TD/TP : intégration numérique, interpolation, équations différentielles

Modalités des TPs

- Ces TPs sont à réaliser essentiellement en classe. Il est néanmoins fortement conseillé de les préparer à domicile.
- Un compte rendu de ces TPs est à rédiger. Ce compte rendu doit comporter les réponses aux questions, les explications, les résultats numériques et une brève description du travail réalisé exercice par exercice.
- Les programmes sources sous scilab doivent être conservés.
- Le rapport et les programmes sources sont à rendre avant le **31 octobre 2020** par courrier électronique à l'adresse
  `tahar.boulmezaoud@uvsq.fr` et avec comme objet : `COMPTE RENDU TD/TP CHPS+ VOS NOMS.`
- Le courriel doit contenir deux pièces jointes :
  1. le compte rendu en format PDF scanné. Le nom de ce fichier PDF **doit comporter vos noms** afin qu'il soit facilement reconnaissable.
     Les listings des programmes sources doivent être insérés en annexe du compte rendu.
  2. un dossier archivé en **un seul fichier** comportant la totalité des programmes (pour l'archivage, on peut utiliser la commande tar par exemple). Le nom de ce fichier **doit comporter vos noms**.

En raison de la crise de Covid-19, le travail en classe doit être effectué individuellement et non en binôme et la distance physique entre vous doit être impérativement respectée. Néanmoins, dans le cas où vous décidez d'écrire un compte rendu en binôme, vous devez m'informer de la composition du binôme par courriel avant le 16 octobre 2020. Dans ce dernier cas, vous pouvez communiquer entre vous deux en dehors des séances de TD/TP par moyens électroniques (comme Zoom, etc) et à distance (en respectant la distance physique).

Une note globale de TP vous sera attribuée individuellement (les notes de deux membres d'un même binôme peuvent différer). Elle dépendra de votre travail en classe et du contenu du compte rendu.

1 Interpolation et Intégration numérique

Exercice 1 -

1. Ecrire sous Scilab une fonction **pointgauche** ayant comme paramètres d'entrée principaux deux réels `a` et `b`, une fonction `func` définie sur $[a, b]$ à valeurs dans $\mathbb{R}$, un nombre (entier) de pas `N` et qui renvoie en sortie une approximation de l'intégrale de `func` sur l'intervalle $[a, b]$ par la méthode des rectangles à gauche.
2. En modifiant légèrement la fonction `pointgauche`, écrire une nouvelle fonction `trapezes` calculant la meme intégrale avec la méthode des trapèzes.
3. Ecrire une fonction `int_simpson` utilisant la méthode de Simpson pour calculer cette même intégrale.
4. Tester les trois méthodes pour la fonction $x \mapsto \sin(\pi x)$ sur l'intervalle $[0, 1]$. On peut étudier le comportement des erreurs relatives en fonction du pas $h = (b - a)/N$.

Exercice 2 - Trouver un polynôme p de degré inférieur ou égal à 3 tel que

$$p(-3) = 3,\ p(-1) = 7, p(3) = 7, p(5) = -3.$$

Exercice 3 - Soit f la fonction de $\mathbb{R}$ dans $\mathbb{R}$ définie par

$$f(x) = \ln\left(2\cos(\frac{\pi x}{4})^2 + 1\right).$$

1. Calculer $f(-2)$, $f(-1)$, $f(0)$, $f(1)$ et $f(2)$.
2. Trouver, s'il existe, un polynôme p de degré inférieur où égal à 3 tel que

$$p(-2) = f(-2),\ p(-1) = f(-1),\ p(1) = f(1), p(2) = f(2).$$

3. Trouver, s'il existe, un polynôme q de degré inférieur où égal à 4 tel que

$$q(-2) = f(-2),\ q(-1) = f(-1), q(0) = f(0),\ q(1) = f(1),\ q(2) = f(2).$$

(on peut considérer la différence des deux polynômes q et p).

Exercice 4 -

1. Ecrire une fonction polyLag qui ayant comme paramètres d'entrée un réel x et un tableau de réels `xi` et qui renvoie un tableau `Lag` comportant les valeurs $\ell_i(x)$, $1 \leq i \leq n$, où les ℓ_i sont les polynômes d'interpolation de Lagrange associés aux points `xi[i]` et n la taille du tableau `xi` (nombre des points).
2. Ecrire une fonction `myinterpol` qui ayant comme paramètres d'entrée une fonction `func`, un réel `x` et un tableau de réels `xi` et qui renvoie la valeur $p(x)$ du polynôme d'interpolation de la function `func` aux points `xi`$[k]$, $1 \leq k \leq n$, où n est la taille de `xi`.

