On importe les bibliotheques dont on va se servir

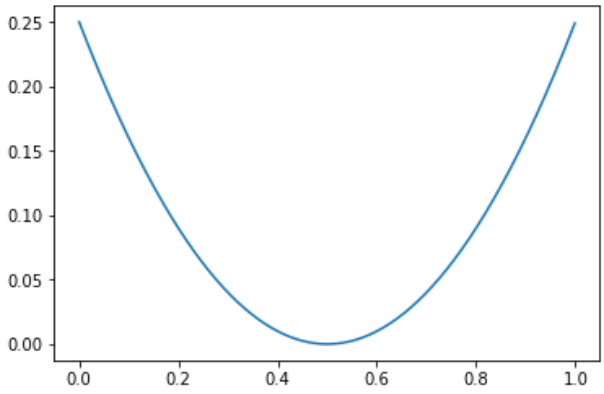
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

On commence par tracer un signal x de $N = 1024$ points qui est la discrétisation de la parabole $t \mapsto (t - \frac{1}{2})^2$ sur l'intervalle $[0, 1]$

In [2]:
```
N=1024
t=np.arange(0,1,1/N)
x=(t-0.5)**2
```

On trace le signal.

In [3]:
```
plt.figure()
plt.plot(t,x)
plt.show()
```

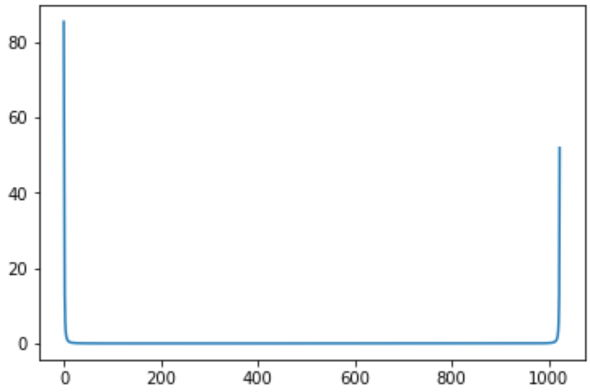


On calcule la transformee de Fourier discrète du signal

In [4]:
```
xchap=np.fft.fft(x)
```

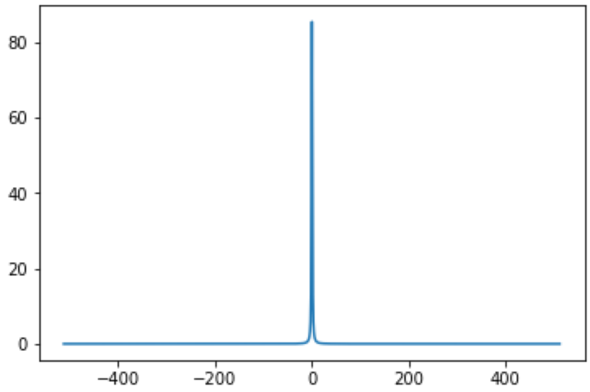
On regarde le resultat. La transformee est complexe ! Il faut donc visualiser le module ou la partie réelle des coefficients.

In [5]:
```
plt.figure()
plt.plot(np.abs(xchap))
plt.show()
```



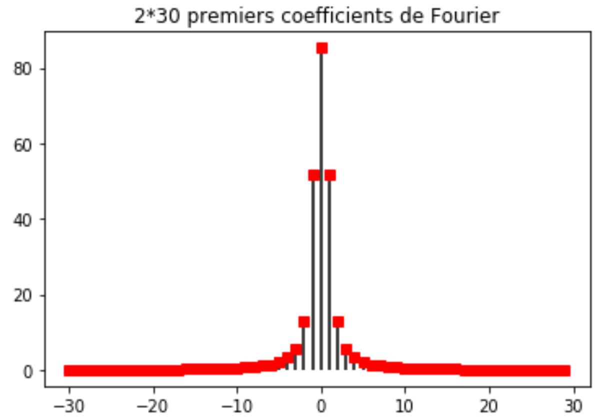
On centre les fréquences en 0.

