Corrigé de TP 4 : Résolution numérique des équations différentielles

I. Travail dirigé

1) D'abord écrivons cette équation sous la forme : $y'(t) = f(t, y(t)) = t + \frac{3}{t} y(t)$

En suite on utilise la formule d'Euler qui permet de trouver un point (t_{i+1}, y_{i+1}) à partir le point précédent (t_i, y_i), elle est donnée par :

$$y_{i+1} = y_i + hf(t_i, y_i) \quad \text{avec} \quad t_0 = a \ , t_n = b \ , t_{i+1} = t_i + h \ , h = \frac{b-a}{n} \ , i = 0,1,2, \dots, n-1$$

où n est le nombre de sous intervalles, h est le pas de t, et les valeurs initiales (t_0, y_0) sont donnés.

On a donc : $h = \frac{2-1}{4} = 0.25$, $(t_0, y_0) = (1\ , 3)$. Pour $n = 4$ on fait uniquement 4 itérations.

Itération 1 :

$t_0 = a = 1$

$y_1 = y_0 + h \times f(t_0\ , y_0) = 3 + 0.25 \times f(1\ , 3) = 3 + 0.25 \times \left(1 + \frac{3}{1} \times 3\right) = 5.5$

$t_1 = t_0 + h = 1 + 0.25 = 1.25$

Itération 2 :

$y_2 = y_1 + h \times f(t_1, y_1) = 5.5 + 0.25 \times f(1.25\ , 5.5) = 5.5 + 0.25 \times \left(1.25 + \frac{3}{1.25} \times 5.5\right)$
$= 9.1125$

$t_2 = t_1 + h = 1.25 + 0.25 = 1.5$

Itération 3 :

$y_3 = y_2 + h \times f(t_2, y_2) = 9.1125 + 0.25 \times f(1.5\ , 9.1125) = 14.0438$

$t_3 = t_2 + h = 1.5 + 0.25 = 1.75$

Itération 4 :

$y_4 = y_3 + h \times f(t_3, y_3) = 20.5000$

$t_4 = t_3 + h = 1.75 + 0.25 = 2$

2) On utilise maintenant la formule de la méthode de RK4, elle permet aussi de trouver un point (t_{i+1}, y_{i+1}) à partir le point précédent (t_i, y_i), elle est donnée par :

$$y_{i+1} = y_i + \frac{1}{6} \times (k_1 + 2k_2 + 2k_3 + k_4)$$

où :

$$k_1 = h \times f(t_i, y_i), k_2 = h \times f\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right), k_3 = h \times f\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right),$$
$$k_4 = h \times f(t_i + h, y_i + k_3)$$

avec :

$$t_0 = a \ , \quad t_n = b \ , \quad t_{i+1} = t_i + h \ , \quad h = \frac{b-a}{n} \ , \quad i = 0,1,2, \dots, n-1$$

où n est le nombre de sous intervalles, h est le pas de t, et les valeurs initiales (t_0, y_0) sont donnés

On a donc : $h = \frac{2-1}{4} = 0.25$, $(t_0, y_0) = (1\ , 3)$. Pour $n = 4$ on fait uniquement 4 itérations. Les résultats sont dans le tableau suivant :

Itération	k_1	k_2	k_3	k_4	y_{i+1}	t_{i+1}

1	2.5000	3.1146	3.3194	4.1042	6.2454	1.25
2	4.0597	4.8575	5.0751	6.0352	11.2387	1.5
3	5.9944	6.9767	7.2034	8.3413	18.3547	1.75
4	8.3038	9.4714	9.7049	11.0223	27.9678	2

Exemple de calcul (première itération)

$t_0 = a = 1$ et $y_0 = 3$ (données). Calculons alors y_1 et t_1 :

$k_1 = h \times f(t_0\,, y_0) = 0.25 \times f(1\,,3) = 2.5$

$k_2 = h \times f\left(t_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right) = 0.25 \times f(1.1250\ \ ,4.2500) = 3.1146$

