MT09-A2016 – Examen final – Questions de cours

Durée : 30mn. Sans documents ni outils électroniques – Rédiger sur l'énoncé

NOM PRÉNOM : Place n° :

ATTENTION, il y a 3 exercices indépendants pour cette partie questions de cours !

Exercice 1 *(barème approximatif : 2 points)*

Soit A une matrice réelle de $\mathcal{M}_{mn}$ avec $m > n$ de rang n. On suppose donnée la factorisation $A = QR$.

1. Donner les dimensions et les propriétés de Q et R.

   Réponse : cf. cours. Comme A est de rang n, R est de rang n donc $\widetilde{R}$ est inversible. ☐

2. Écrire les équations normales et donner un système triangulaire équivalent (on montrera cette équivalence).

   Réponse : cf. cours et TD. On pose $Q^T b = [c, d]^T$, où $c \in \mathbb{R}^n$.

$$\begin{aligned} A^T Ax = A^T b &\iff R^T Q^T QRx = R^T Qb \\ &\iff [\widetilde{R}^T 0] \begin{bmatrix} \widetilde{R} \\ 0 \end{bmatrix} x = [\widetilde{R}^T 0] \begin{bmatrix} c \\ d \end{bmatrix} \\ &\iff \widetilde{R}^T \widetilde{R}x = \widetilde{R}^T c \\ &\iff \widetilde{R}x = c, \end{aligned}$$

   car $Q^T Q = I_m$ et car $\widetilde{R}$ est inversible (carrée triangulaire supérieure de rang n, les r_{ii} sont non nuls), et donc $\widetilde{R}^T$ aussi. ☐

3. Calculer la norme 2 de $A^T A$ en fonction de la norme 2 d'une matrice issue de R, notée $\hat{R}$ que l'on déterminera.

   Réponse : cf. cours et TD. On reprend les manipulations algébriques de la question précédente : $A^T A = \widetilde{R}^T \widetilde{R}$. Comme $A^T A$ est symétrique :

$$\|A^T A\|_2 = \rho(A^T A) = \rho(\widetilde{R}^T \widetilde{R}) = \|\widetilde{R}\|_2^2$$

   d'après la définition de la norme subordonnée à la norme 2. ☐

4. Même question pour la norme 2 de $(A^T A)^{-1}$.

   Réponse : on a : $(A^T A)^{-1} = (\widetilde{R}^T \widetilde{R})^{-1} = \widetilde{R}^{-1} \widetilde{R}^{-T}$ ($\widetilde{R}$ est bien inversible (et carrée! ce qui n'est pas vrai pour A, ni pour R!!). Comme $(A^T A)^{-1}$ est symétrique

$$\|(A^T A)^{-1}\|_2 = \rho((A^T A)^{-1}) = \rho(\widetilde{R}^{-1} \widetilde{R}^{-T}) = \rho(\widetilde{R}^{-T} \widetilde{R}^{-1}) = \|\widetilde{R}^{-1}\|_2^2$$

   car $\rho(AB) = \rho(BA)$ quand ces produits ont un sens, *cf.* TD. ☐

5. En déduire une relation entre le conditionnement de $A^T A$ et celui de $\hat{R}$.

   Réponse : on a :

$$\chi_2(A^T A) = \|A^T A\|_2 \|(A^T A)^{-1}\|_2 = \|\widetilde{R}\|_2^2 \|\widetilde{R}^{-1}\|_2^2 = (\chi_2(\widetilde{R}))^2.$$

☐

Exercice 2 *(barème approximatif : 2 points)*

Soit E un espace euclidien de dimension $n > 0$. On appelle $< u, v >$ son produit scalaire et $\|u\| = \sqrt{< u, u >}$ sa norme, $\forall u, v \in E$. Soit F un sous-espace de E de dimension $k \leq n$. Soit $y \in E$ un vecteur donné.

Soit p_y le projeté orthogonal de y dans F.

1. Écrire les conditions vérifiées par p_y.
2. Montrer que $\|f - y\|^2 > \|p_y - y\|^2$, $\forall f \in F$, $f \neq p_y$.
3. On prend une base $\mathcal{F} = (f_1, f_2, \ldots, f_k)$ de F. Écrire le système que vérifient les composantes de p_y dans la base $\mathcal{F}$.

