

7. SOMMAIRE DES FONCTIONS CRÉÉES SOUS MATLAB

Voici ci-dessous la liste des fonctions que nous avons dû créer. Dans ce tableau figurent les arguments d'entrée et sortie pour chacune d'entre-elles, ainsi que la description de leur rôle.

Fonctions concernant la température et le vent :

Nom	Arguments d'entrée	Arguments de sortie	Rôle
<i>tempdisk</i>	r_{max} : rayon extérieur de l'anneau n_1 : nombre de subdivisions pour le rayon n_2 : nombre de subdivisions pour l'angle Pe : Constante de Péclet Th : Température au sein de la horde Ti : Température au loin de la horde	- z contient les affixes des points de calcul - x contient les solutions aux points dans z - znormal contient les affixes des points sur le cercle - normal approche la dérivée normale aux points dans znormal	Détermine la température en tout point du maillage de l'anneau
<i>temppoly</i>	- zdisk est un vecteur qui contient les coordonnées des points où nous avons effectué nos calculs dans l'anneau - xdisk est un vecteur qui contient les températures aux points dans zdisk - sc est l'application de Schwarz-Christoffel	- zpoly est un vecteur qui contient les affixes concernées autour de p - xpoly est un vecteur qui contient les températures aux points d'affixes contenues dans zpoly	Après avoir obtenu nos résultats autour du disque avec la fonction tempdisk. Cette fonction nous permet via l'application de Schwarz-Christoffel d'obtenir les températures correspondantes aux points correspondants autour du polygone
<i>evaluer</i>	- chaine qui sert à choisir le type d'évaluation que l'on souhaite effectuer (dérivée, inverse...) - scl, c'est l'application créée par le package Tolbin - x est l'affixe du point où l'on souhaite évaluer l'application	- y est l'image de x	Permet d'évaluer l'application Schwarz-Christoffel, ainsi que son inverse, sa dérivée... l'application correspond à celle générée par le package Tolbin rectifiée par la composition avec une rotation

<i>heatloss</i>	- indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique - p est le polygone correspondant à la frontière, tel que p(i) contienne l'abscisse du manchot d'indice indfront(i) -sc est l'application de Schwarz-Christoffel - znormal, vecteur qui contient les abscisses des points de calcul - normal contient les dérivées normales en chaque point de znormal	- Heatloss est un vecteur qui contient les pertes de chaleur des manchots de indfront	Permet de calculer la perte de chaleur de chaque manchot situé sur la frontière
<i>heatloss2</i>	- Mêmes arguments que pour heatloss avec celui-ci en plus : - der est un vecteur tel que der(i) contient pour un manchot indfront(i) l'indice du polygone p correspondant au dernier sommet du manchot indfront(i)	- Heatloss est un vecteur qui contient les pertes de chaleur des manchots de indfront	A la même fonction que heatloss mais est adaptée au 2 ^e modèle étudié
<i>integrale</i>	- zmoins et zplus sont les bornes de l'intégrale - sc est l'application de Schwarz-Christoffel - znormal, vecteur qui contient les abscisses des points de calcul - normal est le vecteur qui contient les approximations des dérivées normales en chaque point de znormal	- Int est la valeur de l'intégrale calculée	Cette fonction permet de calculer l'intégrale entre le long du polygone de la dérivée normale de la température

Notons que c'est en grande partie au sein des fonctions ci-dessus que nous avons besoin du package de Tobin A. Driscoll concernant les applications de Schwarz-Christoffel.

Les fonctions utilisées dans ce package sont les suivantes :

- ★ ***polygon*** : génère un polygone grâce aux données des abscisses des sommets
- ★ ***extermmap*** : construit l'application de Schwarz-Christoffel envoyant l'extérieur d' un polygone donné sur l'extérieur du disque unité

- ★ *evaldiff* : évalue la dérivée de l'application de Schwarz-Christoffel créée en un point
- ★ *evalinv* : évalue l'inverse de l'application de Schwarz-Christoffel créée en un point

Fonctions concernant la génération et les modifications de la horde :

Nom	Arguments d'entrée	Arguments de sortie	Rôle
<i>ping</i>			Ping est la classe sur laquelle on va travailler. Un manchot de la classe ping a son indice, sa position en abscisse x, sa position en ordonnée y, et le vecteur « libre » qui contient les indices des voisins libres
<i>horde</i>	- N est le nombre de manchots que l'on souhaite avoir dans la horde que nous créons	- groupe est un vecteur contenant les caractéristiques de tous les manchots de notre horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique	Créer une horde de N manchots en suivant nos conditions de construction
<i>ajoute</i>	- indnew est l'indice du manchot que l'on souhaite ajouter - indv1 et indv2 sont les manchots auxquels on compte coller le manchot arrivant indnew - dirv2 est la direction (en partant de indv1) dans laquelle on va placer indnew - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique - param vaut 1 si il s'agit de son utilisation pour la simulation, vaut 2 si il s'agit de son utilisation pour la génération de la horde	- groupe est la horde mise à jour après ajout de indnew - indfront est mis à jour lui aussi - ok est un booléen égal à 1 si l'ajout s'est effectué, 0 sinon	Ajoute un manchot à la horde en mettant à jour la horde et le vecteur indfront. (Peut ne pas ajouter de manchot si les 3 conditions pour placer un manchot ne sont pas satisfaites, dans ce cas là il renvoie un booléen ok=0 sans changer la horde)

