0. Théorie.

Le schéma pour résoudre une équation différentielle est souvent le même, donc il est préférable de l'écrire une fois pour tout et de le réutiliser.

On veut résoudre numériquement une équation différentielle : $\boldsymbol{y'(x) = f(x, y(x)), \quad y(x_0) = y_0}$

On veut obtenir nMax+1 valeurs de y répartis régulièrement entre x0 et xFin.

Écrivons une fonction qui résolve numériquement notre équation différentielle :

```
function y=Euler_o1_d1(x0, xFin, y0, nMax, f)
//=============================================
// o1, signifie que l'ordre numérique est de 1, soit peu précis.
// d1, signifie que la dimension de  y0  et de  y  est de  1, soit assez limité.
// Résolution numérique de l'équation différentielle y'(x) = f(x, y(x))
// y(x0) = y0  est la condition initiale
// retourne un vecteur de  nMax+1  valeurs de  y.
// f  est la fonction décrivant l'équ. diff.
y(1) = y0;
dx = (xFin - x0) / nMax;
for j=1:nMax do
   x = x0 + (j-1) * dx;
   y(j+1) = y(j) + dx * f(x, y(j));
   end;
endfunction
```

Pour le problème de l'exercice 1, il suffit de faire ce qui suit :

```
function z=f(x, y)
//================
// z  représente  y'
z = 9.81 - (1.5/3)*y;  // car  y' = f(x, y) = 9.81 - (1.5/3) * y
endfunction

// Résolution numérique
t0 = 0;
tFin = 10;
N = 100;
ay = Euler_o1_d1( t0..    // temps initial
                 ,tFin..   // temps final
                 ,0..    // vitesse initiale
                 ,N..  // nombre de points = nMax
                 ,f);    // f  est la fonction définissant l'équ. diff.

// graphique de la solution numérique
x = linspace(t0, tFin, N + 1)';
clf();
plot(x, y, 'k');
```

1. Évolution de la vitesse d'un corps qui chute avec frottement.

Le but est de calculer numériquement l'évolution de la vitesse d'un corps qui chute avec frottement.

Un corps de masse $m = 3.0$ [kg] chute verticalement, en subissant une force résultante égale à :

$F_{\text{résultante}} = m \cdot g - 1.5 \cdot V(t)$

où V(t) est la vitesse du corps au temps t. $g = 9.81$ [m/s²].

Au temps t = 0 [s], le corps est immobile.

1.1 Juste avant la fonction qui calculera l'évolution de la vitesse en fonction du temps, écrivez en commentaire l'équation différentielle qui régit l'évolution de cette vitesse.

1.2 Écrivez une "function ex1()" qui calcule par la **méthode d'Euler, comme décrit ci-dessus,** l'évolution de la vitesse du corps durant 10 secondes, en prenant nMax = 100.

1.3 À la fin de votre fonction, tracez la courbe d'évolution de la vitesse en fonction du temps.

2. Évolution du niveau d'eau d'un réservoir.

Le but est de calculer numériquement l'évolution du niveau d'eau d'un réservoir qui se remplit.
Le réservoir est cylindrique de section de base valant 1 dm^2.
Le débit d'eau qui entre dans le réservoir vaut :
$D(t) = 2 - h(t)/6$, où $h(t)$ est la hauteur d'eau dans le réservoir au temps t.
La hauteur est exprimée en *dm* et le débit en *litres par seconde*.
Rappelons que le débit est la variation du volume par unité de temps.
Initialement, le réservoir est vide.

2.1 Juste avant la fonction qui calculera l'évolution de la hauteur en fonction du temps, écrivez en commentaire l'équation différentielle qui régit l'évolution de hauteur h(t).

2.2 Écrivez une "function ex2()" qui calcule par la **méthode d'Euler, comme décrit ci-dessus,** l'évolution de la vitesse du corps durant 20 secondes, en prenant nMax = 200.

2.3 À la fin de votre fonction, tracez la courbe d'évolution de la vitesse en fonction du temps.

3_0. Théorie.

Le schéma pour résoudre une équation différentielle écrit dans la théorie précédente est limité, car il ne traite que les systèmes de une dimension.
Ce qui suit est très similaire, mais traite les systèmes de dimension quelconques.
On veut résoudre numériquement une équation différentielle : $\boldsymbol{y'(x) = f(x, y(x)),\quad y(x_0) = y_0}$
On veut obtenir nMax+1 valeurs de y répartis régulièrement entre x0 et xFin.
y0, y et f sont cette fois des vecteurs.

```
function y=Euler_o1_dn(x0, xFin, y0, nMax, f)
//=============================================
// o1, signifie que l'ordre numérique est de 1, soit peu précis.
// dn, signifie que la dimension de  y0  et de  y  peut être quelconque.
// Résolution numérique de l'équation différentielle y'(x) = f(x, y(x))
// y(x0) = y0  est la condition initiale
// retourne une matrice  (nMax+1 ; n)  valeurs de  y.   n = length(y0)
// f  est la fonction décrivant l'équ. diff.
// y0  et  f  doivent être des vecteurs colonne.
y(:, 1) = y0;
dx = (xFin - x0) / nMax;
for j=1:nMax do
   x = x0 + (j-1) * dx;
   y(:, j+1) = y(:, j) + dx * f(x, y(:, j));
   end;
endfunction
```

