In [1]:

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

Out[1]:

M62 TP 2 - calcul numérique VS formel

Vous pouvez commencer à exécuter un serveur de notebook en écrivant dans un terminal :

`jupyter notebook &`

Cela imprimera des informations dans votre console et ouvrira un navigateur Web.
La page d'arrivée est le tableau de bord qui affiche les notebook actuellement disponibles (par défaut, le répertoire à partir duquel le serveur du bloc-notes a été démarré).

Vous pouvez

- soit créer un nouveau notebook à partir du tableau de bord avec le bouton (en haut à droite) `Nouveau`
- soit ouvrir un notebook existant en cliquant sur leur nom (par exemple, vous pouvez télécharger le notebook de présentation ici (http://nbviewer.jupyter.org/url/faccanoni.univ-tln.fr/user/enseignements/20182019/M62-TP1.ipynb).)

Table of Contents

1 Calcul numérique

(approché)

Pour cette première partie, on importera tous les modules necessaires comme suit:

In [4]:

```
%reset -f
%matplotlib inline
%autosave 300
from math import *
from matplotlib.pylab import *
```

```
Autosaving every 300 seconds
```

1.1 Exercice : utiliser `scipy`

En s'appuyant sur le module `scipy.integrate` afficher le graphe de la fonction E entre -2 et 2:

$$E: \mathbb{R} \to \mathbb{R}$$
$$x \mapsto E(x) = \int_0^x e^{-t^2} \mathrm{d}t$$

Correction

In [5]:

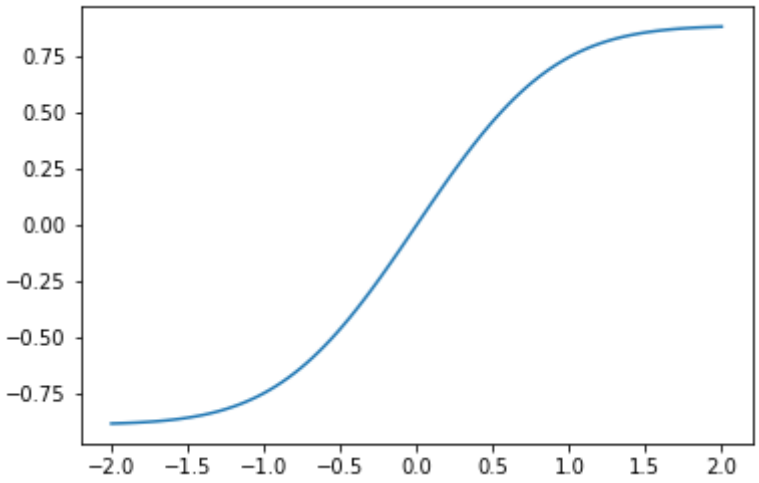
```
from scipy import integrate

# def E(x):
#     return integrate.quad(lambda t: math.exp(-t**2), 0, x)[0
E = lambda x : integrate.quad(lambda t: math.exp(-t**2), 0, x)

x = linspace(-2,2,101)
# y=[E(xi) for xi in x]
y=list(map(E,x))
plot(x,y)

print("Valeur approchee pour x->oo \t= ",E(math.inf))
print("Valeur exacte pour x->oo \t= ",math.sqrt(math.pi)/2)
```

```
Valeur approchee pour x->oo     =  0.886226925452
7579
Valeur exacte pour x->oo        =  0.886226925452
7579
```



1.2 Exercice : méthode de Monte-Carlo

La méthode de Monte-Carlo (du nom des casinos, pas d'une personne) est une approche probabiliste permettant d'approcher la valeur d'une intégrale. L'idée de base est que l'intégrale J peut être vue comme l'espérance d'une variable aléatoire uniforme X sur l'intervalle $[a, b]$ Par la loi des grands nombres cette espérance peut être approchée par la moyenne empirique

$$J \approx J_N = \frac{b-a}{N} \sum_{i=0}^{N-1} f(x_i),$$

où les x_i sont tirés aléatoirement dans l'intervalle $[a, b]$ avec une loi de probabilité uniforme.

