

In [1]:

```
from IPython.core.display import HTML
css_file = './custom.css'
HTML(open(css_file, "r").read())
```

Out[1]:

# M62\_CM5 : problèmes bien posés mathématiquement, bien posé numériquement, A- stables

On comprend bien que, en générale, il ne suffit pas qu'un schéma numérique soit convergent pour qu'il donne des bons résultats sur n'importe quelle équation différentielle. Encore faut-il que le problème soit

- mathématiquement bien posé (existence et unicité de la solution),
- numériquement bien posé (continuité suffisamment bonne par rapport aux conditions initiales)
- bien conditionné (temps de calcul raisonnable)

## Table of Contents

- ▼ [1 Problème mathématiquement mal posé : non unicité de la solution d'u](#)
  - [1.1 Exercice](#)
- ▼ [2 Problème mathématiquement bien posé mais numériquement mal po](#)
  - [2.1 Exercice](#)
  - [3 Stabilité : zéro-stabilité et A-stabilité](#)
- ▼ [4 Étude de la A-stabilité pour les schémas à un pas](#)
  - [4.1 A-stabilité du schéma d'Euler explicite](#)
  - [4.2 A-stabilité du schéma d'Euler implicite](#)
  - [4.3 A-stabilité du schéma d'Euler modifié](#)
  - [4.4 A-stabilité du schéma de Cranck-Nicolson](#)
  - [4.5 A-stabilité du schéma de Heun](#)
  - [4.6 A-stabilité du schéma de Simpson implicite](#)
  - [4.7 A-stabilité du schéma de Simpson explicite](#)
- ▼ [5 Exemple](#)
  - [5.1 Solution exacte](#)
  - [5.2 Solution approchée par la méthode d'Euler explicite](#)

### 5.3 Solution approchée par la méthode d'Euler implicite

#### 6 Exercice (problème raide)

##### ▼ 7 Exercice

7.1 Correction

7.2 Correction

7.3 Correction

7.4 Correction

##### ▼ 8 Exercice

8.1 Correction

8.2 Correction

##### ▼ 9 Exercice

9.1 Correction

9.2 Correction

9.3 Correction

##### ▼ 10 Exercice

10.1 Correction

10.2 Correction

10.3 Correction

10.4 Correction

##### ▼ 11 Exercice

11.1 Correction

11.2 Correction

11.3 Correction

##### ▼ 12 Exercice : A-stabilité du schéma d'Euler explicite dans le cas d'un sys

12.1 Correction

12.2 Correction

12.3 Correction

##### ▼ 13 Exercice : A-stabilité d'un schéma

13.1 Correction

## 1 Problème mathématiquement mal posé : non unicité de la solution d'un problème de Cauchy

On dit qu'un problème de Cauchy est mathématiquement mal posé s'il existe plusieurs solutions.

Par exemple, on cherche une fonction  $y: t \in \mathbb{R}^+ \mapsto y(t) \in \mathbb{R}$  qui satisfait

$$\begin{cases} y'(t) = \sqrt[3]{y(t)}, & \forall t > 0, \\ y(0) = 0. \end{cases}$$

Il est simple de vérifier que les fonctions  $y_1(t) = 0$  et

$y_{2,3}(t) = \pm\sqrt[3]{8t^3/27}$ , pour tout  $t \geq 0$ , sont toutes les trois solution du problème de Cauchy donné.

Dans ce cas, lorsqu'on utilise une méthode numérique, on ne sait pas quelle solution ce schéma approche et différents schémas peuvent approcher différentes solutions.

### 1.1 Exercice

1. Montrer que le problème de Cauchy

$$\begin{cases} y'(t) = y^{1/2}(t), & t > 0 \\ y(0) = 0, \end{cases}$$

admet une infinité de solutions de classe  $C^1(\mathbb{R}^+)$ .

2. Parmi ces solutions, quelle solution approche-t-on si on utilise la méthode d'Euler explicite?
3. et la méthode d'Euler implicite?
4. Que se passe-t-il si la donnée initiale est  $y(0) = y_0 > 0$ ?

### Correction

La fonction  $\varphi(t, y) = \sqrt{y}$  n'est pas lipschitzienne par rapport à  $y$ , donc le théorème d'existence et unicité locale n'est pas valable au voisinage de  $(0, 0)$ .

L'EDO est à variables séparables, on peut donc expliciter toutes les solutions du problème de Cauchy.

- Elle admet une solution constante, la fonction  $y(t) = 0$  pour tout  $t \in \mathbb{R}^+$ ,
- et des solutions de la forme  $y(t) = \frac{1}{4}(t + c)^2$  pour tout  $t \geq c$ .

En imposant la CI on trouve que, pour tout  $b \in \mathbb{R}^+$ , les fonctions

$$y_b(t) = \begin{cases} 0, & \text{si } 0 \leq t \leq b, \\ \frac{1}{4}(t - b)^2, & \text{si } t \geq b, \end{cases}$$

sont de classe  $C^1(\mathbb{R}^+)$  et sont solution du problème de Cauchy donné.

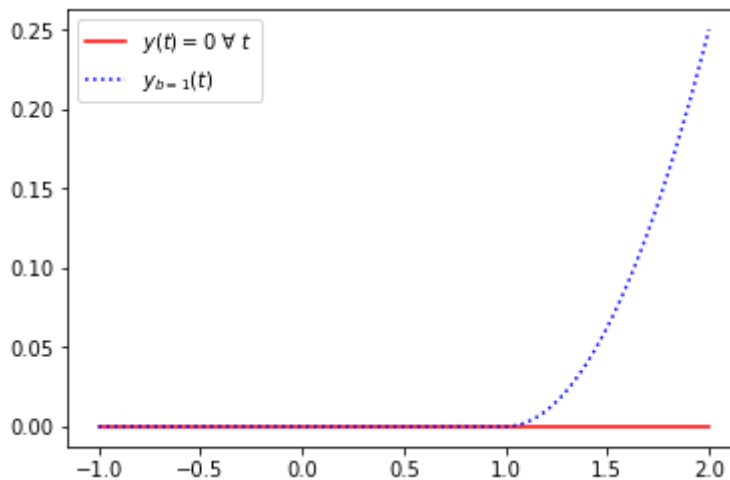
Traçons quelques courbes:

In [2]:

```

%matplotlib inline
from matplotlib.pyplot import *
b=1
xx=linspace(-1,2,101)
yyr=[0 for x in xx]
def f(x,b):
    if x<=b:
        return 0
    else:
        return (x-b)**2/4
yyb=[f(x,b) for x in xx]
plot(xx,yyr,'r-',xx,yyb,'b:');
legend(['r'$y(t)=0$ $\forall t$', '$y_{b=1}(t)$']);

```



La méthode d'Euler explicite construit la suite

$$\begin{cases} u_0 = y_0 = 0, \\ u_{n+1} = u_n + h\varphi(t_n, u_n) = u_n + hu_n^{1/2}, \quad n = 0, 1, 2, \dots, N_h - 1 \end{cases}$$

par conséquent  $u_n = 0$  pour tout  $n$ . La méthode d'Euler explicite approche la solution constante  $y(t) = 0$  pour tout  $t \in \mathbb{R}^+$ .

La méthode d'Euler implicite construit la suite

$$\begin{cases} u_0 = y_0 = 0, \\ u_{n+1} = u_n + h\varphi(t_{n+1}, u_{n+1}) = u_n + hu_{n+1}^{1/2}, \quad n = 0, 1, 2, \dots, N_h - 1. \end{cases}$$

par conséquent  $u_0 = 0$  mais  $u_1$  dépend de la méthode de résolution de l'équation implicite  $x = 0 + h\sqrt{x}$ . Bien sur  $x = 0$  est une solution mais  $x = h^2$  est aussi solution. Si le schéma choisit  $u_1 = h^2$ , alors  $u_n > 0$  pour tout  $n \in \mathbb{N}^*$ .

Notons que le problème de Cauchy avec une CI  $y(0) = y_0 > 0$  admet une et une seule solution, la fonction  $y(t) = \frac{1}{4}(t + 2\sqrt{y_0})^2$ . Dans ce cas, les deux schémas approchent forcément la même solution.

## 2 Problème mathématiquement bien posé mais numériquement mal posé

Une fois calculée la solution numérique  $\{u_n\}_{n=1}^{N_h}$  d'un problème de Cauchy, il est légitime de chercher à savoir dans quelle mesure l'erreur  $|y(t_n) - u_n|$  est petite pour  $n = 1, 2, \dots$ . Cela dépend bien sûr du schéma choisi mais aussi du problème en question. Essayons de voir cela sur un exemple.

On se donne  $\varphi(t, y(t)) = 3t - 3y(t)$  et  $y_0 = \alpha$  (un nombre quelconque). On cherche une fonction  $y: t \in \mathbb{R} \mapsto y(t) \in \mathbb{R}$  qui satisfait

$$\begin{cases} y'(t) = 3t - 3y(t), & \forall t \in \mathbb{R}, \\ y(0) = \alpha. \end{cases}$$

Sa solution, définie sur  $\mathbb{R}$ , est donnée par

$y(t) = (\alpha + 1/3)e^{-3t} + t - 1/3$ . En effet on a bien

$$y(0) = (\alpha + 1/3)e^0 + 0 - 1/3 = \alpha,$$

$$y'(t) = -3(\alpha + 1/3)e^{-3t} + 1 = -3(\alpha + 1/3)e^{-3t} + 1 - 3t + 3t = -3y$$

Si nous cherchons à résoudre le problème de Cauchy jusqu'à  $t = 10$  avec  $\alpha = 1/3$ , nous obtenons  $y(10) = 10 + 1/3 = 31/3$ . Par contre, si nous faisons le calcul avec l'approximation  $\alpha = 0.333333$  au lieu de  $1/3$ , nous avons

$$y(10) = (0.333333 - 1/3)e^{30} + 10 + 1/3 = -e^{30}/3000000 + 31/3$$

ce qui représente une différence avec la précédente valeur de

$$e^{30}/3000000 \approx 10^7/3. \text{ Cet exemple nous apprend qu'une petite erreur}$$

sur la condition initiale (erreur relative d'ordre  $10^{-6}$ ) peut provoquer une

très grande erreur sur  $y(10)$  (erreur relative d'ordre  $10^6$ ). Ainsi, si le

calculateur mis à notre disposition ne calcule qu'avec 6 chiffres significatifs

(en virgule flottante), alors  $\alpha = 1/3$  devient  $\alpha = 0.333333$  et il est inutile

d'essayer d'inventer une méthode numérique pour calculer  $y(10)$ . En effet,

la seule erreur sur la condition initiale provoque déjà une erreur

inadmissible sur la solution. Nous sommes en présence ici d'un problème

**numériquement mal posé**

Voici d'autres exemples :

Considérons le problème de Cauchy

$$\begin{cases} y'(t) = -y(t), & \forall t > 0, \\ y(0) = y_0 + \varepsilon. \end{cases}$$

La solution est  $y(t) = y_0 e^{-t} + \varepsilon e^{-t}$ : l'effet de la perturbation  $\varepsilon$  s'atténue lorsque  $t \rightarrow +\infty$  puisque  $\varepsilon e^{-t} \xrightarrow{t \rightarrow +\infty} 0$ . Cela suggère que si une erreur

est faite dans une étape d'une méthode d'itération, l'effet de cette erreur

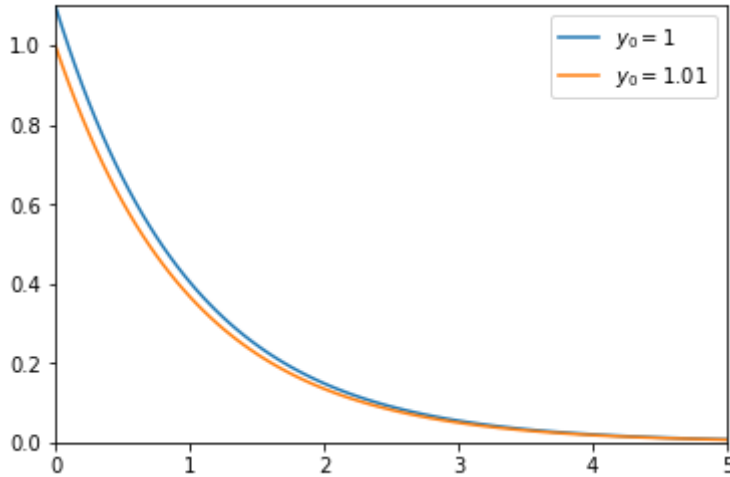
s'atténue au cours du temps: **le problème est numériquement bien posé.**

In [3]:

```

%matplotlib inline
from matplotlib.pyplot import *
tt=linspace(0,5,101)
yye=[(1+0.1)*exp(-t) for t in tt]
yy0=[(1)*exp(-t) for t in tt]
axis([0,5,0,1.1])
plot(tt,yye,tt,yy0);
legend(["$y_0=1$", "$y_0=1.01$"]);

