



Programmation Avancée et Projet Rapport Equation de la chaleur

Lina EL HADDAJ
Ranim BOUSSEMA

Année universitaire : 2023-2024

Contents

1	Introduction	1
2	Partie Technique	1
2.1	Namespace	1
2.2	Structures de données	1
2.3	Cas 1 : La Barre	2
2.4	Cas 2 : La Plaque	5
3	Exécution du Programme	9
3.1	Configuration des Entrées	9
3.2	Résultats de l'Exécution	9
4	Conception	11
5	Défis Rencontrés	12
5.1	La bibliothèque SDL	12
5.2	Résultat de la Plaque	12
5.3	Complexité	12
6	Perspectives d'Amélioration	12
7	Conclusion	13

1. Introduction

Ce projet se donne pour objectif la simulation de l'évolution de la température à travers un support exposé à des sources de chaleur. Ce support peut prendre la forme d'une barre ou d'une plaque, constitué de matériaux tels que le cuivre, le fer, le verre ou le polystyrène. La résolution de ce problème repose sur la solution de l'équation de la chaleur qui décrit la variation temporelle de la température en chaque point du support.

L'équation de la chaleur, au cœur de notre étude, se présente sous la forme :

$$\frac{\partial u}{\partial t} = \frac{\lambda}{\rho c} \Delta u + \frac{F}{\rho c}$$

Avec :

- $t \in \mathbb{R}^+$ le temps d'étude ;
- $x \in [0, L]^d$ le point d'étude ;
- $u(t, x)$ la température en degré Kelvin du matériau étudié, avec $u : \mathbb{R}^+ \times [0, L]^d \rightarrow \mathbb{R}^+$.

On définit également Δ comme le Laplacien sur $[0, L]^d$ et $F : \mathbb{R}^+ \times [0, L]^d \rightarrow \mathbb{R}^+$ comme la source de chaleur appliquée sous le support.

Au fil de ce rapport, nous explorerons la résolution numérique de cette équation pour les cas d'une barre et d'une plaque, analysant ainsi les comportements thermiques inhérents à chaque configuration.

2. Partie Technique

2.1 Namespace

Nous avons adopté un namespace appelé `materials`, qui regroupe quatre autres namespaces, chacun dédié à un matériau spécifique (*copper*, *iron*, *glass* et *polystyrene*). À l'intérieur de chaque namespace dédié à un matériau, on trouve les valeurs des trois grandeurs qui le caractérisent, à savoir λ (la conductivité thermique), ρ (la masse volumique) et c (la chaleur massique du matériau constituant l'objet).

2.2 Structures de données

Pour modéliser les variations spatio-temporelles de la température des deux supports, nous avons choisi d'utiliser des matrices. Ainsi, nous avons défini une classe `Matrix` capable de stocker des valeurs de type double. Cette classe permet également de créer des vecteurs ligne ou colonne en spécifiant le nombre de colonnes ou de lignes égal à 1 lors de la création d'une instance.

Afin d'optimiser le code et de n'avoir qu'un seul programme pour gérer à la fois la barre et la plaque, le fichier `main.cpp` contient les étapes communes aux deux

cas, c'est-à-dire le choix du support et du matériau. Le premier choix détermine l'appel à l'une des deux fonctions externes de calcul, tandis que le deuxième définit les constantes à utiliser pour ces calculs.

Voici la structure du code en question :

```
.
|-- headers
|   |-- animation.hpp
|   |-- bar.hpp
|   |-- constants.hpp
|   |-- interface.hpp
|   |-- material.hpp
|   |-- matrix.hpp
|   |-- surface.hpp
|
|-- animation.cpp
|-- bar.cpp
|-- interface.cpp
|-- main.cpp
|-- matrix.cpp
|-- surface.cpp
```

2.3 Cas 1 : La Barre

Les températures des différents points de la barre sont enregistrées dans des vecteurs colonnes, où chaque élément représente un point spécifique.

