

TP 3 : Gradient conjugué pour un problème d'inpainting.

```
In [1]: %pylab inline
```

Populating the interactive namespace from numpy and matplotlib

1. Mise en place de la méthode du gradient conjugué

```
In [2]: def prodscal(x,y):  
        return sum(x*y)
```

(a) Voici un exemple en utilisant une boucle for plutôt qu'une boucle while, avec une instruction break en fin de boucle.

```
In [3]: def GradConj(fA,b,X0,eps=1e-10,Nmax=200):  
        x=X0  
        r=fA(x)+b  
        p=-r  
        psrr=prodscal(r,r)  
        tol=psrr*eps**2  
        lnr=[sqrt(psrr)]  
        # Pour une boucle while :  
        # initialiser d'abord i=0, puis  
        # while psrr>tol and i<Nmax:  
        #     i+=1  
        #     ...  
        for i in range(Nmax):  
            Ap=fA(p)  
            alpha=psrr/prodscal(p,Ap)  
            x=x+alpha*p  
            r=r+alpha*Ap  
            psrrancien=psrr  
            psrr=prodscal(r,r)  
            lnr.append(sqrt(psrr))  
            if psrr<tol:  
                # C'est cette instruction break  
                # qui fait qu'on quitte la boucle for  
                # quand le critère d'arrêt est satisfait  
                break  
            p=-r+psrr/psrrancien*p  
        return x,lnr
```

(b) On peut utiliser rand (uniforme entre 0 et 1), ou randn (gaussienne standard). Attention, le produit matriciel est effectué grâce à la fonction dot

```
In [4]: N=randn(12,112)  
        A=dot(N,transpose(N))  
        print("Conditionnement de A :",cond(A))  
  
        compteurA=0  
  
        def fA(x):  
            global compteurA
```

```
compteurA+=1
return dot(A,x)
```

```
b=rand(12)
fA(b)
```

Conditionnement de A : 3.07003179823

```
Out[4]: array([ 136.69189691,  -2.09361789,  -7.92710095,  50.90883769,
                2.17845245,  84.30100909,  38.9944903 ,  8.70145126,
                50.44966461,  90.27086491,  68.077767 ,  81.09721966])
```

(c) On prend $x_0 = 0$, ou alors un vecteur aléatoire.

```
In [5]: compteurA=0
x,lnr=GradConj(fA,b,zeros(12))
print("Nombre d'évaluations :",compteurA)

semilogy(lnr)
```

Nombre d'évaluations : 13

```
Out[5]: [<matplotlib.lines.Line2D at 0x7f44fb235fd0>]
```

