

# freeze: documentation

Thomas Foetisch

November 9, 2017

## 1 Introduction

Le package **freeze** implémente une méthode numérique de type éléments finis et une formule de Chernoff pour approximer la solution d'un problème de Stefan en une dimension d'espace.

## 2 Formulation mathématique du problème

Soit  $\rho$  la densité supposée constante d'un matériau qui subit une transition entre le phases solide et liquide à la température  $\Theta_t$ . La chaleur spécifique du matériau dans les phases solide et liquide est notée respectivement  $C_{p,1}$  et  $C_{p,2}$ . Soit  $\lambda$  la chaleur latente de transition de phase, et  $\eta$  la fraction solide en fonction de la température  $\Theta$ :

$$\eta(\Theta) = \begin{cases} 1, & \Theta < \Theta_t, \\ 0, & \Theta \geq \Theta_t. \end{cases}$$

On notera dorénavant la chaleur spécifique en fonction de la température:

$$C_p(\Theta) = \begin{cases} C_{p,1}, & \Theta < \Theta_t, \\ C_{p,2}, & \Theta \geq \Theta_t. \end{cases}$$

De même, on note  $\kappa_1$  et  $\kappa_2$  les coefficients de conduction thermique des phases solide et liquide, et on note  $\kappa$  le coefficient de conduction en fonction de la température:

$$\kappa(\Theta) = \begin{cases} \kappa_1, & \Theta < \Theta_t, \\ \kappa_2, & \Theta \geq \Theta_t. \end{cases}$$

On introduit l'enthalpie par unité de volume du matériau  $h$  en fonction de la température  $\Theta$ :

$$H(\Theta) = \int_0^\Theta \rho C_p(s) ds + \rho \lambda (1 - \eta(\Theta)). \quad (1)$$

Soit  $\beta : \mathbb{R} \rightarrow \mathbb{R}$  la fonction définie par

$$\beta(h) = \begin{cases} \frac{h}{\rho C_{p,1}}, & \forall h < \Theta_t \rho C_{p,1}, \\ \Theta_f, & \forall \Theta_t \rho C_{p,1} \leq h \leq \Theta_t \rho C_{p,1} + \rho \lambda, \\ \Theta_f + \frac{h - (\Theta_t \rho C_{p,1} + \rho \lambda)}{\rho C_{p,2}}, & \forall \Theta_t \rho C_{p,1} + \rho \lambda < h. \end{cases}$$

On note que la fonction  $\beta$  est l'inverse de la fonction  $h$  au sens que:

$$\beta(H(\Theta)) = \Theta, \quad \forall \Theta.$$

On note  $\partial\Omega = \Gamma_D \cup \Gamma_N$ , tel que  $\Gamma_D \cap \Gamma_N = \emptyset$ , où  $\Gamma_N$  et  $\Gamma_D$  sont donnés. On suppose que le matériau occupe le domaine ouvert  $\Omega = (0, \infty)$ . On note  $u(t, x)$  l'enthalpie du matériau au point  $x \in \Omega$  et à l'instant  $t > 0$ . Le problème de Stefan consiste à chercher une fonction  $u : [0, \infty) \times \Omega$  solution de l'équation:

$$\begin{aligned} \frac{\partial u}{\partial t} - \frac{\partial}{\partial x} \left( \kappa \frac{\partial}{\partial x} \beta(u) \right) &= 0, & x \in \Omega, \quad t > 0, \\ u(0, x) &= u_0(x), & x \in \Omega, \\ \beta(h) &= g_D(t, x), & x \in \Gamma_D, \quad t > 0 \\ \frac{\partial \beta(h)}{\partial n}(t, x) &= g_N(t, x), & x \in \Gamma_N, \quad t > 0. \end{aligned} \tag{2}$$

Le problème (2) étant singulier, il est à comprendre au sens faible. Le problème faible correspondant à l'équation (2) est le suivant. On cherche une fonction  $u : [0, \infty) \times \Omega \rightarrow \mathbb{R}$  telle que:

$$\int_{\Omega} \frac{\partial u}{\partial t} v \, dx + \int_{\Omega} \nabla \beta(u) \cdot \nabla v \, dx = \int_{\Gamma_D} g_N v \, d\sigma \tag{3}$$

et

$$\begin{aligned} u(0, x) &= u_0(x), & \forall x \in \Omega, \\ u(t, x) &= g_D(t, x), & \forall x \in \Gamma_D, \end{aligned}$$

pour toute fonction  $v : \Omega \rightarrow \mathbb{R}$  telle que

$$v(t, x) = 0, \quad \forall x \in \Gamma_D$$

suffisamment lisse.