$k_3 = h \times f\left(t_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right) = 0.25 \times f(1.1250\,,4.5573) = 3.3194$

$k_4 = h \times f(t_0 + h, y_0 + k_3) = 0.25 \times f(1.25\,,5.5573) = 4.1042$

$y_1 = y_0 + \frac{1}{6} \times (k_1 + 2k_2 + 2k_3 + k_4) = 3 + \frac{1}{6} \times (2.5 + 2 \times 3.1146 + 2 \times 3.3194 + 4.1042)$
$= 6.2454$

$t_1 = t_0 + h = 1 + 0.25 = 1.25$

(Remarquons que ce calcul est très long en utilisant la calculatrice)

3)

- **a.** Les valeurs $y(1.25), y(1.5), y(1.75), y(2)$ de la solution exacte sont dans le tableau ci-dessous.
- **b.** Comparaison : On écrit les solutions approchées et exacte dans le tableau suivant et on compare.

		Solution approchée		Solution exacte
Itération	t_{i+1}	y_{i+1}(Euler)	y_{i+1} (RK4)	y_{i+1} (exacte)
1	1.25	5.5	6.2454	6.2500
2	1.5	9.1125	11.2387	11.2500
3	1.75	14.0438	18.3547	18.3750
4	2	20.5000	27.9678	28

On remarque que la solution obtenue par la méthode RK4 est très proche à la solution exacte par rapport la solution obtenue par la méthode d'Euler. La méthode RK4 est donc plus précise.

Cependant l'erreur se propage toujours d'une itération à l'autre.

II. <u>Travail pratique</u>

1) On fait la programmation des méthodes numérique d'Euler et de RK4 lorsque le nombre des itérations est grand $n > 4$.

2) Algorithme de calcul pour la méthode d'Euler :

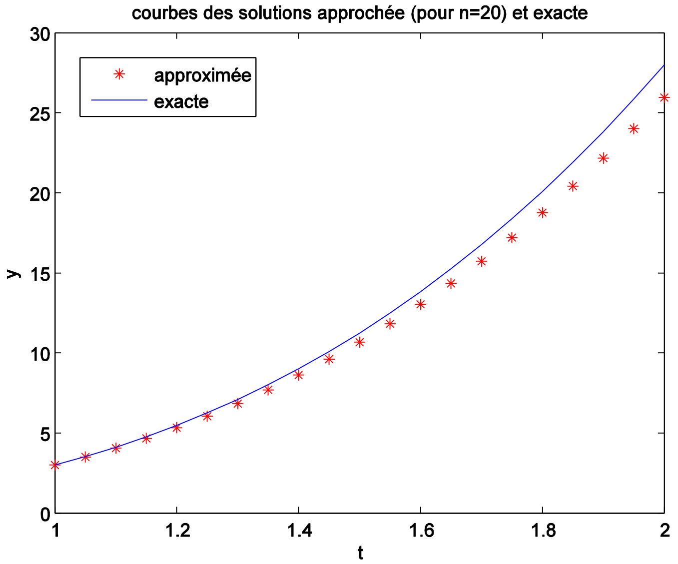
1. Donner la fonction $f(t, y)$, un intervalle $[a, b]$, une condition initiale (t_0, y_0), un nombre de sous intervalles n.
2. Calculer le pas h.
3. Calculer les valeurs (t_{i+1}, y_{i+1}) à partir des valeurs (t_i, y_i) en utilisant la formule de la méthode d'Euler.