En notant N_{space} le nombre de points dans le maillage spatial et N_{time} le nombre de points dans le maillage temporel, la solution à ce problème se présente sous la forme d'une matrice de dimensions $N_{space} \times N_{time}$: chaque ligne et chaque colonne représentant respectivement un point du support et moment particulier d'observation.

2.3.1 Partie théorique : Équation de la chaleur

L'équation régissant la diffusion de la chaleur au sein d'une barre est formulée par :

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\lambda}{\rho c} \frac{\partial^2 u}{\partial x^2} + \frac{F}{\rho c}, & \forall (t, x) \in [0, t_{max}] \times [0, L] \\ u(0, x) = u_0, & \forall x \in [0, L] \\ \frac{\partial u}{\partial x}(t, 0) = 0, & \forall t \in [0, t_{max}] \\ u(t, L) = u_0, & \forall t \in [0, t_{max}] \end{cases}$$

La source de chaleur F est modélisée de manière distincte selon trois intervalles sur la barre, contribuant ainsi à la complexité du système :

$$\begin{cases} F(x) = t_{max} f^2 & \text{sur } [\frac{L}{10}, \frac{2L}{10}] \\ F(x) = \frac{3}{4} t_{max} f^2 & \text{sur } [\frac{5L}{10}, \frac{6L}{10}] \\ F(x) = 0 & \text{sinon} \end{cases}$$

Pour faciliter ce qui suit, introduisons les notations suivantes :

$$\begin{cases} J = 1, \dots, N_{space} \\ N = 1, \dots, N_{time} \\ x_j = x_0 + j \Delta x, \quad \forall j \in J \\ t_n = t_0 + n \Delta t, \quad \forall n \in N \\ u(x_j, t_n) = u_j^n, \quad \forall j \in J, \forall n \in N, \end{cases}$$

On définit Δx et Δt respectivement comme l'intervalle spatial ($\frac{L}{N_{space}}$) et temporel ($\frac{t_{max}}{N_{time}}$).

Afin de résoudre numériquement cette équation, nous adoptons un schéma aux différences finies implicite. En appliquant la formule de Taylor, il vient :

$$\frac{\partial u}{\partial t}(t, x) \approx \frac{u(x, t + \Delta t) - u(x, t - \Delta t)}{2 \Delta t}, \quad \forall (t, x) \in [0, t_{max}] \times [0, L],$$

Donc, en développant $\frac{\partial^2 u}{\partial x^2}$ de la même façon, nous obtenons l'équation de la chaleur sous forme implicite :

$$\begin{aligned} \forall j \in J, \forall n \in N, \quad \frac{u_i^{n+1} - u_i^n}{\Delta t} &= \frac{\lambda}{\rho c} \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} + \frac{F}{\rho c} \\ \forall j \in J, \forall n \in N, \quad u_i^{n+1} - u_i^n &= \frac{\Delta t}{\Delta x^2} \frac{\lambda}{\rho c} (u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}) + \frac{\Delta t}{\rho c} F \\ \forall j \in J, \forall n \in N, \quad (1 + 2 \frac{\Delta t}{\Delta x^2} \frac{\lambda}{\rho c}) u_i^{n+1} - \frac{\Delta t}{\Delta x^2} \frac{\lambda}{\rho c} (u_{i+1}^{n+1} + u_{i-1}^{n+1}) &= u_i^n + \frac{\Delta t}{\rho c} F \end{aligned}$$

L'expression résultante sous forme matricielle est alors :

$$A U_{n+1} = U_n + B$$

On pose la matrice des coefficients :

$$A = \begin{pmatrix} 1 + 2\alpha & -\alpha & 0 & \dots & 0 \\ -\alpha & 1 + 2\alpha & -\alpha & \ddots & \vdots \\ 0 & -\alpha & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha \\ 0 & \dots & 0 & -\alpha & 1 + 2\alpha \end{pmatrix}$$

Et la matrice des sources de chaleur :

$$B = \frac{\Delta t}{\rho c} \begin{pmatrix} F(x_1) \\ F(x_2) \\ \vdots \\ F(x_{N_{space}}) \end{pmatrix}$$

avec $\alpha = \frac{\Delta t}{\Delta x^2} \frac{\lambda}{\rho c}$

A est tridiagonale avec aucun élément diagonal nul, permettant ainsi une résolution itérative efficace.