Dans le cas où on s'intéresse au problème 3D avec une symétrie sphérique, on utilise la formulation faible suivante:

$$\int_{\Omega} \frac{\partial u}{\partial t} v r^2 \, dr + \int_{\Omega} \partial_r \beta(u) \cdot \partial_r v r^2 \, dr = \int_{\Gamma_D} g_N v r^2 \, d\sigma \tag{4}$$

où  $u$  et  $g_N$  sont fonction du rayon  $r$  et du temps  $t$ , et  $v$  est fonction de  $r$ .

### 3 Discrétisation

On discrétise le problème (2) en temps et en espace en suivant le travail de [PSV88]. Soit  $\tau > 0$  le pas de temps,  $t^k = k\tau$ ,  $k = 0, 1, 2, \dots$  la discrétisation en temps et  $\mu$  une constante telle que:

$$0 < \mu \leq \frac{1}{\sup \beta'(h)}. \tag{5}$$

Soit  $k \in \mathbb{N}$ , et on note  $\Theta^k$  l'approximation de  $\Theta(t^k, \cdot)$ , et de même pour  $u^k$  l'approximation de  $u(t^k, \cdot)$ . On pose  $\Theta^0(x) = \Theta_0(x)$  et  $u^0 = H(\Theta^0)$ . A chaque pas de temps, l'équation (2) est linéarisée de la manière suivante.

$$\Theta^{k+1} - \frac{\tau}{\mu} \nabla (\kappa \nabla \Theta^{k+1}) = \beta(u^k), \quad (6)$$

$$u^{k+1} = u^k + \mu (\Theta^{k+1} - \beta(u^k)) \quad (7)$$

avec les conditions de bord:

$$\Theta^{k+1}(x) = g_D(t^{k+1}, x), \quad x \in \Gamma_D \quad (8)$$

$$\frac{\partial \Theta^{k+1}}{\partial n}(t, x) = g_N(t^{k+1}, x), \quad x \in \Gamma_N. \quad (9)$$

L'équation (6) est discrétisée par une méthode éléments finis  $\mathbb{P}_1$ - $\mathbb{P}_1$  sur un maillage uniforme d'un sous-ensemble borné  $\Omega' \subset \Omega$ . Typiquement,  $\Omega' = [0, L]$  avec  $L > 0$ . L'équation (7) est une simple équation algébrique, et la somme est effectuée degré de liberté par degré de liberté, de même que l'évaluation de  $\beta(u^k)$ .

## 4 Le package **freeze**

Le code distribué dans le package **freeze** implémente le schéma numérique décrit à la section 3. Cette section contient des informations quant à l'installation du package, des dépendances requises, de l'exécution du modèle et de l'accès aux résultats.

### 4.1 Installation

La compilation du code nécessite les dépendance suivantes:

- **petsc** version 3.7.6. Les versions ultérieures sont probablement supportées, mais sans garanties. Les version 3.6.3 et antérieures ne sont **pas** supportées.
- **liblapacke-dev** version 3.5.0. Cette version correspond probablement aux packages distribués par les distributions de linux principales (Ubuntu, debian, ...)
- **clang++** version 3.8.0. D'autres compilateurs sont probablement supportés, pour autant qu'ils implémentent le standard **C++14**. En revanche, en raison d'un bug dans **g++**, celui-ci n'est pas supporté.

Pour compiler le package, il suffit de se placer dans le dossier **freeze** et de lancer **make**:

```
[user@localhost ~/dev/freeze> make
```

Une fois la compilation terminée, les exécutables se trouvent dans le sous-dossier **freeze/bin/**. Il y en a deux:

- **freeze**: calcule la solution transitoire d'un problème de Stefan, selon les paramètres spécifiés dans le fichier passé en ligne de commande,

- **neumann**: calcule la solution au temps final d'un problème de Stefan, selon les paramètres spécifiés dans le fichier passé en ligne de commande, puis calcul l'erreur  $L_2$  entre le résultat obtenu et la solution exacte de Neumann du problème adimensionné.

On décrit l'usage de ces deux exécutables dans les parties qui suivent.