4. Afficher les résultats et arrêter.

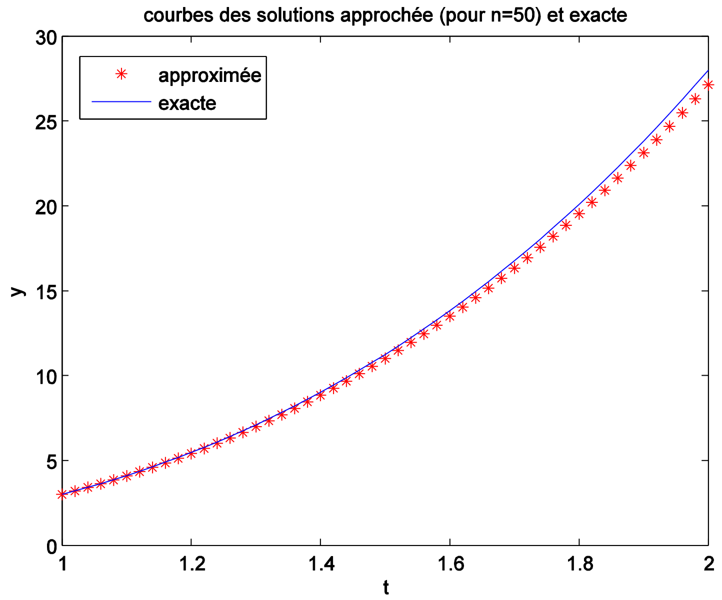
3) Programme Matlab de la méthode d'Euler :

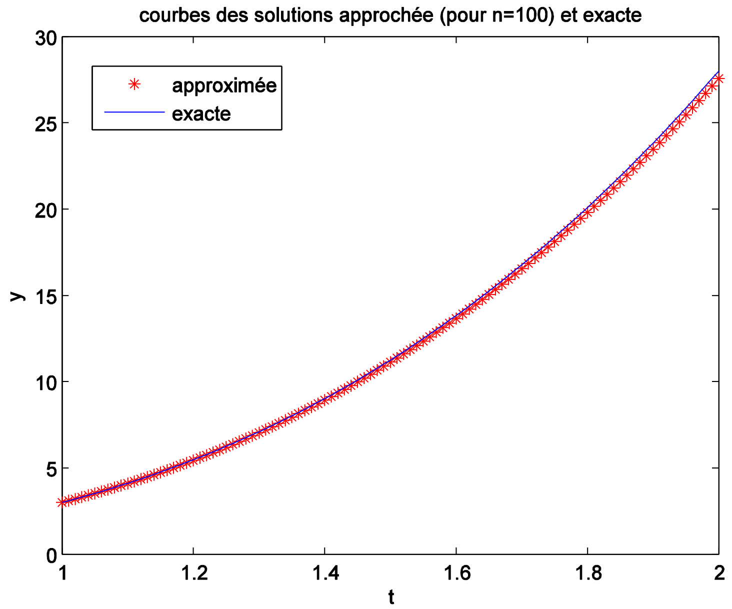
```
clc;clear all;close all;
a=1;
b=2;
n=20
h=(b-a)./n;
t=a:h:b;
y=zeros(size(t));
y(1)=3;
f=@(t,y)t+(3./t)*y;
for i=1:n
t(i)=t(1)+(i-1)*h;
k1=h*f(t(i),y(i));
y(i+1)=y(i)+k1;
end
ys=-(t.^2)+4.*(t.^3);
plot(t,y,'*r',t,ys)
xlabel('t')
ylabel('y')
title('courbes des solutions approchée (pour n=20) et exacte.')
legend('approximée','exacte')
```

