

# MEC6212: GENERATION de MAILLAGES

---

## Travail pratique: Génération de mailles par des équations différentielles elliptiques.

20 février 2024

### Énoncé

L'objectif est de mailler un domaine 2D défini à partir de quatre courbes quelconques utilisant une technique de résolution des équations de mailles issues d'une équation différentielle elliptique.

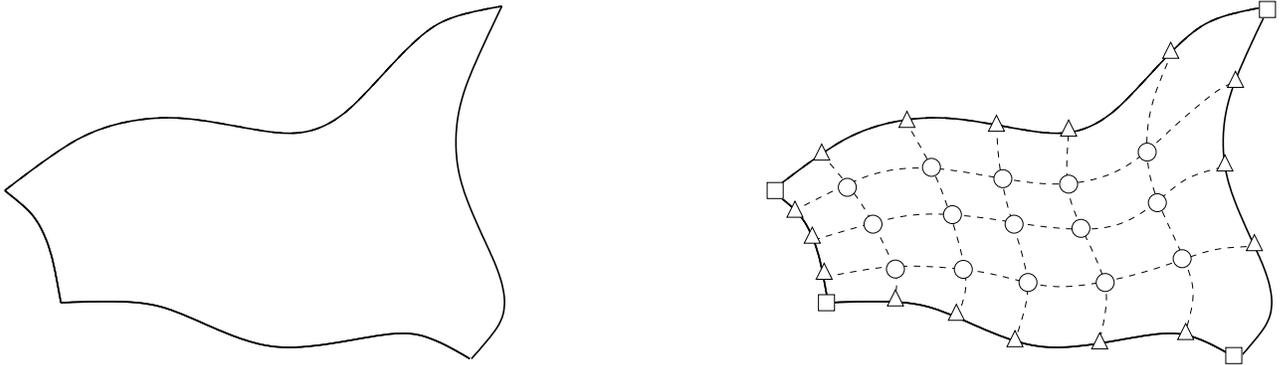


FIGURE 1 – Maillage Elliptique

Dans ce travail, deux fonctions sont à rédiger :

1. une fonction **winslow** qui calcule un réseau de lignes de maillage curvilignes à partir d'un algorithme basé sur la cette méthode, et les structures de données associées ;
2. une fonction **winslowConvergence** qui affiche l'historique de la convergence de la solution itérative.

# 1 La fonction winslow.m

Cette fonction résoud les équations de maille par une méthode de relaxation par point, et donne le résultat illustré à la Fig. 1. Le protocole d'utilisation est donné ci-dessous. où, en argument,

```
function winslow(iterL , iterB , omega , m , n , debutFACE , finFACE)
%-----
global x y nbNOD V
global NormeX NormeY
%-----
alfa = ones(nbNOD,1);
beta = ones(nbNOD,1);
gama = ones(nbNOD,1);
NormeX=zeros(iterL ,1);
NormeY=zeros(iterL ,1);
m=double(m);
n=double(n);
```

FIGURE 2 – Protocole d'appel et variables partagées avec *MARS*

iterL : nombre d'itérations ;

iterB : nombre de balayages ;

omega : facteur de surrelaxation ;

$m \times n$  : dimensions de la discrétisation de la face.

Par exemple à la Fig. 1,  $m = 5$  et  $n = 3$ .

On note que ces dimensions excluent les deux coins.

debutFACE : indice du début des noeuds de la face ;

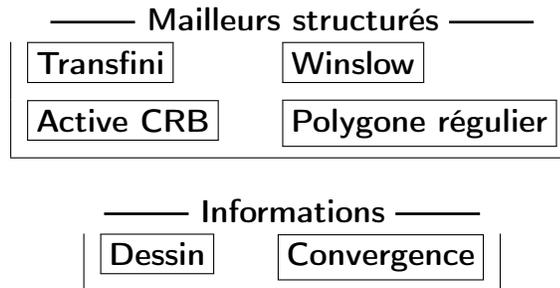
finFACE : indice de la fin des noeuds de la face.

Cette fonction partage les variables  $x$ ,  $y$ ,  $\text{varnbNOD}$  et  $V$  avec les autres fonctions de l'application *MARS*,

```
global x y nbNOD V
global NormeX NormeY
```

En sortie, *MARS* affiche le maillage calculé par **winslow.m**, et calcule également l'historique de la convergence.