## 4.2 Execution des modèles: **freeze**

L'exécutable **freeze** calcule la solution transitoire d'un problème de Stefan. La spécification précise se fait par l'intermédiaire de fichier de configuration qui sont passés en paramètres sur la ligne de commande.

Des exemples de fichier de configuration peuvent être trouvés dans le sous-dossier **freeze/bin**. Par exemple, la commande suivante:

```
[user@localhost ~/dev/freeze/bin> ./freeze cryolite-particle-remelt.conf
```

calcule la solution transitoire d'une situation qui correspond à la formation initiale d'une couche de bain gelé à la surface d'une particule, et qui finit par refondre.

L'exécution de cette commande produit (potentiellement, en fonction des paramètres spécifiés dans **cryolite-particle-remelt.conf**) en output une série de fichiers ASCII dans le sous-dossier **freeze/bin/output/**:

- **cryolite-particle-remelt-beta.dat**: le graph de la fonction  $\beta$  autour de la transition de phase,
- **cryolite-particle-remelt-neumann-exact-solution.dat**: le graph de la solution exacte de Neumann évaluée au temps final,
- **cryolite-particle-remelt-ts-[0-9]+.dat**: la solution numérique transitoire à chaque pas de temps,
- **cryolite-particle-remelt.dat**: la solution numérique au temps final,
- **cryolite-particle-remelt-transitions.dat**: la position de la transition de phase en fonction du temps.

Les données de ces fichiers peuvent être visualisées par **GNUPlot**, **MatLAB**, **GNU Octave** ou équivalent sans post-processing nécessaire.

L'output sur la sortie standard est minimal, et indique uniquement l'instance de la collection de paramètre spécifiée dans le fichier de paramètre qui est en cours de calcul. Pour l'exemple ci-dessus, on obtient:

```
[user@localhost ~/dev/freeze/bin> ./freeze cryolite-particle-remelt.conf
running parameter set collection #1 of 1
[user@localhost ~/dev/freeze/bin>
```

puisque la collection de paramètre est un singleton dans ce cas-ci.

Un exemple de collection non triviale peut être trouvé dans le fichier de paramètre **cryolite-particle-remelt-study**. Le lecteur intéressé par la syntaxe et la sémantique de ces fichiers de paramètres se référera à la documentation du package **parameter**.

### 4.3 Execution des modèles: **neumann**

L'exécutable **neumann** effectue les mêmes opérations que l'exécutable **freeze**, mais calcule en plus l'erreur  $L_2$  entre la solution numérique au temps final et la solution exacte de Neumann. Les paramètres spécifiés dans le fichier de configuration doivent donc correspondre à la solution de Neumann pour que l'erreur calculée ait un sens. Le lecteur intéressé une discussion détaillée de cette solution dans [Hil87].

Le fichier de paramètre **stephan-exact-neumann-solution.conf** correspond à cette configuration. L'output de l'exécutable **neumann** est le même que celui de **freeze** en terme de fichiers ASCII, mais affiche en plus l'erreur  $L_2$  sur la sortie standard à l'issue du calcul.

Le fichier **stephan-exact-neumann-solution.conf** spécifie une collection de paramètres qui correspondent à une étude de convergence de l'erreur  $L_2$ . Par exemple, on obtient:

```
[user@localhost ~/dev/freeze/bin> ./neumann stephan-exact-neumann-solution.conf
running parameter set collection #1 of 6
L2_error(t = 0.5) = 0.00995969
running parameter set collection #2 of 6
L2_error(t = 0.5) = 0.00565716
running parameter set collection #3 of 6
L2_error(t = 0.5) = 0.00343643
running parameter set collection #4 of 6
L2_error(t = 0.5) = 0.00201683
running parameter set collection #5 of 6
L2_error(t = 0.5) = 0.00121125
running parameter set collection #6 of 6
L2_error(t = 0.5) = 0.000713004
[user@localhost ~/dev/freeze/bin>
```

## References

- [Hil87] James M Hill. *One-dimensional Stefan problems : an introduction*. English. Includes index. Harlow, Essex, England : Longman Scientific & Technical ; New York, NY : Wiley, 1987. ISBN: 0470203889.
- [PSV88] M Paolini, G Sacchi, and Claudio Verdi. "Finite element approximations of singular parabolic problems". In: *International Journal for Numerical Methods in Engineering* 26.9 (1988), pp. 1989–2007.