Exercice 3 *(barème approximatif : 2 points)*

Soit $h > 0$. On pose $t_0 = -2h$, $t_1 = 0$ et $t_2 = h$.

1. Écrire les polynômes de la base de Lagrange associée à $t_0,\ t_1,\ t_2$.
2. Soit une fonction continue $f : \mathbb{R} \longrightarrow \mathbb{R}$. Écrire le polynôme interpolant f aux points t_i, $i = 0, 1, 2$.
3. Soit $\theta_0 \in \mathbb{R}$. On prend maintenant les points $\theta_i = \theta_0 + ih$, pour $i = 0, \ldots, N$. Étant donnés $(f_i)_{i=0,\ldots,N}$ et $(c_i)_{i=0,\ldots,N-1}$, on définit les polynômes :

$$p_i(t) = f_i + \frac{f_{i+1} - f_i}{h}(t - \theta_i) + \frac{c_i}{h^2}(t - \theta_i)(t - \theta_{i+1}).$$

   On définit la fonction g par : $g(t) = p_i(t) \quad$ si $t \in [\theta_i, \theta_{i+1}[$.

   Montrer que g est continue sur $[\theta_0, \theta_N[$.

MT09-A2016- Examen final
Durée : 1h30.
Polycopiés de cours et scilab autorisés - pas d'outils numériques

Questions de cours déjà traitées : environ *6 points.*
Il est possible de traiter une question en admettant les résultats précédents.

Exercice 1 : *(barème approximatif : 12 points)* **CHANGEZ DE COPIE**

Les questions **??***),* **??***) et* **??***) sont partiellement indépendantes des précédentes.*

Soient deux réels t_0 et $T > 0$ et un entier $N > 0$. On introduit le pas $h = T/N$ et les points $t_n = t_0 + nh$ pour $n = 0, \ldots, N$.

1. Soit y une fonction de classe C^1 de $[t_0, t_0 + T]$ dans $\mathbb{R}$. On notera $y_n = y(t_n)$ la valeur de y en t_n, pour $n = 0, \ldots, N$. On appelle p_y le polynôme interpolant la fonction y aux points t_{n+1}, t_n, t_{n-1}.

   (a) Écrire p_y dans la base de Newton associée à t_{n+1}, t_n, t_{n-1} (dans cet ordre).
   Réponse : Le polynôme d'interpolation est dans $\mathcal{P}_2$, et s'écrit dans la base de Newton

$$p_y(t) = c_0 + c_1(t - t_{n+1}) + c_2(t - t_{n+1})(t - t_n),$$

   où les coefficients sont donnés par les différences divisées

$$c_0 = y[t_{n+1}] = y_{n+1},\ c_1 = y[t_{n+1}, t_n] = \frac{y_{n+1} - y_n}{h},\ c_0 = y[t_{n+1}, t_n, t_{n-1}] = \frac{y_{n+1} - 2y_n + y_{n-1}}{2h^2}.$$

   Remarque : dans la base de Lagrange :

$$p_y(t) = y_{n+1}\frac{(t - t_{n-1})(t - t_n)}{2h^2} + y_n\frac{(t - t_{n-1})(t - t_{n+1})}{-h^2} + y_{n-1}\frac{(t - t_n)(t - t_{n+1})}{2h^2}.$$

□

   (b) Calculer $p'_y(t_{n+1})$ en fonction de y_{n+1}, y_n, y_{n-1} et de h uniquement.
   Réponse : on trouve

$$p'_y(t) = c_1 + c_2\left((t - t_{n+1}) + (t - t_n)\right),$$

   donc

$$p'_y(t_{n+1}) = \frac{y_{n+1} - y_n}{h} + \frac{y_{n+1} - 2y_n + y_{n-1}}{2h} = \frac{1}{h}(\frac{3}{2}y_{n+1} - 2y_n + \frac{1}{2}y_{n-1}).$$

