

TD IMSP 1 : Résolution d'équations différentielles de conduction thermique par discrétisation spatiale et temporelle

Nous ne chercherons pas à adimensionner les équations ni à « simplifier » les valeurs numériques. Les ordres de grandeur doivent être immédiatement interprétables par le sens commun. Les longueurs seront donc exprimées en m et les durées en secondes. Pour les durées assez élevées des régimes transitoires, on présentera tout de même les dates des profils successifs en minutes en légende de graphe.

Nous nous limiterons au fer comme matériau conducteur thermique. Ses qualités thermiques n'intervenant dans l'équation de la diffusion thermique que par la diffusivité, vous confirmerez puis utiliserez la valeur : $D=2,345 \cdot 10^{-5} \text{ m}^2 \cdot \text{s}^{-1}$

Les bibliothèques à importer seront **numpy** (renommée np), **matplotlib.pyplot** (renommée plt), **time** pour l'estimation de durées de calcul mais également **scipy.special** (renommée sp) lorsque nous devrons appliquer la « fonction d'erreur » à des tableaux.

I. La barre de Fer entre deux thermostats

I.1. Modélisation physique

Une barre de fer à température initiale uniforme $T_0=20^\circ\text{C}$ est calorifugée latéralement et en contact thermique parfait à ses deux extrémités avec deux thermostats : l'un chaud à gauche constitué d'une eau bouillante à température $T_c=100^\circ\text{C}$, l'autre froid à droite maintenu à $T_f=0^\circ\text{C}$ grâce à un mélange eau-glace brassé régulièrement.

L'équation de diffusion de la chaleur au sein de ce solide sera discrétisée :

- spatialement : on choisira une barre de $L=1\text{m}$ de longueur que l'on découpera en $N_x=100$ tranches de $dx=1\text{cm}$ dans un premier temps
- temporellement : on choisira de diviser une durée totale de $\tau=20000$ secondes en $N_t=10000$ pas temporels de $dt=2\text{s}$

I.2. Programme « Barre de fer »

Un profil initial de N_x+2 valeurs de températures sera concaténé.

Les augmentations des températures de chaque tranche i lors d'un pas temporel seront notés $dT[i]$

Les valeurs des températures de thermostats ne doivent pas varier dans le temps.

On demandera un affichage de profil de températures toutes les 20 minutes (de simulation)

On pourra utiliser les « cmap » en associant un dégradé de couleurs à l'écoulement du temps :

```
plt.plot(x,T,'+',label=plotlabel,color = plt.get_cmap('winter')(float(j/Nt)))
```

Le programme devra également restituer le temps de calcul de ces N_t profils

On pourra imprimer ces courbes pour illustrer les régimes transitoires du cours de thermique.

I.3. Et si on faisait directement les opérations sur les tableaux ?

I.3.1. Les possibilités de calcul « vectoriel » de numpy

On peut facilement tirer profit des instructions vectorielles de numpy pour éviter des boucles de programme. Ces « boucles » sont alors gérées par du code compilé et non du code interprété. La bibliothèque numpy est en effet déjà « compilée ».

*La différence centrale entre compilé et interprété est comme suit : là où **le compilateur traduit une bonne fois pour toute un code source en un fichier indépendant exécutable** (donc utilisant du code machine ou du code d'assemblage), **l'interprète est nécessaire à chaque lancement du programme interprété**, pour traduire au fur et à mesure le code source en code machine. Cette traduction à la volée est la cause première de la lenteur des langages dits interprétés.*

Rappelons donc quelques opérations d'indexation sur les tableaux :

```
>>> t = np.array([1,2,3,4,5,6])
>>> t[1:4] # de l'indice 1 à l'indice 4 exclu !!!ATTENTION!!!
array([2, 3, 4])
>>> t[:4] # du début à l'indice 4 exclu
array([1, 2, 3, 4])
>>> t[4:] # de l'indice 4 inclus à la fin
array([5, 6])
>>> t[:-1] # excluant le dernier élément
array([1, 2, 3, 4, 5])
>>> t[1:-1] # excluant le premier et le dernier
array([2, 3, 4, 5])
```

I.3.2. Application à l'obtention du tableau des augmentations de températures

Que représentera $dT[1:-1]$ dans notre programme ?

