r) + \frac{M}{r^2} \frac{1}{\left(1 - \frac{2M}{r}\right)^2} = 0, \quad (83)$$

où nous avons remplacé $f_s(W_j^n)$ par son expression donnée en (79). Faisant maintenant tendre le pas en espace vers 0, nous retrouvons bien l'équation statique (47), c'est-à-dire,

$$\partial_r \left(\frac{(1 - 2M/r)^4 w^2 - 1}{2(1 - 2M/r)} \right) = 0.$$

□

4.1.3 La condition CFL

Nous introduisons à présent la **condition CFL**, nécessaire mais pas forcément suffisante à la **stabilité** de la méthode numérique. Comme nous allons le voir, celle-ci nous donne une contrainte à respecter entre le pas en temps Δt et le pas en espace Δr . Nous l'énonçons dans la sous-section concernant la méthode de Lax-Friedrichs mais celle-ci devra aussi être respectée pour la méthode de Glimm.

Comme nous l'avons mentionné plus tôt, cette condition a été introduite par Courant, Friedrichs et Lax, dans leur article de 1928 [13]⁵. Leur but était alors de prouver l'existence de solutions pour certaines EDP par le biais de méthodes aux éléments finis. De façon analogue à ce que nous avons fait pour prouver la consistance de la méthode de Lax-Friedrichs, leur idée était de discrétiser l'équation pour obtenir une équation aux différences finies dont la solution tendrait vers une fonction limite vérifiant l'EDP de départ, lorsque l'on raffinerait le maillage. Ils établissent alors qu'une condition de stabilité, nécessaire pour n'importe quelle méthode, est que le domaine de dépendance de la méthode aux éléments finis $\mathcal{D}_{\Delta t}(\bar{x}, \bar{t})$, que l'on qualifie de numérique, doit contenir le domaine de dépendance de l'EDP, introduit en 2.2, tout du moins dans la limite $\Delta t, \Delta r \rightarrow 0$. La condition CFL se traduit donc par

$$\mathcal{D}(\bar{x}, \bar{t}) \subset \mathcal{D}_0(\bar{x}, \bar{t}). \quad (84)$$

Nous allons maintenant appliquer ceci à la méthode de Lax-Friedrichs pour le cas de l'équation de Burgers relativiste. D'après la relation (78), la valeur W_j^n dépend des valeurs W_{j-1}^{n-1} , W_j^{n-1} et W_{j+1}^{n-1} qui eux-mêmes dépendent des valeurs W_{j-2}^{n-2} , W_{j-1}^{n-2} , W_j^{n-2} , W_{j+1}^{n-2} et W_{j+2}^{n-2} et ainsi de suite. Si on continue jusqu'au temps $t = 0$, on observe que la solution en t_n ne dépend alors que des valeurs de la donnée initiale aux points x_{j+q} pour $q = -n, \dots, n$. Le domaine de dépendance numérique vérifie donc

$$\mathcal{D}(x_j; t_n) \subset \{x : |x - x_j| \leq nh\}, \quad (85)$$

ou, en terme d'un point fixe (\bar{x}, \bar{t}) ,

$$\mathcal{D}(\bar{x}, \bar{t}) \subset \{x : |x - \bar{x}| \leq (\bar{t}/\Delta t)\Delta r\}. \quad (86)$$

Si on prend maintenant la limite $\Delta t, \Delta r \rightarrow 0$ avec un rapport fixé à $\Delta t/\Delta r = a$, le domaine de dépendance remplit alors complètement l'ensemble, c'est-à-dire

$$\mathcal{D}_0(\bar{x}, \bar{t}) = \{x : |x - \bar{x}| \leq \bar{t}/a\}. \quad (87)$$

La condition CFL imposant que $\mathcal{D}(\bar{x}, \bar{t}) \subset \mathcal{D}_0(\bar{x}, \bar{t})$, cela implique la relation suivante :

$$|(\bar{x} - f'(v)\bar{t}) - \bar{x}| \leq \bar{t}/a, \quad (88)$$

avec $f'(v)$ la pente de la caractéristique passant par (\bar{x}, \bar{t}) comme nous l'avons vu en 2.1. En réécrivant la relation précédente, nous avons donc

$$\left(1 - \frac{2M}{r}\right) \frac{v_L(r) + v_R(r)}{2} \leq \frac{\Delta r}{\Delta t}. \quad (89)$$

Nous pouvons aller plus loin en se souvenant que l'on étudie notre solution dans le domaine extérieur du trou noir $]2M, +\infty[$ où le facteur $(1 - \frac{2M}{r})$ a pour borne supérieure 1. Nous pouvons donc l'oublier tout comme le terme $\frac{v_L(r) + v_R(r)}{2}$ car la norme de la vitesse est aussi bornée à 1. Finalement, sur le domaine extérieur, nous avons la relation suivante entre les pas en temps et en espace :

$$\Delta t \leq \Delta r. \quad (90)$$

Le fait que la condition CFL soit nécessaire provient de la simple observation que si (84) n'est pas vérifiée, il existe alors des points ξ du domaine de dépendance analytique qui ne sont pas dans le domaine de dépendance numérique. Ainsi on pourrait changer la valeur de la donnée initiale au point ξ affectant la solution analytique mais pas la solution numérique. Il est alors évident que cette dernière ne pourrait pas alors converger vers la vraie solution pour toute donnée initiale. Enfin, on peut montrer [8] que pour la méthode de Lax-Friedrichs, la condition CFL est aussi suffisante mais il existe des méthodes pour lesquelles ce n'est pas le cas.