   Remarque : dans la base de Lagrange :

$$\begin{aligned} p'_y(t) &= y_{n+1}\frac{(t - t_{n-1}) + (t - t_n)}{2h^2} + y_n\frac{(t - t_{n-1}) + (t - t_{n+1})}{-h^2} + y_{n-1}\frac{(t - t_n) + (t - t_{n+1})}{2h^2} \quad \textbf{donc :} \\ p'_y(t_{n+1}) &= y_{n+1}\frac{3}{2h} - y_n\frac{2}{h} + y_{n-1}\frac{1}{2h}. \end{aligned}$$

□

2. Soit f une fonction de classe C^1 : $(t, y) \in [t_0, t_0 + T] \times \mathbb{R} \mapsto f(t, y) \in \mathbb{R}$. On cherche à résoudre numériquement l'équation différentielle suivante :

$$\text{trouver } y \in C^1([t_0, t_0 + T], \mathbb{R}),\ \text{telle que} \begin{cases} y'(t) = f(t, y(t)) \text{ pour } t \in [t_0, t_0 + T], \\ y(t_0) = y_0. \end{cases} \quad (1)$$

   Pour cela, on va approcher $y'(t_{n+1})$ par $p'_y(t_{n+1})$.

   (a) Écrire la relation $p'_y(t_{n+1}) \approx y'(t_{n+1})$ et en déduire un schéma du type :

$$z_{n+1} = \alpha_0 z_n + \alpha_1 z_{n-1} + h\beta f(t_{n+1}, z_{n+1}), \quad (2)$$

   pour résoudre le problème (**??**). Déterminer les coefficients α_0, α_1 et β.

Réponse : l'équation approchée

$$f(t_{n+1}, y(t_{n+1})) = y'(t_{n+1}) \approx p_y'(t_{n+1}) = \frac{1}{h}(\frac{3}{2}y_{n+1} - 2y_n + \frac{1}{2}y_{n-1}),$$

suggère le schéma

$$\begin{array}{rcl} \frac{3}{2}z_{n+1} & = & 2z_n - \frac{1}{2}z_{n-1} + hf(t_{n+1}, z_{n+1}), \quad n \geq 1, \\ \iff \quad z_{n+1} & = & \frac{4}{3}z_n - \frac{1}{3}z_{n-1} + h\frac{2}{3}f(t_{n+1}, z_{n+1}), \quad n \geq 1, \end{array}$$

soit $\alpha_0 = \frac{4}{3}$, $\alpha_1 = -\frac{1}{3}$ **et** $\beta = \frac{2}{3}$**.** □

(b) Ce schéma est-il implicite ou explicite? Est-il à un pas ou multi-pas? Expliquer comment déterminer z_1.

Réponse : c'est un schéma implicite à 2 pas (appelé BDF). Pour déterminer z_1**, il faut utiliser un schéma à 1 pas du même ordre que BDF (c'est-à-dire** 2**,** *cf.* **plus bas).** □

3. On cherche à calculer l'ordre du schéma (**??**), pour des valeurs α_0, α_1 et β quelconques. On suppose que y est de classe au moins C^4.

(a) Écrire un développement de Taylor de $y(t_{n+1})$ autour de t_n à l'ordre 3 en h.

(b) Écrire un développement limité de $y(t_{n-1})$ autour de t_n à l'ordre 3 en h.

(c) Écrire un développement limité de $y'(t_{n+1})$ autour de t_n à l'ordre 2 en h.

Réponse : il existe $\xi_1 \in]t_n, t_{n+1}[$**,** $\xi_2 \in]t_{n-1}, t_n[$ **et** $\xi_3 \in]t_n, t_{n+1}[$ **tels que**

$$\begin{array}{rcl} y(t_{n+1}) & = & y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + \frac{h^3}{6}y'''(t_n) + \frac{h^4}{24}y^{(4)}(\xi_1) \\ y(t_{n-1}) & = & y(t_n) - hy'(t_n) + \frac{h^2}{2}y''(t_n) - \frac{h^3}{6}y'''(t_n) + \frac{h^4}{24}y^{(4)}(\xi_2) \\ y'(t_{n+1}) & = & y'(t_n) + hy''(t_n) + \frac{h^2}{2}y'''(t_n) - \frac{h^3}{6}y^{(4)}(\xi_3) \quad \iff \\ hy'(t_{n+1}) & = & hy'(t_n) + h^2y''(t_n) + \frac{h^3}{2}y'''(t_n) - \frac{h^4}{6}y^{(4)}(\xi_3). \end{array}$$

□

(d) Calculer l'erreur de troncature locale et en déduire des relations entre α_0, α_1 et β pour que le schéma (**??**) soit au moins d'ordre 2. Les valeurs trouvées à la question **??**) conviennent-elles?