2 Résolution d'équations différentielles

Exercice 5 - On considère l'équation différentielle ordinaire

$$y'(t) = \frac{1}{2y(t) + 1} \text{ pour } t \geqslant 0, \tag{1}$$

complétée par la condition initiale :

$$y(0) = 0. \tag{2}$$

On admet qu'il existe une solution unique $y : [0, +\infty[\to \mathbb{R}$ et que $y(t) \geqslant 0$ pour tout $t \geqslant 0$.
On aimerait résoudre l'équation (1) avec la condition initiale (2) par un schéma numérique sur un intervalle de temps $[0, T]$. Soit $t_0 = 0 < t_1 < ... < t_N = T$, $N \geqslant 1$, une subdivision uniforme de cet intervalle. On pose $h = T/N$. On a alors $t_k = kh$ pour $0 \leqslant k \leqslant N$. On cherche à approcher chacune des valeurs $y(t_i)$ par une valeur numérique $y_i \geqslant 0$ qu'on calcule selon le schéma utilisé. On peut observer que l'équation (1) est de la forme

$$y'(t) = f(y(t)) \text{ avec } f(x) = \frac{1}{2x + 1}.$$

On rappelle le schéma d'Euler explicite :

$$\begin{cases} y_0 = 0, \\ y_{i+1} = y_i + hf(y_i), \quad \forall i \in \{0, ..., N-1\}. \end{cases} \tag{3}$$

et le schéma d'Euler implicite :

$$\begin{cases} y_0 = 0, \\ y_{i+1} = y_i + hf(y_{i+1}), \quad \forall i \in \{0, ..., N-1\}. \end{cases} \tag{4}$$

Quant au schéma dit de Heun, il s'écrit :

$$\begin{cases} y_0 = 0, \\ y_{i+1} = y_i + \dfrac{h}{2}(f(y_i + hf(y_i)) + f(y_i)), \quad \forall i \in \{0, ..., N-1\}. \end{cases} \tag{5}$$

1. On se place dans le cas d'un schéma d'Euler explicite.
   (a) Trouver explicitement les relations reliant y_{i+1} et y_i, pour $0 \leqslant i \leqslant N-1$.
   (b) Ecrire une fonction `EulerExplicite` ayant comme paramètres d'entrée T et N et qui renvoie les valeurs approchées $y_0, y_1, \cdots, y_N$ calculées par ce schéma d'Euler explicite (stockées dans un tableau).
2. On se place dans le cas d'un schéma de Heun.
   (a) Trouver explicitement les relations reliant y_{i+1} et y_i, pour $0 \leqslant i \leqslant N-1$.
   (b) Ecrire une fonction `Heun` ayant comme paramètres d'entrée T et N et qui renvoie les valeurs approchées $y_0, y_1, \cdots, y_N$ calculées par ce schéma de Heun.
3. On utilise maintenant un schéma d'Euler implicite.
   (a) Montrer que pour chaque $i \leqslant N-1$, y_{i+1} est racine d'un polynôme du second degré dont les coefficients s'expriment en fonction de y_i et h.
   (b) Montrer que le descriminant de ce polynôme est $\Delta = (2y_i + 1)^2 + 8h$.
   (c) En déduire l'expression de y_{i+1} en fonction de y_i et h.
   (d) Ecrire une fonction `EulerImplicite` ayant comme paramètres d'entrée T et N et qui renvoie les valeurs approchées $y_0, y_1, \cdots, y_N$ calculées par ce schéma de Heun.
4. On choisit $T = 0.88$ et $N = 16$ et on affiche
   — les instants t_i, $0 \leqslant i \leqslant 16$, dans la colonne `C0`,
   — les valeurs de la solution exacte dans la colonne `C1` arrondies à 8 décimales,
   — les valeurs des solutions approchées par les trois schémas dans les trois colonnes `C2`, `C3` et `C4`, mais on ne précise pas dans quel ordre.

