1. Consignes générales

1.1. Découpage du module

Le module de modélisation sera découpé en deux phases de 5 séances de 2h de TP chacune. Durant chaque phase, vous serez amenés à étudier un sujet par groupe de 4 étudiants (ou 3-5 s'il n'est pas possible de ne faire que des groupes de 4). Chaque phase sera découpée en :

- une séance de présentation des sujets, formation des groupes et début d'étude ;
- 3 séances d'étude et de développement ;
- 1 séance de présentation orale des résultats.

1.2. Travail attendu

Pour chaque phase, chaque groupe devra réaliser :

- un court rapport ;
- le code source du programme développé ;
- une présentation orale.

1.2.1. Le rapport

Chaque groupe devra rendre un court rapport (5-6 pages de texte) contenant :

- un bref descriptif du sujet ;
- un journal de bord décrivant le travail réalisé durant chaque séance ;
- le modèle mathématique mis en place, ainsi que les éventuels calculs réalisés ;
- une conclusion expliquant les difficultés rencontrées et les outils acquis.

1.2.2. Le code

Quelques remarques sur le code :

- Le ou les programme(s) devront impérativement être réalisés en JAVA.
- Bien qu'aucun rendu ne soit demandé concernant la phase d'analyse, il vous est fortement conseillé d'en faire une.
- Seul le côté fonctionnel du programme sera évalué, mais un code bien écrit facilitera le travail du correcteur en cas de nécessité de débuggage.
- Aucune IHM (hors ligne de commande) n'est demandée.

1.2.3. La présentation

Lors de la dernière séance, chaque groupe disposera de 10 à 15 minutes pour présenter son travail. La présentation devra comporter :

- une présentation claire du sujet ;
- une explication détaillée du modèle mathématique
- une présentation rapide des résultats obtenus ;
- une présentation (non technique et brève) du programme réalisé.

1.3. Modalités de rendu et barème

1.3.1. Modalités de rendu

Le rapport et le code source devront être rendus **avant le jour de la présentation** par mail aux deux enseignants.
Le rapport devra impérativement être rendu au format pdf.
Le code source devra être rendu sous forme d'une archive ne comportant aucun fichier exécutable.

1.3.2. Notation

Pour chaque phase, vous obtiendrez trois notes :

- une pour le rapport, avec le coefficient 1 ;
- une pour le code, avec le coefficient 0,5 ;
- une pour la présentation, avec le coefficient 1,5.

Par défaut, tous les étudiants d'un même groupe se verront attribuer les même notes mais les enseignants sont libres de modifier cette règle si un manque de travail conséquent est observé chez l'un des participants du groupe.

2. Sujets : Phase 2

2.1. Cryptographie

2.1.1. Descriptif du sujet

Alice et Bob, deux scientifiques très réputés, souhaitent pouvoir communiquer régulièrement par mail. Ils souhaitent crypter leurs communications. Après quelques recherches, ils souhaitent utiliser un protocole de cryptographie symétrique réputé extrêmement sûr. Cependant, afin de mettre en place ce protocole, Alice et Bob doivent se mettre d'accord sur une clé (K) qui servira aussi bien à coder qu'à décoder leur message. Ne pouvant se rencontrer, Alice et Bob cherchent donc un moyen de partager une clé K sans que celle-ci puisse être déterminée par une autre personne. Pour cela, Alice propose d'utiliser le principe d'échange de clés de Diffie-Hellman. Bob n'est pas convaincu de l'intérêt de ce principe et propose à Alice de lui envoyer une clé en l'encodant à l'aide d'un chiffrement par transposition qu'ils utilisaient quand ils étaient ensemble à l'université.