<i>enleve</i>	- ind est l'indice du manchot que l'on souhaite enlever - H est la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique	- H est la horde mise à jour - b est le vecteur indfront mis à jour - ok est un booléen qui vaut 1 si le manchot a bien été enlevé, 0 sinon	Permet de retirer un manchot de la horde tout en respectant nos conditions de construction de la horde
<i>deplacement</i>	- pos est la position du manchot d'où l'on part - dir est la direction dans laquelle on part du manchot qui se situe en position pos	- position contient les coordonnées de là où l'on arrive en étant parti du manchot en pos dans la direction dir	- Permet d'obtenir la position d'arrivée après être parti d'un manchot en allant dans une direction choisie
<i>detect</i>	- ind est l'indice d'un manchot - H est la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique	- u est le vecteur contenant les indices dans indfront des voisins frontaliers de ind	- Donne les indices des manchots frontaliers de celui rentré en argument
<i>detect2</i>	- pos est la position (x,y) du manchot que l'on étudie - H est la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés	u est le vecteur contenant les indices dans indfront des voisins frontaliers de ind	- Donne les indices des manchots frontaliers de celui rentré en argument
<i>present</i>	- pos est la position (x,y) que l'on teste - H est la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique	- test est un booléen de valeur 1 si il y a un manchot, 0 sinon	Teste si un manchot occupe la position pos
<i>testdir</i>	- j est la direction dans laquelle on veut vérifier si il y a un manchot - test est le manchot	- bool est un booléen de valeur 1 si un manchot est présent dans la direction j en partant de test, 0 sinon	Vérifie si un manchot est présent dans une direction j en partant d'un manchot test

	duquel on part - H est la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés.		
<i>trou</i>	- pos est la position test - u contient les indices des voisins libres de pos - H est la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique	- bool est un booléen de valeur 1 si l'on crée un trou, vaut 0 sinon	Vérifie si l'ajout d'un manchot à la position pos nous fait créer un trou dans la horde ou non
<i>localise</i>	- H est la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique - P est le manchot duquel on part - j est la direction dans laquelle on part depuis le manchot P	- manchot est de type ping, il s'agit du manchot voisin de P dans la direction j (on travaille uniquement sur les frontaliers)	Détecte le manchot qui est le voisin de P dans la direction j
<i>enferme</i>	- pos est la position que l'on veut tester - j est la direction dans laquelle on part pour nos tests (partant de pos) - H est la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique	- bool est un booléen égal à 1 si la position pos est dans un trou du maillage, 0 sinon	Teste si la position pos est dans un trou du maillage ou non
<i>admi</i>	- pos contient les coordonnées (x,y) de l'endroit où l'on veut tester si un manchot est admissible (selon nos critères de modélisation) - H la horde - indfront, le vecteur	- bool, qui a pour valeur 1 si pos peut accueillir un manchot, 0 sinon - u est un vecteur qui contient les indices correspondants aux voisins libres de pos	Teste si la position pos respecte les 3 conditions pour accueillir un manchot, renvoie le booléen correspondant

	qui contient les manchots frontaliers ordonnés dans le sens trigonométrique		
<i>oppose</i>	- i la direction de base	- j la direction opposée à i	Renvoie la direction opposée à la direction i
<i>poly_contour</i>	- H la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique - p le polygone passant par les centres des manchots frontaliers obtenu par la fonction hordepolycentre.	- p2 le polygone passant par les sommets des hexagones - der , vecteur contenant les indices dans p2 du dernier sommet de l'hexagone du manchot dans p	- Construit le polygone décrivant le contour de la horde (passant par le contour des hexagones et non pas par les centres des hexagones)
<i>hordepolycentre</i>	- H la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique	- p le polygone dont les sommets sont les affixes des centres des manchots frontaliers	Construction du polygone décrivant la frontière. Il s'agit du polygone qui sera soumis souvent aux applications Schwarz-Christoffel
<i>newhorde</i>	- H la horde - Heatloss, le vecteur contenant les pertes de chaleur de tous les manchots frontaliers. - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique	- Hnew la horde mise à jour après déplacement du manchot - indfrontnew, le vecteur indfront mis à jour après déplacement du manchot	Déplace le manchot ayant la plus grande perte de chaleur et le place à côté de celui ayant la plus petite (en respectant les conditions de notre modèle). Puis met à jour la horde et le vecteur indfront
<i>Position</i>	- pos correspond aux coordonnées (x,y) du manchot de référence - dir est la direction dans laquelle on part depuis pos	- position contient les coordonnées de là où l'on arrive en étant parti du manchot pos dans la direction dir	Permet d'avoir la coordonnée de l'emplacement potentiel d'un manchot en partant d'un manchot pos dans une direction dir

