École Polytechnique Fédérale de Lausanne
Semestre de printemps 2026
Joan Decker
Ewout Devlaminck

Physique Numérique I – Exercice 1

A rendre jusqu'au mardi 10 mars 2026 sur le site http://moodle.epfl.ch/mod/assign/view.php?id=835967

1 Modélisation 0D des électrons "runaway" Schémas d'Euler explicite, semi-implicite et implicite

L'objectif de cet exercice est d'étudier le comportement simplifié des électrons dits « runaway» à l'aide d'une équation différentielle ordinaire 0D décrivant l'évolution de leur population au fil du temps. D'un point de vue numérique, nous souhaitons étudier les propriétés de convergence et de stabilité de trois schémas d'Euler différents : explicite, implicite et semi-implicite. Pour résoudre l'équation $dy/dt = f(y,t)$, on peut décrire ces trois schémas de façon unfiée ainsi :

$$y_{n+1} = y_n + [\alpha f(y_n, t_n) + (1-\alpha) f(y_{n+1}, t_{n+1})]\Delta t \tag{1}$$

— Le schéma d'Euler explicite $\alpha = 1$.
— Le schéma d'Euler implicite $\alpha = 0$.
— Le schéma d'Euler semi-implicite $\alpha = 0.5$.

Notez que le travail numérique peut-être effectué sans avoir achevé le travail analytique. Il suffit d'utiliser les équations (3) et (5) ainsi que la solution de 1.1.a.

1.1 Théorie et calculs analytiques [12 pts]

La friction subie par un électron dans un plasma lors de collisions avec d'autres particules n'augmente pas de manière monotone avec leur énergie, comme c'est généralement le cas pour la friction aérodynamique des objets macroscopiques dans notre atmosphère. Au contraire, la friction qu'ils subissent lors des collisions dites coulombiènes tend à diminuer à mesure qu'ils acquièrent suffisamment d'énergie. Si un champ électrique suffisamment intense est présent pour les accélérer le long du champ magnétique, cette accélération peut devenir dominante par rapport à la friction, ce qui signifie que l'électron gagne continuellement de l'énergie.

En ce sens, nous pouvons définir un champ électrique critique (Connor-Hastie) qui est juste assez fort pour accélérer en continu les électrons les plus rapides.

Nous voulons maintenant étudier ce qui arrive à une population de runaways en tenant compte de plusieurs effets. En supposant que le champ électrique soit beaucoup plus élevé que le champ critique de Connor-Hastie, l'évolution de la densité des électrons runaway peut s'écrire comme suit [Decker NF 2024] :

$$\frac{dn}{dt} = d + \left[\gamma\left(1 - \frac{n}{n_R}\right) - \nu\right] n \tag{2}$$

où d est une source indépendant de la densité runaway (génération Dreicer, désintégration bêta du tritium, diffusion Compton, etc), γ est le taux de croissance de l'avalanche (un runaway peut entrer en collision frontale avec un autre électron et lui donner suffisamment d'énergie pour qu'il devienne également un runaway), n_R est la densité de runaways nécessaire pour entraîner l'ensemble du courant plasma toroïdal, et ν est un taux de perte, généralement dû au transport radial.

Définissant $\bar{\nu} \equiv \nu/\gamma$, $\bar{n} \equiv n/n_R$, $\bar{d} \equiv d/(\gamma n_R)$, $\bar{t} \equiv \gamma t$, et $\bar{\gamma} \equiv 1 - \bar{\nu}$, l'équation se normalise à

$$\frac{d\bar{n}}{d\bar{t}} = \bar{d} + [\bar{\gamma} - \bar{n}]\,\bar{n} \tag{3}$$

(a) **[2 pts]** Dérivez la solution à l'état stationnaire $\bar{n}_\infty$.

(b) **[3 pts]** Simplifiez la solution à l'état stationnaire pour le cas $\bar{d} \ll \bar{\gamma}^2$.
Décrivez et commentez ensuite la densité finale dans les cas suivants : (1) les pertes dominent $\bar{\nu} > 1$ ($\bar{\gamma} < 0$) ; (2) les avalanches dominent $\bar{\nu} < 1$ ($\bar{\gamma} > 0$).