C0	C1	C2	C3	C4
0.	0.	0.	0.	0.
0.055	0.0522681	0.05	0.0522748	0.055
0.11	0.1	0.0961308	0.1000097	0.1045495
0.165	0.1442049	0.1391563	0.1442160	0.1500380
0.22	0.1855655	0.1796201	0.1855772	0.1923432
0.275	0.2245688	0.2179249	0.2245808	0.2320634
0.33	0.2615773	0.2543787	0.2615893	0.2696284
0.385	0.2968689	0.2892231	0.2968807	0.3053600
0.44	0.3306624	0.3226516	0.3306740	0.3395062
0.495	0.3631338	0.3548221	0.3631453	0.3722635
0.55	0.3944272	0.3858652	0.3944384	0.4037907
0.605	0.4246621	0.4158906	0.4246731	0.4342181
0.66	0.4539392	0.4449914	0.4539500	0.4636545
0.715	0.4823441	0.4732473	0.4823547	0.4921917
0.77	0.5099505	0.5007273	0.5099608	0.5199081
0.825	0.5368221	0.5274915	0.5368322	0.5468713
0.88	0.5630146	0.5535927	0.5630245	0.5731401

   (a) Préciser pour chacune des colonnes `C2`, `C3` et `C4` à quelle méthode elle correspond (Euler implicite, Euler Explicite ou Heun). On justifiera la réponse.
   (b) Retrouver ces résultats numériquement avec les fonctions écrites précédemment.
   (c) Commenter ces résultats numériques en comparant la convergence des trois méthodes.

Exercice 6 - On étudie ici l'équation différentielle :

$$-u''(t) + c(t)u(t) = f(t), \quad t \in [0,1], \tag{E}$$

où c et f sont deux fonctions de $[0,1]$ dans $\mathbb{R}$ continues et données, avec $c \geqslant 0$. Pour les tests numériques on pourra choisir $c = 0$ ou $c(x) = 2 - \cos(x)$ pour tout $x \in [0,1]$ (par exemple).

Partie I

Dans cette partie, l'équation (E) est complétée avec les conditions initiales

$$u(0) = \alpha,\ u'(0) = \beta, \tag{I}$$

où α et β sont deux réels donnés. On pose pour tout $t \geqslant 0$,

$$U(t) = \begin{pmatrix} u(t) \\ u'(t) \end{pmatrix}.$$

1. Montrer que u si est solution de (E) et de (C) alors U vérifie

$$U'(t) = A(t)U(t) + B(t), \tag{E'}$$

où

$$A(t) = \begin{pmatrix} 0 & 1 \\ c(t) & 0 \end{pmatrix},\ B(t) = \begin{pmatrix} 0 \\ -f(t) \end{pmatrix},\ \ t \in I.$$

et $U(0) = (\alpha \quad \beta)^T$.

2. Soit $t_0 = 0 < t_1 < ... < t_N = 1$, $N \geqslant 1$, une subdivision uniforme de l'intervalle $[0,1]$.
On pose $h = 1/N$. Ainsi, $t_i = ih$ pour $0 \leqslant i \leqslant N$. On cherche à approcher chaque $U(t_i)$, $0 \leqslant i \leqslant N$, par un vecteur $U_i = (u_i,\ \ w_i)^T$ avec, pour $i = 0$, $U_0 = (\alpha,\ \ \beta)^T$. On utilise un schéma d'Euler explicite appliqué à l'équation (E').
   (a) Ecrire la relation reliant U_{i+1} à U_i selon ce schéma, pour $0 \leqslant i \leqslant N-1$.
   (b) En déduire que

$$\begin{cases} -\dfrac{1}{h^2}(u_{i+1} + u_{i-1} - 2u_i) + c(t_i)u_i = f(t_i), \ \ \text{pour } 1 \leqslant i \leqslant N-1, \\ u_0 = \alpha,\ u_1 = u_0 + h\beta. \end{cases} \tag{E_h}$$

   (c) Ecrire une fonction `resouScd` renvoyant les valeurs $u_0, u_1, \cdots, u_N$ et ayant les paramètres d'entrée suivants :
   - `N` : le nombre de pas dans $[0,1]$,
   - `alpha`, `beta` : les paramètres α et β,
   - `cofc` : une fonction représentant le coefficient c,
   - `foncf` : une fonction représentant la fonction f.