Nous obtenons donc :

$$\boxed{U_{n+1} = A^{-1} U_n + A^{-1} B}$$

La résolution itérative de ce système constitue l'étape suivante de notre approche.

2.3.2 Implémentation

Concernant la barre, les frontières dictent la dissipation thermique à l'extrémité gauche avec une condition de Neumann en $x = 0$ et imposent la valeur de la température à u_0 à l'extrémité droite en $x = L$.

D'un côté, l'énergie thermique se libère vers l'environnement à l'extrémité gauche, un phénomène clairement visualisé dans l'animation. De l'autre côté, à l'extrémité droite de la barre, nous fixons la température à u_0 .

Ainsi, la dernière ligne de notre matrice reste constamment égale à $u_0 = 286,15K$.

Nous avons :

$$\begin{aligned}
& \frac{\partial u}{\partial t}(t, 0) = 0, \forall t \in [0, t_{max}] \\
\Rightarrow & \frac{\partial u}{\partial t}(t, 0) = \frac{1}{\rho c} F, \forall t \in [0, t_{max}] \\
\Rightarrow & u(t, 0) = \frac{1}{\rho c} F \times t + c^{te}, \forall t \in [0, t_{max}] \\
& \text{or } u(0, x) = u_0, \forall x \in [0, L] \\
& \text{donc } c^{te} = u_0 \\
\text{donc } & u(t, 0) = \frac{1}{\rho c} F \times t + u_0, \forall t \in [0, t_{max}] \\
& \text{de plus, } F(0) = 0 \\
\text{donc } & u(t, 0) = u_0, \forall t \in [0, t_{max}]
\end{aligned}$$

Le préremplissage de la matrice U avant la résolution intègre déjà les conditions comme contraintes à respecter pendant le calcul.

Pour l'inversion de la matrice A , nous utilisons l'algorithme de la matrice tridiagonale, également connu sous le nom de *tridiagonal matrix algorithm*. Cet algorithme implique le calcul successif des coefficients, suivis d'une substitution de ces derniers dans l'expression des éléments de l'inverse pour en déterminer les valeurs.

Ensuite, le processus de résolution commence par l'initialisation de U_0 , la première colonne de la matrice résultante représentant les températures des points de la barre à $t = 0$. Cela s'accomplit en utilisant la condition initiale $u(0, x) = u_0, \forall x \in [0, L]$.

Les termes subséquents de la suite récurrente U_n sont calculés pour obtenir l'évolution complète des températures de la barre au fil du temps.

2.4 Cas 2 : La Plaque

Les températures des divers points de la plaque sont organisées dans des vecteurs colonnes, où les éléments sont regroupés en paquets, chacun correspondant à un point spécifique. De manière plus détaillée, le vecteur résultant est formé par la concaténation verticale de N_{space} vecteurs colonnes, chacun d'une taille $N_{space}N$.

Cela implique que les

N_{space} points associés à $x = 0$ sont initialisés en premier, suivis par une séquence continue jusqu'à atteindre $x = L$.

2.4.1 Partie théorique : Équation de la chaleur

la structure de base d'une colonne U modélisant la plaque à tout instant t_n :

$$U^n = \begin{pmatrix} u_{1,1}^n \\ \vdots \\ u_{1,N_{space}}^n \\ \vdots \\ u_{N_{space},1}^n \\ \vdots \\ u_{N_{space},N_{space}}^n \end{pmatrix}, \forall n \in N$$

Comme pour la barre, la solution au problème prend la forme d'une matrice de dimensions $N_{space}^2 \times N_{time}$, où les lignes et les colonnes représentent respectivement les maillage spatial et temporel.

