Comparer les deux fonctions `Scilab` suivantes. La première implémente la méthode des différences finies avant et la deuxième implémente la méthode des différences finies centré.

```
function [ U ] = DFA( N, c, f )
// DFA différences finies avant
//   N — le nombre de points du maillage
//   c — une fonction (définie par deff)
//   f  — une autre fonction (définie par deff)

h = 1 / (N+1);       // le pas de discretisation
x = h*(0:(N-1));     // le maillage

A = 1/h^2 * ( 2 * eye(N,N) - diag(ones(N-1, 1), -1) - diag(ones(N-1, 1), 1));

cx = c(x(1:(N-1))); // c(x_i) pour i = 1, 2, ..., N-1
C = diag(cx.', -1);

F = f(x.');          // le membre droit

U = (A + C) \ F;
endfunction
```

```
function [ U ] = DFC( N, c, f )
//DFC différences finies centré
//   N — le nombre de points du maillage
//   c — une fonction inline
//   f — une autre fonction inline

h = 1 / (N+1);       // le pas de discretisation
x = h * (1:N);       // le maillage

A = 1/h^2 * ( 2 * eye(N,N) - diag(ones(N-1, 1), -1) - diag(ones(N-1, 1), 1));

cx = c(x);           // c(x_i) pour i = 1, 2, ..., N-1
C = diag(cx.', 0);

F = f(x.');          // le membre droit

U = (A + C) \ F;
endfunction
```

Le script Scilab suivant (le fichier `test.sce`) teste les deux fonctions précédentes pour le choix $c(x) \equiv 1$ et $f(x) = (4\pi^2 + 1)\sin(2\pi x)$.
Réponds-t-il aux questions de l'exercice 1.4 ? Commenter les résultats. Proposer d'autres choix de $c(x)$ et $f(x)$ tels que la solution exacte du problème (1) soit explicite.

```
clear();         // efface la mémoire
clc();           // efface la console
xdel(winsid())   // ferme toutes les fenêtres

N = 100;
h = 1 / (N+1);
x = h:h:1-h;

deff('y = f(x)', 'y=(4*%pi^2 + 1)*sin(2*%pi*x)');
```

```
deff('y = uex(x)', 'y = sin(2*%pi*x)');
deff('y = c(x)', 'y = ones(x)');

exec('DFA.sci', 0)
exec('DFC.sci', 0)

[UA]=DFA(N, c, f);
[UC]=DFC(N, c, f);

hf = figure
plot(x, uex(x), 'g')
plot(x, UA, 'r:')
plot(x, UC, 'm')
hl = legend('Solution exacte', 'solution DFA', 'Solution DFC')
hl.fill_mode = "off";
hl.font_size=3;

err = sqrt(h)*norm(UC - uex(x)');

// question h
Nvec = 50*(2 .^ (0:4));
hvec = 1 ./ (Nvec+1);

normL2ErrA = zeros(size(Nvec)); // vecteur erreur DFA
normL2ErrC = zeros(size(Nvec)); // vecteur erreur DFC

for i = 1:length(Nvec)
    x = ((1:Nvec(i))*hvec(i)).'; // le maillage
    UA = DFA(Nvec(i), c, f);     // solution DFA
    UC = DFC(Nvec(i), c, f);     // solution DFC
    Uex = sin(2*%pi*x);           // solution exacte evaluee sur le maillage
    normL2ErrA(i) = sqrt(hvec(i)*sum((UA - Uex).^2));
    normL2ErrC(i) = sqrt(hvec(i)*sum((UC - Uex).^2));
end
```

On considère des conditions Neumann pour le problème (1). On obtient :

$$\begin{cases} -\dfrac{d^2u}{dx^2}(x) + c(x)u(x) = f(x), & \text{pour } x \in]0,1[\\ \dfrac{du}{dx}(0) = \dfrac{du}{dx}(1) = 0. \end{cases}$$