Réponse : l'erreur de troncature locale est définie par $\tau_{n+1}(h) = y(t_{n+1}) - \tilde{z}_{n+1}$**, où** y **est la solution du problème (??) et** $\tilde{z}_{n+1}$ **est le résultat du schéma (??) en partant de la solution (**$z_n = y(t_n)$ **et** $z_{n-1} = y(t_{n-1})$**).**

On obtient (on va un cran plus loin que demandé pour s'assurer que l'ordre est exactement 2):

$$\begin{array}{rcl} \tau_{n+1}(h) & = & y(t_{n+1}) - \alpha_0 y(t_n) - \alpha_1 y(t_{n-1}) + h\beta y'(t_{n+1}) \\ & = & y(t_n)(1 - \alpha_0 - \alpha_1) \\ & & +hy'(t_n)(1 + \alpha_1 - \beta) \\ & & +\frac{h^2}{2}y''(t_n)(1 - \alpha_1 - 2\beta) \\ & & +\frac{h^3}{6}y'''(t_n)(1 + \alpha_1 - 3\beta) \\ & & +\frac{h^4}{24}(\ y^{(4)}(\xi_1) - \alpha_1 y^{(4)}(\xi_2) - 4\beta y^{(4)}(\xi_3)\ ). \end{array}$$

Donc en résolvant le système linéaire

$$\left\{ \begin{array}{ccccc} \alpha_0 & +\alpha_1 & & = & 1 \\ & -\alpha_1 & +\beta & = & 1 \\ & \alpha_1 & +2\beta & = & 1 \end{array} \right. ,$$

on annule les termes en $O(1)$**, en** $O(h)$ **et en** $O(h^2)$**. On trouve** $\alpha_0 = \frac{4}{3}$, $\alpha_1 = -\frac{1}{3}$ **et** $\beta = \frac{2}{3}$**. Le coefficient devant terme en** $O(h^3)$ **est alors** $1 + \alpha_1 - 3\beta = -\frac{4}{3} \neq 0$**.**

L'erreur s'écrit dans ce cas

$$\tau_{n+1}(h) \quad = \quad -\frac{2h^3}{9}y'''(t_n) + O(h^4) = -\frac{2h^3}{9}y'''(t_n)(1+O(h)),$$

donc comme $O(h)$ tend vers 0 quand h tend vers 0, il existe $h_0 > 0$ et $K > 0$ tel que pour tout $0 < h < h_0$

$$\frac{|\tau_{n+1}(h)|}{h} \leq Kh^2 \max_{t\in[t_0,t_0+T]} |y'''(t)|,$$

donc le schéma est d'ordre 2. □

4. Soit $\omega > 0$. On définit une suite réelle par la récurrence

$$(3+2\omega)u_{n+1} - 4u_n + u_{n-1} = 0, \quad n \geq 1. \tag{3}$$

(a) On cherche une solution à la suite sous la forme $u_n = Kr^n$. Écrire l'équation caractéristique de (**??**).
Réponse : on remplace u_n par Kr^n dans (??), en supposant que K et r sont non-nuls (sinon la suite est constante et égale à 0). On obtient l'équation de degré 2

$$(3+2\omega)r^2 - 4r + 1 = 0.$$

□

(b) Montrer que la suite converge vers 0. On distinguera les cas où les racines de l'équation caractéristique sont réelles ou complexes.
Réponse : Le discriminant vaut $\Delta = 2^2(4-(3+2\omega)) = 2^2(1-2\omega)$.
Il est positif quand $\omega \leq \frac{1}{2}$. Dans ce cas,

$$r_+ = \frac{2+\sqrt{1-2\omega}}{3+2\omega}, \quad r_- = \frac{2-\sqrt{1-2\omega}}{3+2\omega}.$$