Voici l'équation de la chaleur pour une plaque :

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\lambda}{\rho c} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{F}{\rho c}, & \forall (t, x, y) \in [0, t_{max}] \times [0, L]^2 \\ u(0, x, y) = u_0, & \forall (x, y) \in [0, L]^2 \\ \frac{\partial u}{\partial x}(t, 0, y) = 0, & \forall (t, y) \in [0, t_{max}] \times [0, L] \\ \frac{\partial u}{\partial y}(t, x, 0) = 0, & \forall (t, x) \in [0, t_{max}] \times [0, L] \\ u(t, L, y) = u_0 \forall (t, y) \in [0, t_{max}], & \times [0, L] \\ u(t, x, L) = u_0, & \forall (t, x) \in [0, t_{max}] \times [0, L] \end{cases}$$

Puisque notre système évolue dans le temps, le schéma numérique demeure explicite dans la dimension temporelle, tandis que son caractère implicite se restreint aux variables spatiales.

En procédant à une discrétisation uniforme de l'espace dans les deux directions, nous obtenons $\Delta x^2 = \Delta y^2$. Reprenons les notations précédentes et introduisons $u(x_i, y_j, t_n) = u_{i,j}^n, \forall (i, j) \in J^2, \forall n \in N$.

Il vient alors :

$$\begin{aligned} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} &= \frac{\lambda}{\rho c} \left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta x^2} \right) + \frac{F}{\rho c}, \quad \forall (i, j) \in J^2, \forall n \in N \\ u_{i,j}^{n+1} - u_{i,j}^n &= \frac{\Delta t}{\Delta x^2} \frac{\lambda}{\rho c} (u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n + u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n) + \frac{\Delta t}{\rho c} F, \quad \forall (i, j) \in J^2, \forall n \in N \\ u_{i,j}^{n+1} &= \left(1 - 4 \frac{\Delta t}{\Delta x^2} \frac{\lambda}{\rho c} \right) u_{i,j}^n + \frac{\Delta t}{\Delta x^2} \frac{\lambda}{\rho c} (u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n) + \frac{\Delta t}{\rho c} F, \quad \forall (i, j) \in J^2, \forall n \in N \end{aligned}$$

De la même façon que pour la plaque, on obtient la forme matricielle :

$$\boxed{\mathbf{U}_{n+1} = C \mathbf{U}_n + D}$$

Avec $C = \begin{pmatrix} G & E & 0 & \dots & 0 \\ E & G & E & \ddots & \vdots \\ 0 & E & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & E \\ 0 & \dots & 0 & E & G \end{pmatrix}$, la matrice des coefficients la matrice tridiagonale par blocs.

Ainsi que $E = \alpha Id_{N_{space}}$ et $G = (1 - 4\alpha) Id_{N_{space}}$ des matrices diagonales

Mais aussi $D = \frac{\Delta t}{\rho c} \begin{pmatrix} F(x_1, y_1) \\ F(x_1, y_2) \\ \vdots \\ F(x_2, y_{N_{space}}) \\ F(x_2, y_1) \\ \vdots \\ F(x_{N_{space}}, y_1) \\ F(x_{N_{space}}, y_{N_{space}}) \end{pmatrix}$, la matrice modélisant les sources de chaleur.

2.4.2 Implémentation

En utilisant la condition initiale $\forall (x, y) \in [0, L]^2, u(0, x, y) = u_0$ pour renseigner le vecteur colonne initial U_0 , nous établissons une base pour notre résolution..

Concernant les conditions aux bords, à chaque itération ii, nous imposons la valeur u_0 à tous les éléments du dernier sous-vecteur de U_i (condition de Dirichlet en $y=L$). De manière similaire, cela s'applique au dernier élément de chaque sous-vecteur (condition de Dirichlet en $x=L$).

Les conditions de Neumann indiquent qu'aux bords, pour $x = 0$ et $y = 0$, la température reste constante et ne dépend pas des coordonnées spatiales. De manière analogue au raisonnement précédent, nous concluons que la première et la dernière composante de chaque sous-vecteur sont également égales à u_0 .