Combien d'éléments contient ce tableau ? et $T[1:-1]$?

Comment écrire la liste des N_x termes éliminant la température du premier thermostat et de la première tranche ?

Et la liste éliminant la température du dernier thermostat et de la tranche le précédant ?

Après avoir vérifié que ces trois tableaux (monodimensionnels) avaient la même dimension, imaginez ces trois tableaux les uns sous les autres (les indices ont été décalés par les appels des « coupes » précédentes). Quelle simple opération de combinaison linéaire des tableaux précédents calculera immédiatement le tableau $dT[1:-1]$ des variations de température de toutes les tranches à chaque itération temporelle ?

Copiez-collez votre programme précédent dans un nouveau fichier spyder et remplacez la double boucle de calcul des $T[i]$ par l'opération précédente. Nouveau temps d'exécution ?

II. Deux barres de Fer au contact

Cette fois on ne considère plus la présence de thermostats mais de deux barres calorifugées latéralement, « semi-infinies », de même matériau, présentant à l'instant du contact des températures uniformes mais différentes. On part donc d'un profil initial de longueur infinie avec une température T_{10} pour les $x < 0$ et T_{20} pour les $x > 0$. On admet que l'interface ($x=0$) prend immédiatement sa valeur de température de contact $(T_{10}+T_{20})/2$ (par raison de symétrie). (On se référera à l'exercice 8 donné en annexe 1 pour le cas de deux barres de matériaux différents et le rôle de l'effusivité du matériau)

II.1. La fonction d'erreur : erf(x)

II.1.1. Définition et compatibilité avec notre modèle

Cette fonction a été présentée dans l'exercice 8 de l'annexe 1
$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

On a vérifié qu'il s'agissait d'une solution de l'équation unidimensionnelle de diffusion de la chaleur en remplaçant la variable par $u = \frac{x}{\sqrt{4Dt}}$. On peut vérifier également sa compatibilité avec l'échelon initial de température (discontinuité en x tend vers 0^+ et x tend vers 0^- à $t=0$) et la conservation de l'énergie thermique des barres par intégration de $-\infty$ à $+\infty$.

II.1.2. Evolution des profils de température lors du régime transitoire

Utiliser cette fonction [sous le module **scipy.special** pour l'appliquer à un tableau de valeurs de x] pour sortir les profils successifs de température.

On découpera les deux morceaux de barre de $L=1\text{m}$ chacune en $N_x=100$ tranches chacune soit un profil de 201 valeurs de températures avec la tranche en $x=0$ prenant dès le départ la température connue T_j moyenne des deux côtés. ($T_{10}=40^\circ\text{C}$; $T_{20}=0^\circ\text{C}$ et $T_j=20^\circ\text{C}$)

Pour la durée totale on prendra 2h soit $\tau=7200\text{ s}$ et le nombre d'itérations temporelles sera de $N_t=30000$ pour un incrément temporel de $dt=\tau/N_t = 0,24\text{ s}$

Superposer les graphes de profils de température toutes les 600 itérations (2 minutes 24s).

II.1.3. Comparaison avec les valeurs obtenues par la méthode d'Euler précédente

Nous allons copier-coller le programme de détermination pas à pas de profils de température de la partie I et comparer à la solution donnée par la fonction d'erreur.

Le programme précédent (partie I) va bien prendre le nouveau profil des températures initiales mais considérer que les températures des extrémités sont les valeurs constantes de thermostats. Et donc, contrairement à la réalité d'une évolution des températures de barres à -1m et $+1\text{m}$, il va les figer à T_{10} et T_{20} !!

Superposer sur une même figure les deux réseaux de courbes (avec des choix de gradient de couleur différents !!) :

```
plt.plot(x, Terf, color = plt.get_cmap('winter')(float(j/Nt)))  
plt.plot(x, T, label=plotlabel, color = plt.get_cmap('cool')(float(j/Nt)))
```

Dans quelle zone les deux méthodes donnent-elles des courbes quasi-confondues (Annexe 2) ?

