On importe les bibliotheques dont on va se servir

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

plt.close('all')
```

On commence par tracer un signal creneau x de $N = 2048$ points avec

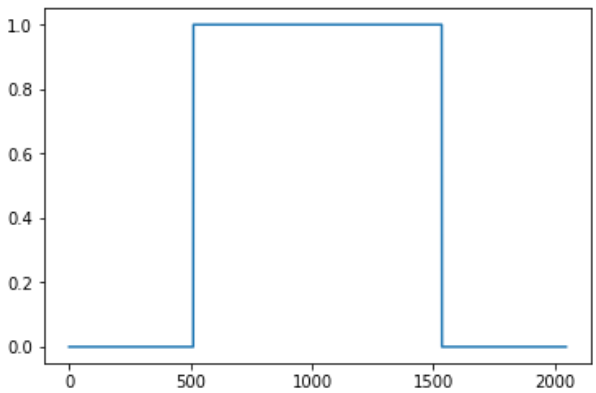
$x_n = 1$ si $N/4 \leq n \leq 3N/4$

$x_n = 0$ sinon.

In [2]:
```
N=2048
x=np.zeros(N)
x[N//4:3*N//4]=1
```

On trace le signal.

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

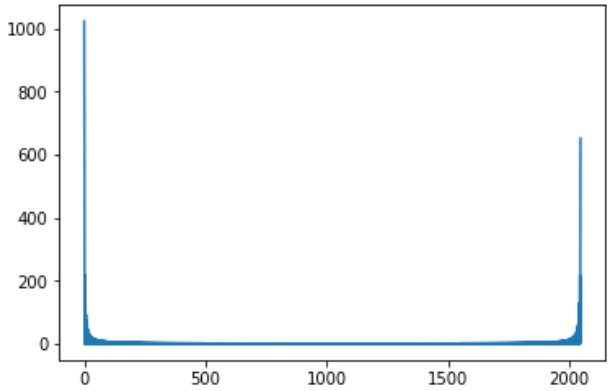


On calcule la transformee discrète du signal

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

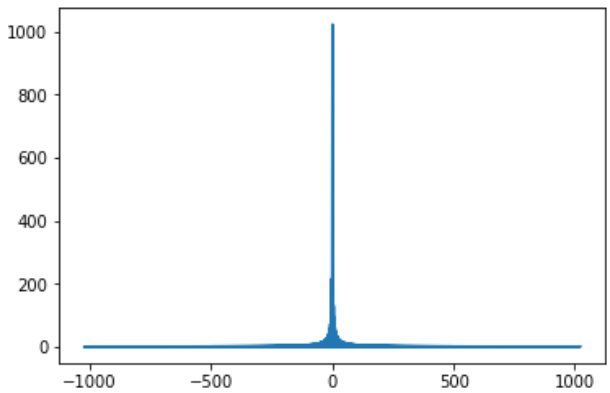
On regarde le resultat. La transformee est complexe ! Il faut donc visualiser le module ou la partie réelle des coefficients.

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



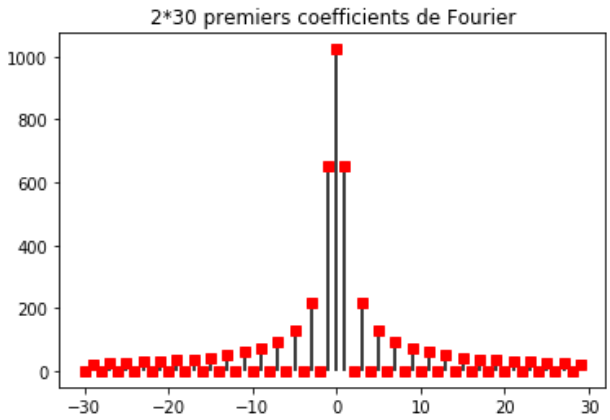
On centre les fréquences en 0.

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()
```



Est ce conforme à ce qu'on attend ? Il faudrait donc calculer $\hat{x}_k = \langle x, e^k \rangle$ avec la formule qui donne les e^k...