5. Pour trouver une traduction de l'article en anglais, voir [14]

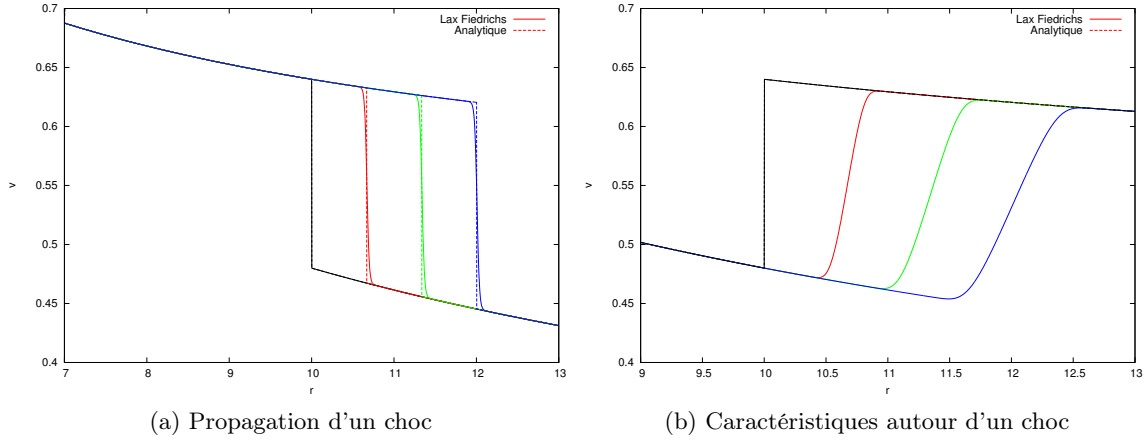


FIGURE 5: Nous avons tracé sur cette figure la solution du problème de Riemann obtenue par la méthode de Lax-Friedrichs, en la comparant avec la solution analytique. On y observe la présence de diffusion numérique au niveau des chocs et un accord parfait entre les deux solutions pour la raréfaction.

4.1.4 Résultats numériques

Pour finir avec la méthode de Lax-Friedrichs, nous présentons les résultats que nous avons obtenus à l'appliquant au problème de Riemann généralisé. Le code source de la méthode est listé en annexe A. Nous avons choisis comme valeur $v_L^0 = 0.64$ et $v_R^0 = 0.48$ dans le cas du choc et l'inverse dans le cas d'une raréfaction. De plus, la valeur de la masse M est posée comme étant égale à 1. Nous avons tracé les résultats sur les figures 5a et 5b pour plusieurs temps et avec un pas de $\Delta t = 1 \times 10^{-3}$ et un pas $\Delta r = 2 \times 10^{-3}$. Nous voyons alors que la méthode de Lax-Friedrichs introduit, comme dans le cas standard, de la diffusion numérique au niveau des chocs c'est-à-dire qu'elle a tendance à lisser les discontinuités. Ceci est du à l'erreur de troncature que l'on a fait en s'arrêtant à l'ordre 1 du développement de Taylor pour définir la méthode. Cependant, la vitesse du choc dans le cas numérique et analytique sont en très bon accord. Pour le cas de la raréfaction, les deux solutions sont par contre confondues et la méthode de Lax-Friedrichs semble bien adaptée dans ce cas.

4.2 La méthode de Glimm

Nous allons maintenant introduire la **méthode de Glimm** ou **méthode de choix aléatoire**, basée sur les travaux de J. Glimm [15], qui a montré notamment la convergence de la méthode, et de A. Chorin [16], [17]. La version généralisée de cette méthode pour le modèle de Burgers relativiste est étudiée dans [11].

4.2.1 La version standard

La méthode de Glimm consiste comme pour la méthode de Lax-Friedrichs à approximer la solution $v(x, t)$ de l'équation de Burgers aux points spatiaux $x_j = j\Delta x$, $j \in \mathbb{Z}$ pour les temps $t_n = n\Delta t$, $n \in \mathbb{N}$. Nous l'appliquons par itérations successives pour faire progresser dans le temps la donnée initiale $u(x_j, 0) = U_j^0$, $j \in \mathbb{Z}$. Pour passer du temps t_n à t_{n+1} , on agit en deux étapes en commençant par faire évoluer la solution de t_n à $t_{n+1/2}$ puis en la faisant évoluer de $t_{n+1/2}$ à t_{n+1} . Ces deux étapes étant similaires, nous ne présentons que le passage de t_n à $t_{n+1/2}$. La méthode consiste premièrement à approximer la solution $v(x, t)$ au temps t_n par une fonction constante par morceaux. Cette dernière est définie telle que pour chaque cellule $[x_{j-1/2}, x_{j+1/2}]$, sa valeur soit donnée par $v(x_j, t_n)$.

On construit alors la solution analytique de l'équation de Burgers pour $t_n \leq t \leq t_{n+1/2}$, ayant pour donnée initiale la fonction constante par morceaux. Cela nous amène à devoir résoudre le problème de

Riemann avec la condition initiale

$$v(x, t_n) = \begin{cases} U_j^n, & \text{pour } x \leq (j + 1/2)\Delta x, \\ U_{j+1}^n, & \text{pour } x > (j + 1/2)\Delta x \end{cases} \quad (91)$$

pour chaque j et pour $t > t_n$. En général, U_j^n et U_{j+1}^n sont différents et la condition initiale contient donc une discontinuité au point $x = (j + 1/2)\Delta x$ qui va se propager en une onde de choc ou bien donner une onde de raréfaction comme nous l'avons vu en 2.4. Si la condition CFL est vérifiée, deux chocs ne pourront pas interagir sur une même cellule nous permettant de recoller les solutions des problèmes de Riemann pour chaque j et d'obtenir ainsi la solution de l'équation de Burgers au temps $t = t_{j+1/2}$. Le cœur de la méthode de Glimm consiste alors à tirer une variable aléatoire $\theta_{n+1/2}$ à valeurs dans $[-1, 1]$ et d'assigner la valeur

$$U_{j+1/2}^{n+1/2} = v \left(\left[j + \frac{1}{2} + \frac{1}{2}\theta_{n+1/2} \right] \Delta x, [n + 1/2] \Delta t \right), \quad (92)$$

à la solution approximée $U_{j+1/2}^{n+1/2}$, $j \in \mathbb{Z}$ dans la cellule $[x_j, x_{j+1}]$ au temps $t = t_{n+1/2}$. On utilise ensuite la même procédure pour avancer du temps $t_{n+1/2}$ au temps t_{n+1} et on répète ce processus en entier pour obtenir la solution approchée aux temps ultérieurs.