2.1.2. Réalisation attendue

- **Objectif principal :**
  Après avoir compris les deux protocoles proposés, vous réaliserez deux programmes permettant le codage et le décodage d'informations numériques suivant les deux principes envisagés par Alice et Bob.
- **Prolongements possibles :**
  Après avoir réalisé ces programmes, voici quelques prolongements possibles :
  1. Conseillez Alice et Bob sur le choix du protocole de partage de clé.
  2. Si Alice et Bob ne s'étaient pas connus à l'université, auraient-ils pu utiliser la méthode proposée par Bob ? Et celle proposée par Alice ?
  3. Diffie-Hellman : expliquez comment fonctionne le principe de l'attaque de l'homme du milieu et mettez en place un programme simulant une telle attaque.
  4. Diffie-Hellman : expliquez le problème du logarithme discret et son lien avec Diffie-Hellman.
  5. Diffie-Hellman : étudiez l'algorithme "baby step giant step" pour la résolution du problème du logarithme discret.
  6. Chiffrement par transposition : proposez et étudiez un protocole d'attaque pour ce chiffrement.

2.2. Code détecteur/correcteur d'erreurs

2.2.1. Descriptif du sujet

Lors de la transmission d'une information, il est assez fréquent que celle-ci soit altérée. Cette altération peut avoir principalement deux causes :

- erreur de l'émetteur (faute de frappe...),
- manque de fiabilité du canal de communication (parasites...).

Afin de minimiser l'impact de ces altérations sur la compréhension d'un message, on utilise deux types d'algorithmes :

- les codes détecteurs d'erreurs, capables de repérer la présence d'une/plusieurs altération(s) dans un message,
- les codes correcteurs d'erreurs, capables de corriger une/plusieurs altérations(s) dans un message.

L'objectif de ce sujet est l'étude de trois algorithmes détecteurs/correcteurs d'erreurs :

1. la clé du numéro de sécurité sociale,
2. le code par redondance triple,
3. le code de Hamming binaire de paramètre $[7, 4, 3]$.

2.2.2. Réalisation attendue

- **Objectif principal :**
  Après avoir compris le principe des deux premiers algorithmes, vous les implémenterez.
- **Prolongements possibles :**
  Après avoir réalisé ces deux programmes, voici quelques prolongements possibles :
  1. Donnez (en les justifiant) les caractéristiques du principe de la clé du numéro de sécurité sociale.
  2. Donnez (en les justifiant) les caractéristiques du code par redondance triple.
  3. Expliquez le principe de la distance de Hamming ainsi que son intérêt dans les codes correcteurs d'erreurs.
  4. Explicitez le corps F_3. Listez les éléments de F_3^3 sous forme d'un graphe où l'on reliera chaque élément à ses voisins les plus proches (au sens de la distance de Hamming).
  5. Implémentez le troisième code correcteur.

2.3. Partage de secret

2.3.1. Descriptif du sujet

Quatre pirates ont réussi, suite à une attaque conjointe, à amasser un important butin. Incapables de se mettre d'accord sur la répartition du trésor, ils décident de le placer dans un coffre disposant de quatre serrures différentes réputées inviolables. Chacun des pirates conservera sa clé jusqu'à sa mort et seul le dernier pirate vivant pourra, après avoir récupéré les trois autres clés, ouvrir le coffre.

Derrière cet exemple romancé se cache une problématique très actuelle : le partage de secret. Le principe général du partage de secret est le suivant :

- Un groupe de n personnes dispose d'un secret K.
- On "découpe" ce secret en n morceaux en en donnant 1 à chacune des personnes du groupe.
- On souhaite que K ne puisse être reconstruit que si au minimum m personnes (avec $m \leq n$) du groupe rassemblent leur morceaux.

Voici quelques protocoles utilisables pour $n = 2$:

1. On suppose que K est composé de $2k$ bits et on donne k bits à chaque personne.
2. On choisit deux grands nombres premiers distincts p et q. La première personne reçoit $K[p]$ et la deuxième $K[q]$.
3. On choisit un nombre premier p, 2 éléments x_1 et x_2 de $\mathbb{Z}/p\mathbb{Z}$ et un polynôme P dont la somme des coefficients est K. On donne $P(x_1)$ à la première personne et $P(x_2)$ à la deuxième.