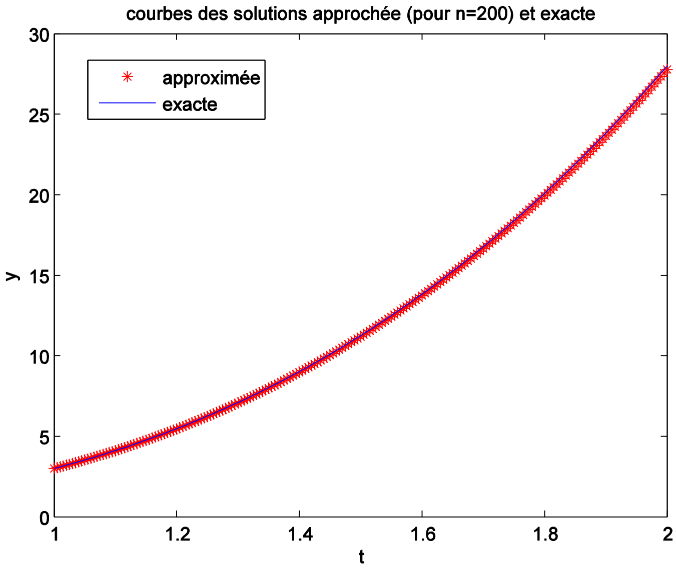
Après exécution on obtient :



4) Exécuter ce programme pour des nombres de sous-intervalles $n = 50\,,100\,,200$. Conclure.


courbes des solutions approchée (pour n=50) et exacte
approximée
exacte
y
t
30
25
20
15
10
5
0
1
1.2
1.4
1.6
1.8
2


courbes des solutions approchée (pour n=100) et exacte
approximée
exacte
y
t
30
25
20
15
10
5
0
1
1.2
1.4
1.6
1.8
2



On conclut que lorsque le nombre de sous-intervalles augmente, la solution approximée se rapproche plus en plus à la solution exacte.

5) Utilisons la méthode RK4 au lieu de la méthode d'Euler :

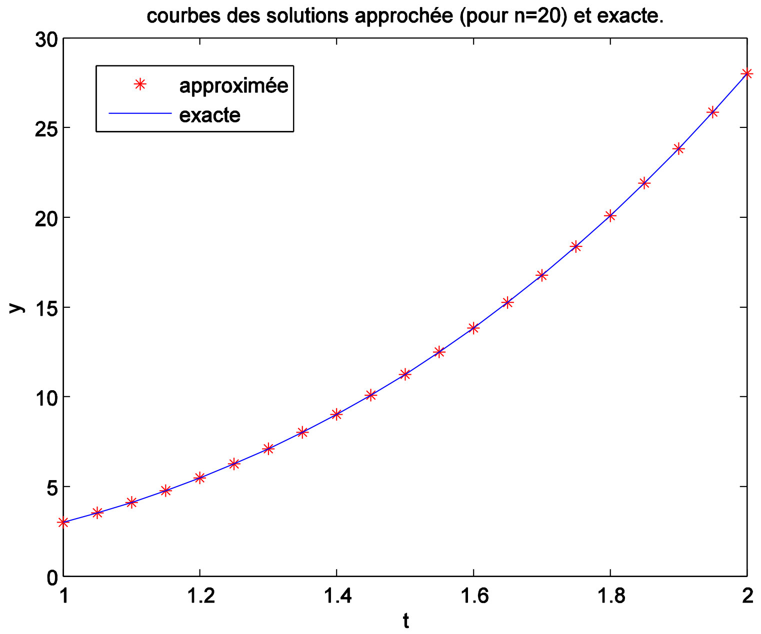
- Algorithme de calcul pour la méthode de RK4 :