- Écrire une fonction `montecarlo(f,a,b,N)` qui détermine une approximation de J par la méthode de Monte-Carlo
- Valider la fonction (i.e. on considère des cas dont on connait la solution exacte et on écrit un test unitaire). Quelle valeur obtient-on

avec le module `scipy.integrate`

Correction

In [6]:

```
import random

# Fonction Monte-Carlo
def montecarlo(f,a,b,N):
    x=[random.uniform(a,b) for i in range(N)]
    # y=[f(x[i])  for i in range(N)]
    y=list(map(f,x))
    return (b-a)*sum(y)/N
```

In [7]:

```
# TESTS
from scipy import integrate

g = lambda x: 1
print("Calcul approché par Montecarlo =",montecarlo(g,0,1,100))
print("Calcul approché par ScyPy =", integrate.quad(g,0,1)[0])

g = lambda x: x**3
print("Calcul approché par Montecarlo =",montecarlo(g,0,1,100))
print("Calcul approché par ScyPy =", integrate.quad(g,0,1)[0])
```

```
Calcul approché par Montecarlo = 1.0
Calcul approché par ScyPy = 1.0
Calcul approché par Montecarlo = 0.24909970447153
65
Calcul approché par ScyPy = 0.25
```

1.3 Exercice: Fond d'investissement

Un compte d'épargne donne un taux $T \in [0; 1]$ d'intérêt par an avec un virement annuel des intérêts sur le compte. Cela signifie que si le premier janvier 2014 on met v euros sur ce compte, à la fin de la n-ème année (i.e. au 31 décembre $2014 + n$) on en retire $v(1 + T)^n$ euros. On décide alors d'ajouter au début de chaque année encore v euros. Cela signifie que si on verse ces v euros le premier janvier $2014 + m$ avec $0 < m < n$, au 31 décembre $2014 + n$ ajoutent $v(1 + T)^{n-m}$ euros. Si à la fin de la n-ème année on en retire un capital de $M > v$ euros, quel est le taux d'intérêt annuel de cet investissement?

Correction

À la fin de la n-ème année, le capital versé au premier janvier 2014 est devenu $v(1+T\%)^n$, celui versé au premier janvier 2015 est devenu $v(1+T\%)^{n-1}$... par conséquente, le capital final M est relié au taux d'intérêt annuel T par la relation

$$M = v\sum_{k=1}^{n}(1+T)^k = v\frac{(1+T)^n-1}{(1+T)-1} = v\frac{1+T}{T}((1+T)^n-1)\,.$$

On en déduit que T est racine de l'équation algébrique non linéaire $f(T)=0$ où

$$f(T) = v\frac{1+T}{T}((1+T)^n-1) - M.$$

Étudions la fonction f:

- $f(T) > 0$ pour tout $T > 0$,
- $\lim_{T\to 0^+} f(T) = nv - M < (n-1)v$, $\lim_{T\to+\infty} f(T) = +\infty$,
- $f'(T) = \frac{v}{T^2}(1+(1+T)^n(Tn-1)) > 0$pour tout $T > 0$
  (comparer le graphe de $-1/(1+T)^n$ et de $nT-1$)

En étudiant la fonction f on voit que, comme $nv < M$ dès que $n > 1$, elle admet un unique zéro dans l'intervalle $]0,+\infty[$ (on peut même prouver qu'elle admet un unique zéro dans l'intervalle $]0, M[$).

Supposons que $v = 1000$€ et qu'après 5 ans M est égal à 6000€. En étudiant la fonction f on voit qu'elle admet un unique zéro dans l'intervalle $]0.01, 0.1[$.

In [5]:

```
from scipy.optimize import fsolve

def f(x):
    return 1000.*(1+x)/x*((1+x)**5-1.)-6000

a = 0.01 # T=1%
b = 0.1  # T=10%

sol = fsolve(f, (a+b)/2)
print('taux=',sol[0])
```

```
taux= 0.061402411536525266
```

On conclut ainsi que le taux d'intérêt T est approximativement égal à {{sol[0]*100}}%.