On fixe maintenant $0 \leq M \leq N/2 - 1$ et on applique le filtre de réponse impulsionnelle $h^M \in \mathbb{C}^N$ telle que pour $k \in \{-N/2, \dots N/2 - 1\}$

- $\widehat{h^M}_{\bar{k}} = 0$ 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 $\langle x, e^n \rangle, n = -M, \dots, M$ et à mettre à zéro les autres. On calcule donc

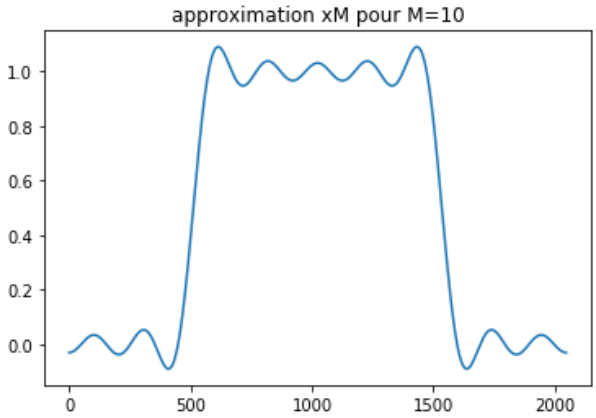
$$x^M = K_{h^M}(x) = \frac{1}{N} \sum_{n=-M}^{M} \langle x, e^n \rangle e^n = \frac{1}{N} \sum_{n=-M}^{M} \hat{x}_n e^n$$

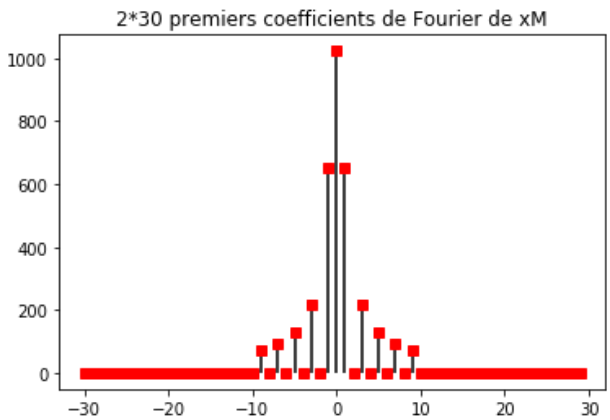
Commençons à voir ce qui se passe pour $M = 10$.

In [8]:
```
M=10
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()
plt.plot(xM)
plt.title('approximation xM pour M=' +str(M))
plt.show()

n0=30
plt.figure()
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()
```

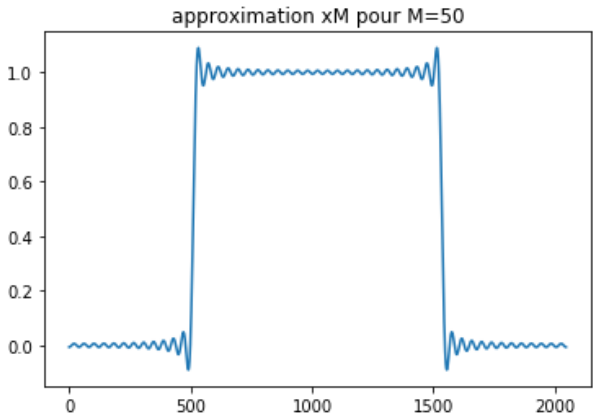




On calcule x^M pour M de plus en plus grand. Qu'observe-t-on ?

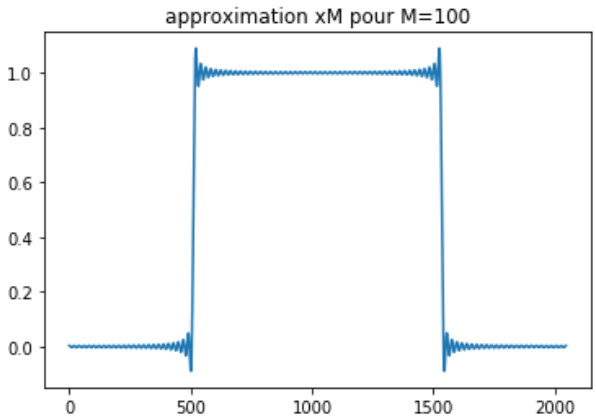
In [9]:
```
M=50
xMchap=np.zeros(xchap.size,dtype=complex)
xMchap[0:M+1]=xchap[0:M+1].copy()
xMchap[N-M:N]=xchap[N-M:N].copy()
xM=np.fft.ifft(xMchap)
xM=xM.real

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



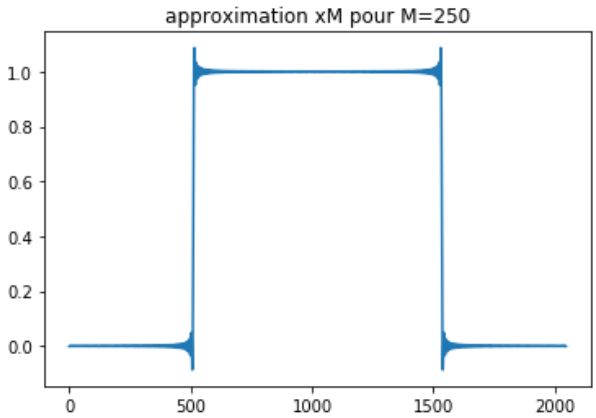
In [10]:
```
M=100
xMchap=np.zeros(xchap.size,dtype=complex)
xMchap[0:M+1]=xchap[0:M+1].copy()
xMchap[N-M:N]=xchap[N-M:N].copy()
xM=np.fft.ifft(xMchap)
xM=xM.real