```



Considérons maintenant le problème de Cauchy

$$\begin{cases} y'(t) = y(t), & \forall t > 0, \\ y(0) = y_0 + \varepsilon. \end{cases}$$

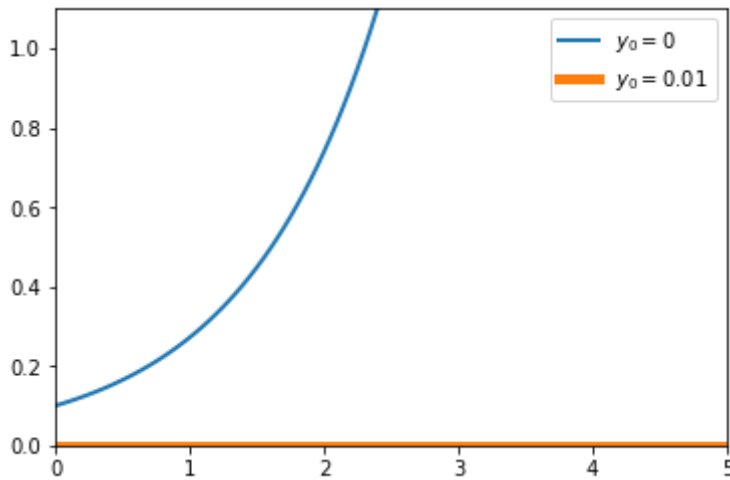
La solution est  $y(t) = y_0 e^t + \varepsilon e^t$ : l'effet de la perturbation  $\varepsilon$  s'amplifie lorsque  $t \rightarrow +\infty$  puisque  $\varepsilon e^t \xrightarrow{t \rightarrow +\infty} +\infty$ . Cela suggère que si une erreur est faite dans une étape d'une méthode d'itération, l'effet de cette erreur s'amplifie au cours du temps: **le problème est numériquement mal posé.**

In [4]:

```

%matplotlib inline
from matplotlib.pyplot import *
tt=linspace(0,5,101)
yye=[(0+0.1)*exp(t) for t in tt]
yy0=[(0)*exp(t) for t in tt]
axis([0,5,0,1.1])
plot(tt,yye,lw='2')
plot(tt,yy0,lw='5');
legend(["$y_0=0$", "$y_0=0.01$"]);

```



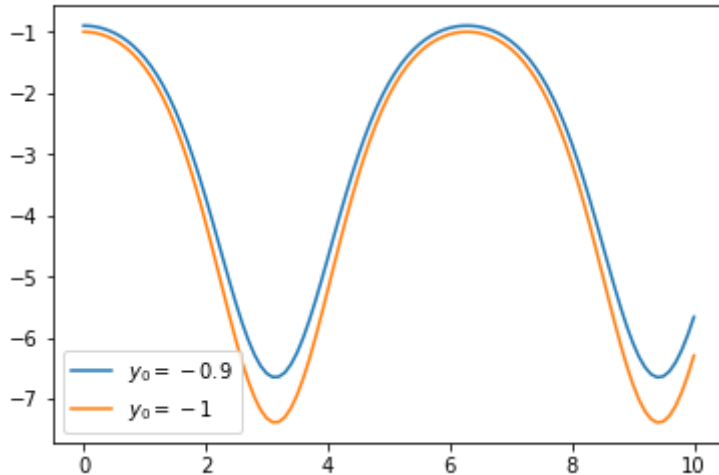
Considérons le problème de Cauchy

$$\begin{cases} y'(t) = y(t) \sin(t), & \forall t > 0, \\ y(0) = -1 + \varepsilon. \end{cases}$$

La solution est  $y(t) = -e^{1-\cos(t)} + \varepsilon e^{1-\cos(t)}$ : l'effet de la perturbation  $\varepsilon$  reste borné lorsque  $t \rightarrow +\infty$  puisque  $|\varepsilon e^{1-\cos(t)}| \leq \varepsilon e^2$ . Cela suggère que si une erreur est faite dans une étape d'une méthode d'itération, l'effet de cette erreur reste borné au cours du temps: **le problème est numériquement bien posé.**

In [5]:

```
%matplotlib inline
from matplotlib.pyplot import *
tt=linspace(0,10,101)
yye=[(-1+0.1)*exp(1-cos(t)) for t in tt]
yy0=[(-1)*exp(1-cos(t)) for t in tt]
# axis([0,5,-5,0])
plot(tt,yye,tt,yy0);
legend(["$y_0=-0.9$", "$y_0=-1$"]);
```



## 2.1 Exercice

L'objectif de cet exercice est de bien mettre en évidence l'influence d'une petite perturbation de la condition initiale sur la solution d'un problème numériquement mal posé et d'appliquer la méthode d'Euler explicite pour en approcher la solution exacte.

Considérons le problème de Cauchy

$$\begin{cases} y'(t) = 3\frac{y(t)}{t} - \frac{5}{t^3}, \\ y(1) = a \end{cases}$$

1. Calculer l'unique solution exacte à ce problème de Cauchy en fonction de  $a$  pour  $t > 0$ . Remarquer qu'avec le changement de variable  $u(t) = \frac{y(t)}{t}$  on obtient une EDO linéaire d'ordre 1
2. Tracer la solution exacte sur l'intervalle  $[1; 5]$  pour  $a = 1$ ,  $a = 1 - \frac{1}{1000}$  et  $a = 1 - \frac{1}{100}$ .
3. Pour  $a = 1$ , sur un autre graphique tracer la solution exacte et les solutions obtenues par la méthode d'Euler explicite avec  $N_h = 20$ ,  $N_h = 100$ ,  $N_h = 300$ ,  $N_h = 5000$ . Comment expliquez-vous ce comportement du schéma?

## Correction



Calculons les solutions de l'edo pour  $t > 0$ . On pose  $u(t) = \frac{y(t)}{t}$  ainsi  $y(t) = tu(t)$  et  $y'(t) = u(t) + tu'(t)$ . On obtient l'edo linéaire d'ordre 1 suivante:

$$tu'(t) - 2u(t) = -5t^{-3}.$$

On a  $a(t) = t$ ,  $b(t) = -2$  et  $g(t) = -5t^{-3}$  donc

- $A(t) = \int -2/t dt = -2 \ln(t) = \ln(t^{-2})$ ,
- $B(t) = \int -5t^{-4} e^{A(t)} dt = \int -5t^{-6} dt = t^{-5}$ . On conclut que  $u(t) = ct^2 + t^{-3}$  et  $y(t) = tu(t) = ct^3 + t^{-2}$ .

En imposant la condition  $a = y(1)$  on trouve l'unique solution du problème de Cauchy donné:

$$y(t) = (a - 1)t^3 + \frac{1}{t^2}.$$

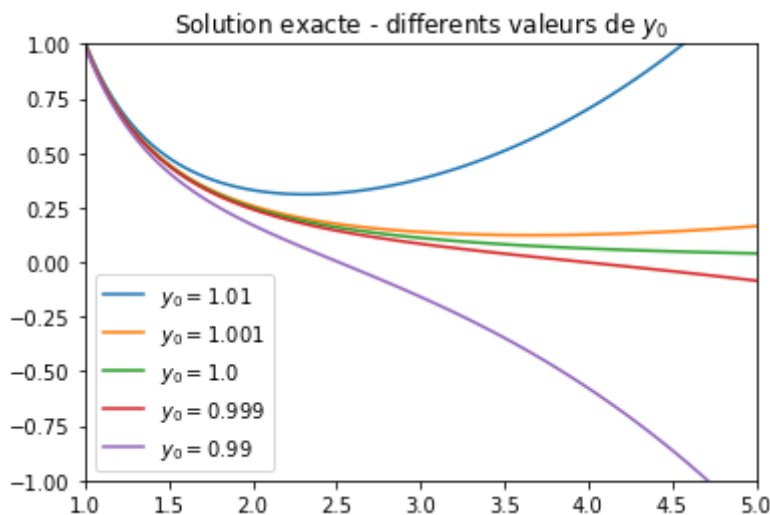
In [6]:

```
%matplotlib inline
from math import *
from matplotlib.pyplot import *

sol_exacte = lambda t,y0 : (y0-1.)*t**3+1./t**2

# INITIALISATION
t0 = 1.
tfinal = 5.
c=1.

tt=linspace(t0,tfinal,101)
Y0=[c+1.e-2,c+1.e-3,c,c-1.e-3,c-1.e-2]
for y0 in Y0:
    yy=[sol_exacte(t,y0) for t in tt]
    plot(tt,yy,label=('$y_0=$'+str(y0)))
legend()
title('Solution exacte - differents valeurs de $y_0$')
axis([t0,tfinal,-c,c]);
```



On constate qu'une petite perturbation de la condition initiale entraîne une

grosse perturbation de la solution, en effet

$$\lim_{t \rightarrow +\infty} y(t) = \begin{cases} +\infty & \text{si } a > 1, \\ 0^+ & \text{si } a = 1, \\ -\infty & \text{si } a < 1. \end{cases}$$

Si  $a = 1$ , la solution exacte est  $y(t) = t^{-2}$  mais **le problème est numériquement mal posé** car toute (petite) erreur de calcul a le même effet qu'une perturbation de la condition initiale: on "réveille" le terme dormant  $t^3$ .

In [7]:

```

from scipy.optimize import fsolve

def euler_progressif(phi,tt):
    h=tt[1]-tt[0]
    uu = [y0]
    for i in range(len(tt)-1):
        uu.append(uu[i]+h*phi(tt[i],uu[i]))
    return uu

def euler_regressif(phi,tt):
    h=tt[1]-tt[0]
    uu = [y0]
    for i in range(len(tt)-1):
        uu.append(fsolve(lambda x : -x +uu[i]+h*phi(tt[i+1],x)
    return uu

phi = lambda t,y : 3*y/t-5./t**3

y0=Y0[2]

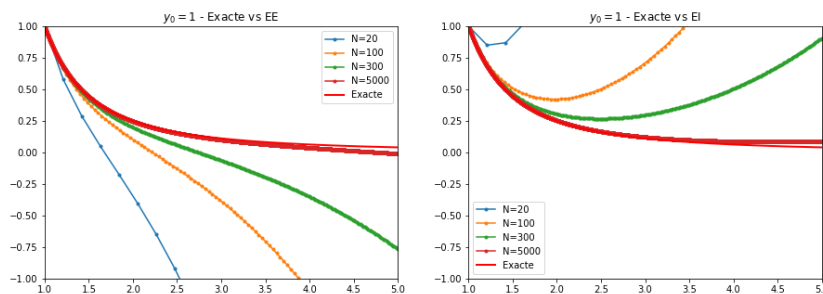
NN=[20,100,300,5000]

figure(1, figsize=(15, 5))

subplot(1,2,1)
for N in NN:
    tt=linspace(t0,tfinal,N)
    uu=euler_progressif(phi,tt)
    plot(tt,uu,'.-')
plot(tt,[sol_exacte(t,y0) for t in tt], 'r-',lw=2)
legend(['N=20', 'N=100', 'N=300', 'N=5000', 'Exacte'])
title('$y_0=1$ - Exacte vs EE')
axis([t0,tfinal,-c,c]);

subplot(1,2,2)
for N in NN:
    tt=linspace(t0,tfinal,N)
    uu=euler_regressif(phi,tt)
    plot(tt,uu,'.-')
plot(tt,[sol_exacte(t,y0) for t in tt], 'r-',lw=2)
legend(['N=20', 'N=100', 'N=300', 'N=5000', 'Exacte'])
title('$y_0=1$ - Exacte vs EI')
axis([t0,tfinal,-c,c]);

```



### 3 Stabilité : zéro-stabilité et A-

# stabilité

Considérons un problème de Cauchy et supposons que l'on ait montré l'existence d'une solution  $y$  (donc il est mathématiquement bien posé) et qu'il est numériquement bien posé. Deux questions naturelles se posent:

- que se passe-t-il lorsqu'on fixe le temps final  $T$  et on fait tendre le pas  $h$  vers 0?
- que se passe-t-il lorsqu'on fixe le pas  $h > 0$  mais on fait tendre  $T$  vers l'infini?

Dans les deux cas le nombre de nœuds tend vers l'infini mais dans le premier cas on s'intéresse à l'erreur en chaque point, dans le deuxième cas il s'agit du comportement asymptotique de la solution et de son approximation.