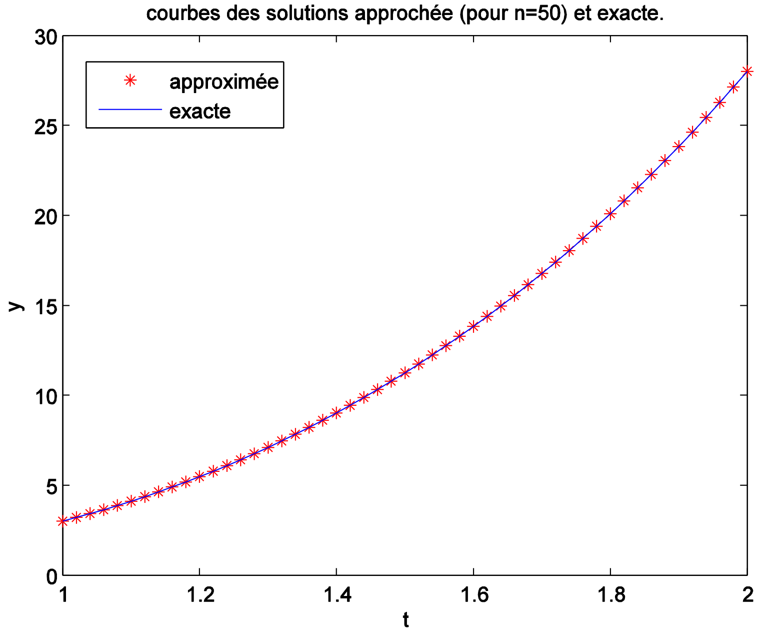
1. Donner la fonction $f(t,y)$, un intervalle $[a,b]$, une condition initiale (t_0,y_0), un nombre de sous intervalles n.
2. Calculer le pas h.
3. Calculer les valeurs (t_{i+1},y_{i+1}) à partir des valeurs (t_i,y_i) en utilisant la formule de la méthode RK4.
4. Afficher les résultats et arrêter.

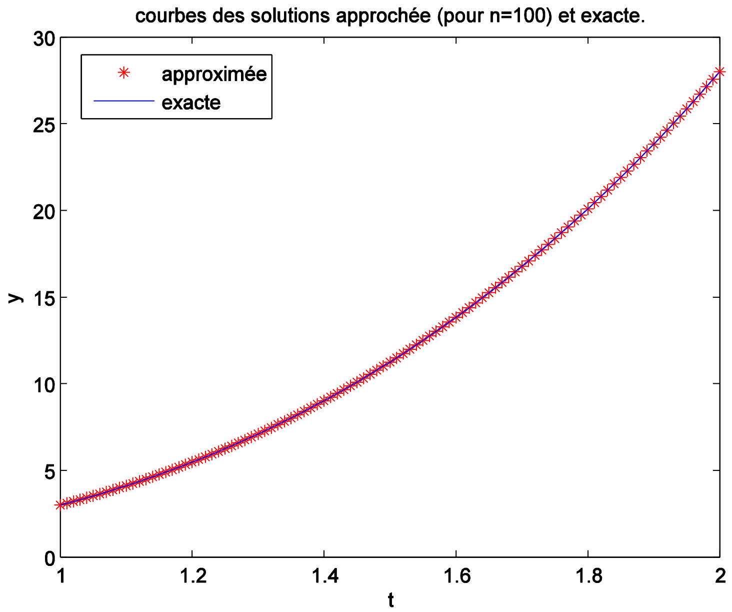
- Programme Matlab de la méthode de RK4 :