Pour le problème de l'exercice 3, il suffit de faire ce qui suit :

```
// Exercice 3.  Évolution d'un oscillateur
// Décrit par une équation différentielle du second ordre :
// x(t)'' = -k * x(t)
// ou par un système de deux équations différentielles du première ordre :
// x'(t) = v(t)
// v'(t) = -k * x(t)
function ex3()
//===========
   function z=f(x, y)
   //===============
   // x représente le temps
   // y(1)  représente la position
   // y(2)  représente la vitesse
   // z(1)  représente la dérivée de la position par rapport au temps
   // z(2)  représente la dérivée de la vitesse par rapport au temps
   k = 2.0;
   z(1) = y(2);     // car  position ' = vitesse
   z(2) = -k*y(1);  // car  vitesse ' = -k * position
   endfunction

   // données :
   t0    = 0;    // temps initial
   tFin  = 20;   // temps final
   y0 = [0.1; 0];  // position ; vitesse initiale, vecteur colonne
   N  = 500;      // nombre de points d'évaluation
   // Résolution numérique
   ay = Euler_o1_dn( t0, tFin, y0, N, f);

   // graphique de la solution numérique
   at = linspace(t0, tFin, N+1);
   scf(3);
   clf();
   plot(at, ay(1, :), 'k');
endfunction
```

3. Oscillateur.

Le but est de calculer numériquement l'évolution d'un oscillateur.
Soit un corps d'une masse de 1 [kg] subissant une force résultante égale à
$F_{résultante} = - k \cdot x(t)$ où $k = 2$ [N / m].
$x(t)$ est la position du corps au temps t.
$V(t)$ est la vitesse du corps au temps t ;
Au temps $t = 0$ [s], le corps est immobile et se trouve en $x(0) = 0.1$ [m].
Reprenez l'énoncé 3 de la série 4.

3.1 Juste avant la fonction qui calculera l'évolution de cet oscillateur en fonction du temps, écrivez en commentaire l'équation différentielle qui régit son l'évolution.

3.2 Écrivez une "function ex3()" qui calcule par la **méthode d'Euler, comme décrit ci-dessus,** l'évolution de la vitesse du corps durant 20 secondes, en prenant nMax = 1000.

3.3 À la fin de votre fonction, tracez la courbe d'évolution de la position en fonction du temps.

4. Oscillateur amorti

Le but est de calculer numériquement l'évolution d'un oscillateur qui subit un frottement
Soit un corps d'une masse de 1 [kg] subissant une force résultante égale à
$F_{résultante} = - k \cdot x(t) - c \cdot v(t)$ où
x(t) est la position du corps au temps t.
V(t) est la vitesse du corps au temps t ;
Au temps t = 0 [s], le corps est immobile et se trouve en x(0) = 0.1 [m].
k = 2 [N / m].
c = 0.3 [N · s / m].
Reprenez l'énoncé 4 de la série 4.

4.1 Juste avant la fonction qui calculera l'évolution de cet oscillateur en fonction du temps, écrivez en commentaire l'équation différentielle qui régit son l'évolution.
4.2 Écrivez une "function ex4()" qui calcule par la **méthode d'Euler, comme décrit ci-dessus,** l'évolution de la vitesse du corps durant 20 secondes, en prenant nMax = 2000.
4.3 À la fin de votre fonction, tracez la courbe d'évolution de la position en fonction du temps.

---

5. Étude de la précision de la méthode d'Euler

Le but est de calculer numériquement la solution d'une équation différentielle, puis de la comparer avec la solution exacte, qui est connue, pour étudier la précision de la méthode d'Euler.
Résolvez numériquement :
$x'(t) = v(t)$
$v'(t) = -x(t)$
$x(0) = 0; \quad v(0) = 1;$
Déterminez (trouvez, devinez) la solution exacte de ce système.

5.1 Juste avant la fonction qui calculera l'évolution de cet oscillateur en fonction du temps, écrivez en commentaire l'équation différentielle qui régit son l'évolution.
5.2 Écrivez une "function ex5()" qui calcule par la **méthode d'Euler, comme décrit ci-dessus,** la solution numérique de ce système en $t = 2 \cdot \pi$.
5.3 À la fin de votre fonction, tracez la courbe obtenue numériquement de v(t) et celle exacte, pour la comparer.
5.4 Pour différentes valeurs de nMax, entre 100 et 10'00, affichez la valeur de
$nMax \cdot (v_{numérique}(2 \cdot \pi) - v_{exacte}(2 \cdot \pi))$.
Constatez qu'on obtient presque chaque fois la même valeur. Déduisez-en une information sur la précision de la méthode d'Euler. On dit qu'elle est d'**ordre 1**.
5.5 Une méthode d'**ordre 2** a la propriété que : $nMax\text{^}2 \cdot (v_{numérique}(2 \cdot \pi) - v_{exacte}(2 \cdot \pi))$ ne varie presque pas en fonction de nMax. Décrivez son avantage.
5.6 La méthode de Runge-Kutta est d'**ordre 4**.
Que cela signifie-t-il ?
Quelle est son avantage ?