Ces deux questions font intervenir deux notions de stabilité:

- **Zéro-stabilité:** la propriété de zéro-stabilité assure que la méthode numérique est peu sensible aux petites perturbations des données (encore faut-il que le problème soit **numériquement bien posé**). L'exigence d'avoir une méthode numérique stable provient avant tout de la nécessité de contrôler les (inévitables) erreurs introduites par l'arithmétique finie des ordinateurs. En effet, si la méthode numérique n'était pas zéro-stable, les erreurs d'arrondi commises sur  $y_0$  et sur le calcul de  $\varphi(t_n, y_n)$  rendraient la solution calculée inutilisable.
- **A-stabilité:** la zéro-stabilité s'intéresse à la résolution du problème de Cauchy sur des intervalles bornés. Dans ce cadre, le nombre  $N_h$  de sous-intervalles ne tend vers l'infini que quand  $h$  tend vers zéro. Il existe cependant de nombreuses situations dans lesquelles le problème de Cauchy doit être intégré sur des intervalles en temps très grands ou même infini. Dans ce cas, même pour  $h$  fixé,  $N_h$  tend vers l'infini. On s'intéresse donc à des méthodes capables d'approcher la solution pour des intervalles en temps arbitrairement grands, même pour des pas de temps  $h$  "assez grands" (si on peut choisir  $h > 0$  quelconque, on dira que la méthode est inconditionnellement A-stable, sinon on aura une condition de stabilité sur  $h$  qui déterminera le temps de calcul). Considérons un problème de Cauchy dont la solution exacte vérifie la propriété  $y(t) \longrightarrow 0$ . Soit  $h > 0$  fixé et considérons la limite  $T \rightarrow +\infty$  (ainsi  $N_h \xrightarrow{t \rightarrow +\infty} +\infty$ ). On dit que la méthode est A-stable si  $u_n^{(h)} \xrightarrow{n \rightarrow +\infty} 0$ .

## 4 Étude de la A-stabilité pour les schémas à un pas

Soit  $\beta > 0$  un nombre réel positif et considérons le problème de Cauchy

$$\begin{cases} y'(t) = -\beta y(t), & \text{pour } t > 0, \\ y(0) = y_0 \end{cases}$$

où  $y_0 \neq 0$  est une valeur donnée. Sa solution est  $y(t) = y_0 e^{-\beta t}$  donc

$$\lim_{t \rightarrow +\infty} y(t) = 0.$$

Soit  $h > 0$  un pas de temps donné,  $t_n = nh$  pour  $n \in \mathbb{N}$  et notons  $u_n \approx y(t_n)$  une approximation de la solution  $y$  au temps  $t_n$ .

Si, sous d'éventuelles conditions sur  $h$ , on a

$$\lim_{n \rightarrow +\infty} u_n = 0,$$

alors on dit que **le schéma est A-stable**.

On peut tirer des conclusions analogues quand  $\beta$  est un complexe ou une fonction positive de  $t$ .

D'autre part, en général il n'y a aucune raison d'exiger qu'une méthode numérique soit absolument stable quand on l'applique à un autre problème. Cependant, on peut montrer que quand une méthode absolument stable sur le problème modèle est utilisée pour un problème modèle généralisé, l'erreur de perturbation (qui est la valeur absolue de la différence entre la solution perturbée et la solution non perturbée) est bornée uniformément (par rapport à  $h$ ). En bref, on peut dire que les méthodes absolument stables permettent de contrôler les perturbations.

**Remarque:** ce résultat peut être généralisé à un système de  $n$  EDO de la forme

$$\mathbf{y}'(t) = -\mathbb{A}\mathbf{y}(t)$$

où  $\mathbb{A}$  est une matrice constante ayant  $n$  valeurs propres positives  $\lambda_i$ ,  $i = 1, 2, \dots, n$ .

La solution s'écrit

$$\mathbf{y}(t) = \sum_i c_i \mathbf{v}_i e^{-\lambda_i t}$$

où  $\mathbf{v}_i$  est le  $i$ -ème vecteur propre associé à la valeur propre  $\lambda_i$  donc

$$\lim_{t \rightarrow +\infty} y_i(t) = 0 \quad \forall i = 1, 2, \dots, n.$$

#### 4.1 A-stabilité du schéma d'Euler explicite

Le schéma d'Euler progressif devient

$$u_{n+1} = (1 - \beta h)u_n, \quad n = 0, 1, 2, \dots$$

et par suite

$$u_n = (1 - \beta h)^n u_0, \quad n = 0, 1, 2, \dots$$

Par conséquent,  $\lim_{n \rightarrow +\infty} u_n = 0$  si et seulement si

$$|1 - \beta h| < 1,$$

ce qui a pour effet de limiter  $h$  à

$$h < \frac{2}{\beta}.$$

Cette condition de A-stabilité limite le pas  $h$  d'avance en  $t$  lorsqu'on utilise le schéma d'Euler progressif. Notons que si  $1 - \beta h > 1$  alors  $u_n$  tend vers  $+\infty$  lorsque  $t$  tend vers l'infini et si  $1 - \beta h < -1$  alors  $u_n$  tend vers l'infini en alternant de signe lorsque  $t$  tend vers l'infini. Nous dirons dans ces cas que le schéma d'Euler progressif est instable.

**Remarque:** dans le cas du système, le schéma d'Euler progressif est A-stable ssi

$$h < \frac{2}{\lambda_{\max}}.$$

## 4.2 A-stabilité du schéma d'Euler implicite

Le schéma d'Euler rétrograde devient dans le cadre de notre exemple

$$(1 + \beta h)u_{n+1} = u_n, \quad n = 0, 1, 2, \dots$$

et par suite

$$u_n = \frac{1}{(1 + \beta h)^n} u_0, \quad n = 0, 1, 2, \dots$$

Dans ce cas nous voyons que pour tout  $h > 0$  nous avons  $\lim_{n \rightarrow \infty} u_n = 0$ , le schéma d'Euler rétrograde est donc toujours stable, sans limitations sur  $h$ .

## 4.3 A-stabilité du schéma d'Euler modifié

Le schéma d'Euler modifié pour notre exemple devient

$$u_{n+1} = u_n + h \left( -\beta \left( u_n + \frac{h}{2} (-\beta u_n) \right) \right) = \left( 1 - \beta h + \frac{(\beta h)^2}{2} \right) u_n$$

Par induction on obtient

$$u_n = \left( 1 - \beta h + \frac{(\beta h)^2}{2} \right)^n u_0.$$

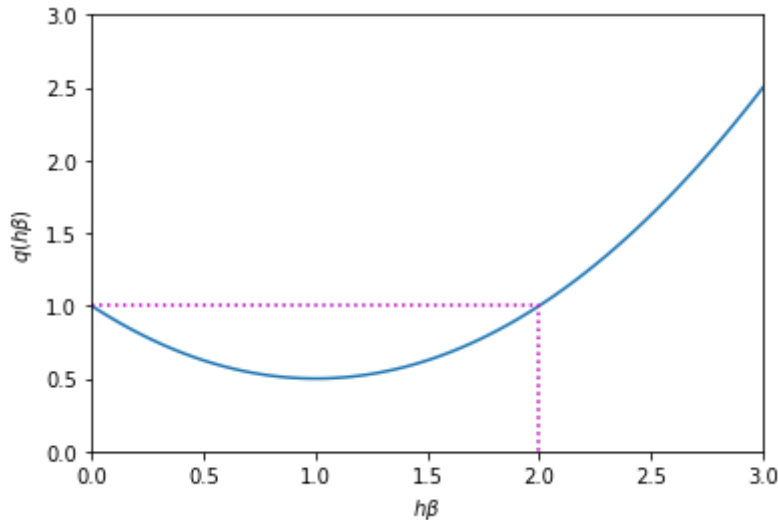
Par conséquent,  $\lim_{n \rightarrow +\infty} u_n = 0$  si et seulement si

$$\left| 1 - \beta h + \beta^2 \frac{(h)^2}{2} \right| < 1.$$

Notons  $x$  le produit  $\beta h$  et  $q$  le polynôme  $q(x) = \frac{1}{2}x^2 - x + 1$  dont le graphe est représenté en figure:

In [8]:

```
%matplotlib inline
from matplotlib.pyplot import *
xx=linspace(0,3,101)
yy=[1-x+0.5*x**2 for x in xx]
plot(xx,yy,[0,2,2],[1,1,0], 'm:')
axis([0,3,0,3])
xlabel(r'$h\beta$')
ylabel(r'$q(h\beta)$');
```



Nous avons  $|q(x)| < 1$  si et seulement si  $0 < x < 2$ . La relation

$\lim_{n \rightarrow +\infty} u_n = 0$  est donc satisfaite si et seulement si

$$h < \frac{2}{\beta}.$$

Cette condition de stabilité limite le pas  $h$  d'avance en  $t$  lorsqu'on utilise le schéma d'Euler modifié.

#### 4.4 A-stabilité du schéma de Cranck-Nicolson

Le schéma de Crank-Nicolson appliqué à notre exemple s'écrit

$$\left(1 + \beta \frac{h}{2}\right) u_{n+1} = \left(1 - \beta \frac{h}{2}\right) u_n$$

et par suite

$$u_n = \left(\frac{2 - \beta h}{2 + \beta h}\right)^n u_0, \quad n = 0, 1, 2, \dots$$

Par conséquent,  $\lim_{n \rightarrow +\infty} u_n = 0$  si et seulement si

$$\left|\frac{2 - \beta h}{2 + \beta h}\right| < 1.$$

Notons  $x$  le produit  $\beta h > 0$  et  $q$  la fonction  $q(x) = \frac{2-x}{2+x} = 1 - 2\frac{x}{2+x}$ .

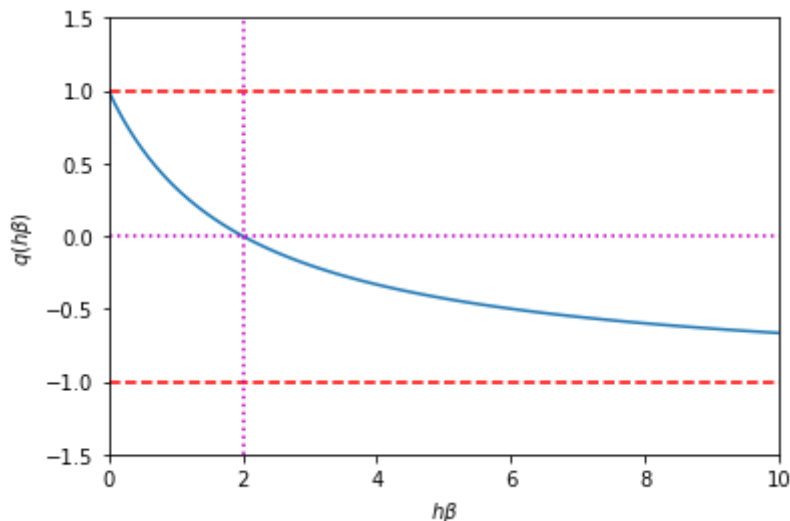
Nous avons  $0 < \frac{x}{2+x} < 1$  pour tout  $x \in \mathbb{R}_+^*$ , donc  $|q(x)| < 1$  pour tout  $x \in \mathbb{R}_+^*$ . La relation  $\lim_{n \rightarrow +\infty} u_n = 0$  est donc satisfaite pour tout  $h > 0$ : le

schéma de Crank-Nicolson est donc toujours stable, sans limitations sur  $h$ .  
Cependant, si on cherche  $0 < q < 1$  on a (pour  $x > 0$ )

$$1 - 2\frac{x}{2+x} > 0 \iff x < 2$$

In [9]:

```
%matplotlib inline
from matplotlib.pyplot import *
xx=linspace(0,10,101)
yy=[1-2*x/(2+x) for x in xx]
plot(xx,yy)
plot([0,10],[1,1], 'r--', [0,10],[-1,-1], 'r--')
plot([0,10],[0,0], 'm:', [2,2],[-1.5,1.5], 'm:')
axis([0,10,-1.5,1.5])
xlabel(r'$h\beta$')
ylabel(r'$q(h\beta)$');
```



#### 4.5 A-stabilité du schéma de Heun

Le schéma de Heun pour notre exemple devient

$$u_{n+1} = u_n + \frac{h}{2}(-\beta u_n - \beta(u_n + h(-\beta u_n))) = \left(1 - \beta h + \frac{(\beta h)^2}{2}\right) u_n$$

ce qui coïncide avec la méthode d'Euler modifiée. Par conséquent,

$\lim_{n \rightarrow +\infty} u_n = 0$  si et seulement si

$$h < \frac{2}{\beta}.$$

Cette condition de stabilité limite le pas  $h$  d'avance en  $t$  lorsqu'on utilise le schéma de Heun.