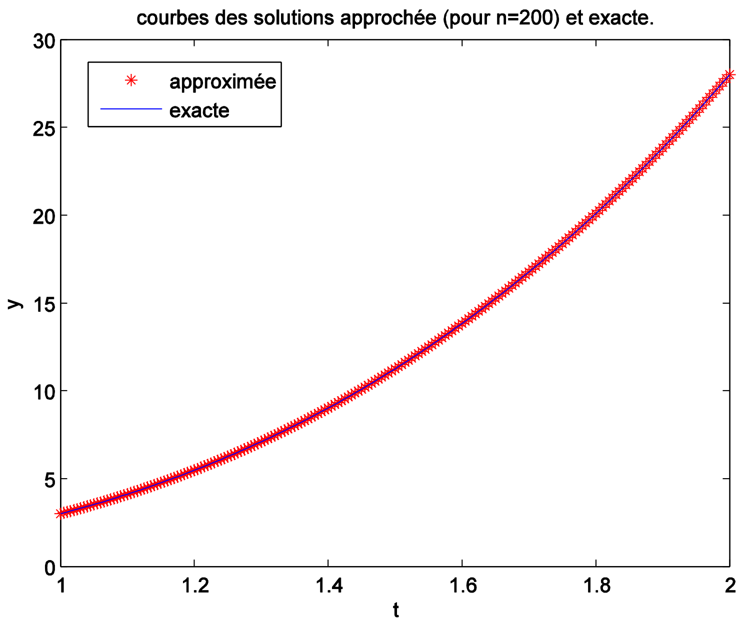
```
clc;clear all;close all;
a=1;
b=2;
n=20
h=(b-a)./n;
t=a:h:b;
y=zeros(size(t));
y(1)=3;
f=@(t,y)t+(3./t)*y;
for i=1:n
t(i)=t(1)+(i-1)*h;
k1=h*f(t(i),y(i));
k2=h*f(t(i)+(h./2),y(i)+(k1./2));
k3=h*f(t(i)+(h./2),y(i)+(k2./2));
k4=h*f(t(i)+h,y(i)+k3);
y(i+1)=y(i)+(1./6)*(k1+2.*k2+2.*k3+k4);
end
ys=-(t.^2)+4.*(t.^3);
plot(t,y,'*r',t,ys)
xlabel('t')
ylabel('y')
title('courbes des solutions approchée (pour n=20) et exacte.')
legend('approximée','exacte')
```

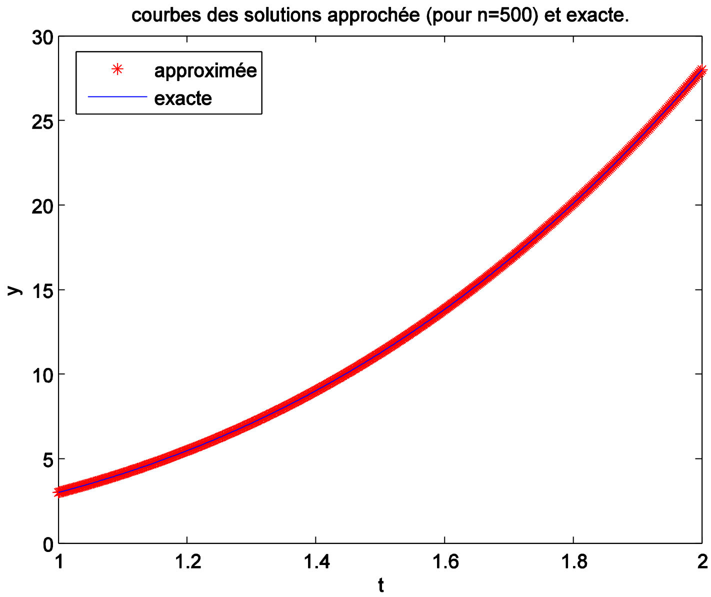
Après exécution on obtient :






courbes des solutions approchée (pour n=100) et exacte.
approximée
exacte
y
t
0
5
10
15
20
25
30
1
1.2
1.4
1.6
1.8
2


courbes des solutions approchée (pour n=200) et exacte.
approximée
exacte
y
t
0
5
10
15
20
25
30
1
1.2
1.4
1.6
1.8
2