On lance le mailleur avec le bouton `Structure` du menu racine, qui donne le panneau suivant, et le bouton `Transfini` suivi de `Winslow`.



## 2 La fonction `structureWinslowConvergence.m`

La méthode de calcul utilisée pour la résolution des équations de maille est une technique itérative, et dans `Winslow` on calcule une norme de l'erreur,

NormeXS : la norme  $L_2$  du résidu de  $x$  ;

NormeYS : la norme  $L_2$  du résidu de  $y$  ;

```
function structureWinslowConvergence  
global NormeX NormeY
```

La fonction `winslowConvergence.m` affiche l'historique de la convergence du processus itératif, i.e. les normes en  $x$  et  $y$  en fonction du nombre d'itérations. Ceci consiste d'un graphe contenant deux courbes, `NormeXS` et `NormeYS` versus le nombre d'itérations. Pour mieux faire ressortir cette convergence, on utilisera des axes log ou semi-log.

Cette information permet de valider et évaluer l'atteinte de la convergence.

### 3 Construction du domaine

Cette étape se fait dans *MARS*.

1. La géométrie comprend quatre courbes quelconques, construites dans le module *Geometrie* ;
2. Le domaine est borné par quatre bords définis sur ces courbes avec le module *Domaine* ;
3. Les bords sont discrétisés avec  $m$  et  $n$  sommets, dans les directions  $\xi$  et  $\tau$ , respectivement, spécifiés dans *Domaine* ;
4. Ces informations constituent le point de départ de la fonction *winslow*,  
Coins : les coordonnées des quatre coins sont stockées dans les vecteurs,  
 $x(1 :nbCOIN)$  et  $y(1 :nbCOIN)$  ;  
Bords : les coordonnées des quatre bords suivent séquentiellement,  
 $x(iniBRD1 :finBRD1)$  et  $y(iniBRD1 :finBRD1)$  pour le premier bord,  
 $x(iniBRD2 :finBRD2)$  et  $y(iniBRD2 :finBRD2)$  pour le second bord, .....

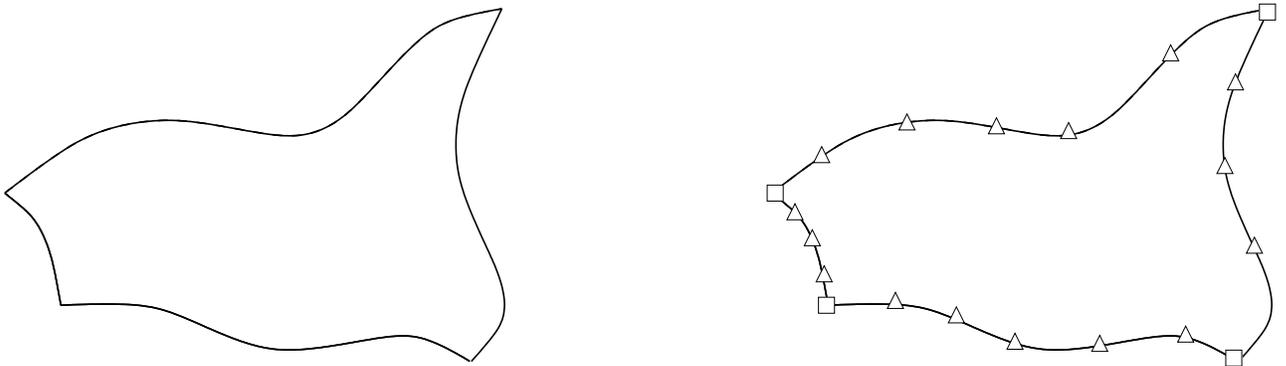


FIGURE 3 – Discrétisation des frontières du domaine

5. Ces données sont accessibles dans la fonction *winslow* via la déclaration,  
***global x y nbNOD V***

Les coordonnées des sommets sur les coins et les bords sont construits dans *MARS*, tandis que les sommets sur les faces (i.e. le maillage) sont calculés par la fonction *winslow*.

L'ensemble de ces coordonnées sont stockées séquentiellement dans deux tableaux  $x$  et  $y$ , dans l'ordre suivant : les coins, suivis des quatre bords, et finalement l'intérieur du domaine, le maillage.

