TP

Conduction thermique

Lycée Thiers - Physique-Chimie - MPI/MPI*- 2024-2025

Table des matières

1 Étude expérimentale du transfert thermique **1**
1.1 Dispositif expérimental. 1
1.2 Analyse d'une image thermique à une date donnée. 2
1.3 Analyse de l'évolution de la température en fonction du temps. 2
1.4 Quelques valeurs tabulées 3
2 Étude numérique de l'équation de la chaleur **3**
2.1 Cas non stationnaire à une dimension 3
2.2 Cas stationnaire en 2 dimensions - équation de Laplace 4

1 Étude expérimentale du transfert thermique

Objectifs du TP

Analyser et **s'approprier** un dispositif expérimental incluant une caméra thermique puis **réaliser** une expérience permettant de **valider** des valeurs expérimentales de conduction thermique.

Pour des raisons de matériel, ce TP ne peut être réalisé que par un groupe par séance.

L'objet de la séance est d'étudier l'évolution des températures des barres dans l'espace et le temps en fonction de la nature du métal grâce à des photographies thermiques prises à différents instants.

1.1 Dispositif expérimental

▷ À $t = 0$, on immerge les extrémités de deux barres cylindriques, qui étaient auparavant à température ambiante, dans un bain thermostaté à la température θ_0. Si cette opération n'est pas déjà réalisée à votre arrivée, la lancer le plus rapidement possible.

1.1.1 Maniement de la caméra

Les photos thermiques sont obtenues avec une caméra thermique qui mesure le rayonnement électromagnétique infra-rouge produit. La résolution de la caméra thermique Chauvin-Arnoux 1886 utilisée est 320x240 pixels. Sur les images thermiques, la distance entre la surface du bain et le haut du récipient (à partir duquel on peut faire les mesures) qui le contient correspond à 20 pixels environ. Les différents réglages correspondent à :

▷ A ajustement (choisit automatiquement l'intervalle de température, indispensable) ;
▷ C annulation, correction, permet d'accéder au menu ;
▷ S mode gelé ou non, enregistrement ;
▷ Retour : menu, validation ; pour sortir : S.

La température de l'objet pointé est écrit en haut à droite. Les images thermiques peuvent être enregistrées sur une carte SD puis analysées via le logiciel « Raycam admin » (le mot de passe est celui de l'ouverture de session).

▷ S'entraîner à réaliser des photographies à l'aide de la caméra.

1.1.2 Expérience

▷ Mesurer la température du bain thermostaté ainsi que la température ambiante au thermomètre.

▷ Prendre une photographie thermique. Il est nécessaire d'attendre la thermalisation pour le traitement proposé.

1.1.3 Images thermiques de travail

Les images thermiques obtenues sont de type « .SAT ». On peut soit travailler en mode « analyse », soit en mode « rapport »

En mode **analyse** de l'image (celui qui s'ouvre automatiquement quand on lance l'application), on a accès à divers outils. Par exemple :

- ▷ avoir la température en un point, les données moyennes sur une ligne, etc...
- ▷ la température maximale, minimale, ...
- ▷ pour choisir la palette, clic droit sur l'image : on choisit les températures associées aux couleurs y compris les valeurs extrêmes, on peut aussi ajuster l'image pour définir les températures extrémale (liées à l'expérience).

En mode **rapport**, des outils équivalents sont disponibles :

- ▷ sélectionner une zone (carrée ou disque ou...) et tracer l'histogramme des températures à l'intérieur de cette zone ;
- ▷ sélectionner une droite, puis avec clic droit obtenir le graphe de la distribution de température sur cette droite.

Les rapports peuvent être sauvegardés et imprimés.

1.2 Analyse d'une image thermique à une date donnée

- ▷ Ouvrir la photographie.
- ▷ Au vue des observations :

▷ Tracer la répartition de température sur chacune des barres : pour cela sélectionner une ligne (outil ligne), et par clic droit sur la ligne sélectionnée, tracer la répartition de température.

▷ Extraire les données de température dans un fichier de données.

1.3 Analyse de l'évolution de la température en fonction du temps

1.3.1 Éléments théoriques

Étudions le problème physique de la thermalisation d'une barre cylindrique de rayon R et de longueur L métallique plongée dans l'air ambiant à la température T_{ext} dont une extrémité est à la température T_0. Cette barre subit une conduction interne ainsi que des pertes de chaleur latérales conducto-convectif.