On montre que $-1 < r_- < r_+ < 1$:

$$\begin{aligned} r_+ < 1 \quad &\Longleftrightarrow \quad 2+\sqrt{1-2\omega} < 3+2\omega \\ &\Longleftrightarrow \quad \sqrt{1-2\omega} < 1+2\omega \\ &\Longleftrightarrow \quad 1-2\omega < 1+4\omega+4\omega^2 \qquad \textbf{(car } 1-2\omega > 0) \\ &\Longleftrightarrow \quad 0 < 6\omega + 4\omega^2 \end{aligned}$$

ce qui est toujours vrai pour $\omega > 0$. De plus :

$$\begin{aligned} r_- > -1 \quad &\Longleftrightarrow \quad 2-\sqrt{1-2\omega} > -3-2\omega \\ &\Longleftrightarrow \quad \sqrt{1-2\omega} < 5+2\omega \\ &\Longleftrightarrow \quad 1-2\omega < 25+20\omega+4\omega^2 \qquad \textbf{(car } 1-2\omega > 0) \\ &\Longleftrightarrow \quad 0 < 24+2\omega+4\omega^2 \end{aligned}$$

ce qui est toujours vrai pour $\omega > 0$. Donc si $\Delta \geq 0$, les racines sont toujours dans $]-1,1[$ donc la suite u_n (dont la solution est $u_n = K_+r_+^n + K_-r_-^n$, ou $(K+nL)r^n$ si $\Delta = 0$), converge vers 0.
On pourrait aussi montrer que $r_- > 0 \Longleftrightarrow 2 > \sqrt{1-2\omega} \Longleftrightarrow 2\omega > -3$, ce qui est vrai...
Autre façon de faire : on remarque

$$\begin{aligned} 0 < \omega < \frac{1}{2} \quad &\Longleftrightarrow \quad 1 < 2-\sqrt{1-2\omega} < 2 \\ &\Longleftrightarrow \quad 2 < 2+\sqrt{1-2\omega} < 3 \\ &\Longleftrightarrow \quad 3 < 3+2\omega < 4 \\ &\Longleftrightarrow \quad \frac{1}{4} < \frac{1}{3+2\omega} < \frac{1}{3} \end{aligned}$$

donc

$$\frac{1}{2} < r_+ < 1 \qquad \frac{1}{4} < r_+ < \frac{2}{3}$$

et pour $\omega = \frac{1}{2}$, $r = \frac{1}{2}$.

Si $\Delta < 0 \Longleftrightarrow \omega > \frac{1}{2}$, alors dans ce cas

$$r_+ = \frac{2 + i\sqrt{2\omega - 1}}{3 + 2\omega} = s_1 + is_2, \quad r_- = \frac{2 - i\sqrt{2\omega - 1}}{3 + 2\omega} = s_1 - is_2.$$

La partie réelle de r_+ et de r_- vaut $s_1 = \frac{2}{3+2\omega} \in]0, \frac{2}{3}[\subset] - 1, 1[$, donc la suite u_n (dont la solution est $u_n = s_1^n(K\cos(s_2 n) + L\sin(s_2 n))$ converge encore vers 0.

□

(c) Dans cette question uniquement, on prend $f(t, u) = -\lambda u$, avec $\lambda > 0$. On prend les valeurs de α_0, α_1 et β trouvées à la question **??**).
Déduire des questions précédentes sous quelle condition sur h le schéma (**??**) est absolument stable.
Réponse : on regarde la stabilité en temps long (stabilité absolue), pour ce f linéaire. Alors le schéma devient : $\frac{3}{2}z_{n+1} = 2z_n - \frac{1}{2}z_{n-1} - h\lambda z_{n+1}$, ce qui donne en posant $\omega = h\lambda > 0$

$$(3 + 2\omega)z_{n+1} = 4z_n - z_{n-1}.$$

On retrouve la récurrence de u_n. On en déduit que pour tout $h > 0$: $z_n \longrightarrow 0$ quand $n \to 0$, donc le schéma est stable sans condition sur h. □

5. Soit p un entier > 0. On suppose dorénavant que la fonction f de classe C^2 est vectorielle : $(t, y) \in [t_0, t_0 + T] \times \mathbb{R}^p \mapsto f(t, y) \in \mathbb{R}^p$. On cherche à résoudre l'équation non-linéaire

$$\text{trouver } x^\star \in \mathbb{R}^p, \text{ tel que } x^\star = \alpha_0 z_n + \alpha_1 z_{n-1} + h\beta f(t_{n+1}, x^\star) \tag{4}$$

par la méthode de Newton.