II.1.4. Rôle du pas spatial dx dans une méthode d'Euler appliquée à une équation aux dérivées partielles du second ordre en x

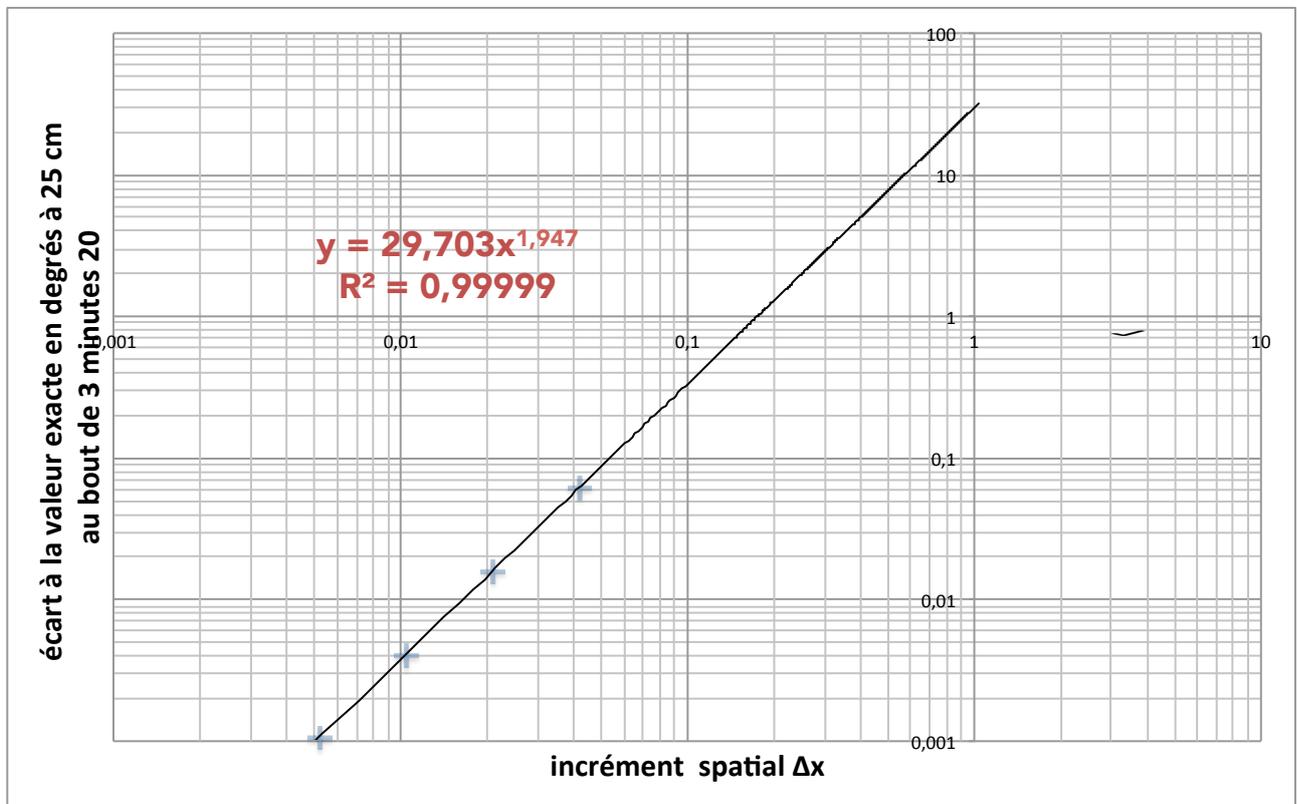
Nous allons observer l'effet du pas spatial sur l'erreur en comparant la valeur de température obtenue par la méthode d'Euler (à $x=25$ cm et à $t=200$ s (3minutes20)) à la valeur obtenue (même position même instant) par la fonction d'erreur (supposée « exacte » même s'il s'agit elle-même d'un calcul numérique d'intégrale)

Vous utilisez toujours le programme précédent dans lequel vous modifiez le pas spatial de la méthode d'Euler.