1.4 Exercice (difficile) : chèvre

On dispose d'un champ circulaire de rayon R. Une chèvre est attaché par une corde de longueur x à un piquet planté sur la circonférence du champs. Calculer x afin que la chèvre broute au maximum la moitié de la surface du champ.

Correction

On note x la longueur de la corde, R le rayon du champs circulaire et α et β les angles en figure. Il s'agit d'un triangle isocèle (cf. Fig. 1) donc

- $\alpha + 2\beta = \pi$,
- $x = 2R\cos(\beta) = 2R\sin\left(\frac{\alpha}{2}\right)$.

L'aire du secteur circulaire orange (cf. Fig. 2) vaut $\frac{1}{2}\ell x$ avec $\ell = x\beta$ si β est mesuré en radiant, soit $\frac{1}{2}x^2\beta$, donc

$$2R^2\beta\cos^2(\beta).$$

L'aire du segment circulaire bleu (cf. Fig. 3) vaut $\frac{1}{2}\ell R - \frac{1}{2}R^2\sin(\alpha)$ avec $\ell = R\alpha$ si α est mesuré en radiant, soit $\frac{1}{2}R^2\alpha - \frac{1}{2}R^2\sin(\alpha)$, donc

$$\frac{1}{2}R^2(\pi - 2\beta) - R^2\cos(\beta)\sin(\beta).$$

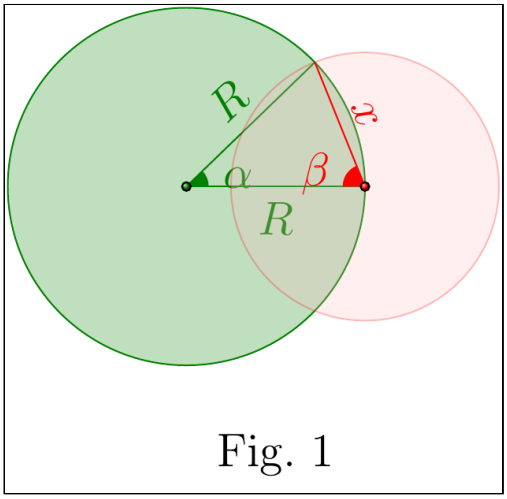


Fig. 1

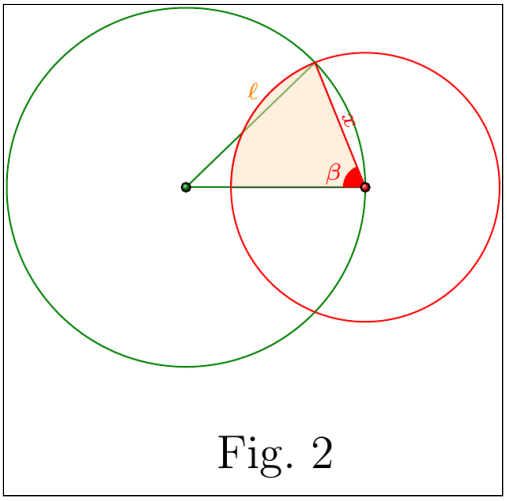


Fig. 2

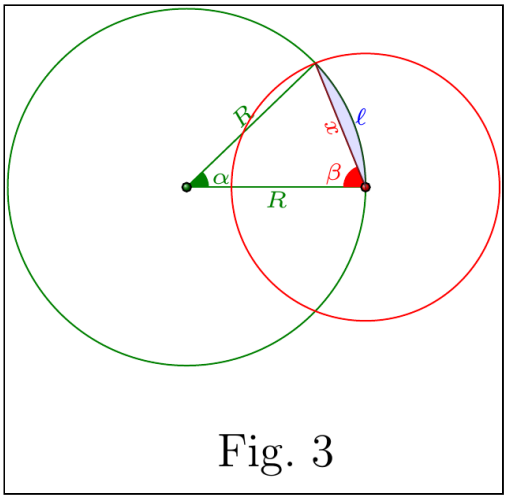


Fig. 3

On cherche β tel que

$$2\left(2R^2\beta\cos^2(\beta) + \frac{1}{2}R^2(\pi - 2\beta) - R^2\cos(\beta)\sin(\beta)\right) = \frac{1}{2}\pi R^2.$$