(a) Reformuler ce problème (**??**) en une équation $G(x) = 0$: déterminer G.
Réponse : il vient :

$$G \ : \ x \in \mathbb{R}^p \mapsto G(x) = x - \alpha_0 z_n - \alpha_1 z_{n-1} - h\beta f(t_{n+1}, x).$$

□

(b) Calculer la matrice jacobienne $DG(x)$ en fonction des dérivées partielles de f.
Réponse : en dérivant, on trouve

$$DG(x) \in \mathcal{M}_{p,p} \ : \ DG(x) = I_{p,p} - h\beta \frac{\partial f}{\partial x}(t_{n+1}, x).$$

où la matrice $\frac{\partial f}{\partial x}(t, x)$ appartient bien à $\mathcal{M}_{p,p}$. □

(c) **Application :** Soit ξ, k, u_0 et v_0 des réels fixés. Soit l'équation différentielle :

$$\text{trouver } u \in C^1([t_0, t_0 + T], \mathbb{R}), \text{ telle que } \begin{cases} u''(t) = \xi u'(t) + ku^2(t) \ \forall t \in [t_0, t_0 + T], \\ u(t_0) = u_0, \ u'(t_0) = v_0. \end{cases} \tag{5}$$

i. Mettre cette équation différentielle (**??**) sous forme d'une équation différentielle du type :

$$\text{trouver } y \text{ telle que } \begin{cases} y'(t) = f(t, y(t)) \text{ pour } t \in [t_0, t_0 + T], \\ y(t_0) = y_0. \end{cases}$$

Expliciter la fonction f de cette équation différentielle, en précisant bien les espaces de départ et d'arrivée.
Réponse : en posant $y(t) = [u(t), u'(t)]^T$, on obtient $y'(t) = [u'(t), u''(t)]^T = [y_2(t), \xi y_2(t) + ky_1^2(t)]^T$. On en déduit que $y \ : \ [t_0, t_0 + T] \to \mathbb{R}^2$ vérifie

$$\begin{cases} y'(t) = f(t, y(t)) \textbf{ pour } t \in [t_0, t_0 + T], \\ y(t_0) = y_0 = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}, \end{cases} \quad \textbf{avec } f \ : \begin{cases} [t_0, t_0 + T] \times \mathbb{R}^2 \longrightarrow \mathbb{R}^2 \\ (t, x) \longrightarrow f(t, x) = \begin{bmatrix} x_2 \\ \xi x_2 + kx_1^2 \end{bmatrix}. \end{cases}$$

□

ii. Calculer G et sa matrice jacobienne $DG(x)$ dans ce cas.
Réponse : il vient :

$$G(x) = x - \alpha_0 z_n - \alpha_1 z_{n-1} - h\beta \begin{bmatrix} x_2 \\ \xi x_2 + kx_1^2 \end{bmatrix}.$$

et sa jacobienne vaut :

$$DG(x) = I_{p,p} - h\beta \begin{bmatrix} 0 & 1 \\ 2kx_1 & \xi \end{bmatrix} = \begin{bmatrix} 1 & -h\beta \\ -2h\beta kx_1 & 1 - h\beta\xi \end{bmatrix}.$$

On peut remarquer que si h est assez petit, la jacobienne sera inversible car

$$\det(DG(x)) = 1 - h\beta\xi - 2h^2\beta^2 kx_1 \quad \longrightarrow 1 \textbf{ quand } h \to 0.$$

□

6. **Programmation du schéma :** On suppose toujours que f est une fonction vectorielle : $f : (t, y) \in [t_0, t_0 + T] \times \mathbb{R}^p \mapsto f(t, y) \in \mathbb{R}^p$, pour $p > 0$.