Afin de multiplier la matrice C avec le vecteur U_n de manière efficace, nous exploitons la nature creuse de la matrice et identifions des relations pour simplifier le calcul, évitant ainsi des opérations inutiles. Avec une matrice de taille $N_{space}^2 \times N_{space}^2$, cette approche accélère considérablement la résolution. Voici les relations universelles, applicables à toute matrice tridiagonale par blocs avec des blocs diagonaux :

$$\begin{cases} U_{i,1}^n = C_{i,i} \times U_{i,1}^n + C_{N_{space}+i,i} \times U_{N_{space}+i,1}^n, & \forall i = 1, \dots, N_{space} \\ U_{i,1}^n = C_{i-N_{space},i} \times U_{i-N_{space},1}^n + C_{i,i} \times U_{i,1}^n, & \forall i = N_{space} + 1, N_{space}^2 - N_{space} \\ \quad + C_{N_{space}+i,i} \times U_{N_{space}+i,1}^n \\ U_{i,1}^n = C_{i,i} \times U_{i,1}^n + C_{i-N_{space},i} \times U_{i-N_{space},1}^n, & \forall i = N_{space}^2 - N_{space} + 1, \dots, N_{space}^2 \end{cases}$$

De cette manière, nous parvenons à déterminer la température en chaque point de la plaque à tout moment t de notre analyse.

```

○ ranim@DESKTOP-FAQS647:/mnt/c/Users/Ranim Boussema/Desktop/projet_chaleur/Equation-de-la-chaleur$ ./projet
Which support do you want ? [1] Bar [2] Surface
1
Which material do you want ? [1] Copper [2] Iron [3] Glass [4] Polystyrene
1

```

3. Exécution du Programme

3.1 Configuration des Entrées

Afin de simplifier les interactions avec l'utilisateur, nous avons mis en place une interface en ligne de commande facilitant le choix du type de support et du matériau. Lors de l'exécution du projet avec la commande suivante :

```
g++ -g -Wall -Wextra -o projet *.cpp `pkg-config --cflags --libs sdl2`
```

L'utilisateur peut sélectionner entre une "Bar" en choisissant le numéro 1 ou une "Surface" en choisissant le numéro 2. De même, le choix du matériau se fait en indiquant 1 pour 'Copper', 2 pour 'Iron', 3 pour 'Glass', et 4 pour 'Polystyrene'.

3.2 Résultats de l'Exécution

3.2.1 Cas 1 : La barre

En ce qui concerne l'affichage de résultat de la barre nous avons fait le choix de créer deux animations.

La première représente une courbe évoluant dans le temps, avec l'ordonnée correspondant à la valeur de température atteinte, et l'abscisse représentant la position x le long de la barre.

La seconde animation illustre une barre, où les variations de température sont visualisées par un changement de couleur, passant du bleu au rouge pour indiquer la plage de températures, de la plus basse à la plus élevée.

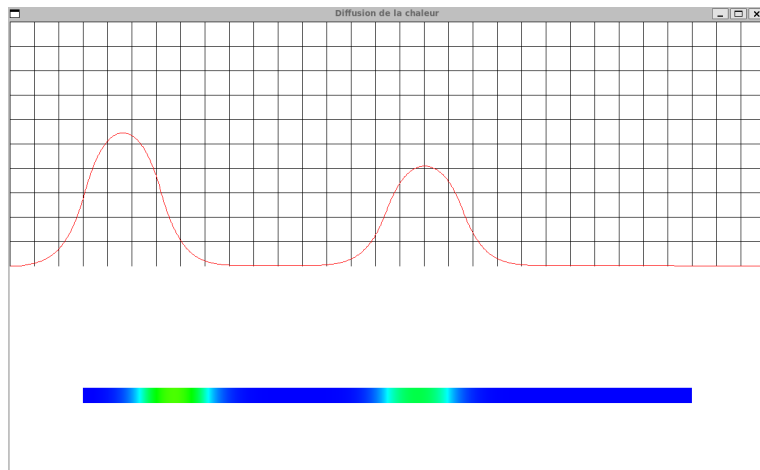
Étant donné que la température maximale diffère pour chaque matériau, nous avons ajusté l'échelle des couleurs pour chaque matériau afin de faciliter la lecture de la solution approchée. Sur l'axe des abscisses, une unité équivaut à un carré, correspondant à environ 3,3 cm.