4.2.2 La version généralisée

Pour la méthode de Glimm généralisée, le principe est exactement le même que celui que l'on vient de voir pour le cas standard, à ceci près que nous n'approximons plus la donnée initiale par une fonction constantes par morceaux, mais **statique** par morceaux

$$v(r, t_n) = \begin{cases} v_L(r), & \text{pour } r \leq (j + 1/2)\Delta r, \\ v_R(r), & \text{pour } r > (j + 1/2)\Delta r. \end{cases} \quad (93)$$

où v_L et v_R sont les solutions statiques du modèle de Burgers relativiste (42). Par conséquent, la méthode s'appuie à présent sur le **problème de Riemann généralisé** pour décrire l'évolution au cours du temps d'une discontinuité entre deux cellules voisines.

4.2.3 Résultats numériques

Comme pour la méthode de Lax-Friedrichs, nous terminons cette section en présentant les résultats obtenus par la méthode de Glimm pour la résolution du modèle de Burgers généralisé. Le code source est présenté en annexe A. Les valeurs de v_L^0 et v_R^0 sont toujours égales à 0.64 et 0.48 et la masse à $M = 1$. Nous avons tracé la solution approchée et la solution analytique sur la figure 6 pour un pas $\Delta t = 1 \times 10^{-3}$ et un pas $\Delta r = 2 \times 10^{-3}$. La méthode n'introduit pas de diffusion numérique comme la méthode de Lax-Friedrichs mais diffère légèrement de la solution analytique. Ce dernier point est du au choix aléatoire que l'on fait pour attribuer sa valeur à la solution approchée au bout d'une (demi) itération en temps. Il faudrait alors encore diminuer le pas Δt pour avoir un meilleur accord entre les deux solutions.

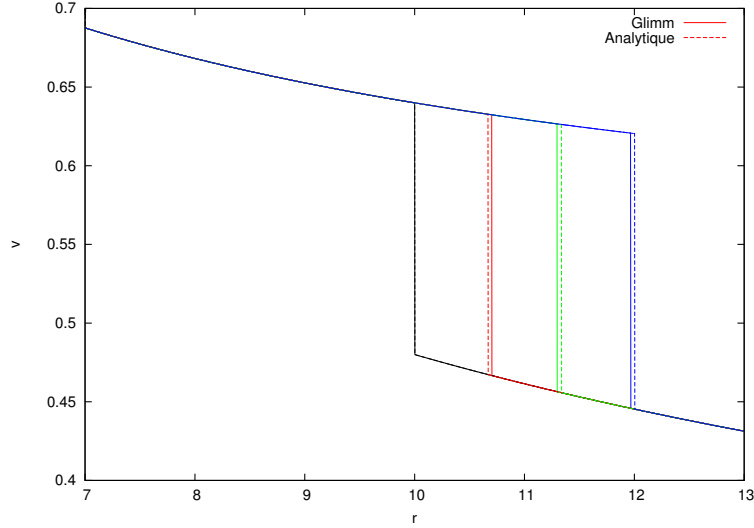


FIGURE 6: Nous avons tracé sur cette figure la solution du problème de Riemann obtenue par la méthode Glimm, en la comparant avec la solution analytique. La légère différence observée entre les deux est due au choix aléatoire que l'on fait pour approximer la solution à chaque itération en temps. En diminuant le pas Δt , on peut arriver à un meilleur accord.

5 Conclusion

Après avoir introduit des concepts généraux sur les lois de conservation hyperboliques, nous avons dérivé le modèle de Burgers relativiste dans un espace de Schwarzschild. Nous avons alors cherché à élaborer une méthode de Lax-Friedrichs ainsi qu'une méthode de Glimm pour résoudre numériquement cette équation. La première comme dans le cas standard introduit de la diffusion numérique au niveau des chocs alors que la deuxième n'en introduit pas. La méthode de Lax-Friedrichs possède tout de même l'avantage de s'exprimer simplement au contraire de la méthode de Glimm qui a elle besoin de la solution du problème de Riemann généralisé pour pouvoir être appliquée. Le problème de Riemann généralisé que nous avons également introduit, voit les caractéristiques ainsi que les chocs de sa solution, qui étaient de simples droites dans le cas standard, devenir des courbes dont l'expression ne peut être obtenue qu'en résolvant une équation non-linéaire. Chacune des deux méthodes qu'on a vu possède donc ses avantages et ses défauts.

Par manque de temps, nous n'avons malheureusement abordé que le cas du choc pour la méthode de Glimm et une suite à ce travail serait donc de s'intéresser au cas de la raréfaction. De plus, nous nous sommes placés uniquement dans le cas où v_L^0 et v_R^0 appartaient au domaine des grandes vitesses. Cependant, nous avons vu que quand elles appartaient au domaine des vitesses faibles, les solutions statiques n'étaient définies que localement. Ceci va donc amener de nouveaux problèmes dans l'implémentation de la méthode de Glimm si on se place dans ce régime. Enfin, tout ceci pourrait être un point de départ pour s'intéresser ensuite au cas vectoriel des équations d'Euler dans un espace courbe.