Partie II

Dans cette seconde partie, on complète l'équation (E) avec les conditions aux limites

$$u(0) = 0,\ u(1) = 0, \tag{C}$$

(qui remplacent donc les conditions initiales (I)). Pour tout i, $1 \leqslant i \leqslant N-1$, on approche $u(t_i)$ par une valeur notée encore u_i avec $u_1, \cdots, u_{N-1}$ solution du problème approché

$$-\frac{1}{h^2}(u_{i+1} + u_{i-1} - 2u_i) + c(t_i)u_i = f(t_i), \ \ \text{pour } 1 \leqslant i \leqslant N-1, \tag{E'_h}$$

et $u_0 = 0,\ u_N = 0$.

1. On pose $V = (u_1, \cdots, u_{N-1})^T$ (vecteur colonne). Montrer que V est solution d'un système linéaire $AV = B$ où A est une matrice tridiagonale qu'on donnera explicitement et B un second membre à expliciter aussi.
2. Soit $W = (w_1, \cdots, w_{N-1})^t$. Montrer que
$$h^2 W^T A W = \sum_{i=1}^{N-1} (2 + h^2 c(ih)) w_i^2 - 2 \sum_{i=2}^{N} w_{i-1} w_i.$$
où on a posé par commodité $w_0 = w_N = 0$.
3. En déduire que A est symétrique définie positive.
4. Programmer cette méthode pour résoudre (E'_h) (après avoir choisi une méthode d'inversion du système linéaire).
5. Effectuer quelques tests numériques avec des solutions exactes connues à l'avance.

Partie III

Dans cette dernière partie, on s'intérèsse à la méthode dite des éléments finis. Dans ce cadre, on veut approcher la solution de le système (E)+(C) par une fonction u_h appartenant à l'espace
$$V_h = \{v_h \in C^0([0,1]) \ | \ v_h(0) = v_h(1) = 0, \ v_{h|[t_k, t_{k+1}]} \in \mathbb{P}_1, \ k = 0, ..., N-1\},$$
où $\mathbb{P}_1$ désigne l'espace des fonctions affines.

1. Ecrire une formulation variationnelle du problème (E)+(C).
2. Ecrire le problème approché (forme de Galerkin) dans V_h.
3. Exhiber une base $\{\omega_k, \ 1 \leqslant k \leqslant N-1\}$ de V_h telle
$$\omega_k(t_j) = \delta_{k,j}, \forall 1 \leqslant k, \ j \leqslant N-1,$$
(on illustrera la courbe de l'une de ces fonctions).
4. En utilisant cette base, montrer que le problème approché se ramène à un système linéaire de la forme $AX = B$, où B est un vecteur colonne donné que vous devez préciser, et $A = (a_{i,j})_{1 \leqslant i,j \leqslant N-1}$ la matrice carrée de coefficient général
$$a_{i,j} = \int_0^1 \omega_i'(x) \omega_j'(x) dx + \int_0^1 c(x) \omega_i(x) \omega_j(x) dx.$$
5. Que peut-on dire de $a_{i,j}$ quand $|i - j| > 1$? justifier.
6. Montrer que A est symétrique définie positive.
7. Ecrire un programme qui approche la solution de (E)+(C) par cette méthode (après avoir proposé une méthode d'inversion).
8. Concevoir un exemple pour tester la méthode des éléments finis numériquement. Calculer l'erreur d'approximation $\max_{0 \leqslant i \leqslant N} |u(x_i) - u_h(x_i)|$ et étudier son comportement en fonction de h.

Exercice 7 - On considère une équation différentielle non linéaire de la forme
$$-u''(x) + f(u(x)) = 0 \text{ pour } x \in [0, \ell], \ u(0) = 0, \ u(\ell) = 1,$$
où $\ell > 0$ désigne un réel fixé et f est une fonction définie sur $[0,1]$ qu'on supposera continue et vérifiant $f(1) \geqslant 0$. On supposera de plus que $t \mapsto f(t)/t$ est strictement croissante sur $]0,1]$. On admettra que ce problème possède une solution unique u de classe $\mathscr{C}^2$ sur $[0,1]$ qu'on cherche à approcher par le schéma de différences finies suivant
$$-\frac{u_{k+1} + u_{k-1} - 2u_k}{h^2} + f(u_k) = 0, \ 1 \leqslant k \leqslant N-1, \ u_0 = 0, \ u_N = 1. \tag{6}$$
Ici N désigne un entier naturel non nul, $h = \ell/N$ et $x_k = kh$ pour $0 \leqslant k \leqslant N$. Ainsi, u_k, $0 \leqslant k \leqslant N$, est une approximation de $u(x_k)$. On propose de résoudre ce système discret en utilisant l'algorithme suivant :

— On fixe $u_k^{(0)} = 1$ pour $0 \leqslant k \leqslant N$.