2.3.2. Réalisation attendue

- **Objectif principal :**
  Après avoir étudié les deux premiers protocoles, vous les implémenterez (découpage et reconstruction de K).
- **Prolongements possibles :**
  Après avoir réalisé ces programmes, voici quelques prolongements possibles :
  1. Proposez une version du premier protocole pour 5 personnes, où 3 personnes sont suffisantes (et nécessaires) pour reconstruire K.
  2. Proposez une version du deuxième protocole pour n personnes et $m = n$.
  3. Proposez une variante du deuxième protocole pour $m < n$.
  4. Expliquez comment retrouver K dans le troisième protocole.
  5. Expliquez comment adapter le troisième protocole pour n personnes et $m < n$.

2.4. Rubik's cube

2.4.1. Descriptif du sujet

Inventé par Erno Rubik en 1974, le Rubik's cube (classique) consiste en un cube constitué de $3 \times 3 \times 3$ petits cubes dont les faces sont colorées. Dans son état résolu, toutes les faces du grand cube sont de couleur uniforme, mais il est possible de perturber cette disposition en faisant tourner les "tranches" du cube.

Bien qu'il existe plus de 43×10^{19} combinaisons possibles, certains cubeurs sont capables de résoudre un Rubik's cube en moins de 6 secondes. La plupart des méthodes de résolution utilisées ont une origine mathématique basée sur la théorie des groupes. L'objectif de ce sujet est l'étude de ces théories sur l'exemple du "pocket cube" (Rubik's cube $2 \times 2 \times 2$).

On considérera l'ensemble G des permutations du cube.

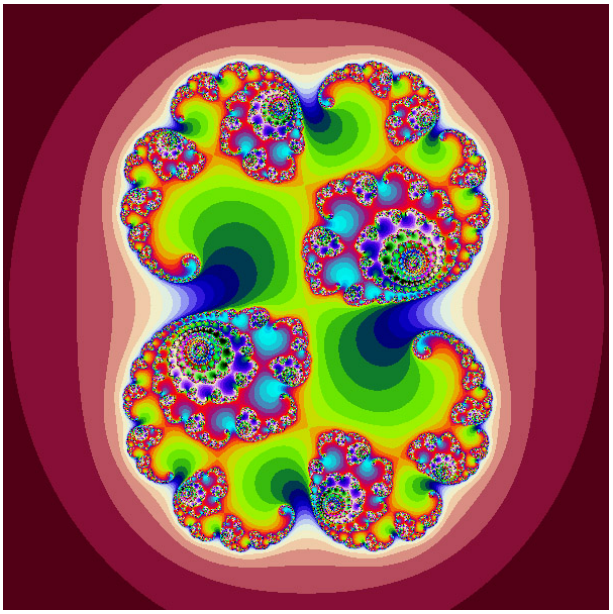
2.4.2. Réalisation attendue

- **Objectif principal :**
  Implémentez un programme générant aléatoirement une configuration possible du "pocket cube" et en donnant une liste de mouvements permettant sa résolution.
- **Prolongements possibles :**
  Après avoir réalisé ce programme, voici quelques prolongements possibles :
  1. Démontrez que G est un groupe.
  2. Déterminez le cardinal de G.
  3. Déterminez une famille génératrice de G.
  4. Déterminez les différents ordres possibles pour les éléments de G en donnant des exemples pour chacun.
  5. Proposez un algorithme de résolution du "pocket cube".
  6. Étendez vos résultats au Rubik's cube classique.

2.5. Les lapins de Douady et l'ensemble de Mandelbrot

2.5.1. Descriptif du sujet

En mathématiques, les ensembles de Julia et de Mandelbrot sont de célèbres ensembles fractals définis à partir du comportement d'une fonction par composition itérée avec elle-même.



Les lapins de Douady sont une famille assez célèbre d'ensembles de Julia définis de la façon suivante. On considère le polynôme complexe :

$$P_c(z) = z^2 + c$$

où c vérifie

- $P_c(0) \neq 0$, $P_c(P_c(0)) \neq 0$ et $P_c(P_c(P_c(0))) = 0$;
- c n'est pas un réel.

On définit alors l'ensemble $J(P_c)$ comme le sous-ensemble de $\mathbb{C}$ constitué des z tels que la suite $P_c^n(z)$ est bornée. Cet ensemble est appelé un lapin de Douady.