$x$  et  $y$  : les coins,  
 $x(1 :nbCOIN)$

$x$  et  $y$  : les bords,  
 $x(debutBORD1 :finBORD1)$   
 $x(debutBORD2 :finBORD2)$

*x(debutBORD3 :finBORD3)*  
*x(debutBORD4 :finBORD4)*

x et y : les faces,

*x(debutFACE1 :finFACE1)*  
*x(debutFACE2 :finFACE2)*

.....

*y(debutFACEnb :finFACEnb)* pour la dernière face.

Dans l'exemple de la Fig. 1, *debutFACE=21* et *finFACE=35*, et pour l'ensemble du domaine *nbNOD=finFACE=35*.

Les sommets (*x(debutFACE1 :finFACE1)*,*y(debutFACE1 :finFACE1)*) sont calculées par la fonction **winslow.m**

## 4 Structure de données

Tous les sommets reposant sur les entités *COIN*, *BORD* et *FACE* sont stockés dans deux tableaux *x* et *y*, et sont regroupés selon ces types comme illustré à la Fig. 4.

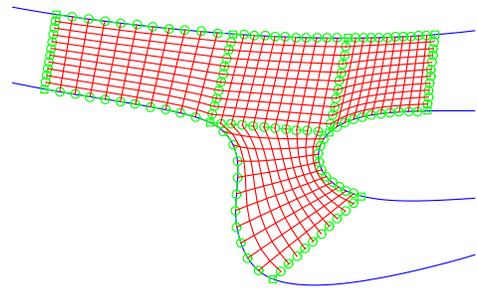
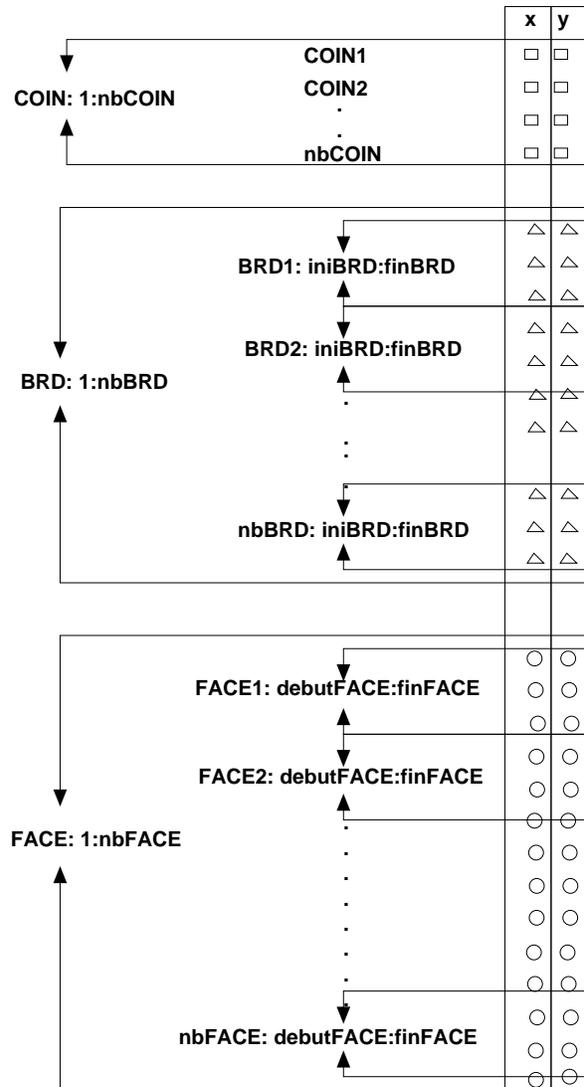


FIGURE 4 – Structure de données pour les sommets du maillage

## 5 Vecteur d'adressage : numérotation des voisins

Avec cette structure, les sommets sont référencés par un seul indice,  $k$ , soit,  $x(k)$  et  $y(k)$ .

Il faut donc établir une correspondance entre un référencement  $x(k)$  et  $x(i,j)$ , c-à-d le voisinage des noeuds  $(i,j)$  dans la numérotation  $k$ . Cette correspondance est illustrée à la Figure 5.