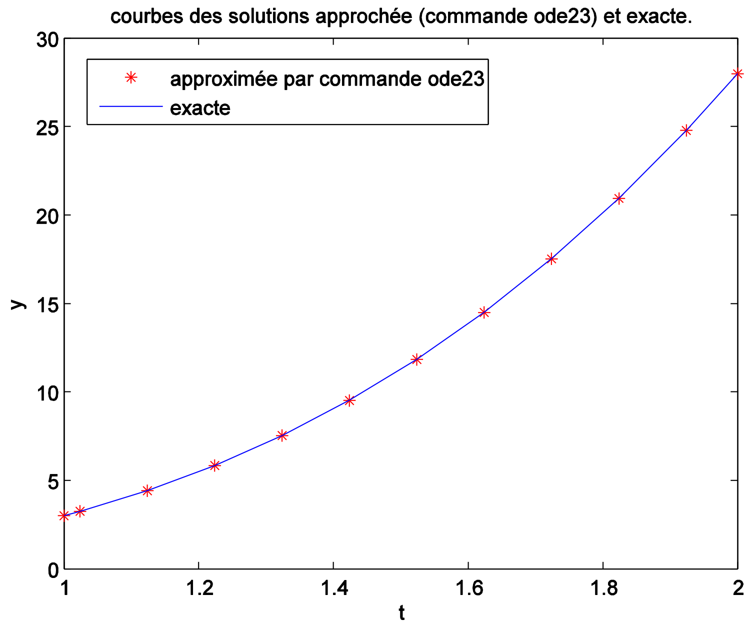
6) Matlab dispose des commandes prédéfinies permutant de calculer la solution approchée des équations différentielles ordinaires. Les commandes **ode23** et **ode45** sont très utiles, leurs syntaxes sont :

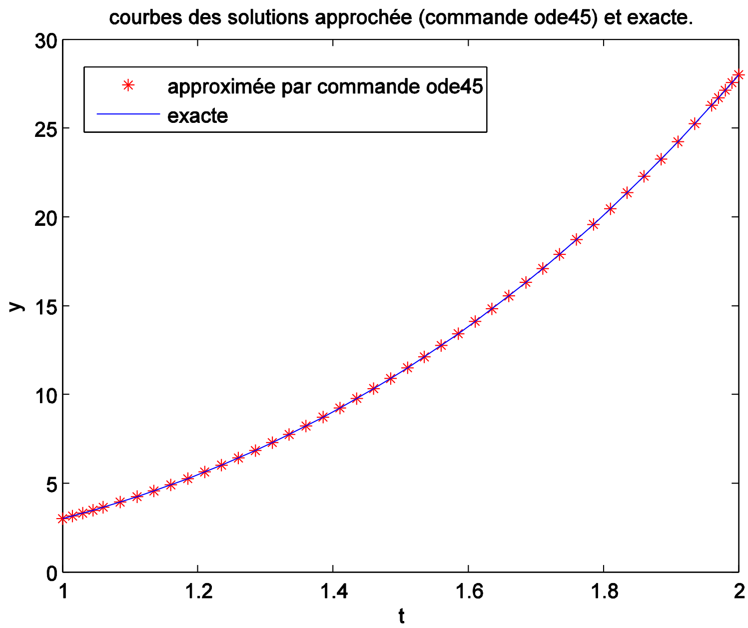
$$[\boldsymbol{t}\,,\boldsymbol{y}] = \mathbf{ode23}(\boldsymbol{f},[\boldsymbol{a}\,,\boldsymbol{b}],\boldsymbol{y0}) \quad \mathbf{et} \quad [\boldsymbol{t}\,,\boldsymbol{y}] = \mathbf{ode45}(\boldsymbol{f},[\boldsymbol{a}\,,\boldsymbol{b}],\boldsymbol{y0})$$

(Ces commandes peuvent être écrits dans un fichier M ou dans la zone de commande directement)

```
a=1;
b=2;
y0=3;
f=@(t,y)t+(3./t)*y;
[t,y]=ode45(f,[a ,b],y0)
ys=-(t.^2)+4.*(t.^3);
plot(t,y,'*r',t,ys)
xlabel('t')
ylabel('y')
title('courbes des solutions approchée (commande ode45) et exacte.')
legend('approximée par commande ode45','exacte')
```

Après exécution on obtient :


courbes des solutions approchée (commande ode23) et exacte.
approximée par commande ode23
exacte
y
t
30
25
20
15
10
5
0
1
1.2
1.4
1.6
1.8
2


courbes des solutions approchée (commande ode45) et exacte.
approximée par commande ode45
exacte
y
t
30
25
20
15
10
5
0
1
1.2
1.4
1.6
1.8
2