   (a) Écrire une fonction scilab [X,k] = newtonSchema(X0, tol, Niter, h, t, Zn, Znm1, f, dfdy) implémentant la méthode de Newton pour la résolution du système d'équations non-linéaires $G(x) = 0$, où la fonction G est définie à la question **??**).

   Expliquer rapidement quel rôle jouent les arguments f et dfdy. Détailler en particulier les dimensions de leurs arguments d'entrée et de sortie.

   Réponse : Exemple d'implémentation :

```
function [X, k] = newtonSchema(X0, tol, N, h, t, Zn, Znm1, f, dfdy)
// Newton pour la resolution de Zn+1 = 4/3*Zn - 1/3*Zn-1 + 2/3*h *f(tn+1,Zn+1)
p = length(X0); X = X0;
if length( Zn )~= p | length( Znm1 )~= p
        error("Size is incompatible with initial conditions.")
end
for k=1:N
        Fx = f(t, X);
        Gx = X - 1/3*(4*Zn - Znm1 + 2*h*Fx); // evaluation de G en X
        DGx = eye(p, p) - 2*h/3 * dfdy(t, X); // differentielle de G en X
        dX = - DGx \ Gx;
        X = X + dX
        if norm(dX) < tol
                return [X, k]
        end
end
disp('NewtonSchema did not converge: Reached maximum number of iterations...')
endfunction
```

□

   (b) Écrire les fonctions scilab ma_fun et ma_dfundy, qui seront appelées par newtonSchema quand G est donnée à la question **??**).

   Réponse : Exemple d'implémentation :

```
function [F] = ma_fun(t, X)
xi = 1; k = 1;
F = [X(2) ; xi * X(2) + k * X(1)*X(1)]
endfunction
```

   et

```
function [DF] = ma_dfundy(t, X)
xi = 1; k = 1;
DF = [ 0 , 1 ; 2*k*X(1) , xi ]
endfunction
```

□

   (c) Écrire une fonction scilab Y = schema(y0, y1, t0, T, N, f, dfdy) implémentant le schéma (**??**).

   Décrire comment vous sauvegardez la solution. Quel point initial X0 choisissez-vous dans l'appel à newtonSchema à chaque pas de temps?

   Réponse : le schéma peut s'écrire :

```
function [Y] = schema(y0, y1, t0, T, N, f, dfdy)
exec('newtonSchema.sci', -1)
p = length(y0)
if ( length( y1 )~= p )
        error("y1 and y0 have incompatible dimensions (initial conditions).")
end
Y = zeros(p, N+1); Y(:,1) = y0; Y(:,2) = y1; // initialisations
h = T / N; tnp1 = t0 + h;
Znm1 = y0; Zn = y1;
for iter = 2:N
        tnp1 = tnp1 + h // time at tn+1
        // Solveur non lineaire. Newton part de X0 = Zn.
        [Znp1 , k] = newtonSchema(Zn, 1e-8, 1000, h, tnp1, Zn, Znm1, f, dfdy);
        Y(: , iter+1) = Znp1;
        Znm1 = Zn; Zn = Znp1;
end
endfunction
```

□

Exercice 2 : *(barème approximatif : 5 points)* **CHANGEZ DE COPIE**
Soit $n \geq 3$ un entier. Soient $(t_i)_{i=1,...,n}$ n points distincts ($t_i \neq t_j$ si $i \neq j$) dans l'intervalle $[-10, 10]$ et $(y_i)_{i=1,...,n}$ n réels.