On prendra $N_x=24$ puis 48, 96 et 192.

Introduisez les dx et les écarts de températures obtenues (entre les deux estimations erf et euler) dans un tableau excel et confirmez graphiquement une méthode d'ordre 2 en dx : soit un écart fonction du carré du dx choisi pour la discrétisation.

(On pourrait également confirmer un ordre 1 en pas temporel)



III. Matériau non linéaire : conductivité dépendante de la température.

Dans cette troisième partie, on choisit d'étudier un régime unidirectionnel **permanent**. Ainsi nous abandonnons une équation aux dérivées partielles pour une simple équation différentielle. Le but (avoué ici) est de travailler avec une **équation différentielle non-linéaire du premier ordre en x** dont on peut déterminer la solution exacte par intégration. On comparera alors les résultats de 2 méthodes itératives à pas constant : la **méthode d'Euler explicite** précédemment utilisée et la **méthode de Runge Kutta** (ordre 4 !).

III.1. Modélisation physique

MUR EN REGIME PERMANENT AVEC CONDUCTIVITE VARIABLE

Pour de nombreux matériaux soumis à des écarts de température importants, il faut prendre en compte la variation de la conductivité avec la température. Cette variation est donnée généralement par une loi linéaire

$$\lambda = \lambda_0 (1 + \alpha (T - T_0))$$

On considère une plaque d'épaisseur e soumise sur ses deux faces à un contact parfait avec deux milieux de températures $T(0) = T_0$ $T(e) = T_e$

En supposant que la conductivité du matériau constitutif de la plaque varie linéairement avec la température, déterminer la répartition interne des températures, ainsi que la valeur du flux de chaleur traversant cette paroi.

$$\text{A.N.} \quad \begin{array}{lll} e = 5 \text{ cm} & \lambda_0 = 1 \text{ W/m.K} & \alpha = 2.10^{-3} \text{ }^\circ\text{C}^{-1} \\ T_0 = 50^\circ\text{C} & T_e = 550^\circ\text{C}. & \end{array}$$

Cette situation (et ces valeurs numériques) conduisent à une valeur de flux de chaleur (surfactive !) indépendante de x (position dans le matériau) et négative (flux positif dans le sens des hautes vers les basses températures donc dans le sens $-x$) : $\varphi = -15 \text{ kW } m^{-2} = -15000 \text{ W } m^{-2}$

et à une expression numérique exacte de : $\theta(x) \equiv T(x) - T_0 = 500.(\sqrt{1 + 60x} - 1)$ avec x en m

En utilisant l'hypothèse de régime permanent, écrire l'équation différentielle en $\theta(x) \equiv T(x) - T_0$ avec les paramètres caractéristiques constants φ, λ_0 et α .

III.2. Comparaison Euler « versus » Runge-Kutta

Vous allez évaluer les ordres spatiaux des deux méthodes :

- Euler explicite : Ordre attendu en fonction du pas spatial ?
- Runge-Kutta : ?

III.2.1. Méthodes : d'Euler à Runge-Kutta 4

La **méthode d'Euler** (à l'ordre 1) peut se présenter de la manière suivante :

On cherche à approximer la solution d'une équation différentielle de la forme :

$$\left\{ \begin{array}{l} \frac{dy}{dx}(x) = f(x, y(x)) \\ y(x_0) = y_0 \end{array} \right. \text{ en écrivant : } y(x_{i+1}) \equiv y(x_i + h) \approx y(x_i) + h \cdot \frac{dy}{dx}(x_i) = y_i + h \cdot f(x_i, y_i)$$

h représente le pas de chaque itération : $h = dx = e/N_x$

La **méthode du point milieu** est une des méthodes de Runge-Kutta d'ordre 2 qui utilise le nouveau schéma d'approximation :

$$y(x_{i+1}) = y_i + h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} \cdot k_1\right) \text{ avec } k_1 = f(x_i, y_i)$$

Enfin la **méthode de Runge Kutta d'ordre 4** utilise le schéma d'approximation suivant :

$$y(x_{i+1}) = y_i + h \cdot \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \text{ avec } \begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} \cdot k_1\right) \\ k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} \cdot k_2\right) \\ k_4 = f(x_i + h, y_i + h \cdot k_3) \end{cases}$$

Vous allez donc proposer un programme Python qui sortira le graphe profil des températures : d'abord avec Euler pour un pas donné $dx=e/N_x$, puis par implémentation de Runge Kutta pour le même pas.