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



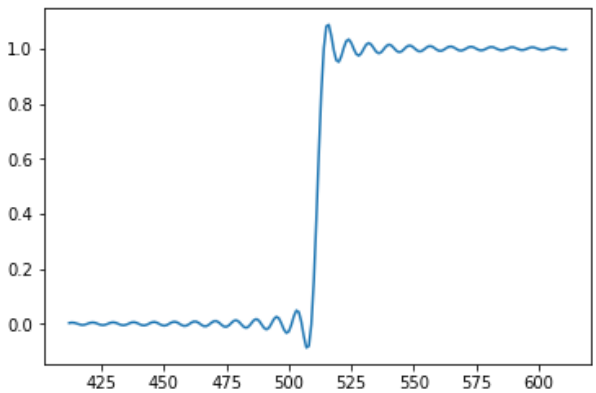
In [11]:
```
M=250
xMchap=np.zeros(xchap.size,dtype=complex)
xMchap[0:M+1]=xchap[0:M+1].copy()
xMchap[N-M:N]=xchap[N-M:N].copy()
xM=np.fft.ifft(xMchap)
xM=xM.real

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



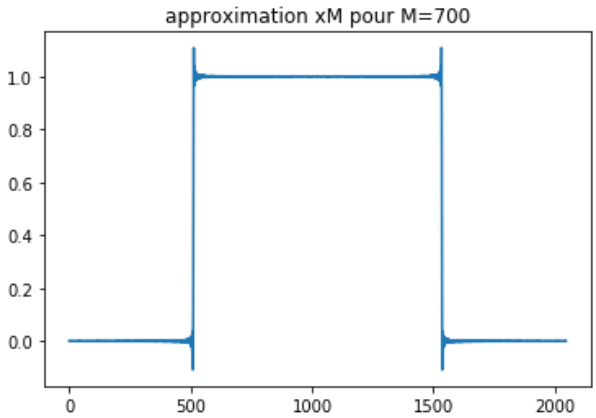
Un zoom autour de la discontinuité

In [12]:
```
plt.figure()
plt.plot(np.arange(N//4-100,N//4+100),xM[N//4-100:N//4+100])
plt.show()
```



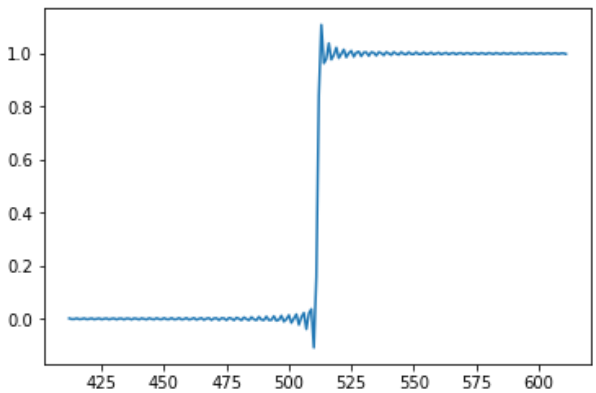
In [13]:
```
M=700
xMchap=np.zeros(xchap.size,dtype=complex)
xMchap[0:M+1]=xchap[0:M+1].copy()
xMchap[N-M:N]=xchap[N-M:N].copy()
xM=np.fft.ifft(xMchap)
xM=xM.real

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



Encore un zoom autour de la discontinuité

In [14]:
```
plt.figure()
plt.plot(np.arange(N//4-100,N//4+100),xM[N//4-100:N//4+100])
plt.show()
```



Les oscillations parasites persistent au voisinage des discontinuités même si on a M de plus en plus grand. C'est ce qu'on appelle le phénomène de Gibbs.