Comme $\cos(2\beta) = 2\cos^2(\beta) - 1$ et $2\cos(\beta)\sin(\beta) = \sin(2\beta)$, on cherche β tel que

$$2\beta\cos(2\beta) - \sin(2\beta) + \frac{1}{2}\pi = 0.$$

Posons $f(\beta) = 2\beta\cos(2\beta) - \sin(2\beta) + \frac{1}{2}\pi$. On cherche $\beta \in]0;\pi/2[$ tel que $f(\beta) = 0$. On a $f(0) > 0, f(\pi/2) < 0$ et $f'(\beta) = -4\beta\sin(2\beta) < 0$ pour tout $\beta \in]0;\pi/2[$: il existe un et un seul $\beta \in]0;\pi/2[$ tel que $f(\beta) = 0$. On n'est pas capable de calculer analytiquement la solution de cette équation, on va alors l'approcher numériquement avec une précision de 10^{-4}.

In [8]:

```
from scipy.optimize import fsolve

def f(x):
    return 2.*x*cos(2.*x)-sin(2.*x)+pi/2

a = 0.0
b = 0.5*pi

sol = fsolve(f, (a+b)/2)
print('beta=',sol[0],', f(beta)=',f(sol[0]))
temp= 2.*cos(sol[0])
print('x/R =',temp)
```

```
beta= 0.9528478646549418 , f(beta)= 4.44089209850
0626e-16
x/R = 1.1587284730181218
```

La corde doit mesurer environ {{temp}} fois le rayon.

2 Calcul formel (exact)

Pour cette deuxième partie, de preference dans un nouvau document (pour éviter les conflits avec matplotlib & co.) on importe le module de calcul symbolique:

In [9]:

```
%reset -f
%matplotlib inline
%autosave 300

from sympy import *
init_printing(use_unicode=False, wrap_line=False, no_global=Tr

x,f,d1f,d2f = symbols('x,f,d1f,d2f')
```

```
Autosaving every 300 seconds
```

2.1 Exercice : étude d'extrema

Soit $f(x) = x^3 - 3x + 1$. Calculer les points stationnaires et y évaluer la dérivée seconde.

Correction

In [10]:

```
f=x**3-3*x+1
f
```

Out[10]:

$x^3 - 3x + 1$

In [11]:

```
d1f=diff(f,x)
d1f
```

Out[11]:

$3x^2 - 3$

In [12]:

```
d2f=diff(f,x,2)
d2f
```

Out[12]:

$6x$

In [13]:

```
extr=solve(d1f)
extr
```

Out[13]:

$[-1, \quad 1]$

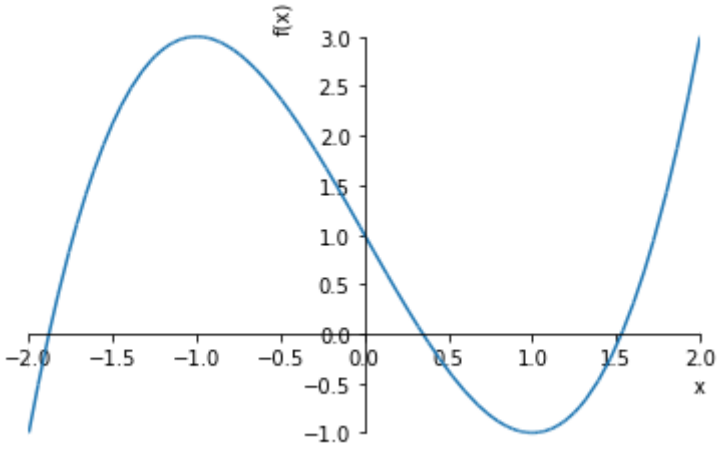
In [14]:

```
for pt in extr:
    print("f''(",pt,")=",d2f.subs(x,pt))
```

```
f''( -1 )= -6
f''( 1 )= 6
```