A Codes sources des différentes méthodes utilisées

Nous listons ici l'ensemble des codes sources que nous avons écrit en C99 pour la mise en œuvre des méthodes de Lax-Friedrichs et de Glimm. Nous listons premièrement les fichiers contenant les fonctions auxiliaires qui ne participent pas directement à la résolution des équations, avec certains des headers qui définissent les variables globales. Sont ensuite listés les fichiers pour la méthode de Lax-Friedrichs, puis ceux de la méthode de Glimm.

A.1 Headers et fonctions auxiliaires

```

/*****
 * Nom ..... : real_precision.h
 * Role ..... : Le type real peut correspondre au choix a un
 *               double ou un float.
 *               En procedant ainsi nous n'avons
 *               pas besoin de changer a la main le type
 *               des variables dans tous les fichiers.
 *****/

#ifndef REAL_PRECISION_H
#define REAL_PRECISION_H

typedef float real;

#endif /* REAL_PRECISION_H */

/*****
 * Nom ..... : io.c
 * Role ..... : La fonction entete ecrit en debut de fichier
 *               le nombre de points en x et en temps, les
 *               bornes du domaine en espace, le temps final
 *               ainsi que la valeur des pas.
 *               La fonction ecrit ecrit un tableau de taille
 *               size_x dans un fichier en écrivant en
 *               debut et en fin respectivement la condition au
 *               bord gauche et la condition au bord droit
 *****/

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

#include "real_precision.h"
#include "io.h"

void entete(const char* filename, const int nb_x, const int nb_t,
            const real x_left, const real x_right, const real t_f,
            const real x_step, const real t_step){

    FILE* file = NULL;
    file = fopen(filename, "w");

    fprintf(file, "%d %d %g %g %g %g\n", nb_x, nb_t, x_left,
        x_right, t_f, x_step, t_step);

    fclose(file);
    return;
}

```

```

void ecrit(const char* filename, const int size_x, real* mat){

    int i = 0;
    FILE* file = NULL;

    file = fopen(filename, "a");

    for (i=1;i<size_x-1;i++){
        fprintf(file, "%g ", mat[i]);
    }
    fprintf(file, "\n");

    fclose(file);
}

/*****
* Nom ..... : scan_real.c
* Role ..... : Fonction de lecture clavier pour le type real
*               definit dans real_precision.
*****/

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

#include "real_precision.h"
#include "scan_real.h"

void scan_real(real* in){

    if (sizeof(real)==sizeof(float))
        scanf("%f", (float*)in);
    else if (sizeof(real)==sizeof(double))
        scanf("%lf", (double*)in);
    else
        fprintf(stderr, "Erreur sur la taille de real!\n");
}

/*****
* Nom ..... : parametre.h
* Role ..... : Definie les variables globales a tous les
*               fichiers.
*****/

#ifndef DEFVAR
#define EXTERN extern
#else
#define EXTERN
#endif /* DEFVAR */

EXTERN real M, EPSILON;

```

A.2 Lax-Friedrichs

```

/*****
 * Nom ..... : burgers.c
 * Role ..... : Resolution de l'equation de Burgers relativiste
 *               par une methode de Lax-Friedrichs.
 *****/

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

#include "real_precision.h"
#include "scan_real.h"
#include "methodes.h"
#include "io.h"

/* Nom du fichier ou on ecrit la solution pour w*/
#define FILENAME_W "lax_friedrichs_w.dat"

/* Nom du fichier ou on ecrit la solution pour la vitesse v*/
#define FILENAME_V "lax_friedrichs_v.dat"

#define EPSILON 1.

int main(void){

    /* ----- DECLARATION DES VARIABLES ----- */

    int j = 0;                /* compteur pour l'espace */
    int n = 0;                /* compteur pour le temps */

    int nr = 0;               /* nombre de points en abscisses
                             pour un t donne */
    int nt = 0;               /* nombre de points en temps
                             pour un x donne */

    real M = 0.;              /* Masse de l'objet */
    real h = 0.;              /* pas pour l'abscisses */
    real k = 0.;              /* pas pour le temps */
    real alpha = 0.;          /* alpha = k/h */

    real r_left = 0.;         /* Borne inferieure en r */
    real r_right = 0.;        /* Borne superieure en r */
    real t_f = 0.;            /* Temps final */

    real r_dis = 0.;          /* Rayon de la discontinuite */
    real l_border = 0.;       /* Condition au bord gauche */
    real r_border = 0.;       /* Condition au bord droit */

    real u_left = 0.;         /* Valeur a gauche */
    real u_right = 0.;        /* Valeur a droite */

    real* vect_r = NULL;      /* Discretisation de l'axe r */
    real* unew = NULL;        /* vecteur contenant U^{n+1} */
    real* uold = NULL;        /* vecteur contenant U^n */

    real* vect_v = NULL;      /* vecteur contenant la vitesse */

    /* ----- FIN DE LA DECLARATION ----- */