On définit de façon similaire l'ensemble de Mandelbrot comme le sous-ensemble de $\mathbb{C}$ des valeurs de c tel que $P_c^n(0)$ soit bornée.

2.5.2. Réalisation attendue

- **Objectif principal :**
  Après avoir montré que si le module de z dépasse 2, $P_c^n(z)$ diverge, vous implémenterez un programme représentant graphiquement l'ensemble des z tel que le module $P_c^n(z)$ ne dépasse pas 2 pour tout n inférieur à un entier M (M et c seront saisis par l'utilisateur).
- **Prolongements possibles :**
  Après avoir réalisé ce programme, voici quelques prolongements possibles :
  1. Utilisez votre programme pour représenter les deux lapins de Douady. Quel lien entre les deux lapins remarquez-vous ? Démontrez-le.
  2. En vous inspirant du programme principal, implémentez un programme permettant de dessiner l'ensemble de Mandelbrot.
  3. Démontrez que l'ensemble de Mandelbrot est borné.
  4. Déterminez l'ensemble de Julia pour $c = 0$.
  5. Déterminez les valeurs de c telles que P_c admette un point fixe. Quelle conséquence cela implique-t-il sur l'ensemble de Julia associé ?

2.6. La fourmi

2.6.1. Descriptif du sujet

Une fourmi se balade sur les dalles de la terrasse carrée de la famille Tartempion. Félix, le jeune Tartempion, observe cette fourmi. Il s'est fixé un repère au milieu des 100 dalles carrées dont les axes suivent les rangées de carreaux et il constate que la fourmi passe successivement par les points de coordonnées $(-1, -3/2)$, $(-1/2, 0)$, $(0, 1/4)$, $(1/2, 0)$, $(1, 0)$.

Le jeune Tartempion lance un défi à sa grande sœur Anabelle : déterminer la trajectoire de la fourmi pour comparer les résultats avec ses observations.

Après réflexion, Anabelle décide d'utiliser 4 méthodes d'interpolation différentes pour déterminer la trajectoire de la fourmi :

- l'interpolation linéaire,
- les polynômes de Lagrange,
- les splines cubiques,
- les courbes de Béziers.

2.6.2. Réalisation attendue

- **Objectif principal :**
  Implémentez les deux premiers algorithmes et comparez leurs résultats.
- **Prolongements possibles :**
  Après avoir réalisé ce programme, voici quelques prolongements possibles :
  1. Quelles conséquences aurait l'augmentation du nombre de points d'observation sur les deux premières méthodes ? Justifiez et donnez des exemples.
  2. La méthode des splines cubiques est-elle directement applicable ici ? Peut-on l'adapter ?
  3. Implémentez la méthode des splines cubiques. Comparez son résultat aux deux précédents.
  4. La méthode des courbes de Béziers est-elle directement applicable ici ? Peut-on l'adapter ?
  5. Implémentez la méthode des courbes de Béziers. Comparez son résultat aux trois précédents.
  6. La fourmi rencontre sur son chemin le ballon du jeune Tartempion. Elle grimpe sur le ballon et fait un tour jusqu'à retrouver le sol. Comment simuler la trajectoire sur le ballon ?

2.7. Ça chauffe

2.7.1. Descriptif du sujet

Une tige de métal est chauffée à une extrémité par une flamme tandis que son autre extrémité est maintenue à température constante.

La température de la tige dépend non seulement du temps t mais de la position x dans la tige. Par conséquent, cette température sera représentée par une fonction à deux variables $T(x,t)$. L'évolution de la température dans la tige est donnée par l'équation :

$$\frac{\partial T(x,t)}{\partial t} = a\frac{\partial^2 T(x,t)}{\partial x^2}.$$

La fonction T étant à deux variables, on peut la dériver par rapport à x (une fois : $\frac{\partial T(x,t)}{\partial x}$, deux fois : $\frac{\partial^2 T(x,t)}{\partial x^2}$...) ou par rapport à t ($\frac{\partial T(x,t)}{\partial t}$).