#### 4.6 A-stabilité du schéma de Simpson implicite

Le schéma de Simpson implicite appliqué à notre exemple s'écrit



$$\left(1 + \beta \frac{h}{6}\right) u_{n+1} = \left(1 - \frac{5}{6}\beta h + \frac{1}{3}(\beta h)^2\right) u_n$$

et par suite

$$u_n = \left(\frac{1 - \frac{5}{6}\beta h + \frac{1}{3}(\beta h)^2}{1 + \beta \frac{h}{6}}\right)^n u_0, \quad n = 0, 1, 2, \dots$$

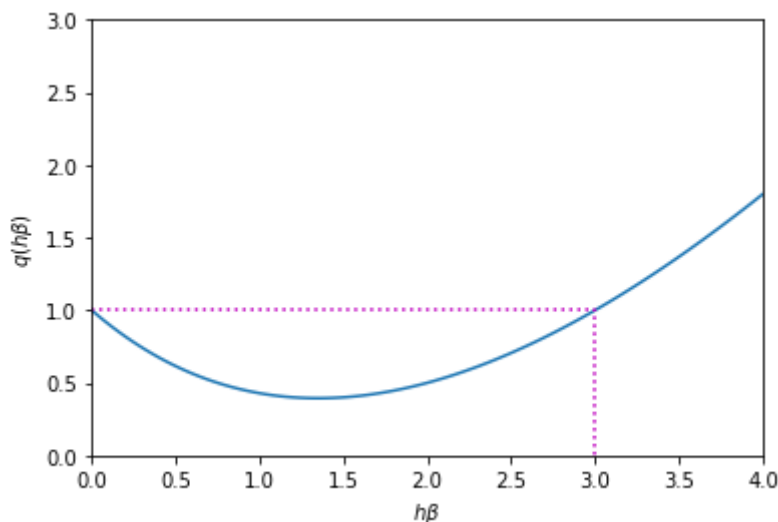
Par conséquent,  $\lim_{n \rightarrow +\infty} u_n = 0$  si et seulement si

$$\left|\frac{6 - 5\beta h + 2(\beta h)^2}{6 + \beta h}\right| < 1.$$

Notons  $x$  le produit  $\beta h > 0$  et  $q$  la fonction  $q(x) = \frac{6-5x+2x^2}{6+x}$  dont le graphe est représenté en figure.

In [10]:

```
%matplotlib inline
from matplotlib.pyplot import *
xx=linspace(0,4,101)
yy=[(6-5*x+2*x**2)/(6+x) for x in xx]
plot(xx,yy,[0,3,3],[1,1,0], 'm:')
axis([0,4,0,3])
xlabel(r'$h\beta$')
ylabel(r'$q(h\beta)$');
```



Nous avons  $q(x) > 0$  pour tout  $x \in \mathbb{R}_+^*$  et  $q(x) < 1$  si et seulement si  $x < 3$ , donc  $|q(x)| < 1$  pour tout  $x \in ]0; 3[$ . Par conséquent,

$\lim_{n \rightarrow +\infty} u_n = 0$  si et seulement si

$$h < \frac{3}{\beta}.$$

Cette condition de stabilité limite le pas  $h$  d'avance en  $t$ .

#### 4.7 A-stabilité du schéma de Simpson explicite

Le schéma de Simpson explicite pour notre exemple devient

$$u_{n+1} = u_n + \frac{h}{6} \left( -\beta u_n - 4\beta \left( u_n + \frac{h}{2}(-\beta u_n) \right) - \beta (u_n + h(-\beta u_n)) \right)$$

Par induction on obtient

$$u_n = \left( 1 - \beta h + \frac{3}{2}(\beta h)^2 \right)^n u_0.$$

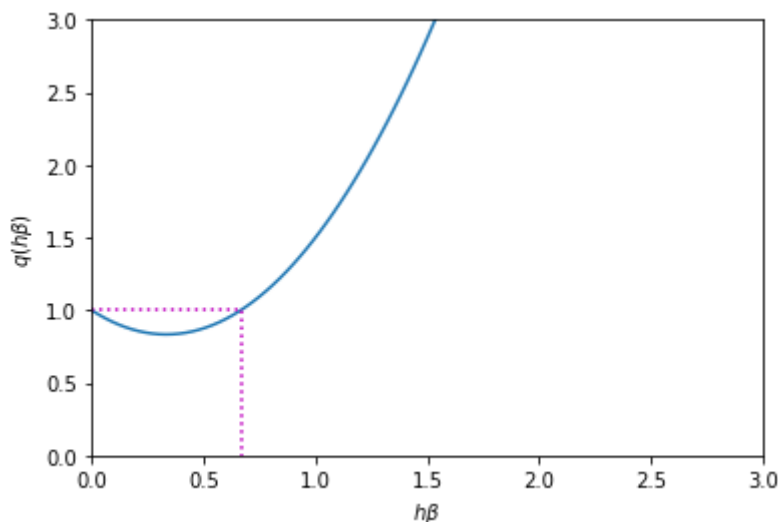
Par conséquent,  $\lim_{n \rightarrow +\infty} u_n = 0$  si et seulement si

$$\left| 1 - \beta h + \frac{3}{2}(\beta h)^2 \right| < 1.$$

Notons  $x$  le produit  $\beta h$  et  $q$  le polynôme  $q(x) = \frac{3}{2}x^2 - x + 1$  dont le graphe est représenté en figure.

In [11]:

```
%matplotlib inline
from matplotlib.pyplot import *
xx=linspace(0,3,101)
yy=[1-x+1.5*x**2 for x in xx]
plot(xx,yy,[0,2/3,2/3],[1,1,0], 'm:')
axis([0,3,0,3])
xlabel(r'$h\beta$')
ylabel(r'$q(h\beta)$');
```



Nous avons  $|q(x)| < 1$  si et seulement si  $0 < x < \frac{2}{3}$ . La relation

$\lim_{n \rightarrow +\infty} u_n = 0$  est donc satisfaite si et seulement si

$$h < \frac{2}{3\beta}.$$

Cette condition de stabilité limite le pas  $h$  d'avance en  $t$ .

## 5 Exemple

On considère le problème de Cauchy

$$\begin{cases} y'(t) = -y(t), \\ y(0) = 1, \end{cases}$$

sur l'intervalle  $[0; 12]$ .

### 5.1 Solution exacte

Il s'agit d'une EDO à variables séparables. L'unique solution constante de l'EDO est la fonction  $y(t) \equiv 0$ , toutes les autres solutions sont du type  $y(t) = Ce^{-t}$ . Donc l'unique solution du problème de Cauchy est la fonction  $y(t) = e^{-t}$  définie pour tout  $t \in \mathbb{R}$ .

### 5.2 Solution approchée par la méthode d'Euler explicite

La méthode d'Euler explicite pour cette EDO s'écrit

$$u_{n+1} = (1 - h)u_n.$$

En procédant par récurrence sur  $n$ , on obtient

$$u_{n+1} = (1 - h)^{n+1}.$$

De la formule  $u_{n+1} = (1 - h)^{n+1}$  on déduit que

- si  $0 < h < 1$  alors la solution numérique est stable et convergente,
- si  $h = 1$  alors la solution numérique est stationnaire  $u_n = 0$  pour tout  $n \in \mathbb{N}^*$ ,
- si  $1 < h < 2$  alors la solution numérique oscille mais est encore convergente,
- si  $h = 2$  alors la solution numérique oscille, plus précisément on a  $u_{2n} = 1$  et  $u_{2n+1} = -1$  pour tout  $n \in \mathbb{N}^*$ ,
- si  $h > 2$  alors la solution numérique oscille et diverge. En effet, la méthode est A-stable si et seulement si  $|1 - h| < 1$ .

Remarque: la suite obtenue est une suite géométrique de raison  $q = 1 - h$ . On sait qu'une telle suite

- diverge si  $|q| > 1$  ou  $q = -1$ ,
- est stationnaire si  $q = 1$ ,
- converge vers 0 si  $|q| < 1$ .
- converge de façon monotone vers 0 si  $0 < q < 1$ .

Voyons graphiquement ce que cela donne avec différentes valeurs de  $h > 0$ :

- si  $h = 4$  alors  $t_n = 4n$  et  $u_n = (-4)^n$  tandis que  $y(t_n) = e^{-4n}$ ,
- si  $h = 2$  alors  $t_n = 2n$  et  $u_n = (-1)^n$  tandis que  $y(t_n) = e^{-2n}$ ,
- si  $h = 1$  alors  $t_n = n$  et  $u_n = 0$  tandis que  $y(t_n) = e^{-n}$ ,
- si  $h = \frac{1}{2}$  alors  $t_n = n/2$  et  $u_n = \left(\frac{1}{2}\right)^n$  tandis que  $y(t_n) = e^{-n/2}$ ,

- si  $h = \frac{1}{4}$  alors  $t_n = n/4$  et  $u_n = \left(\frac{3}{4}\right)^n$  tandis que  $y(t_n) = e^{-n/4}$ .

Ci-dessous sont tracées sur l'intervalle  $[0; 10]$ , les courbes représentatives de la solution exacte et de la solution calculée par la méthode d'Euler explicite. En faisant varier le pas  $h$  nous pouvons constater que si  $h > 1$  l'erreur commise entre la solution exacte et la solution calculée est amplifiée d'un pas à l'autre.

(Remarque: la première itérée a la même pente quelque soit le pas  $h$  (se rappeler de la construction géométrique de la méthode d'Euler)).

In [12]:

```

%matplotlib inline
from matplotlib.pyplot import *

# SCHEMA
def euler_progressif(phi,tt):
    h=tt[1]-tt[0]
    uu = [y0]
    for i in range(len(tt)-1):
        uu.append(uu[i]+h*phi(tt[i],uu[i]))
    return uu

phi = lambda t,y : -y
sol_exacte = lambda t : exp(-t)

# INITIALISATION
t0, y0, tfinal = 0 , 1 , 12

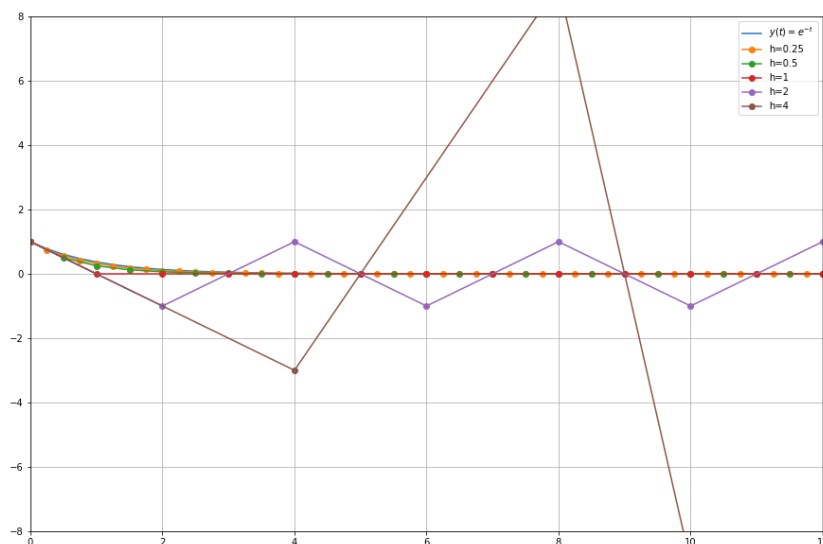
# CALCUL
H = [ 2**(k-2) for k in range(5) ]
tt = [] # liste de liste
uu = [] # liste de liste

for h in H:
    Nh = int((tfinal-t0)/h)
    tt.append( [ t0+i*h for i in range(Nh+1) ] )
    uu.append(euler_progressif(phi,tt[-1]))

# AFFICHAGE
figure(1, figsize=(15, 10))
yy = [sol_exacte(t) for t in tt[0]] # affichage de la sol exac
axis([t0, tfinal, -8, 8])

plot(tt[0],yy, '- ',label=('$y(t)=e^{-t}$'))
for k in range(5):
    plot(tt[k],uu[k], '-o',label=('h='+str(H[k])))
legend()
grid();

```



### 5.3 Solution approchée par la méthode d'Euler implicite

La méthode d'Euler implicite pour cette EDO s'écrit

$$u_{n+1} = \frac{1}{1+h} u_n.$$

En procédant par récurrence sur  $n$ , on obtient

$$u_{n+1} = \frac{1}{(1+h)^{n+1}}.$$

De la formule  $u_{n+1} = (1+h)^{-(n+1)}$  on déduit que la solution numérique est stable et convergente pour tout  $h$ . En effet, la méthode est inconditionnellement A-stable.

Remarque: la suite obtenue est une suite géométrique de raison  $q = 1/(1+h) \in ]0; 1[$ .