In [15]:

```
plot(f,(x,-2,2))
```



Out[15]:

```
<sympy.plotting.plot.Plot at 0x7fc906d1ae48>
```

2.2 Exercice : le jeu d'échecs

Selon la légende, le jeu d'échecs fut inventé en Inde par un savant. Le roi, séduit par ce nouveau loisir, le convoqua au palais: "Ton jeu m'a redonné la joie de vivre! Je t'offre ce que tu désires!" lui dit-il. Le sage ne voulait rien et ne dit mot. Le roi offensé s'énerva: "Parle donc, insolent! Tu as peur que je ne puisse exaucer tes souhaits?" Le sage fut blessé par ce ton et décida de se venger: "J'accepte votre présent. Vous ferez déposer un grain de blé sur la première case de l'échiquier. Vous ferez mettre ensuite 2 grains sur la deuxième case, 4 sur la troisième et ainsi de suite..." Le roi s'énerva pour de bon: "Puisque tu honores si mal ma générosité, vas-t-en! Ton sac de blé te sera porté demain et ne me dérange plus!" Le lendemain matin, le roi fut réveillé par son intendant affolé: "Sire, c'est une catastrophe! Nous ne pouvons pas livrer le blé! Nos mathématiciens ont travaillé toute la nuit: il n'y a pas assez de blé dans tout le royaume pour exaucer le souhait du savant!" Pourquoi une telle affirmation?

Correction

Notons b_n le nombre de grains de blé sur la case n, n allant de 0 à 63. La suite (g_n) est géométrique de raison 2 car $g_{n+1} = 2g_n$ donc $g_n = 2^n g_0 = 2^n$. Ainsi la somme totale des grains de blé sera

$$\sum_{n=0}^{63} g_n = \sum_{n=0}^{63} 2^n = \frac{1-2^{64}}{1-2} = 18446744073709551615.$$

In [16]:

```
%reset -f

from sympy import *
init_printing(use_unicode=False, wrap_line=False, no_global=Tr

a,n,N= symbols('a n N')
```

In [17]:

```
somme=Sum(a**n,(n,0,N))
somme
```

Out[17]:

$$\sum_{n=0}^{N} a^n$$

In [18]:

```
somme=somme.subs({a:2,N:63})
somme
```

Out[18]:

$$\sum_{n=0}^{63} 2^n$$

In [19]:

```
somme.evalf()
```

Out[19]:

$$1.84467440737096 \cdot 10^{19}$$

2.3 Exercice : étude d'une suite

Étant donnés trois nombres réels a, b et μ, on considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par

$$\begin{cases} u_0 = \mu, \\ 4u_{n+1} = ab + 2(a+b) - 2(a+b)u_n + 3u_n^2, \quad \text{pour tout } n \in \mathbb{N}. \end{cases}$$

> Dans cette partie on suppose $a < b < 2$.

1. Montrer que, lorsqu'elle existe, la limite ℓ de la suite est une solution de l'équation
$$ab + 2(a+b) - 2(2+a+b)x + 3x^2 = 0.$$
2. On considère la fonction polynomiale définie par
$$f: \mathbb{R} \to \mathbb{R}$$
$$x \mapsto ab + 2(a+b) - 2(2+a+b)x + 3x^2.$$
Déterminer la primitive de f qui s'annule pour $x = 2$ et montrer qu'elle a exactement trois zéros que l'on précisera.
3. En déduire que, lorsqu'elle existe, la limite de la suite vérifie l'une des deux inégalités suivants:
$$a < \ell < b \quad \text{ou} \quad b < \ell < 2.$$

> Dans cette partie on suppose $a = b = 2$.

