In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import Image#permet l'affihage de photos dans jupyter
from scipy.integrate import odeint# permet la résolution d'éuation différentielle
```

I) Présentation de l'expérience

a) présentation du système

In [2]:

```
Image(filename='systeme.jpg')
```

Out[2]:

In [3]:

```
m=0.396# masse du système en kg
```

b) conditions initiales et déclaration des variables

In [4]:

```
# Conditions initiales connues
g=9.8# accélération du champ de pesanteur
z0=5.8# hauteur du lancé en m # mesuré à l'aide d'un mètre ruban
v0=0# système laché sans vitesse initiale
az0=-g# chute libre : à l'instant initial v=0 seul le poids s'exerce sur notre système
```

In [5]:

```
Image(filename='CI2.png')
```

Out[5]:

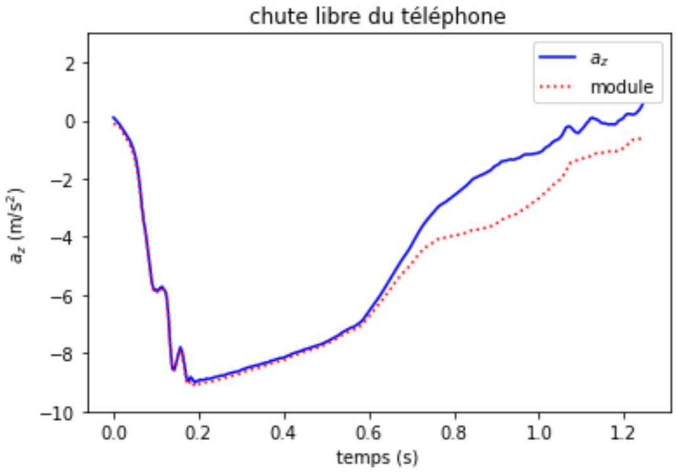
c) Importation des données

In [6]:
```
donnees=np.loadtxt('lance2_m405.csv', skiprows=1,delimiter=';')
```

In [7]:
```
td=donnees[:,0]# extraction du temps
ideb=4300# identification de l'indice correpondant à un instant avant le début de l'expérience
ifin=4562# identification de l'indice correpondant à la fin de l'exercice
t1=td[ideb:ifin]
t=t1-t1[0]
```

In [8]:
```
az=donnees[:,3]# extraction de l'accélération suivant z
az1=az[ideb:ifin]# extraction des données utiles
ma=donnees[:,4]# extraction du module de l'accélération
ma1=ma[ideb:ifin]# donnée utilies par rapport aux indices précédent
```

In [9]:
```
plt.figure()
plt.plot(t,az1,'b-',t,-ma1,'r:')
plt.xlabel('temps (s)')
plt.legend((r'$a_z$','module'),loc=0)
plt.ylabel(r'$a_z$ (m/s$^2$)')
plt.title('chute libre du téléphone')
plt.ylim((-10,3))
plt.show()
```



d) Premières observations

- Nous savons que l'acélération initiale est éagle à -g, on obsevre ainsi qu'à cause du temps de réponse du capteur, on n'a pas une accélération instantanée.
- On conste tout de suite que cette accélération n'est pas constante, on ne peut donc pas négligé les frottements
- Le module de l'accélération et l'accélération suivant uz diffère à partir de 0.6s. Ceci peut être expliqué par une rotation légère du système observé également sur la vidéo. Pour rester dans le modèle du point matériel, on ne considèrera que les instants inférieurs à 0.6s
- L'accéleration tend à tendre vers zéro, le système a presque atteint sa vitesse limite en fin de chute sauf si la rotation du système est à l'origine de cette décélération plus rapide dans la deuxième phase.

II) Résolution de l'équation différentielle

a) Cas d'un coefficient de frottement proportionnel à la vitesse