Pour le cuivre, une unité (équivalente à un carré) représente 0,7 degré Kelvin, pour le fer, une unité équivaut à 0,95 degré Kelvin, pour le verre, une unité correspond à 1,56 degré Kelvin, et pour le polystyrène, une unité équivaut à 2,62 degré Kelvin.

Le programme s'exécute pendant 57 secondes. À l'ouverture de l'exécutable, un clic sur n'importe quelle touche lance l'animation pour une durée de 15 secondes, correspondant à t_{\max} . À la fin de l'animation, la dernière image reste affichée pendant 3 secondes avant de passer à un écran blanc. Vous avez alors la possibilité de fermer la fenêtre en cliquant sur la croix ou de relancer l'animation en appuyant sur n'importe quelle touche.

3.2.2 Cas 2 : La plaque

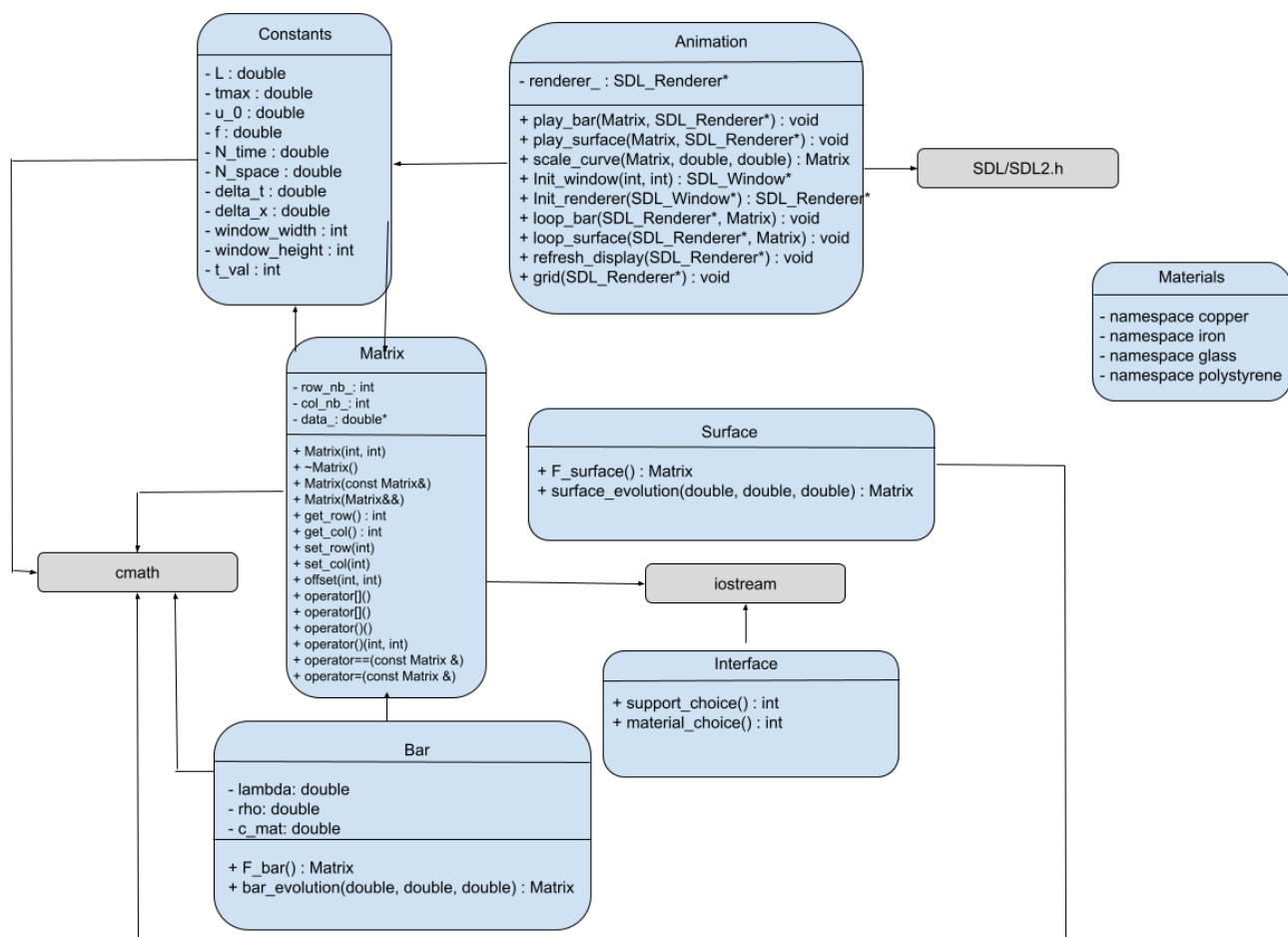
Pour animer la plaque, nous avons opté pour une approche qui la considère comme une série de barres. Ainsi, nous avons extrait les valeurs des premiers N_{space}