In [13]:

```
%matplotlib inline
from matplotlib.pyplot import *
from scipy.optimize import fsolve

# SCHEMA
def euler_regressif(phi,tt):
    h=tt[1]-tt[0]
    uu = [y0]
    for i in range(len(tt)-1):
        uu.append(fsolve(lambda x:-x+uu[i]+h*phi(tt[i+1],x),uu
    return uu

phi = lambda t,y : -y
sol_exacte = lambda t : exp(-t)

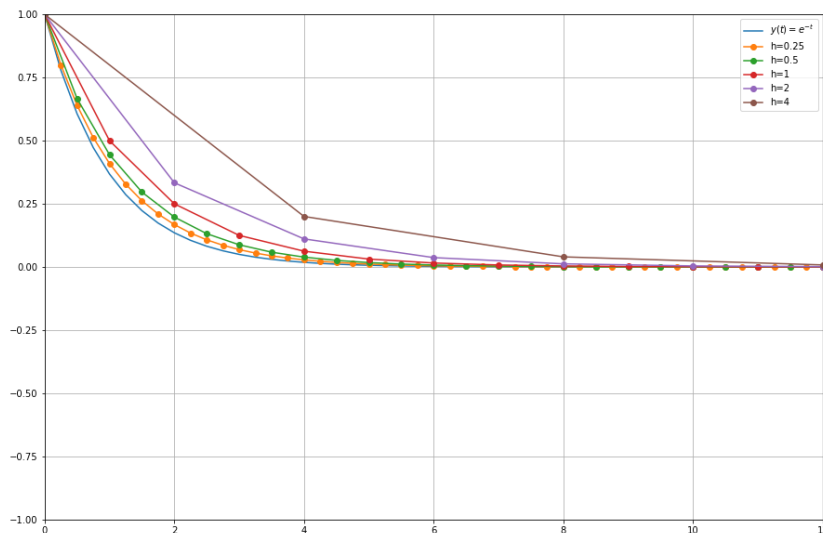
# INITIALISATION
t0, y0, tfinal = 0 , 1 , 12

# CALCUL
H = [ 2**(k-2) for k in range(5) ]
tt = [] # liste de liste
uu = [] # liste de liste

for h in H:
    Nh = int((tfinal-t0)/h)
    tt.append( [ t0+i*h for i in range(Nh+1) ] )
    uu.append(euler_regressif(phi,tt[-1]))

# AFFICHAGE
figure(1, figsize=(15, 10))
yy = [sol_exacte(t) for t in tt[0]] # affichage de la sol exacte
axis([t0, tfinal, -1, 1])

plot(tt[0],yy,'-',label=(' $y(t)=e^{-t}$ '))
for k in range(5):
    plot(tt[k],uu[k], '-o',label=('h='+str(H[k])))
legend()
grid();
```



## 6 Exercice (problème raide)

Soient  $b, g > 0$  et considérons le problème de Cauchy

$$\begin{cases} y'(t) = -by(t) + g, & t \in [0; 1] \\ y(0) = y_0. \end{cases}$$

### 1. Solution exacte

- Donner l'unique solution exacte à ce problème de Cauchy et tracer le graphe  $t \mapsto y(t)$  pour  $t \geq 0$  en fonction de  $y_0$ .
- Soient  $b = g = 10$ . Tracer avec python les solutions exactes sur l'intervalle  $[0; 1]$  pour  $y_0 = \frac{g}{b}$  et  $y_0 = \frac{g}{b} + 10^{-8}$  pour  $b = g = 10$ .
- Sur un graphe à coté tracer les mêmes courbes avec  $b = g = 40$ . Que peut-on dire lorsque  $b$  devient de plus en plus grand?

### 2. Méthode d'Euler explicite

- Vérifier que la méthode génère une suite arithmético-géométrique et calculer analytiquement pour quelles valeurs de  $h$  la suite ainsi construite converge et pour quelles valeurs la convergence est monotone.
- Soient  $b = g = 40$  et  $y_0 = \frac{g}{b}$ . Tracer les solutions approchées obtenues avec  $N_h = 19, N_h = 23, N_h = 45, N_h = 100$ .
- Répéter pour  $y_0 = \frac{g}{b} + 10^{-8}$  et expliquer chaque résultat.

### 3. Méthode d'Euler implicite

- Répéter le même exercice pour la méthode d'Euler implicite. Noter que, étant  $\varphi$  linéaire, la méthode peut être rendu explicite par un calcul élémentaire.

## Correction

### Solution exacte

Il s'agit d'une edo linéaire d'ordre 1 dont la solution est

$$y(t) = \left(y_0 - \frac{g}{b}\right) e^{-bt} + \frac{g}{b}$$

et

$$\lim_{t \rightarrow +\infty} y(t) = \frac{g}{b} \text{ pour tout } y_0 \in \mathbb{R}$$

le problème est numériquement bien posé, c'est-à-dire que l'erreur sur la solution sera au pire de l'ordre de l'erreur sur la condition initiale.

Lorsque  $b$  devient de plus en plus grand, la solution devient de plus en plus raide:



In [14]:

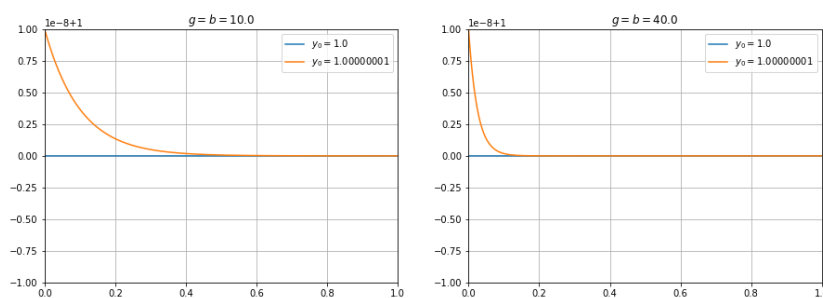
```
%reset -f
%matplotlib inline
from matplotlib.pyplot import *

exacte = lambda t,b,g,y0 : (y0-g/b)*exp(-b*t)+g/b

t0 = 0.
tfinal = 1.0
tt=linspace(t0,tfinal,10001)

figure(1, figsize=(15, 5))
subplot(1,2,1)
g=10.
b=10.
Y0=[g/b,g/b+1.e-8]
for y0 in Y0:
    yy=[exacte(t,b,g,y0) for t in tt]
    plot(tt,yy,label=(' $y_0=$'+str(y0)));
title("$g=b=$"+str(g))
axis([0,1,g/b-1.e-8,g/b+1.e-8])
legend()
grid()

subplot(1,2,2)
g=40.
b=40.
Y0=[g/b,g/b+1.e-8]
for y0 in Y0:
    yy=[exacte(t,b,g,y0) for t in tt]
    plot(tt,yy,label=(r'$y_0=$'+str(y0)));
title("$g=b=$"+str(g))
axis([0,1,g/b-1.e-8,g/b+1.e-8])
legend()
grid()
```



## Méthode d'Euler explicite

1. La méthode d'Euler explicite donne la suite définie par récurrence  $u_{n+1} = Au_n + B$  et  $u_0 = y_0$  avec  $A = 1 - bh$  et  $B = gh$ . Pour calculer  $u_n$  sans passer par la récurrence, on introduit une suite auxiliaire définie par  $v_n = u_n - \alpha$  ainsi  $u_n = v_n + \alpha$  pour tout  $n \in \mathbb{N}$ . On a  $v_{n+1} + \alpha = A(v_n + \alpha) + B = Av_n + A\alpha + B$ . En posant  $\alpha = \frac{B}{1-A}$  ( $A \neq 1$ ), la suite  $(v_n)_{n \in \mathbb{N}}$  est géométrique de raison  $A$  donc  $v_n = v_0 A^n$ . En remplaçant  $v_n$  par  $u_n - \alpha$  et  $v_0$  par  $u_0 - \alpha$  on en déduit que  $u_n = A^n(u_0 - \alpha) + \alpha$ ,  $\alpha = \frac{B}{1-A}$  soit encore

$$u_n - \frac{g}{b} = (1 - hb)^n \left( y_0 - \frac{g}{b} \right)$$

2. Pour que  $\lim_{t \rightarrow +\infty} |u_n - \frac{g}{b}| = 0$ , il est nécessaire de prendre  $|1 - hb| < 1$ , c'est-à-dire  $N_h > b/2$ . Pour que la convergence soit monotone il est nécessaire de prendre  $0 \leq (1 - hb) < 1$ , i.e  $N_h \geq b$ . Dans cet exemple, pour obtenir une bonne convergence il est nécessaire de prendre un pas  $h$  de plus en plus petit que  $b$  est grand, c'est-à-dire un coût en calcul plus élevé. On parle de **problème raide** ou mal conditionné.

In [15]:

```
%reset -f
%matplotlib inline
from matplotlib.pyplot import *

phi = lambda t,y,b,g : -b*y+g

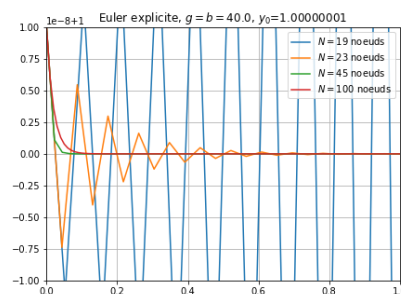
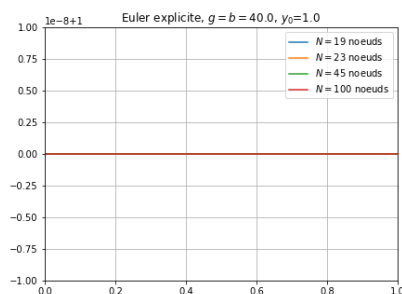
def euler_progressif(phi,tt,y0,b,g):
    h=tt[1]-tt[0]
    uu = [y0]
    for i in range(len(tt)-1):
        uu.append(uu[i]+h*phi(tt[i],uu[i],b,g))
    return uu

t0 = 0.
tfinal = 1.0

g=40.
b=40.
Y0=[g/b,g/b+1.e-8]
NN=[19,23,45,100]

figure(1, figsize=(15, 5))
subplot(1,2,1)
y0=Y0[0]
for N in NN:
    tt=linspace(t0,tfinal,N+1)
    yy=euler_progressif(phi,tt,y0,b,g)
    plot(tt,yy,label=('$N=$'+str(N)+' noeuds'));
title("Euler explicite, $g=b=$"+str(g)+ ", $y_0$="+str(y0))
axis([0,1,g/b-1.e-8,g/b+1.e-8])
legend()
grid()

subplot(1,2,2)
y0=Y0[1]
for N in NN:
    tt=linspace(t0,tfinal,N+1)
    yy=euler_progressif(phi,tt,y0,b,g)
    plot(tt,yy,label=('$N=$'+str(N)+' noeuds'));
title("Euler explicite, $g=b=$"+str(g)+ ", $y_0$="+str(y0))
axis([0,1,g/b-1.e-8,g/b+1.e-8])
legend()
grid()
```



## Méthode d'Euler implicite

1. La méthode d'Euler implicite donne la suite définie par récurrence

$$u_{n+1} = Au_n + B \text{ et } u_0 = y_0 \text{ avec } A = \frac{1}{1+bh} \text{ et } B = \frac{gh}{1+bh}. \text{ On}$$

introduit une suite auxiliaire définie par  $v_n = u_n - \alpha$  ainsi

$u_n = v_n + \alpha$  pour tout  $n \in \mathbb{N}$ . On a

$$v_{n+1} + \alpha = A(v_n + \alpha) + B = Av_n + A\alpha + B \text{ En posant}$$

$\alpha = \frac{B}{1-A}$  ( $A \neq 1$ ), la suite  $(v_n)_{n \in \mathbb{N}}$  est géométrique de raison  $A$

donc  $v_n = v_0 A^n$ . En remplaçant  $v_n$  par  $u_n - \alpha$  et  $v_0$  par  $u_0 - \alpha$  on

en déduit que  $u_n = A^n(u_0 - \alpha) + \alpha$ ,  $\alpha = \frac{B}{1-A}$  soit encore

$$u_n - \frac{g}{b} = \frac{1}{(1+hb)^n} \left( y_0 - \frac{g}{b} \right)$$

Pour que  $\lim_{t \rightarrow +\infty} |u_n - \frac{g}{b}| = 0$ , il est nécessaire de prendre

$\left| \frac{1}{1+hb} \right| < 1$ , c'est-à-dire pour n'importe quelle valeur de  $h$ . Pour que la convergence soit monotone il est nécessaire de prendre

$0 \leq \frac{1}{1+hb} < 1$ , c'est-à-dire pour n'importe quelle valeur de  $h$ . Dans

cet exemple, pour obtenir une bonne convergence il n'est plus

nécessaire de prendre un pas  $h$  de plus en plus petit que  $b$  est grand.

Cependant, pour une edo quelconque, à chaque pas de temps on doit résoudre une équation non-linéaire.