In [6]:
```
plt.figure()
plt.plot(np.arange(-N//2,N//2),np.fft.fftshift(np.abs(xchap)))
plt.show()
```



On zoome sur les coefficients autour de 0

In [7]:
```
n0=30
plt.figure()
Xchap=np.fft.fftshift(np.abs(xchap))
plt.plot(np.arange(-n0,n0),Xchap[N//2-n0:N//2+n0],'rs')
plt.vlines(np.arange(-n0,n0),[0],Xchap[N//2-n0:N//2+n0])
plt.title('2*'+str(n0)+' premiers coefficients de Fourier')
plt.show()
```



On fixe maintenant M tel que $0 \leq M \leq N/2-1$ et on définit la réponse impulsionnelle d'un filtre notée h^M tel que

- $\widehat{h^M}_{\bar{k}} = 0$ si $|k| > M$
- $\widehat{h^M}_{\bar{k}} = 1$ si $|k| \leq M$

Appliquer ce filtre sur un signal x revient à ne garder que les M premiers coefficients $\langle x, e^n \rangle, n = -M, \ldots, M$ et à mettre à zéro les autres. On calcule donc

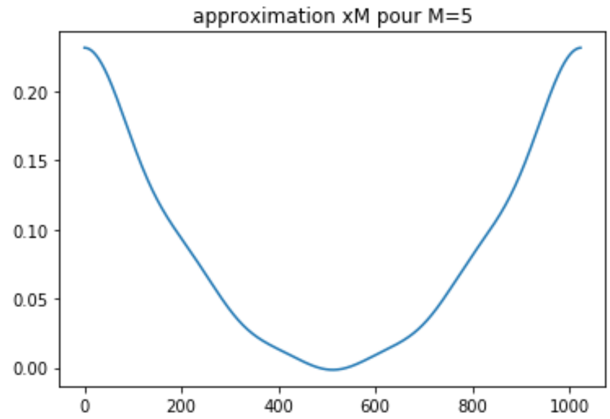
$$x^M = K_{h^M}(x) = \frac{1}{N} \sum_{n=-M}^{M} \langle x, e^n \rangle e^n = \frac{1}{N} \sum_{n=-M}^{M} \hat{x}_n e^n$$

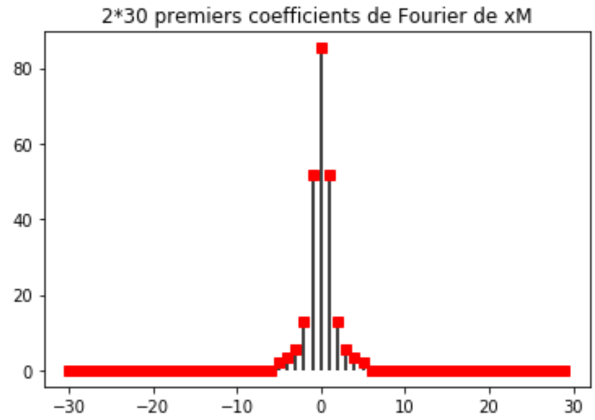
Commençons à voir ce qui se passe pour $M = 5$.

In [8]:
```
M=5
xMchap=np.zeros(xchap.size,dtype=complex)
xMchap[0:M+1]=xchap[0:M+1].copy()
xMchap[N-M:N]=xchap[N-M:N].copy()
xM=np.fft.ifft(xMchap)
xM=xM.real

plt.figure(5)
plt.plot(xM)
plt.title('approximation xM pour M=' +str(M))
plt.show()

n0=30
plt.figure(6)
XMchap=np.fft.fftshift(np.abs(xMchap))
plt.plot(np.arange(-n0,n0),XMchap[N//2-n0:N//2+n0],'rs')
plt.vlines(np.arange(-n0,n0),[0],XMchap[N//2-n0:N//2+n0])
plt.title('2*'+str(n0)+' premiers coefficients de Fourier de xM')
plt.show()
```

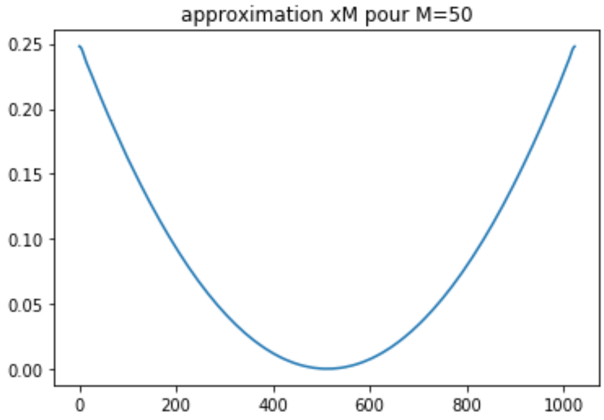




On calcule x^M pour M de plus en plus grand. Qu'observe-t-on ?