Fonctions permettant la visualisation des résultats :

Nom	Arguments d'entrée	Arguments de sortie	Rôle
<i>dessin</i>	- H la horde - indfront, le vecteur qui contient les manchots frontaliers ordonnés dans le sens trigonométrique - p le polygone représentant la frontière de la horde		Représente la horde de manchots (avec les hexagones)
<i>dessin2</i>	- H la horde - p le polygone représentant la frontière de la horde - s la taille du point au centre de l'hexagone représentant le manchot (dépend du nombre de manchots)		Représente la horde de manchots (en dessinant le contour et représentant chaque manchot avec un point)
<i>hexagone</i>	- x la partie réelle de z - y la partie imaginaire de z		Crée un hexagone situé autour du point d'affixe z de coordonnées (x,y)
<i>hexagone2</i>	- x la partie réelle de z - y la partie imaginaire de z - v vecteur à trois composantes servant à coder une couleur		Crée un hexagone situé autour du point d'affixe z de coordonnées (x,y), et coloriant l'intérieur d'une couleur codée par v. (utilisée pour représenter les manchots frontalier avec leur heatloss)
<i>point</i>	- z_1 affixe du centre de l'hexagone dans lequel on travaille - dir la direction du sommet dont on veut avoir l'affixe	- z_2 l'affixe du sommet de l'hexagone centré en z_1 en étant parti dans la direction dir	Donne l'affixe du sommet choisi d'un hexagone en ayant rentré en paramètre le centre de l'hexagone
<i>dessinventpoly</i>	- p le polygone représentant la horde - sc l'application de Schwarz-Christoffel		Représente le vent autour du polygone (La force du vent ne rentre pas dans les paramètres parce que la représentation resterait inchangée car

			le champ de vecteur serait proportionnel et la fonction quiver donnerait le même résultat)
<i>dessintemppoly</i>	- z contient tous les points du maillage - t est un vecteur contenant les températures des affixes stockés dans z - n représente le nombre de subdivisions pour les deux axes - H la horde		Représente la température en couleur autour du polygone (rouge pour les zones chaudes, bleu pour les zones froides)
<i>Heat_color</i>	- p le polygone - Heatloss le vecteur contenant les pertes de chaleur de tous les manchots frontaliers		Représente par une échelle de couleurs les valeurs du Heatloss pour les manchots de la frontière. (rouge = Heatloss faible, bleu = Heatloss élevé)
<i>circus</i>	- n le nombre de côtés que l'on souhaite que le polygone ait	- p le polygone dessiné	Construit un polygone régulier à n côtés
<i>maillage</i>	- p le polygone frontalier		Construit un maillage autour du disque et renvoie le dessin de l'image du maillage autour du polygone p
<i>simulation</i>	- nb_ping le nombre de manchots que l'on souhaite avoir dans la horde -nb_etapes le nombre d'étapes que l'on souhaite exécuter - Pe la valeur de la constante de Péclet	- M1 et M2 génèrent toutes les images permettant de faire une animation. M1 stocke les images avec les hexagones, M2 stocke les images avec représentation du vent et de la température à l'extérieur de la horde	Fait une simulation de nb_etapes étapes avec nb_ping manchots et une constante de Péclet égale à Pe Ci-dessous, nous allons plus développer cette fonction

Fonction simulation :

La fonction simulation commence par générer une horde avec le nombre de manchots rentré en paramètre et crée aussi le vecteur indfront stockant les indices des manchots frontaliers. Puis appelle la fonction hordepolycentre pour créer le polygone frontalier.

Ensuite nous effectuons la discrétisation de l'équation de la température grâce à la fonction tempdisk.

Comme nous allons nous intéresser à certaines statistiques de nos simulations, nous créons aussi les vecteurs heatmat, heatmean, heatmax, front_compteur et ratio qui nous permettent d'extraire des données intéressantes (heatloss moyen à chaque étape, etc...).

On entre maintenant dans la boucle qui va effectuer le nombre d'étapes demandé.

Une étape se détaille de la façon suivante :

- On calcule les température autour du polygone à l'aide de la fonction temppoly.
- On calcule le vecteur Heatloss stockant les perte de chaleur des manchots frontaliers.
- On entre dans les vecteurs heatmat, heatmean, heatmax, front_compteur et ratio les informations dont nous avons besoin pour pouvoir les récupérer en sortie de boucle.
- On règle les axes pour avoir notre horde bien dessinée dans nos fenêtres.
 - On représente sur une figure la horde avec les hexagones, ceux frontaliers étant coloriés avec des couleurs montrant le Heatloss de chaque manchot. Sur une deuxième figure nous représentons le polygone avec les manchots représentés par des points, et autour du polygone sont affichés le vent et la température hors de la horde.