Pour simplifier le problème, on supposera que la tige mesure un mètre ($x \in [0,1]$). L'extrémité située en $x = 1$ est chauffée à une température T_f et l'extrémité située en $x = 0$ est maintenue à la température T_c. Initialement ($t = 0$) la tige est entièrement à une température constante T_0.

Il est très fréquent de rencontrer ce genre d'équations en physique, en chimie ou en biologie. Hélas, leur étude est souvent délicate. Bien souvent, il est possible de démontrer théoriquement l'existence d'une solution mais bien plus rarement de pouvoir donner une formule pour celle-ci. Dans le cas où une formule n'est pas obtenable (ou trop complexe), on privilégie une approche numérique du problème en essayant de déterminer des "valeurs approchées" de la fonction afin de pouvoir simuler informatiquement le phénomène étudié.

2.7.2. Réalisation attendue

- **Objectif principal :**
  Implémentez la méthode des différences finies pour résoudre l'équation (E) avec les conditions initiales suivantes : $T_c = 35°C$, $T_f = 400°C$, $T_0 = 25°C$ et $a = 9 \times 10^{-5}$.
- **Prolongements possibles :**
  Après avoir réalisé ce programme, voici quelques prolongements possibles :
  1. On suppose de plus que le flux de chaleur propagé par la flamme suit la loi de refroidissement de Newton :
     $$\frac{\partial T(1,t)}{\partial t} = h\left(T_f - T(1,t)\right).$$
     Adaptez la méthode précédente à ce nouveau problème.
  2. Quelles difficultés peuvent apparaître pour un grand nombre n de subdivisions ?
  3. Comment adapter la méthode pour une plaque mince ? Dans ce cas, l'équation E devient :
     $$\frac{\partial T(x,y,t)}{\partial t} = a\left(\frac{\partial^2 T(x,y,t)}{\partial x^2} + \frac{\partial^2 T(x,y,t)}{\partial y^2}\right).$$

2.8. Épidémie

2.8.1. Descriptif du sujet

Un modèle pour la diffusion d'une épidémie se base sur l'hypothèse que la vitesse de propagation est proportionnelle au nombre d'individus infectés et au nombre d'individus sains.
On note $I(t)$ le nombre d'individus infectés à la date t et P le nombre total d'individus de la population. Ceci revient à poser l'existence d'une constante k telle que :

$$I'(t) = kI(t)(P - I(t)) \text{ et } I(0) = I_0.$$

Il est très fréquent de rencontrer ce genre d'équations en physique, en chimie ou en biologie. Hélas, leur étude est souvent délicate. Bien souvent, il est possible de démontrer théoriquement l'existence d'une solution mais bien plus rarement de pouvoir donner une formule pour celle-ci. Dans le cas où une formule n'est pas obtenable (ou trop complexe), on privilégie une approche numérique du problème en essayant de déterminer des "valeurs approchées" de la fonction afin de pouvoir simuler informatiquement le phénomène étudié.

2.8.2. Réalisation attendue

- **Objectif principal :**
  Déterminez une solution approchée de la fonction I en implémentant la méthode d'Euler explicite pour une population de 10000 individus, 2% d'individus infectés à la date $t = 0$ et pour différentes valeurs de k. Comment appelleriez-vous k ?
- **Prolongements possibles :**
  Après avoir réalisé ce programme, voici quelques prolongements possibles :
  1. Comparer la solution précédente avec celle obtenue en prenant le schéma de discrétisation semi-implicite suivant :

     $$\frac{I_{n+1} - I_n}{\Delta t} = kI_n(P - I_{n+1})$$

  2. On suppose que les malades guérissent et sont immunisés pour une certaine période. On obtient alors l'équation différentielle suivante :

     $$I'(t) = kI(t)(P - I(t)) - rI(t)$$

     Appliquez les méthodes précédentes à cette nouvelle équation. Que représente r ?
  3. Pourrait-on utiliser d'autres méthodes numériques que celle d'Euler pour résoudre ce type d'équation différentielle ? Si oui, implémentez-en une pour l'équation différentielle initiale.