```

```

/* Dialogue avec l'utilisateur */

printf("Donner la borne inferieure en r: ");
scan_real(&r_left);
printf("r_left = %g\n", r_left);

printf("Donner la borne superieure en r: ");
scan_real(&r_right);
printf("r_right = %g\n\n", r_right);

printf("Donner le temps final: ");
scan_real(&t_f);
printf("t_f = %g\n", t_f);

printf("Entrer la valeur de la masse M de l'objet: ");
scan_real(&M);
printf("M = %g\n", M);

printf("Entrer la valeur du pas h en espace: ");
scan_real(&h);
printf("h = %g\n", h);

printf("Entrer la valeur du pas k en temps: ");
scan_real(&k);
printf("k = %g\n", k);

printf("Entrer la valeur de la donnee initiale a gauche u_left: ");
scan_real(&u_left);
printf("u_left = %g\n", u_left);

printf("Entrer la valeur de la donnee initiale a droite u_right: ");
scan_real(&u_right);
printf("u_right = %g\n", u_right);

printf("Entrer le rayon de la discontinuite: ");
scan_real(&r_dis);
printf("r_dis = %g\n", r_dis);

/* Calcul du nombre de points en x */
nr = (int) (ceil((r_right - r_left)/h + 1));
printf("nr = %d\n\n", nr);

/* Calcul du nombre de points en t */
nt = (int) (ceil(t_f/k + 1));
printf("nt = %d\n\n", nt);

/* Calcul et affichage de alpha */
alpha = k/h;
printf("alpha = %g\n\n", alpha);

/* Allocation memoire des tableaux et matrices */
vect_v = (real*)calloc((size_t)nr, sizeof(real));
vect_r = (real*)calloc((size_t)nr, sizeof(real));
unew = (real*)calloc((size_t)nr, sizeof(real));
uold = (real*)calloc((size_t)nr, sizeof(real));

/* Remplissage du vecteur r */
vect_r[0] = r_left;
for(j=1;j<nr-1;j++){
    vect_r[j] = r_left + (real)j * h;

```



```

}
vect_r[nr-1] = r_right;

/* Evaluation en r_left de la branche gauche */
if(u_left <= 0){
    l_border = -(real)sqrt(1./EPSILON/EPSILON
        - (1./EPSILON/EPSILON-u_left*u_left
            * (1.-2*M/r_dis)*(1.-2*M/r_dis)
            * (1.-2*M/r_dis)*(1.-2*M/r_dis))
        / (1.-2*M/r_dis)
        * (1.-2*M/r_left))
        / (real)(1.-2*M/r_left)/(real)(1.-2*M/r_left);
}
else if(u_left > 0){
    l_border = (real)sqrt(1./EPSILON/EPSILON
        - (1./EPSILON/EPSILON-u_left*u_left
            * (1.-2*M/r_dis)*(1.-2*M/r_dis)
            * (1.-2*M/r_dis)*(1.-2*M/r_dis))
        / (1.-2*M/r_dis)
        * (1.-2*M/r_left))
        / (real)(1.-2*M/r_left)/(real)(1.-2*M/r_left);
}
else{
    fprintf(stderr, "Erreur !\n");
    exit(EXIT_FAILURE);
}

/* Evaluation en r_right de la branche droite */
if(u_right <= 0){
    r_border = -(real)sqrt(1./EPSILON/EPSILON
        - (1./EPSILON/EPSILON-u_right*u_right
            * (1.-2*M/r_dis)*(1.-2*M/r_dis)
            * (1.-2*M/r_dis)*(1.-2*M/r_dis))
        / (1.-2*M/r_dis)
        * (1.-2*M/r_right))
        / (real)(1.-2*M/r_right)/(real)(1.-2*M/r_right);
}
else if(u_right > 0){
    r_border = (real)sqrt(1./EPSILON/EPSILON
        - (1./EPSILON/EPSILON-u_right*u_right
            * (1.-2*M/r_dis)*(1.-2*M/r_dis)
            * (1.-2*M/r_dis)*(1.-2*M/r_dis))
        / (1.-2*M/r_dis)
        * (1.-2*M/r_right))
        / (real)(1.-2*M/r_right)/(real)(1.-2*M/r_right);
}
else{
    fprintf(stderr, "Erreur !\n");
    exit(EXIT_FAILURE);
}

/* ----- CALCUL + ECRITURE DE LA SOLUTION ----- */

remove(FILENAME_W); /* Si le fichier existe deja on le supprime */
remove(FILENAME_V);

/* Ecriture de l'entete contenant le nombre
de points en x et d'instant */
entete(FILENAME_W, nr-2, nt, r_left, r_right, t_f, h, k);
entete(FILENAME_V, nr-2, nt, r_left, r_right, t_f, h, k);

```

```

/* Initialisation du probleme */
uold[0] = l_border;
for(j=1;j<nr;j++){
    if(vect_r[j] <= r_dis && u_left < 0){
        uold[j] = -(real)sqrt(1./EPSILON/EPSILON
            - (1./EPSILON/EPSILON-u_left*u_left
            *(1-2*M/r_dis)*(1-2*M/r_dis)
            *(1-2*M/r_dis)*(1-2*M/r_dis))
            / (1-2*M/r_dis)
            * (1-2*M/vect_r[j]))
            / (1-2*M/vect_r[j])/(1-2*M/vect_r[j]);
    }
    else if(vect_r[j] <= r_dis && u_left >= 0){
        uold[j] = (real)sqrt(1./EPSILON/EPSILON
            - (1./EPSILON/EPSILON-u_left*u_left
            *(1-2*M/r_dis)*(1-2*M/r_dis)
            *(1-2*M/r_dis)*(1-2*M/r_dis))
            / (1-2*M/r_dis)
            * (1-2*M/vect_r[j]))
            / (1-2*M/vect_r[j])/(1-2*M/vect_r[j]);
    }
    else if(vect_r[j] > r_dis && u_right < 0){
        uold[j] = - (real)sqrt(1./EPSILON/EPSILON
            - (1./EPSILON/EPSILON-u_right*u_right
            *(1-2*M/r_dis)*(1-2*M/r_dis)
            *(1-2*M/r_dis)*(1-2*M/r_dis))
            / (1-2*M/r_dis)
            * (1-2*M/vect_r[j]))
            / (1-2*M/vect_r[j])/(1-2*M/vect_r[j]);
    }
    else if(vect_r[j] > r_dis && u_right >= 0){
        uold[j] = (real)sqrt(1./EPSILON/EPSILON
            - (1./EPSILON/EPSILON-u_right*u_right
            *(1-2*M/r_dis)*(1-2*M/r_dis)
            *(1-2*M/r_dis)*(1-2*M/r_dis))
            / (1-2*M/r_dis)
            * (1-2*M/vect_r[j]))
            / (1-2*M/vect_r[j])/(1-2*M/vect_r[j]);
    }
    else{
        fprintf(stderr, "Erreur !\n");
        exit(EXIT_FAILURE);
    }
}
uold[nr-1] = r_border;

for(j=0;j<nr;j++){
    vect_v[j] = (1-2*M/vect_r[j])*(1-2*M/vect_r[j])*uold[j];
}

ecrit(FILENAME_W, nr, uold); /* Ecriture de la donnee initiale */
ecrit(FILENAME_V, nr, vect_v);

for(n=1;n<nt;n++){
    lax_friedrichs(nr, M, h, k, l_border, r_border,
        vect_r, uold, unew);
    for(j=0;j<nr;j++){
        vect_v[j] = (1-2*M/vect_r[j])*(1-2*M/vect_r[j])*unew[j];
    }
}