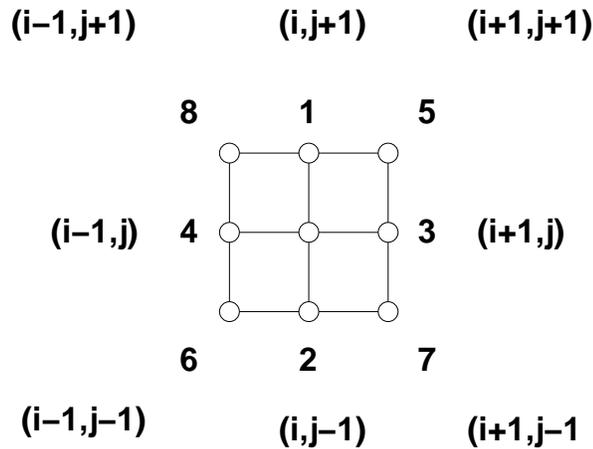


FIGURE 5 – Vecteur d'adressage du noeud  $(k) \iff (i,j)$

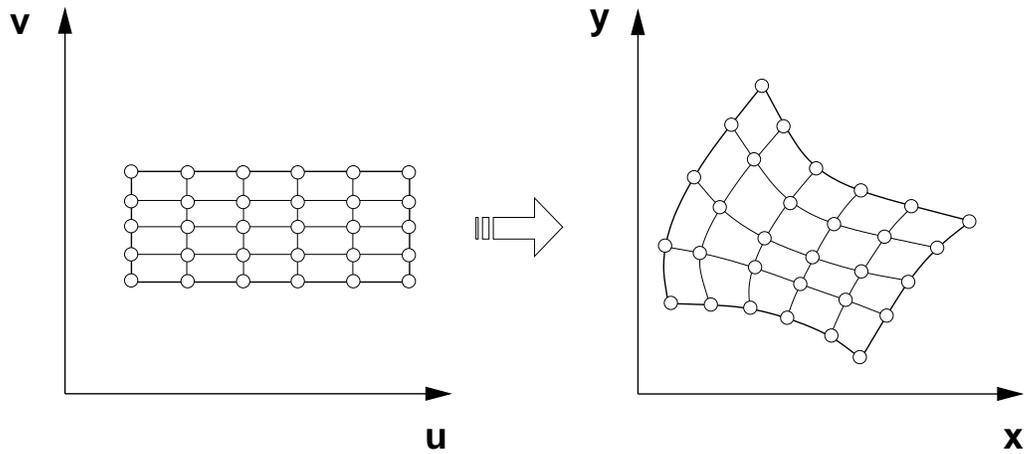
Les pointeurs pour le voisinage du noeud  $(i,j)$  se trouvent dans le vecteur d'adressage au tableau ci-dessous,

$(i,j)$	$k$	$V$
$(i+1,j)$	$ip1j$	$V(k,3)$
$(i-1,j)$	$im1j$	$V(k,4)$
$(i,j+1)$	$ijp1$	$V(k,1)$
$(i,j-1)$	$ijm1$	$V(k,2)$
$(i+1,j+1)$	$ip1jp1$	$V(k,5)$
$(i-1,j-1)$	$im1jm1$	$V(k,6)$
$(i-1,j+1)$	$im1jp1$	$V(k,8)$
$(i+1,j-1)$	$ip1jm1$	$V(k,7)$

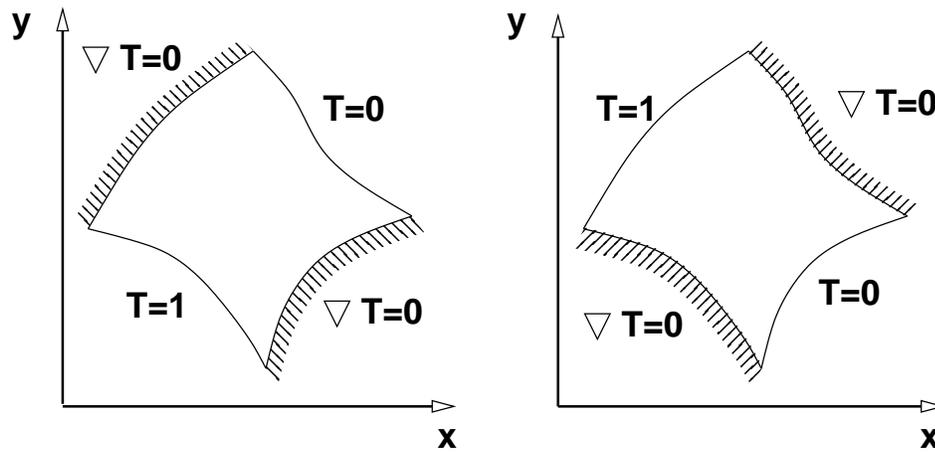
## 6 Modèle de maille

Un maillage est généré dans l'espace paramétrique comme un maillage régulier cartésien, et le maillage dans l'espace physique est obtenu par une transformation de la forme :