In [16]:

```
%reset -f
%matplotlib inline
from matplotlib.pyplot import *
from scipy.optimize import fsolve

phi = lambda t,y,b,g : -b*y+g

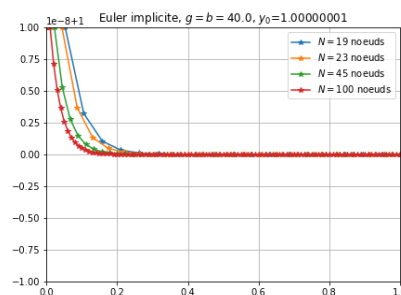
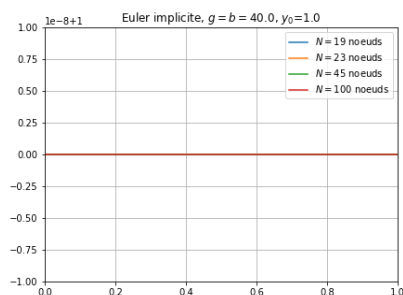
def euler_regressif(phi,tt,y0,b,g):
    h=tt[1]-tt[0]
    uu = [y0]
    for i in range(len(tt)-1):
        uu.append(g/b + (y0-g/b)/(1.+h*b)**i)
        #uu.append(fsolve(lambda x: x-uu[i]-h*phi(tt[i+1],x,b,
    return uu

t0 = 0.
tfinal = 1.0

g=40.
b=40.
Y0=[g/b,g/b+1.e-8]
NN=[19,23,45,100]

figure(1, figsize=(15, 5))
subplot(1,2,1)
y0=Y0[0]
for N in NN:
    tt=linspace(t0,tfinal,N+1)
    yy=euler_regressif(phi,tt,y0,b,g)
    plot(tt,yy,label=('$N=$'+str(N)+' noeuds'));
title("Euler implicite, $g=b=$"+str(g)+ " , $y_0$="+str(y0))
axis([0,1,g/b-1.e-8,g/b+1.e-8])
legend()
grid()

subplot(1,2,2)
y0=Y0[1]
for N in NN:
    tt=linspace(t0,tfinal,N+1)
    yy=euler_regressif(phi,tt,y0,b,g)
    plot(tt,yy,'-*',label=('$N=$'+str(N)+' noeuds'));
title("Euler implicite, $g=b=$"+str(g)+ " , $y_0$="+str(y0))
axis([0,1,g/b-1.e-8,g/b+1.e-8])
legend()
grid()
```



## 7 Exercice

Soit le problème de Cauchy:

$$\begin{cases} y'(t) + 10y(t) = 0, & \forall t \in \mathbb{R}, \\ y(0) = y_0 > 0. \end{cases}$$

1. Montrer qu'il existe une unique solution globale  $y \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R})$  que vous préciserez explicitement.
2. Soit le schéma numérique de Crank-Nicolson défini par la suite  $(u_n)_{n \in \mathbb{N}}$  vérifiant

$$\frac{u_{n+1} - u_n}{h} + 5(u_{n+1} + u_n) = 0, \quad \forall n \in \mathbb{N},$$

pour  $h > 0$  fixé.

- Montrer que la suite  $(u_n)_{n \in \mathbb{N}}$  est une suite géométrique dont on précisera la raison. Donner l'expression de  $u_n$  en fonction de  $n$ .
3. Montrer que la raison  $r$  de la suite vérifie  $|r| < 1$  pour tout  $h > 0$ . Ce schéma est-il inconditionnellement A-stable?
  4. Sous quelle condition sur  $h > 0$  le schéma génère-t-il une suite positive?

### 7.1 Correction

C'est un problème de Cauchy du type

$$\begin{cases} y'(t) = \varphi(t, y(t)), & \forall t \in \mathbb{R}, \\ y(0) = y_0 > 0, \end{cases}$$

avec  $\varphi(t, y) = g(y) = -10y$ .

On montre d'abord qu'il existe une et une seule solution locale (ie sur  $[-T; T]$ ) de classe  $\mathcal{C}^1([-T, T], \mathbb{R})$ .

On montre ensuite que cette solution est de classe  $\mathcal{C}^\infty([-T, T], \mathbb{R})$ .

On montre enfin que la solution admet un prolongement sur  $\mathbb{R}$ .

- Comme  $g \in \mathcal{C}^1(\mathbb{R}, \mathbb{R})$ , d'après le théorème de Cauchy-Lipschitz il existe  $T > 0$  et une unique solution  $y \in \mathcal{C}^1([-T, T], \mathbb{R})$ .
- Par récurrence, en exploitant l'EDO et la régularité de  $g$ , on grimpe en régularité sur  $y$  ainsi  $y \in \mathcal{C}^\infty([-T, T], \mathbb{R})$ .
- La fonction nulle est solution de l'EDO (mais non du problème de Cauchy donné). Par l'unicité de la solution du problème de Cauchy on en déduit que soit  $y(t) > 0$  pour tout  $t \in [-T, T]$  (ie lorsque  $y_0 > 0$ ) soit  $y(t) < 0$  pour tout  $t \in [-T, T]$  (ie lorsque  $y_0 < 0$ ). De plus,  $y$  est décroissante si  $y_0 > 0$  et croissante si  $y_0 < 0$ . On en déduit par le théorème des extrémités que la solution  $u$  admet un prolongement sur  $\mathbb{R}$  solution de l'EDO.

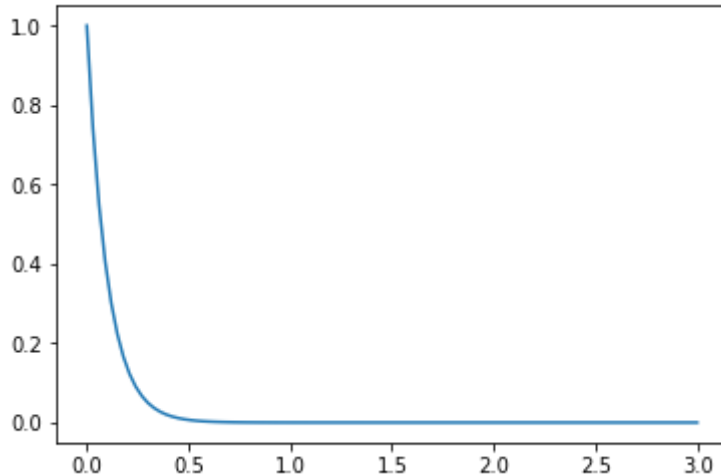
Pour en calculer la solution, on remarque qu'il s'agit d'une EDO à variables séparables. L'unique solution constante est  $y(t) \equiv 0$ , toutes les autres solutions sont du type  $y(t) = Ce^{-10t}$ . En prenant en compte la condition initiale on conclut que l'unique solution du problème de Cauchy est

$$y(t) = y_0 e^{-10t}$$

définie pour tout  $t \in \mathbb{R}$ .

In [17]:

```
y0=1
exacte = lambda t : y0*exp(-10*t)
tt=linspace(0,3,101)
yy=[exacte(t) for t in tt]
plot(tt,yy);
```



## 7.2 Correction

Soit le schéma numérique de Crank-Nicolson défini par la suite  $(u_n)_{n \in \mathbb{N}}$  vérifiant

$$\frac{u_{n+1} - u_n}{h} + 5(u_{n+1} + u_n) = 0, \quad \forall n \in \mathbb{N},$$

pour  $h > 0$  fixé. On obtient une formule de récurrence rendue explicite par un calcul élémentaire:

$$u_{n+1} = -5hu_{n+1} - 5hu_n + u_n$$

d'où

$$u_{n+1} = \frac{1 - 5h}{1 + 5h} u_n.$$

Par récurrence on obtient

$$u_n = \left( \frac{1 - 5h}{1 + 5h} \right)^n u_0.$$

Il s'agit d'une suite géométrique de raison

$$r = \frac{1 - 5h}{1 + 5h}.$$

## 7.3 Correction

Pour tout  $h > 0$  on a

$$r = \frac{1 - 5h}{1 + 5h} = 1 - \frac{10h}{1 + 5h}$$

et

$$-1 < 1 - \frac{10h}{1+5h} < 1.$$

Ce schéma est donc inconditionnellement A-stable car  $|u_{n+1}| = |r^{n+1} u_0| \leq |u_0|$

## 7.4 Correction

Le schéma génère une suite positive ssi

$$1 - \frac{10h}{1+5h} > 0$$

c'est-à-dire ssi

$$h < \frac{1}{5}.$$

In [18]:

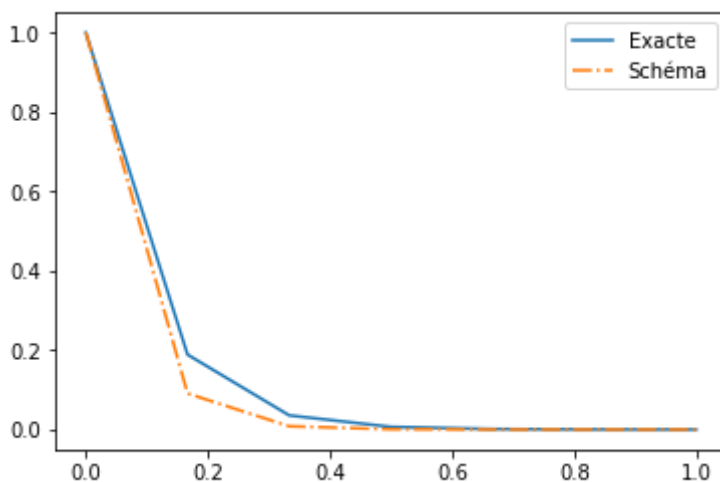
```
%matplotlib inline
from matplotlib.pyplot import *

N=6
tt=linspace(0,1,N+1)

y0=1
exacte = lambda t : y0*exp(-10*t)
yy=[exacte(t) for t in tt]

phi = lambda t,y : -y**5
def schema(phi,tt):
    h=tt[1]-tt[0]
    uu=[y0]
    for i in range(len(tt)-1):
        uu.append((1-5*h)/(1+5*h)*uu[i])
    return uu

plot(tt,yy,label=("Exacte"))
plot(tt,schema(phi,tt),"-.",label=("Schéma"))
legend();
```





## 8 Exercice

Soit le problème de Cauchy:

$$\begin{cases} y'(t) + \frac{\sqrt{y(t)}}{2} = 0, & \forall t \in \mathbb{R}^+, \\ y(0) = y_0 > 0. \end{cases}$$

1. Soit le schéma numérique défini par la suite  $(u_n)_{n \in \mathbb{N}}$  suivante

$$\frac{u_{n+1} - u_n}{h} + \frac{u_{n+1}}{2\sqrt{u_n}} = 0, \quad \forall n \in \mathbb{N},$$

pour  $h > 0$  fixé. Expliciter l'expression de  $u_{n+1}$  en fonction de  $u_n$ .

2. Montrer que la suite  $(u_n)_{n \in \mathbb{N}}$  est une suite positive, décroissante et convergente vers 0.

### 8.1 Correction

C'est un problème de Cauchy du type

$$\begin{cases} y'(t) = \varphi(t, y(t)), & \forall t \in \mathbb{R}, \\ y(0) = y_0 > 0, \end{cases}$$

avec  $\varphi(t, y) = g(y) = -\frac{\sqrt{y}}{2}$ .

Pour  $h > 0$  fixé on obtient une formule de récurrence rendue explicite par un calcul élémentaire:

$$u_{n+1} = \frac{u_n}{1 + \frac{h}{2\sqrt{u_n}}} = \frac{2(u_n)^{3/2}}{2\sqrt{u_n} + h}, \quad \forall n \in \mathbb{N}.$$

### 8.2 Correction

On étudie la suite

$$\begin{cases} u_0 > 0, \\ u_{n+1} = \frac{2(u_n)^{3/2}}{2\sqrt{u_n} + h}, \end{cases} \quad \forall n \in \mathbb{N},$$

pour  $h > 0$  fixé.

Par récurrence on montre que si  $u_0 > 0$  alors  $u_n > 0$  pour tout  $n \in \mathbb{N}$ .

De plus,  $\frac{u_{n+1}}{u_n} = \frac{1}{1 + \frac{h}{2\sqrt{u_n}}} < 1$  pour tout  $n \in \mathbb{N}$ : la suite est monotone

décroissante. On a alors une suite décroissante et bornée par zéro, donc

elle converge. Soit  $\ell$  la limite de cette suite, alors  $\ell \geq 0$  et  $\ell = \frac{2\ell^{3/2}}{2\sqrt{\ell} + h}$

donc  $\ell = 0$ .

In [19]:

```
%matplotlib inline
from matplotlib.pyplot import *

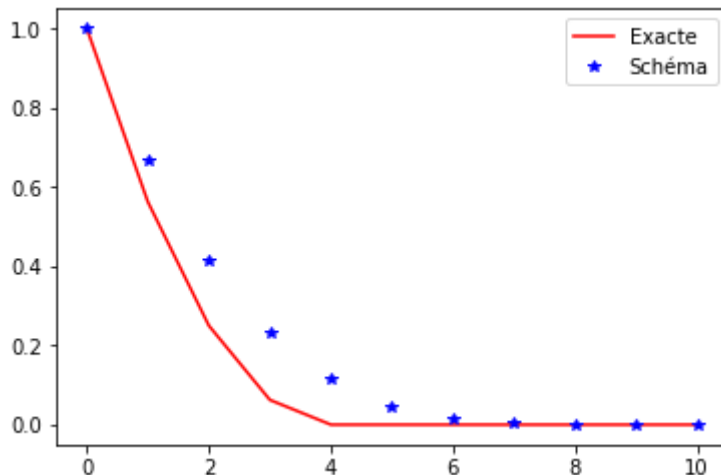
y0=1

phi = lambda t,y : -0.5*sqrt(y)
exacte = lambda t: 0 if t>=4*sqrt(y0) else (sqrt(y0)-t/4)**2

N=10
tt=linspace(0,10,N+1)

def schema(phi,tt):
    h=tt[1]-tt[0]
    uu=[y0]
    for i in range(len(tt)-1):
        uu.append(2*(uu[i])**3/2/(h+2*sqrt(uu[i])))
    return uu

yy=[exacte(t) for t in tt]
plot(tt,yy,'r-',label='Exacte')
plot(tt,schema(phi,tt),'b*',label="Schéma")
legend();
```



## 9 Exercice

Soit le problème de Cauchy:

$$\begin{cases} y'(t) + y^5(t) = 0, & \forall t \in \mathbb{R}^+, \\ y(0) = y_0 > 0. \end{cases}$$

1. Montrer qu'il existe une unique solution globale  $y \in \mathcal{C}^\infty(\mathbb{R}^+, \mathbb{R}^+)$ .
2. Soit le schéma numérique défini par la suite  $(u_n)_{n \in \mathbb{N}}$  suivante

$$\frac{u_{n+1} - u_n}{h} + u_{n+1}u_n^4 = 0, \quad \forall n \in \mathbb{N},$$

pour  $h > 0$  fixé. Expliciter l'expression de  $u_{n+1}$  en fonction de  $u_n$ .