La modélisation du problème conduit à l'équation différentielle

$$\frac{\partial T}{\partial t} = \frac{2h}{R\rho c}\left(\frac{R\lambda}{2h}\frac{\partial T^2}{\partial x^2} - (T - T_{\text{ext}})\right)$$

avec h le coefficient de transfert conducto-convectif, c la capacité calorifique massique du métal, ρ sa masse volumique et λ sa conductivité thermique.

On pose $\delta = \sqrt{\dfrac{R\lambda}{2h}}$ et la durée caractéristique de thermalisation $\tau = \dfrac{R\rho c}{2h}$. La solution stationnaire de cette équation est

$$T(x) = T_{\text{ext}} + (T_0 - T_{\text{ext}})\ \mathrm{e}^{-x/\delta} \tag{1.1}$$

en ayant utilisé le fait que $\delta \ll L$.

▷ Donner l'ordre de grandeur de la durée caractéristique de la thermalisation de la barre. On rappelle que $D = \lambda/(\rho c)$.

1.3.2 Analyse des données

▷ Avec l'outil de votre choix, rechercher les paramètres d'une exponentielle décroissante ajustant au mieux les données. Le modèle (1.1) vous semble-t-il compatible avec les données expérimentales ?

> *Avec python, on peut utiliser par exemple la fonction* `curve_fit` *provenant de la bibliothèque* `scipy.optimize`.

▷ En prenant soin de convertir la taille des pixels en distance physique, en déduire les valeurs de δ du modèle (1.1) pour les deux barres.

▷ Comparer les valeurs expérimentales avec les données tabulées du paragraphe suivant. En particulier, comparer $\delta_{\text{cuivre}}/\delta_{\text{aluminium}}$ avec $\sqrt{\lambda_{\text{cuivre}}/\lambda_{\text{aluminium}}}$. Conclure.

▷ En cas de temps, enlever les barres de la source de température et observer le retour à l'équilibre thermique.

1.4 Quelques valeurs tabulées

Métal	Conductivité thermique en W/m/K	Coefficient de diffusion thermique en m^2/s
Aluminium	237	99×10^{-6}
Cuivre	390	117×10^{-6}
Acier	46	15×10^{-6}
Laiton	121	33×10^{-6}

Coefficient de transport conducto-convectif métal/air : de l'ordre de $10\,\mathrm{W \cdot m^{-2} \cdot K^{-1}}$.

2 Étude numérique de l'équation de la chaleur

Objectifs du TP

Analyser et **s'approprier** les schémas numériques de résolution de l'équation de la chaleur. **Réaliser** des codes correspondants et **valider** les résultats numériques correspondants.

2.1 Cas non stationnaire à une dimension

2.1.1 Situation étudiée

Étudions le problème physique de la thermalisation d'une barre cylindrique de rayon R et de longueur L métallique plongée dans l'air ambiant à la température T_{ext} dont une extrémité est à la température T_0. Cette barre subit une conduction interne ainsi que des pertes de chaleur latérales conducto-convectif.

La modélisation du problème conduit à l'équation différentielle

$$\frac{\partial T}{\partial t} = \frac{2h}{R\rho c}\left(\frac{R\lambda}{2h}\frac{\partial T^2}{\partial x^2} - (T - T_{\text{ext}})\right)$$

avec h le coefficient de transfert conducto-convectif, c la capacité calorifique massique du métal, ρ sa masse volumique et λ sa conductivité thermique.

On pose $\delta = \sqrt{\dfrac{R\lambda}{2h}}$ et $\tau = \dfrac{R\rho c}{2h}$. Ces données caractéristiques du problème permettent d'adimensionner l'équation précédente en posant $\widetilde{x} = x/\delta$ et $\widetilde{t} = t/\tau$. Il vient alors

$$\frac{\partial T}{\partial \widetilde{t}} = \frac{\partial T^2}{\partial \widetilde{x}^2} - (T - T_{\text{ext}})\,. \tag{2.1}$$

La solution stationnaire de cette équation est

$$T(x) = T_{\text{ext}} + (T_0 - T_{\text{ext}})\,\mathrm{e}^{-\widetilde{x}} \tag{2.2}$$

en ayant utilisé le fait que $\delta \ll L$.