$$x_i = x_i(u^1, u^2, u^3)$$



Dans ce travail, on utilise un modèle de maille basé sur des équations différentielles à partir d'une analogie thermique. Soit un domaine borné par quatre cotés : on pose que les lignes curvilignes d'un maillage peuvent être obtenues comme le réseau d'isothermes résultant de la solution d'un problème thermique.



Deux cotés sont posés à un différentiel de température, tandis que les deux autres sont adiabatiques.

Ainsi formulé, le problème de génération de mailles se transforme en un problème différentiel aux valeurs frontières. Formellement, les coordonnées curvilignes  $\xi^i$  où  $\xi^1 = \tau$  et  $\xi^2 = \eta$  sont obtenues en solutionnant le système,

$$\nabla^2 \xi^i = 0 \quad i = 1, 2$$

avec les conditions limites suivantes :

Le long de la frontière :	La variable $\xi^1$	La variable $\xi^2$
$\Gamma_1$ (premier coté)	$\xi^1(x, y) = 0$	$\partial\xi^2(x, y)/\partial n = 0$
$\Gamma_2$ (second coté)	$\xi^1(x, y) = 1$	$\partial\xi^2(x, y)/\partial n = 0$
$\Gamma_3$ (troisième coté)	$\partial\xi^1(x, y)/\partial n = 0$	$\xi^2(x, y) = 0$
$\Gamma_4$ (quatrième coté)	$\partial\xi^1(x, y)/\partial n = 0$	$\xi^2(x, y) = 1$

## 7 Inversion des variables

Le problème est reformulé par l'inversion des variables dépendantes et indépendantes du problème. C'est-à-dire que le problème ne sera pas résolu dans l'espace physique mais plutôt dans l'espace des "températures" ou plus formellement dans l'espace paramétrique. Ce qui revient à chercher le point  $(x, y)$  de l'espace physique correspondant aux deux températures  $(\eta, \tau)$ , plutôt que résoudre les températures  $(\eta, \tau)$  en fonction du point  $(x, y)$ .

Ceci équivaut à un changement de variables de  $\tau = \tau(x, y)$  et  $\eta = \eta(x, y)$  vers  $x = x(\eta, \tau)$  et  $y = y(\eta, \tau)$ .

Pour un problème en dimension 2, on pose  $x_1 = x$  et  $x_2 = y$ , et  $\xi_1 = \eta$  et  $\xi_2 = \tau$

$$\alpha \begin{bmatrix} x_{\eta\eta} \\ y_{\eta\eta} \end{bmatrix} - 2\beta \begin{bmatrix} x_{\tau\eta} \\ y_{\tau\eta} \end{bmatrix} + \gamma \begin{bmatrix} x_{\tau\tau} \\ y_{\tau\tau} \end{bmatrix} = 0$$

où les coefficients :

$$\begin{aligned} \alpha &= J^2(\eta_x^2 + \eta_y^2) &= (x_\tau^2 + y_\tau^2) \\ \beta &= -J^2(\eta_x\tau_x + \eta_y\tau_y) &= (x_\eta x_\tau + y_\eta y_\tau) \\ \gamma &= J^2(\tau_x^2 + \tau_y^2) &= (x_\eta^2 + y_\eta^2) \end{aligned}$$