— Pour $n \geqslant 0$, $u_k^{(n+1)}$ est solution du système linéaire

$$-\frac{u_{k+1}^{(n+1)} + u_{k-1}^{(n+1)} - 2u_k^{(n+1)}}{h^2} + Ku_k^{(n+1)} = Ku_k^{(n)} - f(u_k^{(n)}),\ 1 \leqslant k \leqslant N-1,$$

$$u_0^{(n+1)} = 0,\ u_N^{(n+1)} = 1.$$

Ici $K > 0$ désigne un paramètre réel fixé.
Implémenter ce schéma (avec un test d'arrêt pour la convergence).
Tester la méthode dans les cas suivants (visualisation des solutions exacte et approchée, estimation de l'erreur aux noeuds) :

— $f(t) = 2(t^3 - t)$ et $u(x) = \tanh(x)$,

— $f(t) = -\frac{1}{\pi}\sin(\pi t)$ et $u(x) = \frac{2}{\pi}\arcsin(\tanh(x))$.

On choisira plusieurs valeurs de ℓ et de N.

3 Résolution d'une équation aux dérivées partielles instationnaire

Exercice 8 - On considère l'équation d'évolution sur un intervalle $[a, b]$ $(b > a)$:

$$\frac{\partial u}{\partial t}(t,x) - \lambda\frac{\partial^2 u}{\partial x^2}(t,x) = 0,\ t > 0,\ x \in]a,b[,$$

complétée avec la condition initiale à $t = 0$

$$u(0,x) = v(x),\ x \in]a,b[,$$

et les conditions au bord

$$u(t,a) = \alpha,\ u(t,b) = \beta,$$

où α, β et $\lambda > 0$ sont des constantes réels et v une fonction donnée sur $]a,b[$ (donnée initiale).

On se propose de résoudre numériquement cette équation pour $0 \leqslant t \leqslant T$ avec $T > 0$ donné. Pour ce faire, on subdivise l'intervalle $[a,b]$ en M petits intervalles $[x_k, x_{k+1}]$, $0 \leqslant k \leqslant M-1$, de longueur $h = (b-a)/M$, où $M \geqslant 1$ est un entier et

$$x_k = a + kh \text{ pour } 0 \leqslant k \leqslant M.$$

On découpe aussi l'intervalle de temps $[0,T]$ en N petits intervalles, $N \geqslant 1$, de longueur $\Delta t = T/N$ où $N \geqslant 1$ est un entier et

$$t_n = n\Delta t \text{ pour } 0 \leqslant n \leqslant N.$$

On aimerait construire ensuite une approximation $u_k^{(n)}$ de $u(t_n, x_k)$ pour $0 \leqslant k \leqslant M$ et $0 \leqslant n \leqslant N$.

1. Une première méthode pour approcher u consiste à considérer le schéma :

$$\begin{cases} \dfrac{u_k^{(n+1)} - u_k^{(n)}}{\Delta t} - \lambda\dfrac{u_{k+1}^{(n)} + u_{k-1}^{(n)} - 2u_k^{(n)}}{h^2} = 0, \text{ pour } 1 \leqslant k \leqslant M-1 \text{ et } 0 \leqslant n \leqslant N-1, \\ u_0^{(n+1)} = \alpha,\ u_M^{(n+1)} = \beta, \\ u_0^{(0)} = \alpha,\ u_M^{(0)} = \beta,\ u_k^{(0)} = v(x_k) \text{ pour } 1 \leqslant k \leqslant M-1. \end{cases}$$

   (a) Expliquer brièvement les idées essentielles de la construction de ce schéma.