points au fil du temps pour les représenter graphiquement en utilisant la méthode précédemment employée. Cependant, nous avons rencontré des problèmes d'affichage qui surviennent fréquemment, et malheureusement, nous n'avons pas pu résoudre l'erreur qui apparaît de manière intermittente.

4. Conception

La conception UML complète de notre projet est modélisée par le diagramme suivant :



5. Défis Rencontrés

5.1 La bibliothèque SDL

Nous avons eu du mal à installer et configurer SDL sur WSL, et cela nous a pris un certain temps pour que tout fonctionne correctement.

5.2 Résultat de la Plaque

Pour la partie de la plaque, nous avons obtenu des résultats prometteurs au cours du développement. Cependant, par la suite, le programme a cessé de fonctionner, affichant une erreur de type `std::bad_array_new_length`. Malgré nos efforts pour trouver une solution, le manque de temps nous a empêchés de résoudre ce problème.

5.3 Complexité

L'inversion de la matrice en utilisant l'algorithme classique de Gauss s'est révélée être excessivement chronophage, nécessitant 57 secondes sur les 160 secondes totales d'exécution du programme. Devant cette inefficacité, nous avons fait le choix de passer à l'algorithme de la matrice tridiagonale, mieux adapté à la résolution de systèmes tridiagonaux. Cette transition a significativement réduit la complexité de l'opération, passant de $O(n^3)$ à $O(n)$.

6. Perspectives d'Amélioration

Bien que le projet ait atteint ses objectifs principaux, des axes d'amélioration subsistent. Des optimisations du code, des améliorations de l'interface utilisateur et l'exploration de méthodes numériques plus avancées pourraient être envisagées pour renforcer la robustesse et l'efficacité du programme.

Nous avons l'intention d'optimiser davantage l'interface utilisateur en se concentrant spécifiquement sur l'amélioration de l'affichage des courbes.

Cette amélioration consistera à rendre les graphiques plus visuellement attrayants et informatifs, tout en offrant des fonctionnalités avancées pour une meilleure compréhension des données par exemple en exploitant l'accès à d'autres bibliothèques et fonctionnalités, il devient possible d'étendre le programme pour inclure une représentation en trois dimensions (3D).

L'allocation de mémoire aurait pu être simplifiée en utilisant un type différent pour la structure `data_` au lieu de son implémentation actuelle. Cela aurait éliminé la nécessité de la fonction `offset` et simplifié la récupération d'éléments. Néanmoins, la mise en œuvre actuelle offre une plus grande flexibilité, permettant l'utilisation de la classe pour définir des vecteurs également, sans gestion de cas particuliers d'allocation de mémoire.

7. Conclusion

En conclusion, ce projet de simulation de l'évolution thermique a été mené à bien. La résolution de l'équation de la chaleur pour une barre et une plaque, avec différentes propriétés thermiques, a donné des résultats satisfaisants.

L'implémentation a permis de visualiser efficacement les comportements thermiques des matériaux.

Bien que le projet ait répondu à ses objectifs, des axes d'amélioration sont envisageables, notamment dans l'optimisation du code et l'amélioration de l'interface utilisateur. Globalement, cette expérience a offert un aperçu pratique de la modélisation numérique des phénomènes thermiques.