```

```

        ecrit(FILENAME_W, nr, unew);
        ecrit(FILENAME_V, nr, vect_v);
        for (j=0;j<nr;j++){
            /* Transfert de unew dans
               uold pour la */
            uold[j] = unew[j]; /* prochaine iteration */
        }
    }

    /* ----- */

    /* Desallocation memoire des tableaux et matrices */
    free(unew);
    free(uold);
    free(vect_v);
    free(vect_r);

    exit(EXIT_SUCCESS);
}

/*****
 * Nom ..... : methodes.c
 * Role ..... : Implementation de la methode de Lax-Friedrichs
 *****/

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

#include "real_precision.h"
#include "methodes.h"

/* Fonction de flux numerique */
real num_flux_function(real alpha, real a, real b){
    return (a*a + b*b)/4 - (b - a)/(2*alpha);
}

/* Methode de Lax-Friedrichs */
void lax_friedrichs(const int n, real M, real dr, real dt,
                    real l_border, real r_border, real* r, real* anc,
                    real* nouv){

    int j = 0;
    real alpha = dt/dr;

    for (j=1;j<n-1;j++){
        nouv[j] = anc[j]
            +alpha*(1./(2*(1.-2*M/(r[j]+(real)0.5*dr)))
                -1./(2*(1.-2*M/(r[j]-(real)0.5*dr))))
            -alpha*((1.-2*M/(r[j]+(real)0.5*dr))
                *(1.-2*M/(r[j]+(real)0.5*dr))
                *(1.-2*M/(r[j]+(real)0.5*dr))
                *num_flux_function(alpha, anc[j], anc[j+1])
                -(1.-2*M/(r[j]-(real)0.5*dr))
                *(1.-2*M/(r[j]-(real)0.5*dr))
                *(1.-2*M/(r[j]-(real)0.5*dr))
                *num_flux_function(alpha, anc[j-1], anc[j]));
    }
    nouv[0] = l_border;
    nouv[n-1] = r_border;
}

```

A.3 Glimm

```

/*****
* Nom ..... : burgers.c
* Role ..... : Resolution de l'equation de Burgers relativiste
*              avec une methode de Glimm generalisee.
*****/

#include <stdio.h>
#include <stdlib.h>
#include <tgmath.h>
#include <time.h>

#include "real_precision.h"
#include "functions.h"
#include "scan_real.h"
#include "methodes.h"
#include "io.h"

/* Nom du fichier ou on ecrit la solution */
#define FILENAME "glimm.dat"

#define DEFVAR
#include "parametres.h"

int main(void){

    EPSILON = 1.;

    /* Initialisation de la "graine" pour le tirage de la
       variable aleatoire*/
    srand((unsigned int)time(NULL));

    /* ----- DECLARATION DES VARIABLES ----- */

    int j = 0;                /* compteur pour l'espace */
    int n = 0;                /* compteur pour le temps */

    int nr = 0;               /* nombre de points en abscisses
                               pour un t donne */
    int nt = 0;               /* nombre de points en temps
                               pour un x donne */

    real h = 0.;              /* pas pour l'abscisses */
    real k = 0.;              /* pas pour le temps */

    real r_left = 0.;         /* Borne inferieure en x */
    real r_right = 0.;        /* Borne superieure en x */
    real t_0 = 0.;            /* Temps initial */
    real t_f = 0.;            /* Temps final */

    real r_dis = 0.;          /* Abscisse de la discontinuite */
    real l_border = 0.;       /* Condition au bord gauche */
    real r_border = 0.;       /* Condition au bord droit */

    real v_l = 0.;            /* Valeur a gauche de la discontinuite */
    real v_r = 0.;            /* Valeur a droite de la discontinuite */

    real* vnew = NULL;        /* vecteur contenant  $V^{n+1}$  */
    real* vold = NULL;        /* vecteur contenant  $V^n$  */