Ce système peut être exprimé sous la forme d'un opérateur :

$$\mathcal{L}\mathbf{x} = \mathbf{0}$$

où  $\mathcal{L}$  est donné par :

$$\mathcal{L} = \alpha \frac{\partial^2}{\partial \eta^2} - 2\beta \frac{\partial^2}{\partial \eta \partial \tau} + \gamma \frac{\partial^2}{\partial \tau^2}$$

et

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

## 8 Résolution numérique

Les équations de maille forment un système couplé, nonlinéaire qui pour des conditions frontières générales ne peuvent être résolues que par des méthodes numériques. La substitution des expressions aux différences divisées dans les équations de maille,

$$\mathcal{L} = \alpha \frac{\partial^2}{\partial \eta^2} - 2\beta \frac{\partial^2}{\partial \eta \partial \tau} + \gamma \frac{\partial^2}{\partial \tau^2}$$

donne pour un noeud  $(i, j)$  du réseau de l'espace paramétrique une relation algébrique pour les coordonnées  $x$  et  $y$ ,

$$\begin{aligned} \alpha' [x_{i+1,j} - 2x_{i,j} + x_{i-1,j}] + \gamma' [x_{i,j+1} - 2x_{i,j} + x_{i,j-1}] \\ - 2\beta' [x_{i+1,j+1} - x_{i-1,j+1} - x_{i+1,j-1} + x_{i-1,j-1}] = 0 \end{aligned}$$

$$\begin{aligned} \alpha' [y_{i+1,j} - 2y_{i,j} + y_{i-1,j}] + \gamma' [y_{i,j+1} - 2y_{i,j} + y_{i,j-1}] \\ - 2\beta' [y_{i+1,j+1} - y_{i-1,j+1} - y_{i+1,j-1} + y_{i-1,j-1}] = 0 \end{aligned}$$

où

$$\begin{aligned} \alpha' &= \frac{(x_{i,j+1} - x_{i,j-1})^2 + (y_{i,j+1} - y_{i,j-1})^2}{(2\Delta\tau\Delta\eta)^2} \\ \gamma' &= \frac{(x_{i+1,j} - x_{i-1,j})^2 + (y_{i+1,j} - y_{i-1,j})^2}{(2\Delta\tau\Delta\eta)^2} \\ \beta' &= \frac{(x_{i+1,j} - x_{i-1,j})(x_{i,j+1} - x_{i,j-1})}{(4\Delta\tau\Delta\eta)^2} \\ &\quad + \frac{(y_{i+1,j} - y_{i-1,j})(y_{i,j+1} - y_{i,j-1})}{(4\Delta\tau\Delta\eta)^2} \end{aligned}$$

Comme résolveur, on utilisera un schéma de relaxation par point qui consiste à corriger successivement les inconnues par un balayage lexicographique du domaine discret, tel qu'illustré à la Fig. 6.

La fonction itérante est dérivée du système algébrique écrit au noeud  $(i, j)$  à l'étape courante du processus itératif en tenant compte de l'état, c-à-d valeur ancienne ou corrigée, des variables du voisinage. On isole les valeurs des inconnues au noeud  $(i, j)$  et les valeurs aux noeuds voisins sont placées à droite de l'équation. Ce qui donne la fonction itérante, qui exprime  $x_{i,j}$  et  $y_{i,j}$  en fonction des valeurs voisines.

$$\begin{aligned} 2(\alpha' + \gamma')x_{i,j} &= \alpha' (x_{i+1,j} + x_{i-1,j}) + \gamma' (x_{i,j+1} + x_{i,j-1}) \\ &\quad - 2\beta' (x_{i+1,j+1} - x_{i-1,j+1} - x_{i+1,j-1} + x_{i-1,j-1}) \\ 2(\alpha' + \gamma')y_{i,j} &= \alpha' (y_{i+1,j} + y_{i-1,j}) + \gamma' (y_{i,j+1} + y_{i,j-1}) \\ &\quad - 2\beta' (y_{i+1,j+1} - y_{i-1,j+1} - y_{i+1,j-1} + y_{i-1,j-1}) \end{aligned}$$

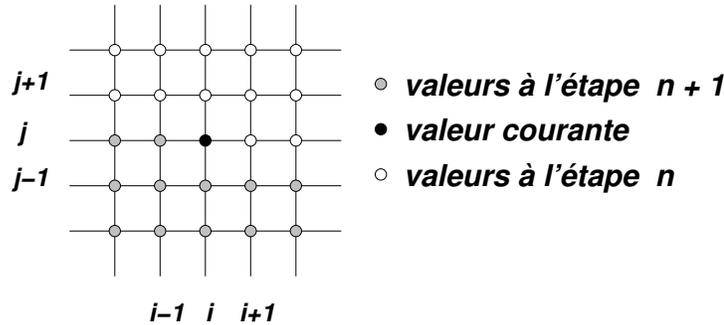


FIGURE 6 – Relaxation par point

Au fur et à mesure des itérations, le membre de droite est mis à jour et le calcul des  $x_{i,j}$  et  $y_{i,j}$  converge pour un ensemble de valeurs  $\alpha'$ ,  $\gamma'$  et  $\beta'$ . Comme ces dernières sont fonction des inconnues  $x_{i,j}$  et  $y_{i,j}$ , on doit recalculer ces coefficients à mesure que les  $x_{i,j}$  et  $y_{i,j}$  changent. Les résidus,