   (b) Compléter la fonction `SolveEquation` (voir feuille ci-jointe) afin qu'elle renvoie en sortie un tableau bidimensionnel `res` de taille $(N+1) \times (M+1)$ où pour $1 \leqslant i \leqslant N+1$ et $1 \leqslant k \leqslant M+1$, `res(i, k)` contient la valeur $u_{k-1}^{(i-1)}$ calculée avec le schéma ci-dessus (le décalage de 1 dans l'indice est dû au fait que les indices dans les tableaux du programme commencent par 1 et non par 0).
   On notera que les paramètres d'entrée `lambda, a, b, M, Tf, N, alpha, beta, vinit` représentent respectivement les paramètres λ, a, b, M, T, N, α, β et la fonction v.

2. Une deuxième méthode pour approcher u consiste à considérer le schéma :

$$\begin{cases} \dfrac{u_k^{(n+1)} - u_k^{(n)}}{\Delta t} - \lambda \dfrac{u_{k+1}^{(n+1)} + u_{k-1}^{(n+1)} - 2u_k^{(n+1)}}{h^2} = 0, \text{ pour } 1 \leqslant k \leqslant M-1 \text{ et } 0 \leqslant n \leqslant N-1, \\ u_0^{(n+1)} = \alpha,\ u_M^{(n+1)} = \beta, \\ u_0^{(0)} = \alpha,\ u_M^{(0)} = \beta,\ u_k^{(0)} = v(x_k) \text{ pour } 1 \leqslant k \leqslant M-1. \end{cases}$$

   (a) On pose $U^{(n)} = (u_1^{(n)}, \cdots, u_{M-1}^{(n)})^t$ pour $0 \leq n \leq N$. Montrer que ce schéma s'écrit sous la forme :

$$AU^{(n+1)} = U^{(n)} + B, 0 \leqslant n \leqslant N-1,\ U^{(0)} \text{ donné},$$

   où A est une matrice carrée et B un vecteur colonne. On précisera les coefficients de A et de B en fonction de α, β et de

$$\tau = \frac{\lambda \Delta t}{h^2}.$$

   (b) Ecrire sous Scilab une fonction `ResouEquation` analogue à `SolveEquation` et qui renvoie en sortie un tableau `res` de taille $(N+1)\times(M+1)$ où pour $1 \leqslant i \leqslant N+1$ et $1 \leqslant k \leqslant M+1$, `res(i, k)` contient la valeur $u_{k-1}^{(i-1)}$ calculée avec ce dernier schéma.

3. Effectuer des tests numériques de résolution de l'équation d'évolution de départ avec les deux fonctions `SolveEquation` et `ResouEquation` dans le cas suivant : $a = 0$, $b = \pi$, $\alpha = \beta = 0$ et

$$v(x) = \begin{cases} \dfrac{12}{5}\dfrac{x}{b} & \text{si } 0 \leq x \leq \frac{b}{2}, \\ \dfrac{12}{5}\dfrac{b-x}{b} & \text{si } \frac{b}{2} \leq x \leq b. \end{cases}$$

   On visualise $U^{(n)}$ pour $n = 0,\ 5,\ 10,\ 20$, *etc.* On choisira $\lambda = 1$, $M = 100$ et Δt tel que $\tau = 0.4$, puis tel que $\tau = 0.6$. On commentera les résultats numériques.

NOM ET PRENOM :

```
/ /*****************************************************************

   function [res] = detTridiag(va, vb, vc)

    n = max(size(va)) ; // taille du vecteur diagonal (size of the diagonal vector)
    x = 1;
    y = va(1);

    for i = 2:n
      z = y;
      y = _____________________________________________ ;
      x = _____ ;
    end;

    res = y;
   endfunction

//*****************************************************************

   function [res] = SolveEquation(lambda, a, b, M, Tf, N, alpha, beta, vinit)

    dx = (b-a)/M;
    dt = Tf/N;
    tau = dt*lambda/(dx*dx);

    // La solution a t=0
    // (the solution at t=0)
    res(1,1) = alpha;
    res(1, M+1) = beta;
    for k = 1: _____ ;
      xk = _____________________ ;
      res(1, k+1) = vinit(xk)
    end

    // Calcul de la solution a t = t_i, 1 <= i <= N
    // (computing the solution for t=t_i, 1 <= i <= N)

    for i = 1:N
      res(i+1,1) = ___________;
      res(i+1, M+1) = ___________;
      for k = 1:M-1
        res(i+1, k+1) = ____________________________________________________________
;
      end
    end

   endfunction
```