1. Montrer que si la suite converge, sa limite ℓ est égale à 2.
2. Montrer que pour tout $\mu \in \mathbb{R}$, la suite est croissante.
3. Montrer que la suite est convergente lorsque $\mu \in \left]\frac{2}{3}; 2\right[$.
4. Montrer que la suite est divergente lorsque $\mu \notin \left[\frac{2}{3}; 2\right]$.
5. Préciser les cas pour $\mu = \frac{2}{3}$ et pour $\mu = 2$.

Correction

In [20]:

```
%reset -f

from sympy import *
init_printing(use_unicode=False, wrap_line=False, no_global=Tr

x,t, h,a,b,f,g,fp,hp= symbols('x t h a b f g fp hp')
```

On commence par introduire la fonction $h: \mathbb{R} \to \mathbb{R}$ telle que $u_{n+1} = h(u_n)$:
$$h(x) = \frac{ab + 2(a+b) - 2(a+b)x + 3x^2}{4}$$

In [21]:

```
h=(3*x**2-2*(a+b)*x+a*b+2*(a+b))/4
```

> Dans cette partie on suppose $a < b < 2$.

Si la suite converge vers $\ell \in \mathbb{R}$ alors, sachant que l'on a $u_{n+1} = h(u_n)$ et que h est une fonction continue, nécessairement $\ell = h\ell$.

In [22]:

```
f=4*(h-x)
f
```

Out[22]:

$$ab + 2a + 2b + 3x^2 - x(2a + 2b) - 4x$$

On détermine g la primitive de f qui s'annule pour $x = 2$:

In [23]:

```
g=integrate(f,(x,2,t))
g
```

Out[23]:

$$-2ab + t^3 + t^2(-a - b - 2) + t(ab + 2a + 2b)$$

On factorise l'expression obtenue:

In [24]:

```
g.factor()
```

Out[24]:

$$(-a + t)(-b + t)(t - 2)$$

donc g s'annule en $x = 2$, $x = a$ et $x = b$.

Le théorème de Rolle s'applique à g sur les deux segments $[a; b]$ et $[b; 2]$, sa dérivée f s'annule donc au moins une fois à l'intérieur de ces deux segments et comme elle ne s'annule au plus que deux fois sur $\mathbb{R}$ (polynôme de degré 2), on a là ses deux zéros distincts et ℓ est l'un d'eux.

> Dans cette partie on suppose $a = b = 2$.

On particularise les fonctions f et h dans ce cas:

In [25]:

```
fp=f.subs({a:2, b:2})
fp.factor()
```

Out[25]:

$$3(x-2)^2$$

In [26]:

```
hp=h.subs({a:2, b:2})
hp.factor()
```

Out[26]:

$$\frac{1}{4}\left(3x^2-8x+12\right)$$

Comme on le voit dans la factorisation précédente, l'équation $f_p(x)=0$ ou encore $h_p(x)=x$ n'admet qu'une seule solution: $x=2$, ce qui permet de justifier que lorsque la suite converge, elle converge nécessairement vers $\ell=2$.

Le signe de $h_p(x)-x$ étant toujours positif, on peut en déduire que la suite est croissante quelle que soit la valeur de μ.

Une étude des variations de h_p sur $\mathbb{R}$ montre que l'intervalle $\left]\frac{2}{3};2\right[$ est stable. Si μ est dans cet intervalle alors on montre par récurrence que tous les termes de la suite y seront, la suite est donc bornée. Comme elle est croissante et bornée, elle converge et sa limite est $\ell=2$.

D'après l'étude des variations de h_p sur $\mathbb{R}$, on peut établir que si $\mu<\frac{2}{3}$ ou $\mu>2$ alors tous les termes de la suite à partir du rang 1 sont supérieurs à 2. Comme la suite est croissante et tous les termes de la suite à partir du rang 1 sont supérieurs à 2, la suite ne peut alors pas converger vers 2, seule limite possible. Dans ce cas, elle est divergente.

Si $\mu=\frac{2}{3}$ ou $\mu=2$, alors la suite est stationnaire à partir du rang 1 et vaut 2.