1. On cherche la parabole qui passe au plus près (au sens des moindres carrés) des points $(t_i, y_i)_{i=1,...,n}$.

   (a) Poser le problème de moindres carrés et les équations à résoudre.
   Réponse : on écrit le polynôme recherché dans la base canonique : $p = \alpha_0 + \alpha_1 t + \alpha_2 t^2$ **On cherche donc** $x = [\alpha_0, \alpha_1, \alpha_2]^T \in \mathbb{R}^3$ **tel que l'erreur quadratique**

   $$E(x) = \sum_{i=1}^{n} (p(t_i) - y_i)^2 = \sum_{i=1}^{n} (\alpha_0 + \alpha_1 t_i + \alpha_2 t_i^2 - y_i)^2$$

   soit minimale. L'erreur se réécrit

   $$E(x) = \sum_{i=1}^{n} ((Ax)_i - y_i)^2 = \|Ax - y\|_2^2, \quad \textbf{avec } A = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ \vdots & \vdots & \vdots \\ 1 & t_n & t_n^2 \end{bmatrix} \in \mathcal{M}_{n,3} \textbf{ et } y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

   Le vecteur x **est solution des équations normales**

   $$A^T A x = A^T y, \quad \textbf{où } A^T A = \begin{bmatrix} n & \sum_{i=1}^n t_i & \sum_{i=1}^n t_i^2 \\ \sum_{i=1}^n t_i & \sum_{i=1}^n t_i^2 & \sum_{i=1}^n t_i^3 \\ \sum_{i=1}^n t_i^2 & \sum_{i=1}^n t_i^3 & \sum_{i=1}^n t_i^4 \end{bmatrix} \in \mathcal{M}_{3,3} \textbf{ et } A^T y = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n y_i t_i \\ \sum_{i=1}^n y_i t_i^2 \end{bmatrix}.$$

   □

   (b) Prouver que ces équations admettent une solution unique.
   Réponse : on montre que $\operatorname{rank}(A) = 3$. **On extrait les 3 premières lignes de** A **pour obtenir une matrice** $\hat{A}$ **qui est une matrice de Van der Monde dont le déterminant est d'après le cours** $\det(\hat{A}) = (t_3 - t_2)(t_3 - t_1)(t_2 - t_1)$ **qui est donc non-nul car les** t_i **sont distincts deux à deux. Comme on a extrait de** $A \in \mathcal{M}_{n,3}$ **avec** $n \geq 3$ **une matrice carrée inversible de taille** 3**, le rang de** A **est** 3**.**
   Donc les équations normales admettent une unique solution.

   □

2. On suppose données les fonctions scilab suivantes :
   `[p] = hornv(a, t, theta)`,
   qui, étant donnés les vecteurs $(a_1, a_2, ..., a_n)$, $(t_1, t_2, ..., t_{n-1})$, $(\theta_1, ..., \theta_m)$, calcule le vecteur $p = (p_1, p_2, ..., p_m)$ dont les termes sont définis par

   $$p_j = a_1 + a_2(\theta_j - t_1) + a_3(\theta_j - t_1)(\theta_j - t_2) + ... + a_n(\theta_j - t_1)...(\theta_j - t_{n-1}), \text{ pour } j = 1, ..., m.$$

   à l'aide de l'algorithme de Horner.
   `[Q, R] = qr(A)`, qui effectue la factorisation $A = QR$,
   `[x] = solsup(A, b)`, qui résout le système triangulaire supérieur $Ax = b$,
   `[x] = solinf(A, b)`, qui résout le système triangulaire inférieur $Ax = b$.
   Écrire une fonction scilab : `trace(t, y, N)` qui trace le polynôme de degré ≤ 2 passant au plus près des points $(t_i, y_i)_{i=1,...,n}$. On tracera la courbe avec N points dans l'intervalle $[-10, 10]$.
   Réponse : Exemple d'implémentation :

```
function [x] = trace(t, y N)
exec("solsup.sci", -1); exec(hornv.sci", -1);
n = length(t);
if ( length( y )~= n )
    error("y and t have incompatible dimensions.")
end
t = t(:); tt = t .* t; // t doit etre un vecteur colonne.
A = [ ones(n, 1) , t, tt ]; [ Q , R ] = qr(A);
rhs = Q'*y;
x = solsup( R(1:3, 1:3), rhs(1:3) ); // car A^T Ax = A^T y <=> R~x = (Q^T y)(1 : 3)
```

```
theta = linspace(-10, 10, N);
p = hornv(x, [0, 0], theta); // le polynome s'ecrit: p=x1 + x2 (t-0) + x3 (t-0)^2.
plot(theta , p, 'b-'); // trace du polynome
plot(t , y, 'ro'); // trace des points de mesures
endfunction
```

□