```

```

real* vect_r = NULL; /* Vecteur contenant l'intervalle discretise */
real* vect_R = NULL; /* Vecteur contenant le rayon pour lequel
                       est calculee la solution. Il ne
                       correspond plus avec la grille. */

/* ----- FIN DE LA DECLARATION ----- */

/* Dialogue avec l'utilisateur */

printf("Donner la borne inferieure en r: ");
scan_real(&r_left);
printf("r_left = %g\n", r_left);

printf("Donner la borne superieure en r: ");
scan_real(&r_right);
printf("r_right = %g\n\n", r_right);

printf("Donner le temps initial: ");
scan_real(&t_0);
printf("t_0 = %g\n", t_0);

printf("Donner le temps final: ");
scan_real(&t_f);
printf("t_f = %g\n", t_f);

printf("Donner la valeur de la masse M: ");
scan_real(&M);
printf("M = %g\n", M);

printf("Entrer la valeur du pas h en espace: ");
scan_real(&h);
printf("h = %g\n", h);

printf("Entrer la valeur du pas k en temps: ");
scan_real(&k);
printf("k = %g\n", k);

printf("Entrer la valeur a gauche de la discontinuite v_left: ");
scan_real(&v_l);
printf("v_l = %g\n", v_l);

printf("Entrer la valeur a droit de la discontinuite v_right: ");
scan_real(&v_r);
printf("v_right = %g\n", v_r);

printf("Entrer l'abscisse de la discontinuite: ");
scan_real(&r_dis);
printf("r_dis = %g\n", r_dis);

/* Calcul du nombre de points en x */
nr = (int) (ceil((r_right - r_left)/h + 1));
printf("nr = %d\n\n", nr);

/* Calcul du nombre de points en t */
nt = (int) (ceil((t_f - t_0)/k + 1));
printf("nt = %d\n\n", nt);

/* Allocation memoire des tableaux et matrices */
vnew = (real*)calloc((size_t)nr, sizeof(real));

```

```

vold = (real*)calloc((size_t)nr, sizeof(real));
vect_r = (real*)calloc((size_t)nr, sizeof(real));
vect_R = (real*)calloc((size_t)nr, sizeof(real));

/* ----- CALCUL + ECRITURE DE LA SOLUTION ----- */

remove(FILENAME); /* Si le fichier existe deja on le supprime */

/* Ecriture de l'entete contenant le
nombre de points en x et d'instant */
entete(FILENAME, nr-2, nt, r_left, r_right, t_0, t_f, h, k);

/* Remplissage de vect_r */
vect_r[j] = r_left;
for(j=1;j<nr-1;j++){
    vect_r[j] = r_left + (real)j*h;
}
vect_r[nr-1] = r_right;

/* Calcul de la valeur aux bornes */
l_border = V(v_l, r_dis, r_left);
r_border = V(v_r, r_dis, r_right);

/* Initialisation du probleme */
vold[0] = l_border;
for(j=1;j<nr-1;j++){
    if(vect_r[j]<=r_dis){
        vold[j] = V(v_l, r_dis, vect_r[j]);
    }
    else if(vect_r[j]>r_dis){
        vold[j] = V(v_r, r_dis, vect_r[j]);
    }
    else{
        fprintf(stderr, "Erreur !\n");
        exit(EXIT_FAILURE);
    }
}
vold[nr-1] = r_border;

for(j=0;j<nr;j++){
    vect_R[j] = vect_r[j];
}

printf("Temps n = 0\n");
for(j=0;j<nr;j++){
    printf("anc[%d] = %g\n", j, vold[j]);
}
printf("\n");

ecrit(FILENAME, nr, vold); /* Ecriture de la solution */
/* au temps initial */

for(n=1;n<nt;n++){
    printf("Temps n = %d\n", n);
    glimm(nr, k, h, r_dis, n*k, vect_R, vect_r, vold, vnew);
    ecrit(FILENAME, nr, vnew);
    for(j=0;j<nr;j++){
        /* Transfert de vnew dans
        vold pour la */
        vold[j] = vnew[j]; /* prochaine iteration */
    }
}

```

```

    }

    /* ----- */

    /* Desallocation memoire des tableaux et matrices */
    free(vect_r);
    free(vect_R);
    free(vnew);
    free(vold);

    exit(EXIT_SUCCESS);
}

/*****
 * Nom ..... : methodes.c
 * Role ..... : Implementation de la methode de Glimm.
 *****/

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

#include "real_precision.h"
#include "methodes.h"
#include "functions.h"
#include "parametres.h"

/* Second membre de l'EDO pour trouver la forme du choc */
real scd_mbr(real t, real y, real v_l, real v_r, real r_dis){
    return (1-2*M/y)*(V(v_l, r_dis, y)+V(v_r, r_dis, y))/2;
}

/* Methode de Runge-Kutta d'ordre 4 */
real runge_kutta4(real u_i, real t_i, real dt, real v_l, real v_r,
                  real r_dis){
    real k1 = 0.;
    real k2 = 0.;
    real k3 = 0.;
    real k4 = 0.;

    k1 = scd_mbr(t_i, u_i, v_l, v_r, r_dis);
    k2 = scd_mbr(t_i + dt/2, u_i + k1*dt/2, v_l, v_r, r_dis);
    k3 = scd_mbr(t_i + dt/2, u_i + k2*dt/2, v_l, v_r, r_dis);
    k4 = scd_mbr(t_i + dt, u_i + k3*dt, v_l, v_r, r_dis);

    return u_i + (k1 + 2*(k2 + k3) + k4)*dt/6;
}

void glimm(const int nr, real dt, real dr, real r_dis, real t,
           real* vect_R, real* vect_r, real* anc, real* nouv){

    int i = 0;          /* Compteur pour l'espace */

    real K = 0;
    real l_border = anc[0];
    real r_border = anc[nr-1];

    real left = 0;
    real right = 0;