2.1.2 Résolution numérique

Pour résoudre numériquement l'équation précédente, la température T est représentée par un tableau `T` à N éléments. On pose le schéma numérique suivant

$$T_{k+1}[i] = T_k[i] + h_\tau \left(\frac{T_k[i+1] + T_k[i-1] - 2T_k[i]}{h_x^2} - T_k[i] + T_{\text{ext}} \right) .$$

▷ Justifier ce schéma numérique et préciser le rôle des grandeurs h_τ et h_x. On prendra $h_\tau = 0.001$ et $h_x = 0.1$.

Pour limiter le temps de calcul, on se limite à une longueur de barre telle que $\widetilde{x} < 6$ contenant en tout 60 points. En effet, on sait que la solution stationnaire est une exponentielle de la forme $\exp(-\widetilde{x})$, il n'est pas nécessaire d'aller plus loin. De plus, pour assurer la convergence de l'algorithme, il est nécessaire de prendre $h_\tau / h_x^2 \lesssim 0.1$.

▷ Rédiger une fonction `evolution_1(T)` prenant en entrée le tableau `T` à un instant donné et modifiant sur place ce tableau pour donner le tableau à l'instant suivant. Pour rappel, on peut réaliser une copie d'un tableau à l'aide de la commande `T.copy()`.

▷ On prend initialement la température à $T_0 = 400\,\text{K}$ en $x = 0$ et $T_{\text{ext}} = 300\,\text{K}$ partout ailleurs. À l'aide de la fonction précédente, calculer numérique la température sur toute la barre après un temps suffisamment long.

▷ Vérifier que la solution stationnaire correspond bien à la solution théorique donnée par l'équation (2.2).

Une proposition de correction est disponible en cliquant sur le lien suivant [1].

Sur certaines distributions python, on peut utiliser le module suivant pour créer une animation, où `animate` est la fonction affichant la courbe :

```
from matplotlib import animation

def animate(i) :
    evolution_1(T)
    line1.set_data(x,T)

fig = plt.figure(figsize=(8,6))
line1, = plt.plot(x,T)
ani = animation.FuncAnimation(fig, animate, frames=100000, interval=10, repeat=False)
```

2.2 Cas stationnaire en 2 dimensions - équation de Laplace

Une fonction suit l'équation de Laplace si son laplacien est nul, $\Delta f = 0$. Cette équation est rencontrée dans de nombreux domaines de la physique : en électrostatique pour le potentiel dans une zone vide de charges, en conduction thermique pour la température en régime stationnaire, en diffusion moléculaire avec la densité de particules, en gravitation pour le potentiel gravitationnel dans une zone vide de masses, en hydrodynamique pour le potentiel des vitesses dans un écoulement bidimensionnel irrotationnel et incompressible, pour décrire les déformations d'une membrane, etc. On peut écrire des solutions analytiques au prix de plus ou moins d'efforts dans des situations simples, mais il est parfois plus visuel ou même nécessaire de recourir à une intégration numérique dans des cas plus généraux [2].

2.2.1 Algorithme de résolution

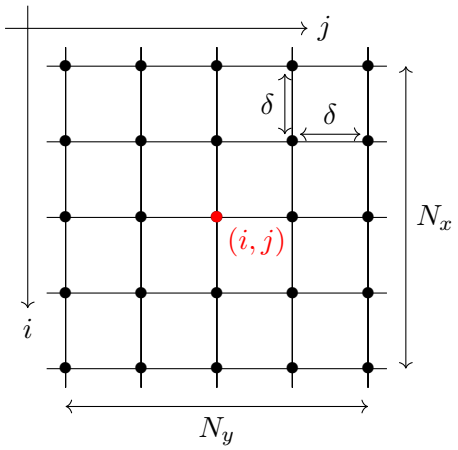
L'équation de Laplace (ou celle de Poisson $\Delta f + g = 0$ avec g fonction connue) est le prototype des équations aux dérivées partielles dites elliptiques. Ces équations servent à décrire un problème aux limites (*boundary value problem* en anglais) où les valeurs de la fonction dans le domaine doivent être déduites de celles sur les bords du domaine. Par opposition aux problèmes de Cauchy (équation de propagation, équation de la chaleur...), il n'y a pas d'évolution temporelle.

1. https://bit.ly/thermalisation_barre
2. Ce TP est fortement inspiré du travail de Mickaël Melzani http://www.mmelzani.fr/documents/divers/mmelzani_resolution_Laplace.pdf

Nous nous intéressons ici à une méthode numérique par différences finies, c'est-à-dire que la fonction f inconnue est discrétisée aux points (i, j) d'un maillage (tel que celui de la figure ci-dessous). Il faut alors également discrétiser l'équation à résoudre, ce qui mène à un système d'équations portant sur les $f(x_i, y_j)$ qu'il faut résoudre.