(c) **[5 pts] Solution exacte dépendante du temps.**
Résolvez l'équation dépendante du temps :

$$\frac{d\bar{n}}{d\bar{t}} = \bar{d} + \bar{\gamma}\bar{n} - \bar{n}^2, \tag{4}$$

pour montrer, en introduisant $\beta \equiv \sqrt{\bar{\gamma}^2 + 4\bar{d}}$:

$$\bar{n}(\bar{t}) = \frac{(\beta + \bar{\gamma})\left[1 - e^{-\beta\bar{t}}\right]/2}{1 + e^{-\beta\bar{t}}\,(\beta + \bar{\gamma})\,/\,(\beta - \bar{\gamma})} \tag{5}$$

ou de manière équivalente

$$\bar{n}(\bar{t}) = \frac{2\bar{d}\left[1 - e^{-\beta\bar{t}}\right]}{\beta - \bar{\gamma} + (\beta + \bar{\gamma})\,e^{-\beta\bar{t}}} \tag{6}$$

Indice : en utilisant les racines $\bar{n}_\pm$ trouvées dans la question 1.1(a) pour exprimer le côté droit comme un produit de deux termes et utilisez la technique de séparation des variables et les fractions partielles. Gardez à l'esprit que $\bar{n}_- < 0 \leq \bar{n} < \bar{n}_+$.

(d) **[2 pts] Limites et comportements asymptotiques**
Verifiez qu'on retrouve l'état stationnaire pour $\bar{t} \to \infty$.
Inspectez le comportement initial, lorsque $\bar{t}\beta \ll 1$.
Approximez l'évolution temporelle pour le cas $\bar{d} \ll \bar{\gamma}^2$ et $\bar{\gamma} < 0$ (pertes dominantes).

1.2 Implémentation en C++

Télécharger le fichier Exercise1_student.zip du site Moodle. Dans le code, il faut implémenter le schéma d'Euler, Eq.(1). Vous devez compléter le code partout où vous trouvez « TODO » dans le fichier .cpp et le script Python associé.
Important : il vous faut au moins une fois sur les deux sessions d'exercices montrer et expliquer votre code à votre assistant.

1.3 Simulations et Analyses [42+7 pts]

On effectue des simulations avec le programme que l'on vient d'écrire et de compiler. Si l'éxécutable est `engine`, taper à la ligne de commande d'un Terminal : `> ./engine configuration.in.example`
La visualisation des résultats numériques se fait avec Python (ou `Matlab`). Voir le fichier parameterscan.py que vous modifierez selon les besoins.

(a) **[16 pts]** *Cas $\bar{\gamma} > 0$ où les avalanches dominent.*
On prend $\bar{t}_{final} = 32$, $\bar{n}(0) = 0$, $\bar{\gamma} = 0.5$, $\bar{d} = 0.01$.
Pour chacun des trois schémas numériques implémentés (Euler explicite, implicite et semi-implicite), effectuer des simulations avec des nombres de pas de temps (nsteps) différents, vérifier la convergence numérique de la densité finale et déterminer l'ordre de convergence.
Indication : on définit l'erreur comme la différence relative entre la densité finale de runaway

calculée numériquement et analytiquement.
Nous nous intéressons au temps caractéristique $\bar{\tau}$, défini ici comme le temps nécessaire à la population runaway pour atteindre 20 % de sa valeur d'équilibre. Effectuez les mêmes tests de convergence pour cette quantité et commentez.
Indication : nous pouvons trouver le valeur $\bar{\tau}$ correspondant à $\bar{n}(\bar{\tau})/\bar{n}_\infty = 0.2$ en utilisant la fonction np.interp$(0.2, \bar{n}(t)/\bar{n}_\infty, t)$ de Python. Pour obtenir de meilleures performances, vous pouvez également utiliser CubicSpline *du package scipy.interpolate.*
Le code pour produire les figures correspondantes est déjà disponible.

(b) **[4 pts]** *Comportement pour $\bar{t}\beta \ll 1$.*
Étudiez l'évolution initiale $\bar{t}\beta \ll 1$ dans un cas à haute précision (i.e. nsteps = 1024) et comparez à vos calculs analytiques.