$$\begin{aligned}
 R_x i, j &= \alpha' [x_{i+1,j} - 2x_{i,j} + x_{i-1,j}] + \gamma' [x_{i,j+1} - 2x_{i,j} + x_{i,j-1}] \\
 &\quad - 2\beta' [x_{i+1,j+1} - x_{i-1,j+1} - x_{i+1,j-1} + x_{i-1,j-1}] \\
 R_y i, j &= \alpha' [y_{i+1,j} - 2y_{i,j} + y_{i-1,j}] + \gamma' [y_{i,j+1} - 2y_{i,j} + y_{i,j-1}] \\
 &\quad - 2\beta' [y_{i+1,j+1} - y_{i-1,j+1} - y_{i+1,j-1} + y_{i-1,j-1}]
 \end{aligned}$$

donnent une mesure de l'atteinte de la convergence.

On prendra soin dans l'écriture de la fonction itérante et des résidus de remplacer les  $(i,j)$  par l'indice  $(k)$  tel que montré à la Fig. 5.

## 9 Algorithme global

L'algorithme global comprend plusieurs composantes :

**Le balayage :** Cette étape consiste à visiter chaque noeud (sommet) " $k$ " de l'espace paramétrique, correspondant au sommet " $(i, j)$ " du domaine physique. Un balayage va de  $k = \text{debutFace}$  à  $k = \text{finFace}$  et est répété **iterB** fois.

À chaque sommet on met à jour les valeurs de  $x(k)$  et  $y(k)$  en appliquant la fonction itérante où la correspondance entre  $k$  et  $(i, j)$  est donnée à la Fig. 5. On utilisera une méthode itérative par points de Gauss-Seidel, avec ou sans surrelaxation ;

Au cours d'un balayage, les valeurs des coefficients sont gelés.

**La linéarisation :** À la fin d'un cycle de **iterB** balayages, les valeurs de  $x(k)$  et  $y(k)$  ont changé. Donc, les coefficients qui en dépendent doivent être mis à jour.

Une linéarisation suivie de **iterB** balayages, est une itération, que l'on répète **iterL** fois jusqu'à l'atteinte de la convergence souhaitée.

**La convergence et critère d'arrêt :** On mesure l'erreur par une norme sur les résidus  $R_x(k)$  et  $R_y(k)$ . À cette étape, on calcule les résidus et la norme, et selon une cible, on poursuit ou arrête les calculs. L'historique de la convergence est affiché.

**Le maillage initial** Le maillage initial est obtenu par la méthode d'interpolation transfinie, qui est en fait le point de départ de **Winslow**

## 10 Remise des résultats

1. Ecrire et valider la fonction `winslow.m` sur des configurations simples (ex. un rectangle ..... et progressivement valider sur des configurations plus complexes. La mise en oeuvre et la structure du code devra utiliser les structures de données décrites à la Section 4, avec une fonction itérante du type relaxation par point et une linéarisation des coefficients  $\alpha$ ,  $\beta$  et  $\gamma$ .
2. Ecrire la fonction `structureWinslowConvergence.m` pour afficher les normes des résidus,  $R_x(k)$  et  $R_y(k)$ , et ainsi valider le comportement de la convergence du calcul itératif.
3. Varier le nombre d'itérations *iterB* et *iterL* afin d'établir un lien entre la convergence en fonction de ces paramètres.  
Commenter !
4. Remarques sur la structure de données, le temps de calcul , .complexité .....
5. Dans ce TP, ne pas faire de comparaisons avec la technique transfinie : cette analyse fait l'objet du TP suivant.
6. Peut-on envisager la possibilité d'automatiser le choix de ces paramètres ?
7. Déposer dans Moodle un fichier `identifiantTP3.zip` contenant :  
`winslow.m`  
`winslowConvergence.m`  
`discussion.pdf`