On testera aisément l'écart de chacune de ces méthodes à la valeur exacte en $x=e$ puisque cette valeur vaut exactement :

$$\theta(e) \equiv T(e) - T_0 = 500 \cdot (\sqrt{1 + 60 \cdot (0,05)} - 1) = 500^\circ\text{C}$$

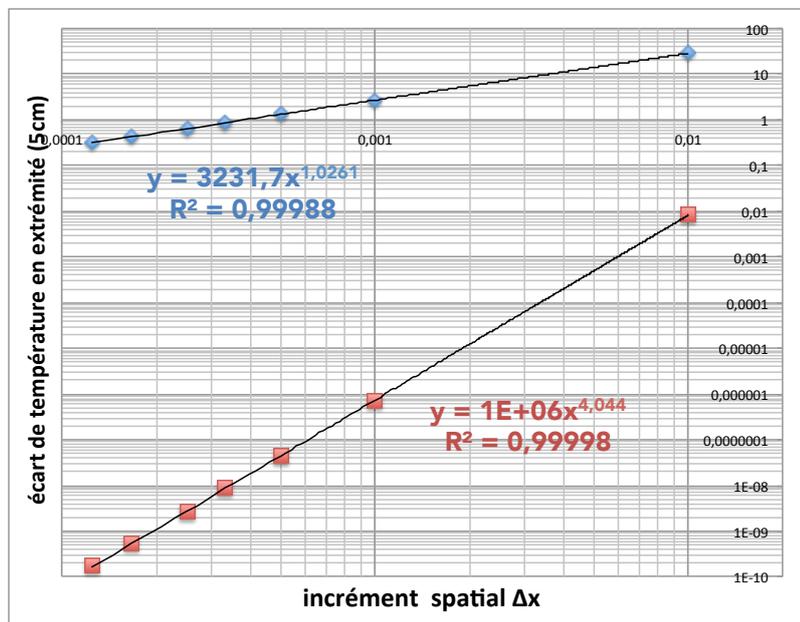
pour un $T(e) = T_e = 550^\circ\text{C}$

Attention à bien préparer une liste de « tranches de matériau » de N_x+1 valeurs : première valeur ($x=0, \theta(0)=0$) et dernière ($x=(N_x+1) \cdot dx=e+dx$ pour $\theta=?$)

III.2.2. Confirmation de l'ordre de chacune des méthodes

On proposera des discrétisations très grossières en commençant à $N_x=5$ (5 tranches d'1 cm chacune). Admirez la faiblesse de l'écart à la valeur exacte avec Runge-Kutta contrairement à Euler !

On rassemblera les écarts en $x=e$ pour les découpages suivants : $N_x=5; 50; 100; 150; 200; 300; 400$ dans un tableur Excel et on confirmera les ordres en dx par une régression en puissance sur un tracé log-log.



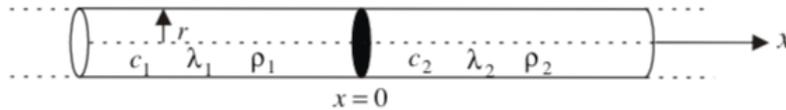
Annexe 1 : PARTIE II : Deux barres au contact : Solution approchée puis exacte

8. Température de contact

On considère deux barres à base circulaire de petit rayon r , accolées, calorifugées sur leur surface latérale, constituées de matériaux différents :

- pour $x < 0$, on a une capacité thermique massique c_1 , une conductivité thermique λ_1 , une masse volumique ρ_1 ; la barre étant initialement à la température T_1^0 ;
- pour $x > 0$, on a une capacité thermique massique c_2 , une conductivité thermique λ_2 , une masse volumique ρ_2 ; la barre étant initialement à la température T_2^0 .