Adapter la fonction `DFC` pour prendre en charge les conditions frontière de type Neumann en réalisant une nouvelle fonction d'entête : `function [U] = DFCNeumann(N, c, f)`.
Indication : Le vecteur U va être de taille $(N+2)$ (car u_0 et u_{N+1} ne sont plus nuls).

```
function [ U ] = DFCNeumann( N, c, f )
//DFCNeumann différences finies centré
//   N - le nombre de points du maillage
//   c - une fonction (définie par deff)
//   f - une autre fonction (définie par deff)

h = 1 / (N+1);      // le pas de discrétisation
x = h*(1:N);        // le maillage

A = 1/h^2*(2*eye(N+2,N+2) - diag(ones(N+1, 1), -1) - diag(ones(N+1, 1), 1));
A(1, 1) = -1/h;  A(1, 2) = 1/h; A(N+2, N+1) = -1/h;  A(N+2, N+2) = 1/h;

cx = [0, c(x), 0];  C = diag(cx.', 0);
F = [0; f(x.'); 0];         // le membre droit
U = (A + C) \ F;
endfunction
```

Le script suivant (`test2.sce`) réponds à la question 1.4.i)

```
clear();
clc();
xdel(winsid())

deff('y=f(x)', 'y=(1 + %pi^2)*cos(%pi*x)');
deff('y=c(x)', 'y=ones(x)');

exec('DFCNeumann.sci')

// question g)
N = 100;                // points dans le maillage
h = 1 / (N+1);          // pas de discretisation
x = ((0:(N+1))*h).';    // le maillage

UN = DFCNeumann( N, c, f );
Uex = cos(%pi*x);

// affichage des graphiques
figure('BackgroundColor', [1, 1, 1])
plot(x, UN, 'b');  // graph de la sol DFNeumann avec bord
plot(x, Uex, 'r'); // graph de la sol exacte
hl=legend('Solution DFCNeumann', 'Solution exacte');
hl.fill_mode = "off";
hl.font_size=3;
xlabel('x');
ylabel('u(x)');

// Re—faire le graphique de l'erreur en fonction de h dans ce cas!!!
```

La fonction suivante résout le problème (2) en utilisant le schéma (3). **Attention aux conditions sur la frontière du domaine !**

```
function [ U ] = DFCInelastique( N, c, f )
//DFCInelastique différences finies centré pour le probleme inelastique
//   N — le nombre de points du maillage
//   c — une fonction (définie par deff)
//   f — une autre fonction (définie par deff)

h = 1 / (N+1);      // le pas de discretisation
x = h*(1:N);      // le maillage

A = 1/h^4 * ( 6 * eye(N,N) — 4*diag(ones(N—1, 1), —1) ...
    —4* diag(ones(N—1, 1),1) + diag(ones(N—2,1), —2) + diag(ones(N—2,1),2));
A(1,1) = 7;
A(N,N) = 7;

cx = c(x);          // c(x_i) pour i = 2, ..., N—1
C = diag(cx.', 0);

F = f(x.');         // le membre droit

U = (A + C) \ F;
endfunction
```

Le script suivant (`test3.sce`) réponds partiellement à la question 2.3.

```
clear();
clc();
xdel(winsid())

deff('y=f(x)', 'y=16*(24 + x.^2 .* (1 — x).^2)');
deff('y=c(x)', 'y=ones(x)');

exec('DFCInelastique.sci')

N = 400;                // points dans le maillage
h = 1 / (N+1);          // pas de discretisation
x = ((0:(N+1))*h).';    // le maillage

UN = DFCInelastique( N, c, f );
Uex = 16*x.^2 .* (1—x).^2;

UN = [0; UN; 0]; // on ajoute les x0, x1, xN et x(N+1)

figure // affichage des graphiques
plot(x, UN, 'b');  // graph de la sol DFCInelastique avec bord
plot(x, Uex, 'r'); // graph de la sol exacte
legend('Solution DFCInelastique', 'Solution exacte');
xlabel('x');
ylabel('u(x)');

err = sqrt(h)*norm(UN — Uex)
// Re—faire le graphique de l'erreur en fonction de h dans ce cas!!!
// Quel est l'ordre du schéma?
// Pour le même f comparer les solutions des problèmes (1) et (2).
// Comment affecte la valeur de la fonction c la solution de (1) ou (2)?
```