3. Montrer que la suite  $(u_n)_{n \in \mathbb{N}}$  est une suite décroissante et convergente vers 0 pour tout  $h > 0$  fixé.

## 9.1 Correction

C'est un problème de Cauchy du type

$$\begin{cases} y'(t) = \varphi(t, y(t)), & \forall t \in \mathbb{R}, \\ y(0) = y_0 > 0, \end{cases}$$

avec  $\varphi(t, y) = g(y) = -y^5$ .

On montre d'abord qu'il existe une et une seule solution locale (ie sur  $[0; T]$ ) de classe  $\mathcal{C}^1([0, T], \mathbb{R})$ .

On montre ensuite que cette solution est de classe  $\mathcal{C}^\infty([0, T], \mathbb{R})$ .

On montre enfin que la solution admet un prolongement sur  $\mathbb{R}$ .

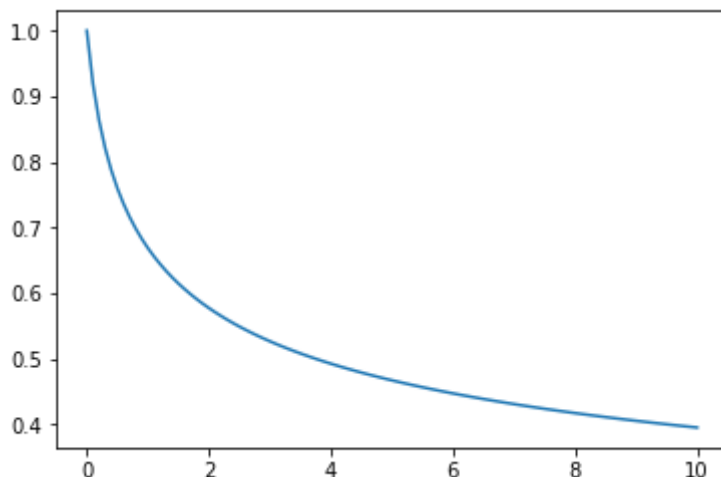
- Comme  $g \in \mathcal{C}^1(\mathbb{R}^+, \mathbb{R}^+)$ , d'après le théorème de Cauchy-Lipschitz il existe  $T > 0$  et une unique solution  $y \in \mathcal{C}^1([0, T], \mathbb{R})$ .
- Par récurrence, en exploitant l'EDO et la régularité de  $g$ , on grimpe en régularité sur  $y$  ainsi  $y \in \mathcal{C}^\infty([0, T], \mathbb{R})$ .
- La fonction nulle est solution de l'EDO (mais non du problème de Cauchy donné). Comme  $y_0 > 0$ , par l'unicité de la solution du problème de Cauchy on a  $y(t) > 0$  pour tout  $t \in [0, T]$  (car deux trajectoires ne peuvent pas se croiser). De plus,  $y$  est décroissante, ainsi la solution est bornée ( $y(t) \in ]0, y_0[$ ). On en déduit par le théorème des extrémités que la solution  $y$  admet un prolongement sur  $\mathbb{R}^+$  solution de l'EDO.

Pour en calculer la solution, on remarque qu'il s'agit d'une EDO à variables séparables. L'unique solution du problème de Cauchy est

$$y(t) = y_0(4ty_0^4 + 1)^{-1/4}$$

In [20]:

```
y0=1
exacte = lambda t : y0*(4*t*y0**4+1)**(-0.25)
tt=linspace(0,10,101)
yy=[exacte(t) for t in tt]
plot(tt,yy);
```



## 9.2 Correction

Pour  $h > 0$  fixé on obtient une formule de récurrence rendue explicite par un calcul élémentaire:

$$u_{n+1} = \frac{u_n}{1 + u_n^4 h}, \quad \forall n \in \mathbb{N}.$$

## 9.3 Correction

On étudie la suite

$$\begin{cases} u_0 = y_0 > 0, \\ u_{n+1} = \frac{u_n}{1 + u_n^4 h}, \end{cases} \quad \forall n \in \mathbb{N},$$

pour  $h > 0$  fixé.

Par récurrence on montre que si  $u_0 > 0$  alors  $u_n > 0$  pour tout  $n \in \mathbb{N}$ . De plus,  $\frac{u_{n+1}}{u_n} = \frac{1}{1 + u_n^4 h} < 1$  pour tout  $n \in \mathbb{N}$ : la suite est monotone décroissante. On a alors une suite décroissante et bornée par zéro, donc elle converge. Soit  $\ell$  la limite de cette suite, alors  $\ell \geq 0$  et  $\ell = \frac{\ell}{1 + \ell^4 h}$  donc  $\ell = 0$ . Ce schéma est donc inconditionnellement A-stable.

In [21]:

```

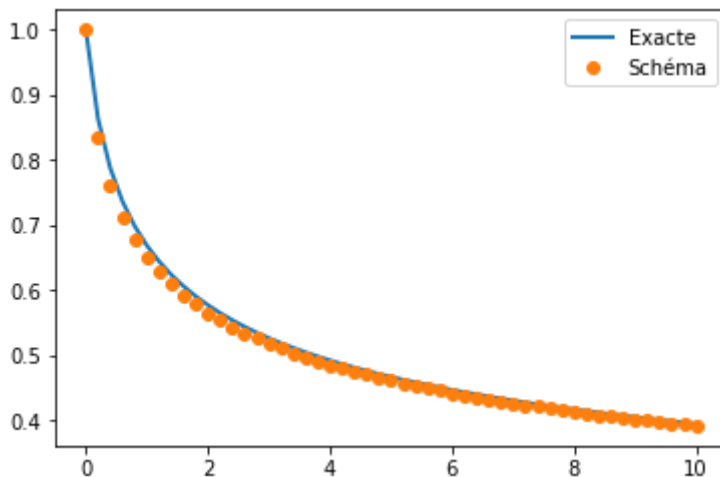
N=50
tt=linspace(0,10,N+1)

y0=1
exacte = lambda t : y0*(4*t*y0**4+1)**(-0.25)
yy=[exacte(t) for t in tt]

phi = lambda t,y : -y**5
def schema(phi,tt):
    h=tt[1]-tt[0]
    uu=[y0]
    for i in range(len(tt)-1):
        uu.append(uu[i]/(1+h*uu[i]**4))
    return uu

plot(tt,yy,label="Exacte",lw=2)
plot(tt,schema(phi,tt),'o',label="Schéma")
legend();

```



## 10 Exercice

Soit le problème de Cauchy:

$$\begin{cases} y'(t) + \sin(y(t)) = 0, & \forall t \in \mathbb{R}, \\ y(0) = y_0 > 0. \end{cases}$$

1. Montrer qu'il existe une unique solution globale  $y \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R})$ .
2. Écrire le schéma le schéma d'Euler explicite pour ce problème.
3. Montrer que la suite  $(u_n)_{n \in \mathbb{N}}$  construite par ce schéma vérifie

$$|u_{n+1}| \leq |u_n| + h, \quad \forall n \in \mathbb{N},$$

où  $h > 0$  est le pas de la suite.

4. En déduire que

$$|u_n| \leq |u_0| + nh, \quad \forall n \in \mathbb{N}.$$

### 10.1 Correction

[...]

C'est un problème de Cauchy du type

$$\begin{cases} y'(t) = \varphi(t, y(t)), & \forall t \in \mathbb{R}, \\ y(0) = y_0 > 0, \end{cases}$$

avec  $\varphi(t, y) = g(y) = -\sin(y)$ .

On montre d'abord qu'il existe une et une seule solution locale (ie sur  $[0; T]$ ) de classe  $\mathcal{C}^1([0, T], \mathbb{R})$ .

On montre ensuite que cette solution est de classe  $\mathcal{C}^\infty([0, T], \mathbb{R})$ .

On montre enfin que la solution admet un prolongement sur  $\mathbb{R}$ .

- Comme  $g \in \mathcal{C}^1(\mathbb{R}, \mathbb{R})$ , d'après le théorème de Cauchy-Lipschitz il existe  $T > 0$  et une unique solution  $y \in \mathcal{C}^1([-T, T], \mathbb{R})$ .
- Par récurrence, en exploitant l'EDO et la régularité de  $g$ , on grimpe en régularité sur  $y$  ainsi  $y \in \mathcal{C}^\infty([-T, T], \mathbb{R})$ .
- Toutes les fonctions constante  $y(t) = k\pi$  pour  $k \in \mathbb{Z}$  sont solutions de l'équation différentielle car  $g(k\pi) = 0$ . Pour  $y_0$  donné, soit  $\tilde{k} \in \mathbb{Z}$  tel que  $y_0 \in [\tilde{k}\pi; (\tilde{k} + 1)\pi]$ ; par l'unicité de la solution du problème de Cauchy on a  $y(t) \in [\tilde{k}\pi; (\tilde{k} + 1)\pi]$  pour tout  $t \in [-T, T]$  (car deux trajectoires ne peuvent pas se croiser), i.e. la solution est bornée. On en déduit par le théorème des extrémités que la solution  $y$  admet un prolongement sur  $\mathbb{R}$  solution de l'EDO.

## 10.2 Correction

Soit  $h > 0$  fixé et  $t_n = nh$  pour tout  $n \in \mathbb{Z}$ . Le schéma d'Euler explicite pour l'EDO donnée construit la suite

$$u_{n+1} = u_n - h \sin(u_n), \quad \forall n \in \mathbb{N}.$$

## 10.3 Correction

Comme  $|a + b| \leq |a| + |b|$  et comme  $|\sin(x)| \leq 1$  pour tout  $x \in \mathbb{R}$ , on conclut que

$$|u_{n+1}| = |u_n - h \sin(u_n)| \leq |u_n| + |h \sin(u_n)| \leq |u_n| + h$$

pour  $h > 0$  fixé.

## 10.4 Correction

Par récurrence:

$$|u_{n+1}| \leq |u_n| + h \leq |u_{n-1}| + 2h \leq \dots \leq |u_0| + (n + 1)h$$

In [22]:

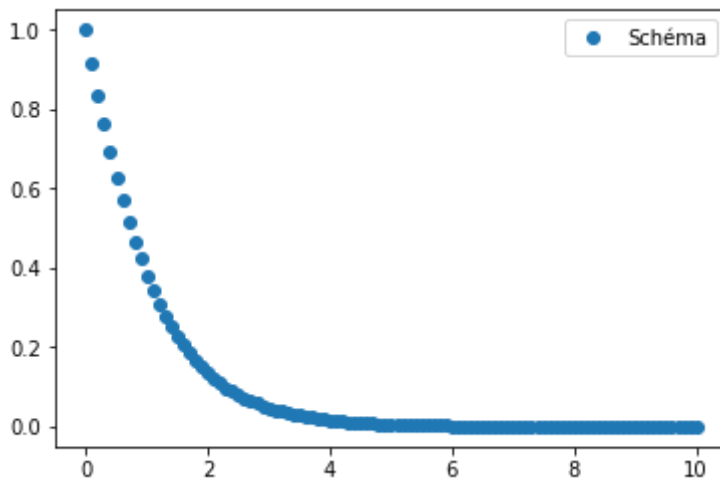
```

N=100
tt=linspace(0,10,N+1)

phi = lambda t,y : -sin(y)
def schema(phi,tt):
    h=tt[1]-tt[0]
    uu=[y0]
    for i in range(len(tt)-1):
        uu.append(uu[i]-h*sin(uu[i]))
    return uu

plot(tt,schema(phi,tt),'o',label=("Schéma"))
legend();

```



## 11 Exercice

Un modèle pour la diffusion d'une épidémie se base sur l'hypothèse que sa vitesse de propagation est proportionnelle au nombre d'individus infectés et au nombre d'individus sains.

Si on note  $I(t) \geq 0$  le nombre d'individus infectés à l'instant  $t \geq 0$  et  $A > 0$  le nombre d'individus total, il existe une constante  $k \in \mathbb{R}^+$  telle que  $I'(t) = kI(t)(A - I(t))$ .