Définition de la grille et du domaine :

On se restreint au cas à deux dimensions. Le cas à trois dimensions ne pose pas de difficultés supplémentaires, mais requiert des temps de calculs plus longs et des méthodes de visualisation des résultats plus complexes.



L'espace est donc un rectangle de dimensions $L_x \times L_y$, représenté par une grille de taille $N_x \times N_y$.

La distance entre deux points de la grille est le pas spatial δ. On a ainsi $L_x = (N_x - 1)\delta$ et $L_y = (N_y - 1)\delta$.

On note (x_i, y_j) les coordonnées du point (i, j) de la grille, i allant de 0 à $N_x - 1$ et j de 0 à $N_y - 1$. Un carré de côté δ est délimité par quatre points de la grille est une cellule.

La grandeur physique f (qui satisfait $\Delta f = 0$) est représentée par un tableau `f` avec `f[i,j]` donnant la valeur de $f(x_i, x_j)$ au point (i, j) de la grille.

Une propriété importante de l'équation de Laplace (et de celle de Poisson également) est qu'elle admet une unique solution dans un domaine $\mathcal{D}$ de l'espace si on connaît en tout point du bord de $\mathcal{D}$: soit la valeur de f (condition dite de Dirichlet), soit la valeur de la dérivée normale de f (condition dite de Neumann). On se restreindra pour notre part à la condition de Dirichlet.

Les bords du domaine peuvent simplement être le cadre de la grille, mais ils peuvent aussi inclure d'autres points de la grille pouvant représenter les armatures d'un condensateur, un conducteur métallique à un potentiel V_0 fixe, une surface à une température imposée, etc.

Nous introduisons donc le tableau `B`, de taille $N_x \times N_y$, tel que :

- si `B[i,j] = 1`, alors le point (i, j) de la grille appartient au bord du domaine, et la valeur `f[i,j]` est imposée soit par une condition de Dirichlet ;
- si `B[i,j] = 0`, alors le point (i, j) n'appartient pas au bord du domaine. Il faudra alors utiliser les équations du paragraphe suivant pour déterminer sa valeur.

2.2.2 Discrétisation de l'équation

La fonction f étant connue uniquement aux points de la grille, il est nécessaire de discrétiser l'opérateur laplacien, ici en coordonnées cartésiennes et à deux dimensions.

Un développement limité à l'ordre 3 de f autour du point (i, j) donne

$$f(x_i \pm \delta, y_i) = f(x_i, y_j) \pm \delta \left(\frac{\partial f}{\partial x}\right)_y (x_i, y_i) + \frac{\delta^2}{2} \left(\frac{\partial^2 f}{\partial x^2}\right)_y (x_i, y_i) \pm \frac{\delta^3}{3!} \left(\frac{\partial^3 f}{\partial x^3}\right)_y (x_i, y_i) + \mathcal{O}(\delta^4) \, .$$

En ajoutant $f(x_i - \delta, y_i)$ et $f(x_i + \delta, y_i)$, on en déduit une écriture discrète de la dérivée seconde

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_y (x_i, y_i) = \frac{f(x_i + \delta, y_j) + f(x_i - \delta, y_j) - 2f(x_i, y_j)}{\delta^2} + \mathcal{O}(\delta^2) \, .$$

On procède de même dans la direction y, et on obtient ainsi, en simplifiant de façon transparente les notations,

$$\Delta f[i, j] = \frac{f[i + 1, j] + f[i - 1, j] + f[i, j + 1] + f[i, j - 1] - 4f[i, j]}{\delta^2} + \mathcal{O}(\delta^2) \, .$$

L'équation de Laplace impose $\Delta f = 0$. Ceci peut donc s'écrire, pour tout i, j dans le domaine et à l'ordre deux en δ,

$$\boxed{f[i, j] = \frac{f[i + 1, j] + f[i - 1, j] + f[i, j + 1] + f[i, j - 1]}{4}} \, . \tag{2.3}$$

Cette expression provient du fait que pour une fonction harmonique (qui satisfait $\Delta f = 0$), $f(M)$ est égal à la valeur moyenne de f sur une sphère entourant M.

2.2.3 Méthode itérative de Jacobi

La méthode de Jacobi consiste à appliquer l'équation (2.3) pour construire par itérations les valeurs de f sur la grille, et ceci jusqu'à ce que les valeurs de f n'évoluent quasiment plus.