(c) **[6 pts]** *Cas $\bar{\gamma} < 0$ où les pertes dominent.*
Étudiez le cas $\bar{\gamma} = -0.2$ pour vérifier si vos estimations concernantes ce régime étaient valides. Commentez les différences éventuelles entre le comportements analytiques (physique) et numériques obtenus dans les trois schémas d'Euler.

(d) **[6 pts]** *Etude de stabilité.*
Exécuter le code dans le schéma d'Euler explicite en partant du cas 1.3-a et en réduisant le nombre d'itérations (i.e. nsteps = 16, 8, 4). Quel comportement observez-vous ? Pouvez-vous prédire ce comportement à partir de l'analyse de Von Neumann de la stabilité pour ce problème ?

(e) **[10 pts]** *Schéma (semi-)implicite sans itération.*
Cette équation différentielle ordinaire particulière possède une propriété intéressante. Pour les schémas d'Euler implicites et semi-implicites (Eq. 1 appliquée à Eq. 3), une solution analytique à l'équation discrète permet d'exprimer $\bar{n}_{i+1}$ en terme des quantités connues, sans recourir à un processus itératif. Trouvez ces expressions, copiez le code de engine.cpp dans un nouveau fichier engine2.cpp et implémentez cette solution analytique de l'équation discrète. Appliquez au calcul de $\bar{\tau}$ dans le cas décrit dans 1.3-a avec une précision temporelle modérée (i.e. nsteps = 32) et vérifiez que la solution obtenue par le processus itératif converge vers cette solution analytique quand la tolérance diminue.

(f) **[max 7 pts] Facultatif.** Le but de cette question facultative est d'aborder brièvement la question de l'optimisation. Pour ce problème 0D, ni les ressources informatiques ni le temps de calcul ne posent de problème, mais pour des simulations complexes, l'optimisation du schéma utilisé est l'un des choix les plus importants à faire. Le nombre de fois où notre fonction « compute_f » est appelée dans le script est inclus dans la sortie et utilisé dans l'un des graphiques.
De toute évidence, le schéma explicite n'utilise qu'une seule évaluation par pas de temps. Le schéma semi-implicite utilise plusieurs évaluations par pas de temps, mais avec une meilleure précision que le schéma explicite pour un pas de temps donné. La question de déterminer quel schéma est le plus efficace (en terme de nombre d'appels à « compute_f ») pour un objectif de précision fixé.

(i) Dans le fichier .cpp fourni, nous avons introduit un minimum fixe de 3 itérations si une méthode implicite est choisie. Si vous supprimez cette condition, que se passe-t-il dans le schéma semi-implicite lorsque nous réduisons considérablement les pas de temps pour une tolérance relativement élevée (par exemple 1e-4) ? Commentez le comportement de convergence de la méthode semi-implicite dans ce cas.

(ii) En partant du cas 1.3-a, essayez de trouver une combinaison optimale entre le schéma,

la tolérance, le nombre d'itérations forcées et le pas de temps afin d'obtenir une erreur sur tau inférieure à 1e-4 avec le moins d'évaluations possible de compute_f.

1.4 Rédaction du rapport en LaTeX, soumission du rapport en pdf et du code source C++

(a) Rédiger un rapport dans lequel les résultats des simulations ainsi que les réponses aux questions ci-dessus sont présentées et analysées. On peut partir du fichier source sur Moodle LaTeX(SqueletteRapport.tex) sur "Moodle". Voir aussi le dossier ressources sur Moodle : Dossier LaTeX.

(b) Préparer le fichier du rapport en format `pdf` portant le nom `RapportExercice1_Nom1_Nom2.pdf`.

(c) Préparer les fichiers source C++ `Exercice1_Nom1_Nom2_x.cpp`. (Le x correspond à 1,2,... si vous en avez plusieurs).

(d) Le lien de soumission est ici.

En plus des points mentionnés ci-dessus, **[6 pts]** *sont attribués pour la participation en classe : montrez et discutez vos résultats avec votre assistant.*