On cherche à déterminer à une date t quelconque la température de contact $T(0,t)$. Les barres sont supposées infiniment longues.



- a) On choisit un modèle simple (il n'est pas solution de l'équation de la diffusion thermique) pour lequel à la date t on a $T = T_1^0$ pour $x < -\ell_1(t)$; $T = T_2^0$ pour $x > \ell_2(t)$, T affine pour $-\ell_1(t) < x < 0$ et pour $0 < x < \ell_2(t)$.

Déterminer $T(0,t)$ en fonction de T_1^0 , T_2^0 , $\Gamma_1 = \sqrt{\rho_1 c_1 \lambda_1}$ et $\Gamma_2 = \sqrt{\rho_2 c_2 \lambda_2}$, en écrivant :

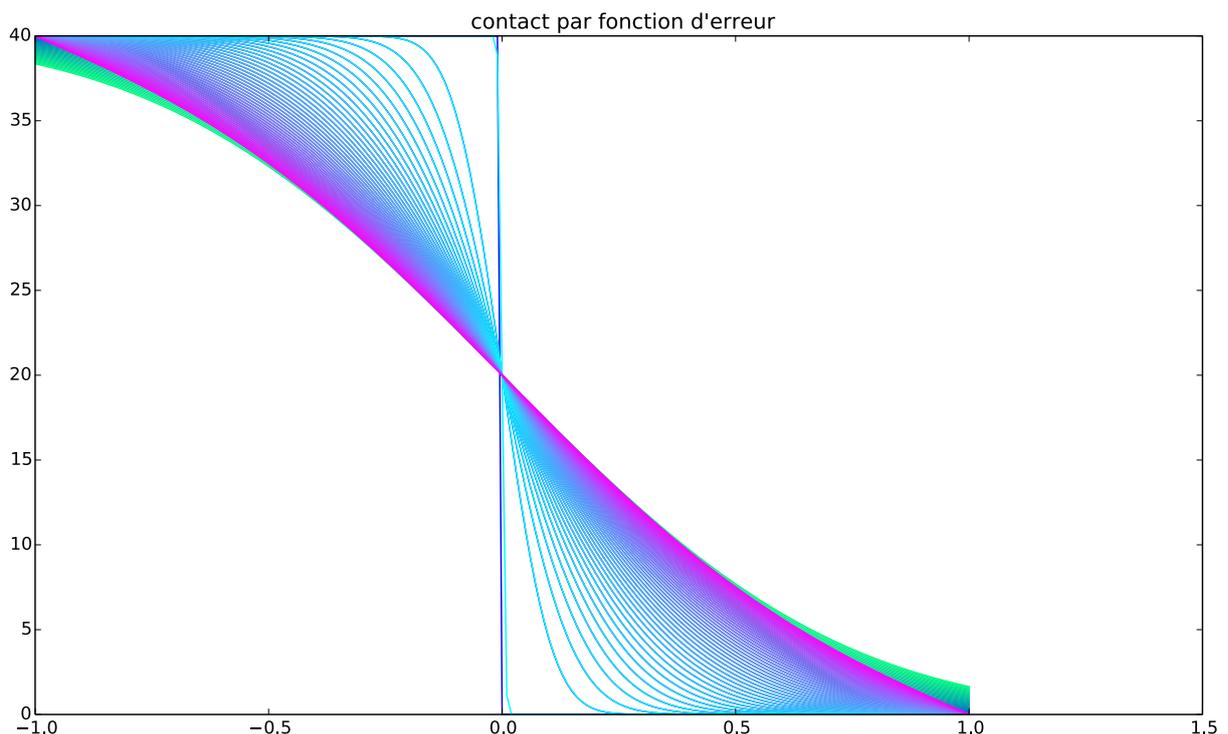
- la continuité du flux thermique en $x = 0 \ \forall t > 0$;
- la conservation de l'énergie totale des deux barres.