L'algorithme est le suivant :

- **Initialisation** :
  - On crée la matrice `B` et on l'initialise avec des 0 ou des 1 pour décrire les bords du domaine souhaité.
  - On crée la matrice `f` et on l'initialise avec les valeurs voulues sur les bords du domaine et avec la valeur initiale ailleurs.
- **Itérations** : Une itération consiste à effectuer, pour tout point (i, j) qui n'appartient pas à un bord, le calcul d'une nouvelle valeur de f selon l'équation (2.3). Ainsi f à l'itération k est obtenue en fonction de f à l'itération $k-1$ via la relation

$$\boxed{f_k[i,j] = \frac{f_{k-1}[i+1,j] + f_{k-1}[i-1,j] + f_{k-1}[i,j+1] + f_{k-1}[i,j-1]}{4}}.$$

  En python ceci nécessite de faire une copie de la matrice f, comme suit

```
fcopy = f.copy()
for i in range(Nx):
  for j in range(Ny):
    if B[i,j] == 0:
      f[i,j] = (fcopy[i+1,j]+fcopy[i,j+1]+fcopy[i-1,j]+fcopy[i,j-1])/4
```

- **Critère de terminaison** : On stoppe la simulation lorsque les itérations ne font quasiment plus évoluer les valeurs de f. On définit pour cela l'écart

$$e = \sqrt{\frac{1}{N_x N_y} \sum_{i,j} [f_k(x_i, y_j) - f_{k-1}(x_i, y_j)]^2} \tag{2.4}$$

  qui représente l'écart entre la nouvelle valeur de f et l'ancienne en moyenne sur toute la grille. On arrête le programme lorsque e devient inférieur à une valeur ε fixée à l'avance, que l'on appellera critère de convergence.

Pour un critère de convergence ε fixé, la méthode de Jacobi converge en $\mathcal{O}(N^2)$ itérations (si $N_x = N_y = N$). Comme chaque itération requiert le parcours de la grille et donc $\mathcal{O}(N^2)$ opérations, la complexité globale est en $\mathcal{O}(N^4)$.

En réalité, on peut montrer que la méthode itérative de Jacobi s'apparente à une convergence vers un état stationnaire via l'intégration de l'équation de diffusion avec un choix de pas de temps maximal. En particulier, la solution converge en partant de l'état initialisé au départ vers la solution du problème par diffusion à partir des bords où f est imposée. Ainsi, tracer successivement les tableaux `f` permet d'observer la dynamique de la thermalisation.

2.2.4 Amélioration par les méthodes de Gauss-Seidel et de sur-relaxation

En pratique, la complexité en $\mathcal{O}(N^4)$ (si $N_x = N_y = N$) ne permet guère de dépasser des domaines de taille N de l'ordre de 100 tout en gardant des temps de simulations raisonnables pour une séance de TP. Il se trouve qu'il existe des améliorations qui permettent de faire tomber la complexité en $\mathcal{O}(N^3)$, tout en conservant un algorithme simple.

Les améliorations présentées permettent de gagner du temps d'exécution, toutefois les tableaux intermédiaires obtenus ne sont plus ceux correspondant à la dynamique du problème. Il n'est donc plus possible d'observer celle-ci.

Ces améliorations sont de deux natures :

▷ D'abord, on utilise la méthode de Gauss-Seidel, qui remplace l'équation (2.2.3) de Jacobi par

$$\boxed{f_k[i,j] = \frac{f_{k-1}[i+1,j] + f_k[i-1,j] + f_{k-1}[i,j+1] + f_k[i,j-1]}{4}} .$$

où les valeurs de f en $(i-1,j)$ et $(i,j-1)$ sont celles de l'itération courante k. Ceci nécessite de parcourir la grille à i et j croissants, afin que ces valeurs aient été déjà actualisées lors de cette itération. Comme les valeurs courantes ont déjà été mises à jour, cela permet d'améliorer la vitesse de convergence, mais avec un gain d'un facteur 2 seulement. En pratique dans l'algorithme il ne faudra plus utiliser une matrice copie de celle de f, mais directement itérer avec f avec

```
f[i,j] = (f[i+1,j]+f[i,j+1]+f[i-1,j]+f[i,j-1])/4.
```