```

```

real theta = 0.;    /* Variable aleatoire a valeurs
                    dans [-0.5, 0.5] */
real sigma_l = 0;
real sigma_r = 0;
real sigma = 0;    /* solution des trajectoires pour chaque choc
                    entre deux cellules */

real* Rnew = NULL;
real* R = NULL;

/* Allocation memoire des tableaux */
R = (real*)calloc((size_t)nr-1, sizeof(real));
Rnew = (real*)calloc((size_t)nr-2, sizeof(real));

/* Tirage d'un nombre entre -0.5 et 0.5 */
theta = (real)(rand()) / (real)(RAND_MAX) - (real)0.5;

/* Remplissage du vecteur R */
for(i=0;i<nr-1;i++){
    R[i] = vect_r[i] + dr/2 + theta*dr;
}

/* ----- METHODE DE GLIMM ----- */

/* Passage de t a t+1/2 */
for(i=0;i<nr-1;i++){
    left = V(anc[i], vect_R[i], vect_r[i] + dr/2);
    right = V(anc[i+1], vect_R[i+1], vect_r[i+1] - dr/2);

    /* ***** CAS D'UN CHOC ***** */

    if(left >= right || fabs(left-right) < 0.0000000001){
        sigma = runge_kutta4(vect_r[i] + dr/2, 0, dt/2,
                             left, right,
                             vect_r[i] + dr/2);
        if(R[i]<=sigma){
            nouv[i] = V(anc[i], vect_R[i], R[i]);
        }
        else{
            nouv[i] = V(anc[i+1], vect_R[i+1], R[i]);
        }
    }

    /* ***** CAS D'UNE RAREFACTION ***** */

    else{}
}

for(i=0;i<nr-1;i++){
    anc[i] = nouv[i];
}

/* Tirage d'un autre nombre entre -0.5 et 0.5 */
theta = (real)(rand())/(real)(RAND_MAX) - (real) 0.5;
for(i=0;i<nr-2;i++){
    Rnew[i] = vect_r[i+1] + theta*dr;
}

```



```

/* Passage de t+1/2 a t+1 */
nouv[0] = l_border;
for (i=1;i<nr-1;i++){
    left = V(anc[i-1], R[i-1], vect_r[i]);
    right = V(anc[i], R[i], vect_r[i]);

    /* ***** CAS D'UN CHOC ***** */

    if (left >= right || fabs(left-right) < 0.000000000001){
        sigma = runge_kutta4(vect_r[i], dt/2, dt/2,
                             left, right, vect_r[i]);
        if (Rnew[i-1]<=sigma){
            nouv[i] = V(anc[i-1], R[i-1], Rnew[i-1]);
        }
        else {
            nouv[i] = V(anc[i], R[i], Rnew[i-1]);
        }
    }

    /* ***** CAS D'UNE RAREFACTION ***** */

    else {}
}
nouv[nr-1] = r_border;

vect_R[0] = vect_r[0];
for (i=1;i<nr-1;i++){
    vect_R[i] = Rnew[i-1];
}
vect_R[nr-1] = vect_r[nr-1];

free(R);
free(Rnew);
}

```

Références

- [1] M. Kardar, G. Parisi, and Y.-C. Zhang, “Dynamic scaling of growing interfaces,” *Physical Review Letters*, vol. 56, no. 9, p. 889, 1986.
- [2] S. F. Shandarin, “Three-dimensional burgers’ equation as a model for the large-scale structure formation in the universe,” in *Stochastic Models in Geosystems*, pp. 401–413, Springer, 1997.
- [3] P. G. LeFloch, H. Makhlof, and B. Okutmustur, “Relativistic burgers equations on curved space-times. derivation and finite volume approximation,” *SIAM Journal on Numerical Analysis*, vol. 50, no. 4, pp. 2136–2158, 2012.
- [4] T. Ceylan, P. G. LeFloch, and B. Okutmustur, “The relativistic burgers equation on a flrw background and its finite volume approximation,” *arXiv preprint arXiv :1512.08142*, 2015.
- [5] T. Ceylan and B. Okutmustur, “Derivation of relativistic burgers equation on de sitter background,” in *Proceedings of 19th International Conference on Applied Mathematics, WSEAS-AMATH’14, Mathematics and Computers in Science and Engineering Series*, vol. 38, pp. 41–47, 2014.
- [6] M. Ben-Artzi and P. G. Le Floch, “Well-posedness theory for geometry-compatible hyperbolic conservation laws on manifolds,” in *Annales de l’IHP Analyse non linéaire*, vol. 24, pp. 989–1008, 2007.
- [7] P. G. LeFloch, “Hyperbolic conservation laws on spacetimes,” in *Nonlinear Conservation Laws and Applications*, pp. 379–391, Springer, 2011.
- [8] R. J. LeVeque, *Numerical methods for conservation laws*, vol. 132. Springer, 1992.
- [9] O. A. Oleinik, “Discontinuous solutions of non-linear differential equations,” *Uspekhi Matematicheskikh Nauk*, vol. 12, no. 3, pp. 3–73, 1957.
- [10] S. Weinberg, *Gravitation and cosmology : Principle and applications of general theory of relativity*. John Wiley and Sons, Inc., New York, 1972.
- [11] P. G. LeFloch and S. Xiang, “The relativistic burgers model on the schwarzschild spacetime : existence theory and time-asymptotics,” En préparation.
- [12] P. G. LeFloch and S. Xiang, “Weakly regular fluid flows with bounded variation on the domain of outer communication of a schwarzschild black hole spacetime,” *Journal de Mathématiques Pures et Appliquées*, 2016.
- [13] R. Courant, K. Friedrichs, and H. Lewy, “Über die partiellen differenzengleichungen der mathematischen physik,” *Mathematische annalen*, vol. 100, no. 1, pp. 32–74, 1928.
- [14] R. Courant, K. Friedrichs, and H. Lewy, “On the partial difference equations of mathematical physics,” *IBM journal*, vol. 11, no. 2, pp. 215–234, 1967.
- [15] J. Glimm, “Solutions in the large for nonlinear hyperbolic systems of equations,” *Communications on Pure and Applied Mathematics*, vol. 18, no. 4, pp. 697–715, 1965.
- [16] A. J. Chorin, “Random choice solution of hyperbolic systems,” *Journal of Computational Physics*, vol. 22, no. 4, pp. 517–533, 1976.
- [17] A. J. Chorin, “Random choice methods with applications to reacting gas flow,” *Journal of Computational physics*, vol. 25, no. 3, pp. 253–272, 1977.