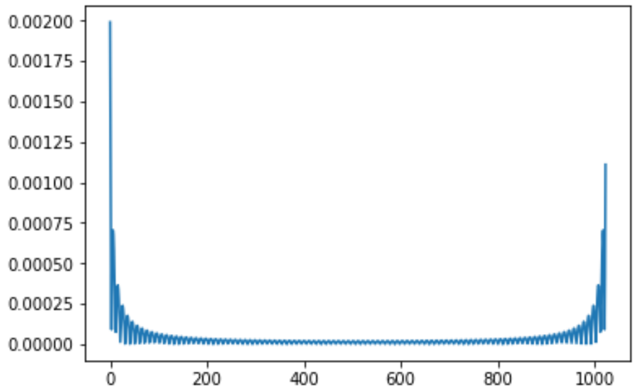
In [9]:
```
M=50
xMchap=np.zeros(xchap.size,dtype=complex)
xMchap[0:M+1]=xchap[0:M+1].copy()
xMchap[N-M:N]=xchap[N-M:N].copy()
xM=np.fft.ifft(xMchap)
xM=xM.real

plt.figure(7)
plt.plot(xM)
plt.title('approximation xM pour M=' +str(M))
plt.show()
```



Il se pourrait qu'on a presque convergé. Vérifions en traçant $|x^M[n] - x[n]|$

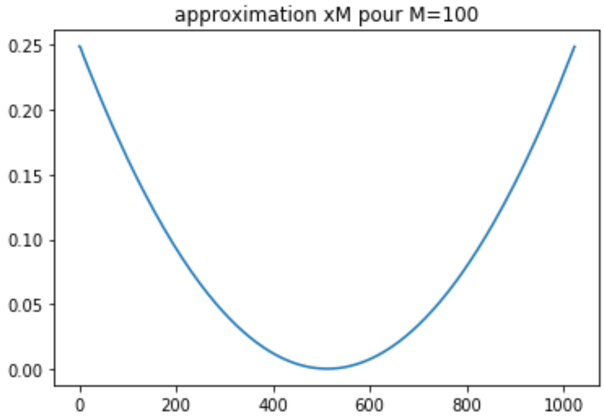
In [10]:
```
plt.figure()
plt.plot(np.abs(xM-x))
plt.show()
```



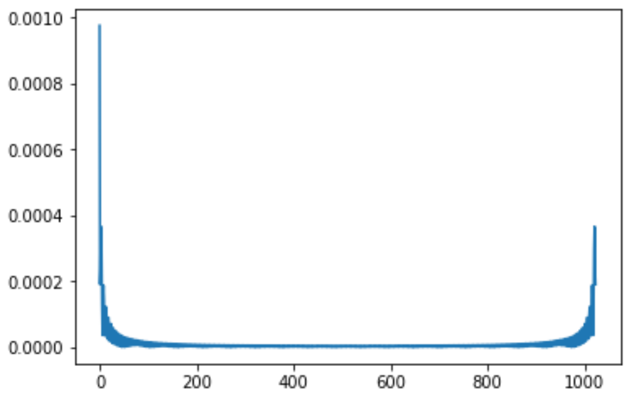
Hum.. Ce n'est pas si petit.. Continuons

In [11]:
```
M=100
xMchap=np.zeros(xchap.size,dtype=complex)
xMchap[0:M+1]=xchap[0:M+1].copy()
xMchap[N-M:N]=xchap[N-M:N].copy()
xM=np.fft.ifft(xMchap)
xM=xM.real

plt.figure(8)
plt.plot(xM)
plt.title('approximation xM pour M=' +str(M))
plt.show()
```



In [12]:
```
plt.figure()
plt.plot(np.abs(xM-x))
plt.show()
```



Calculons $||x^M - x||_2$

In [13]:
```
np.linalg.norm(xM-x,ord=2)
```

Out[13]: 0.0014291938018850747