▷ Ensuite, on utilise une méthode dite de sur-relaxation successive, dans laquelle la nouvelle valeur est obtenue comme une moyenne pondérée entre l'estimation donnée par Gauss-Seidel et l'ancienne valeur de f au point considéré :

$$\boxed{f_k[i,j] = (1-\omega) f_{k-1}[i,j] + \omega \frac{f_{k-1}[i+1,j] + f_k[i-1,j] + f_{k-1}[i,j+1] + f_k[i,j-1]}{4}} . \tag{2.5}$$

Le facteur de pondération ω est appelé le facteur de relaxation. La méthode est dite de sur-relaxation si $\omega > 1$ (et de sous-relaxation si $\omega < 1$, mais ce ne sera jamais notre cas car ceci ralentit la convergence).

On a les propriétés suivantes admises :

▷ L'algorithme converge si et seulement si $0 < \omega < 2$.
▷ Il existe une valeur optimale de ω qui permet d'atteindre le critère de convergence en $\mathcal{O}(N)$ itérations pour une grille de taille $N_x = N_y = N$ soit

$$\omega_{\text{opt}} \approx \frac{2}{1 + \pi/N} .$$

> ***Remarque :*** *Si la grille est de taille $N_x \times N_y$, remplacer N par $N_x N_y \sqrt{2/(N_x^2 + N_y^2)}$. Ce résultat est valide pour l'équation de Laplace ou de Poisson, avec conditions aux bords de Dirichlet ou de Neumann, et est obtenue par un développement asymptotique en $N_x, N_y \gg 1$.*

Dans ce cas, la complexité totale de l'algorithme est donc en $\mathcal{O}(N^3)$ au lieu de $\mathcal{O}(N^4)$, ce qui est un gain significatif et même nécessaire pour réaliser des simulations intéressantes.
▷ Pour $\omega = 1$ on retrouve la méthode de Gauss-Seidel.
▷ Cette méthode nécessite un ordre particulier de parcours de i et j pour fonctionner. Le sens de parcours de la méthode de Gauss-Seidel fonctionne, mais d'autres possibilités existent.

Enfin, notons que d'autres méthodes d'optimisation sont possibles (accélération de Chebyshev qui change la valeur de ω au cours des itérations, schéma multigrilles qui utilise des grilles de plus en plus fines...), mais sont significativement plus complexes à mettre en place.

2.2.5 Conditions aux bords de Neumann

Dans certains cas, il est nécessaire de prendre des conditions aux bords de conservation du flux (comme dans les cas de transferts thermiques conducto-convectif). Dans ce cas, il est nécessaire de connaître les valeurs du flux φ. La condition aux bords est alors simplement de la forme $f[i, N_y - 1] = f[i, N_y - 2] + \delta\varphi[i]$.

> ***Remarque :*** *Dans le cas conducto-convectif, $\varphi[i]$ dépend des températures $T[i, N_y - 1]$ et $T[i, N_y - 2]$.*

Dans le cas particulier (et important) d'une paroi calorifugée le flux doit être nul, donc $\varphi = 0$ et on a simplement l'égalité des températures.

2.2.6 Application

▷ Coder les méthodes précédentes (optimisées selon l'équation (2.5) et non optimisées selon l'équation (2.3)) dans des fonctions python prenant en entrée le tableau des température `T` et le tableau `B` des conditions aux limites (préalablement créé). Ces fonctions modifient sur place le tableau `T` selon le schéma numérique.

▷ Coder une fonction renvoyant la valeur de l'écart entre deux tableaux tel que défini équation (2.4).

▷ En utilisant les fonctions précédentes, coder une fonction prenant en entrée une fonction d'évolution, un tableau initial et une précision. Cette fonction modifie sur place le tableaux des températures jusqu'à la convergence de l'algorithme. La fonction affichera la carte de température sous forme de ligne de niveau. On donne les commandes suivantes

```
import matplotlib.pyplot as plt
x = [i for i in range(Nx)]
y = [i for i in range(Ny)]

plt.contourf(y,x,T,20)
plt.colorbar()
plt.show()
```

permettant l'affichage du tableau `T` de taille $N_x \times N_y$ sous forme de carte de niveau.

▷ Tester ces fonctions pour des conditions aux bords diverses de votre choix : température fixe sur tous les bords différentes de celle initiale, température différentes sur les bords, gradient de température, point central chaud, etc. Comparer en temps les méthodes d'évolution.

Une proposition de correction est disponible en cliquant sur le lien suivant [3].

3. https://bit.ly/Laplace_thermique