- b) On cherche une solution exacte. Montrer que la fonction $\operatorname{erf}(u) = \frac{2}{\sqrt{\pi}} \int_0^u e^{-y^2} dy$, avec $u = \frac{x}{\sqrt{4at}}$ est solution de l'équation

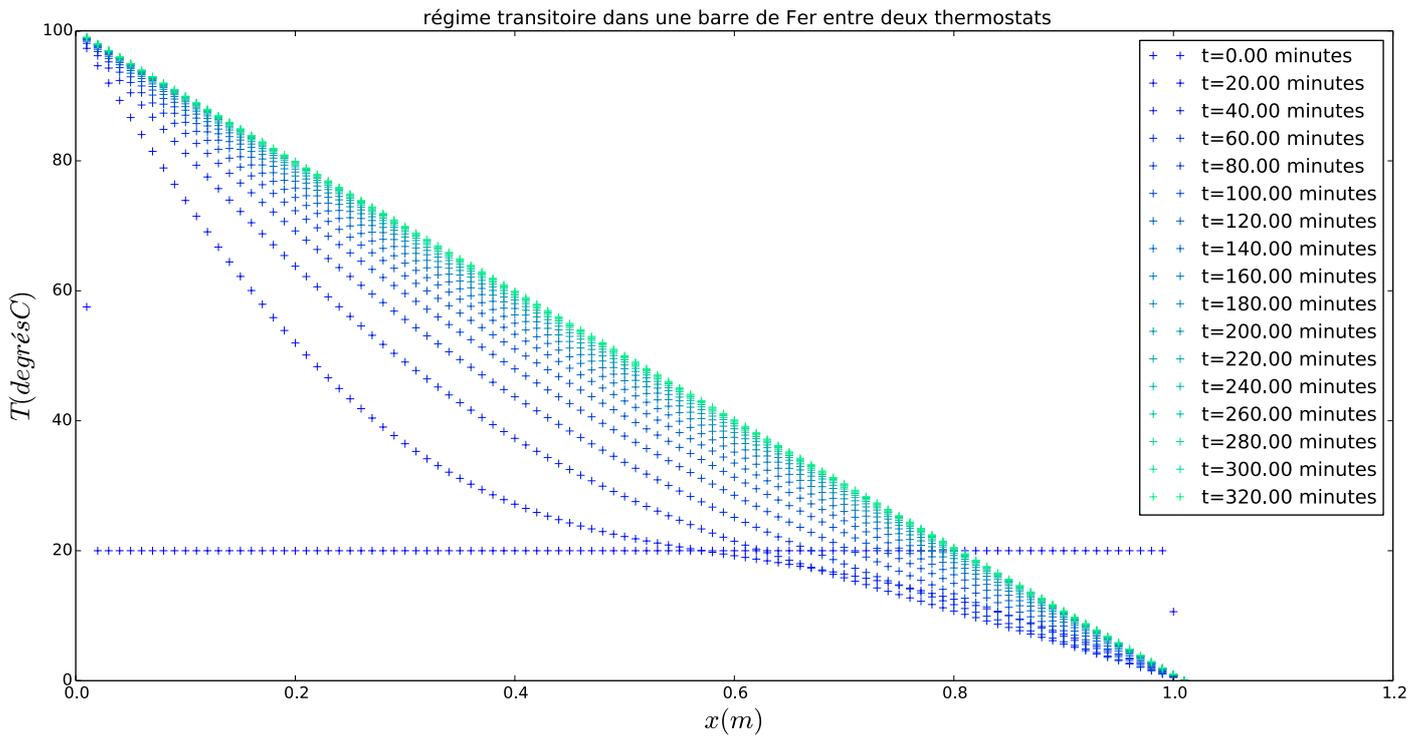
de la diffusion thermique, $a = \frac{\lambda}{\rho c}$ étant la diffusivité thermique du matériau. On donne $\int_0^{+\infty} e^{-y^2} dy = \frac{\sqrt{\pi}}{2}$

Calculer $T(x,t)$. En déduire $T(0,t)$ en fonction de T_1^0 , T_2^0 , Γ_1 et Γ_2 .

Annexe 2 : PARTIE II : Deux barres au contact : superposition de deux modélisations



Annexe 3 : PARTIE I : Barre de Fer entre 2 thermostats : approche discrète



Annexe 4 : PARTIE III : Mur de conductivité dépendant de la température