1. Montrer qu'il existe  $T > 0$  et une unique solution  $I \in \mathcal{C}^\infty([0, T])$  au problème de Cauchy:

$$\begin{cases} I'(t) = kI(t)(A - I(t)), \\ I(0) = I_0 > 0. \end{cases}$$

Montrer que si  $0 < I_0 < A$  alors  $0 < I(t) < A$  pour tout  $t > 0$ .

Montrer que si  $0 < I_0 < A$  alors  $I(t)$  est croissante sur  $\mathbb{R}^+$ .

2. Soit  $0 < I_0 < A$ . On considère le schéma semi-implicite

$$\frac{I_{n+1} - I_n}{h} = kI_n(A - I_{n+1}).$$

Montrer que  $I_n \rightarrow A$  lorsque  $n \rightarrow +\infty$  indépendamment du pas  $h > 0$  fixé.

## 3. Calculer la solution exacte

## 11.1 Correction

C'est un problème de Cauchy du type

$$\begin{cases} I'(t) = \varphi(t, I(t)), & \forall t \in \mathbb{R}^+, \\ I(0) = I_0 > 0, \end{cases}$$

avec  $\varphi(t, I(t)) = g(I(t)) = kI(t)(A - I(t))$ .

- Comme  $g \in \mathcal{C}^1(\mathbb{R}, \mathbb{R})$ , d'après le théorème de Cauchy-Lipschitz il existe  $T > 0$  et une unique solution  $I \in \mathcal{C}^1([0, T], \mathbb{R})$ .
- Par récurrence, en exploitant l'EDO et la régularité de  $g$ , on grimpe en régularité sur  $I$  ainsi  $I \in \mathcal{C}^\infty([0, T], \mathbb{R})$ .
- Puisque la fonction nulle et la fonction constante  $I(t) = A$  sont solutions de l'équation différentielle, si  $0 < I_0 < A$  alors  $0 < I(t) < A$  pour tout  $t \in [0, T]$  (car, par l'unicité de la solution du problème de Cauchy, deux trajectoires ne peuvent pas se croiser).
- Puisque  $I'(t) = kI(t)(A - I(t))$ , si  $0 < I_0 < A$  alors  $I$  est croissante pour tout  $t \in [0, T]$ . On en déduit par le théorème des extrémités que la solution  $I$  admet un prolongement sur  $\mathbb{R}^+$  solution de l'EDO et que  $I$  est croissante pour tout  $t \in \mathbb{R}^+$ .

## 11.2 Correction

Soit  $0 < I_0 < A$ . On considère le schéma semi-implicite

$$\frac{I_{n+1} - I_n}{h} = kI_n(A - I_{n+1})$$

pour  $h > 0$  fixé. On obtient une formule de récurrence rendue explicite par un calcul élémentaire:

$$I_{n+1} = \frac{1 + kAh}{1 + kI_n h} I_n.$$

Si  $0 < I_0 < A$  alors

- $I_n > 0$  quelque soit  $n$ ;
- $I_n$  est majorée par  $A$  car  $I_{n+1} \leq A \iff (1 + kAh)I_n \leq (1 + kI_n h)A \iff I_n \leq A$  donc par récurrence  $I_{n+1} \leq A$  quelque soit  $n$ ;
- si  $I_n \rightarrow \ell$  alors  $\ell = \frac{1+kAh}{1+k\ell h} \ell$  donc  $\ell = 0$  ou  $\ell = A$ ;
- $I_n$  est une suite monotone croissante (encore par récurrence on montre que  $|I_{n+1}| \geq |I_n| \geq \dots \geq |I_0|$ );

donc  $I_n \rightarrow A$  lorsque  $n \rightarrow +\infty$  indépendamment du pas  $h > 0$  choisi.

## 11.3 Correction



On a déjà observé qu'il y a deux solutions constantes de l'EDO: la fonction  $I(t) \equiv 0$  et la fonction  $I(t) \equiv A$ .

Pour chercher toutes les solutions non constantes on remarque qu'il s'agit d'une EDO à variables séparables donc on a

$$I(t) = \frac{A}{De^{-Akt} + 1}$$

La valeur numérique de la constante d'intégration  $D$  est obtenue grâce à la CI:

$$D = \frac{A - I_0}{I_0}$$

Exemple avec  $A = 5000$ ,  $I_0 = 160$ ,  $k = \frac{\ln(363/38)}{35000}$  et  $h = 1$ :

In [23]:

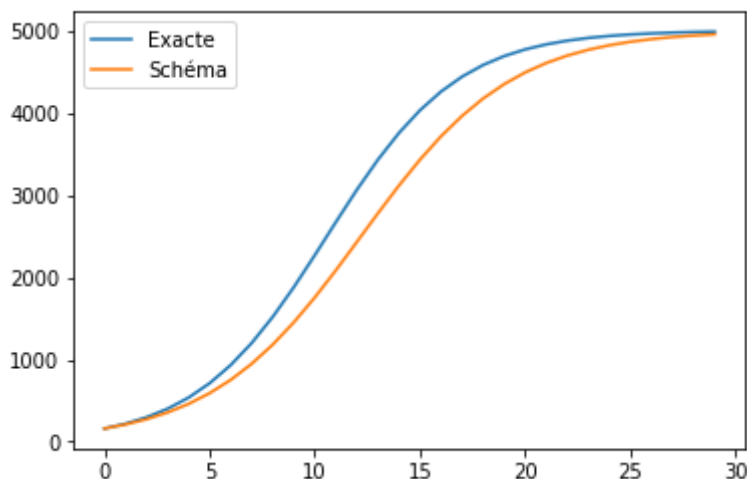
```
A=5000
I0=160
k=log(363/38)/35000

D=(A-I0)/I0

exacte = lambda t : A/(D*exp(-A*k*t)+1)
h=1
tt=arange(0,30,h)
yy=[exacte(t) for t in tt]

phi = lambda t,I : k*I*(A-I)
def schema(phi,tt):
    uu=[I0]
    for i in range(len(tt)-1):
        uu.append((1+k*A*h)/(1+k*uu[-1]*h)*uu[-1])
    return uu

plot(tt,yy,label=("Exacte"))
plot(tt,schema(phi,tt),label=("Schéma"))
legend();
```



## 12 Exercice : A-stabilité du schéma d'Euler explicite dans le cas d'un système

Soit le problème de Cauchy:

$$\begin{cases} y''(t) + 1001y'(t) + 1000y(t) = 0, & \forall t \in \mathbb{R}, \\ y(0) = 1, \\ y'(0) = 0. \end{cases}$$

1. Écrire le problème comme un système de 2 EDO d'ordre 1 et en calculer la solution.
2. Écrire le schéma d'Euler explicite pour ce problème.
3. Pour quelles valeurs de  $h$  le schéma est A-stable?

### 12.1 Correction

Soit  $\mathbf{z}(t) = (y(t), y'(t))^T$  alors  $y''(t) + 1001y'(t) + 1000y(t) = 0$  se réécrit

$$\mathbf{z}'(t) = - \begin{pmatrix} 0 & -1 \\ 1000 & 1001 \end{pmatrix} \mathbf{z}(t)$$

Cette matrice a pour valeurs propres  $\lambda_1 = 1$  et  $\lambda_2 = 1000$  et pour vecteurs propres associés  $\mathbf{v}_1 = (-1, 1)^T$  et  $\mathbf{v}_2 = (0, 1000)^T$  (ci dessous le calcul formel des valeurs et vecteurs propres). Ainsi

$$\mathbf{z}(t) = c_1 \mathbf{v}_1 e^{-\lambda_1 t} + c_2 \mathbf{v}_2 e^{-\lambda_2 t}$$

et

$$y(t) = -c_1 e^{-t} - c_2 e^{-1001t}.$$

Il s'agit d'un problème stiff!

In [24]:

```
%reset -f
import sympy as symb
symb.init_printing()

symb.var('t')
M = symb.Matrix(((0, -1), (1000, 1001)))
M
```

Out[24]:

$$\begin{bmatrix} 0 & -1 \\ 1000 & 1001 \end{bmatrix}$$

In [25]:

```
p=M.charpoly(t)
symb.factor(p)
```

Out[25]:

$$(t - 1000)(t - 1)$$

In [26]:

```
P, D = M.diagonalize()
P,D
```

Out[26]:

$$\left( \begin{bmatrix} -1 & -1 \\ 1 & 1000 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1000 \end{bmatrix} \right)$$

In [27]:

```
symb.simplify(P*D*P**-1)
```

Out[27]:

$$\begin{bmatrix} 0 & -1 \\ 1000 & 1001 \end{bmatrix}$$

## 12.2 Correction

Soit  $h > 0$  fixé et  $t_n = nh$  pour tout  $n \in \mathbb{Z}$ . Le schéma d'Euler explicite pour le système donné construit la suite

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n - h \begin{pmatrix} 0 & -1 \\ 1000 & 1001 \end{pmatrix} \mathbf{u}_n \\ &= \begin{pmatrix} 1 & h \\ -1000h & 1 - 1001h \end{pmatrix} \mathbf{u}_n \\ &= \begin{pmatrix} 1 & h \\ -1000h & 1 - 1001h \end{pmatrix}^{n+1} \mathbf{u}_0, \\ &= \begin{pmatrix} -1 & -1 \\ 1000 & 1 \end{pmatrix} \begin{pmatrix} 1 - 1000h & 0 \\ 0 & 1 - h \end{pmatrix}^{n+1} \begin{pmatrix} -1 & -1 \\ 1000 & 1 \end{pmatrix}^{-1} \mathbf{u}_0 \\ &= \begin{pmatrix} -1 & -1 \\ 1000 & 1 \end{pmatrix} \begin{pmatrix} (1 - 1000h)^{n+1} & 0 \\ 0 & (1 - h)^{n+1} \end{pmatrix} \begin{pmatrix} -1 & -1 \\ 1000 & 1 \end{pmatrix}^{-1} \mathbf{u}_0 \end{aligned}$$

In [28]:

```
symb.var('h')
M = symb.Matrix(((1,h), (-1000*h,1-1001*h)))
M
```

Out[28]:

$$\begin{bmatrix} 1 & h \\ -1000h & -1001h + 1 \end{bmatrix}$$

In [29]:

```
P, D = M.diagonalize()
P, D
```

Out[29]:

$$\left( \begin{bmatrix} -1 & -1 \\ 1000 & 1 \end{bmatrix}, \begin{bmatrix} -1000h + 1 & 0 \\ 0 & -h + 1 \end{bmatrix} \right)$$

### 12.3 Correction

Le schéma est A-stable si  $\lim_{n \rightarrow +\infty} \mathbf{u}_{n+1} = \mathbf{0}$ . Il faut donc que

$$\begin{cases} |1 - 1000h| < 1 \\ |1 - h| < 1 \end{cases}$$

d'où la condition  $h < \frac{2}{1000}$ .

## 13 Exercice : A-stabilité d'un schéma

Pour calculer la solution approchée considérons la  $\vartheta$ -méthode suivante:

$$\begin{cases} u_0 = y_0, \\ u_{n+1} = u_n + h(\vartheta \varphi(t_{n+1}, u_{n+1}) + (1 - \vartheta)\varphi(t_n, u_n)). \end{cases}$$

avec  $\vartheta \in [0; 1]$ .

Soit  $\varphi(t, y(t)) = -\beta y(t)$  avec  $\beta > 0$ . Donner une condition sur  $h$  pour que la  $\vartheta$ -méthode soit A-stable. Pour quelles valeurs de  $\vartheta$  la méthode est inconditionnellement A-stable?

*Remarque:* pour  $\vartheta = 0$  on retrouve la méthode d'Euler explicite, pour  $\vartheta = 1$  la méthode d'Euler implicite, pour  $\vartheta = \frac{1}{2}$  la méthode de Crank-Nicholson.

### 13.1 Correction

$$u_{n+1} = u_n - \beta h(\vartheta u_{n+1} + (1 - \vartheta)u_n) \iff u_{n+1} = \frac{1 - \beta h(1 - \vartheta)}{1 + \beta h\vartheta} u_n =$$

Par induction

$$u_n = \left(1 - \frac{\beta h}{1 + \beta h\vartheta}\right)^n y_0$$

Il s'agit d'une suite géométrique de raison  $q = 1 - \frac{\beta h}{1 + \beta h\vartheta}$ .

La suite tend à zéro ssi  $|q| < 1$  (la convergence est monotone ssi  $0 \leq q < 1$ ):

$$-1 < 1 - \frac{\beta h}{1 + \beta h\vartheta} < 1 \iff 0 < \frac{\beta h}{1 + \beta h\vartheta} < 2 \iff (1 - 2\vartheta)\beta h <$$

Pour  $\vartheta \geq \frac{1}{2}$  la méthode est inconditionnellement A-stable,

pour  $\vartheta < \frac{1}{2}$  la méthode est A-stable ssi  $h < \frac{2}{(1-2\vartheta)\beta}$ .