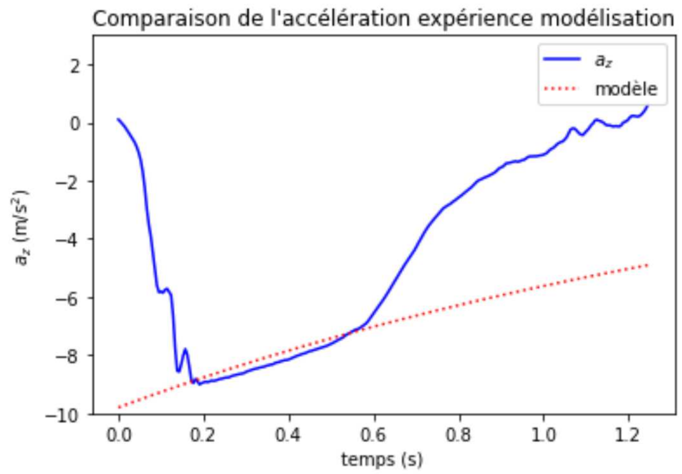
In [10]:
```
alpha=0.22 # valeur obtenue en kg/s pour faire coller la courbe à l'expérience ainsi que la durée de la chute
```

In [11]:
```
def Phi(X,t):
    """ X comprend à la fois la position la vitese et l'accélération"""
    return np.array([X[1],X[2],-alpha/m*X[2]])
```

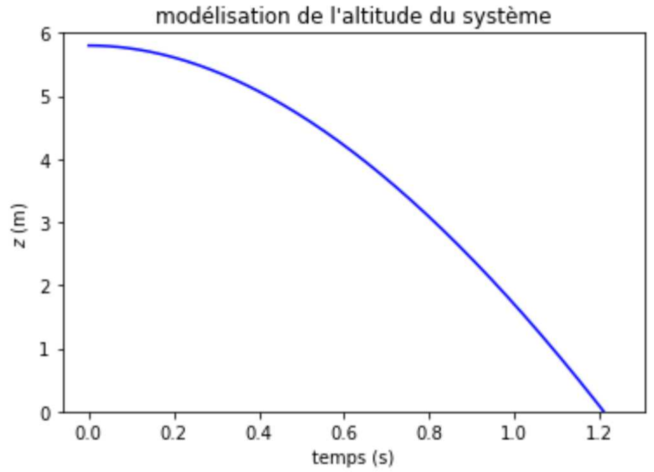
In [12]:
```
X0=[z0,v0,az0]
sol=odeint(Phi,X0,t)
```

In [13]:
```
zmodel=sol[:,0]
vmodel=sol[:,1]
azmodel=sol[:,2]
```

In [14]:
```
plt.figure()
plt.plot(t,az1,'b-',t,azmodel,'r:')
plt.xlabel('temps (s)')
plt.legend((r'$a_z$',r'modèle'),loc=0)
plt.ylabel(r'$a_z$ (m/s$^2$)')
plt.title("Comparaison de l'accélération expérience modélisation")
plt.ylim((-10,3))
plt.show()
```



In [15]:
```
plt.figure()
plt.plot(t,zmodel,'b-')
plt.xlabel('temps (s)')
plt.ylabel(r'$z$ (m)')
plt.ylim((0,6))
plt.title("modélisation de l'altitude du système")
plt.show()
```



Commentaires :

- L'accélération modèle colle avec le capteur pendant l'intervalle de temps considéré
- Le temps de chute obtenue avec l'évolution de $z(t)$ est compatible avec l'expérience
- La vitesse limite n'est finalement pas atteinte contrairement à ce qu'on a identifié lors de la première mesure

b) Cas d'un coefficient de frottement proportionnel à la vitesse au carré

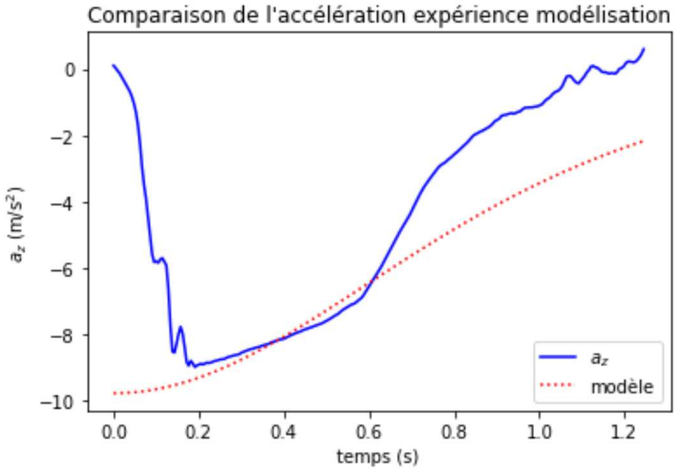
In [16]:
```
beta=0.05# en kg/m
```

In [17]:
```
def Phi2(X,t):
    """ X comprend à la fois la position la vitese et l'accélération"""
    return np.array([X[1],X[2],2*beta/m*X[2]*X[1]])# la force s'oppose à la vitesse est ici orienté suivant +z
```

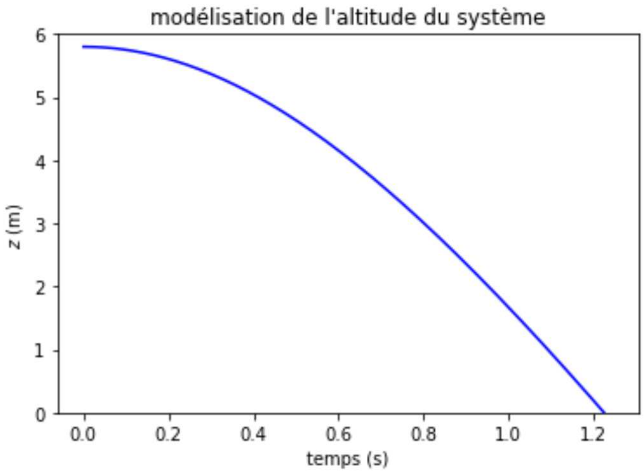
In [18]:
```
X20=[z0,v0,az0]
sol2=odeint(Phi2,X20,t)
```

In [19]:
```
zmode2=sol2[:,0]
vmode2=sol2[:,1]
azmode2=sol2[:,2]
```

In [20]:
```
plt.figure()
plt.plot(t,az1,'b-',t,azmode2,'r:')
plt.xlabel('temps (s)')
plt.legend((r'$a_z$',r'modèle'),loc=0)
plt.ylabel(r'$a_z$ (m/s$^2$)')
plt.title("Comparaison de l'accélération expérience modélisation")
plt.show()
```



In [21]:
```
plt.figure()
plt.plot(t,zmode2,'b-')
plt.xlabel('temps (s)')
plt.ylabel(r'$z$ (m)')
plt.ylim((0,6))
plt.title("modélisation de l'altitude du système")
plt.show()
```



Commentaires

- En ajustant la valeur de beta pour que la courbe expérimentale colle à la modélisation d'une part et à la hauteur de chute d'autre part, on constate que la courbe modèle "colle" moins d'avec une force de frottement proportionnelle à la vitesse
- On conclue qu'il est plus probable de modéliser la force de frottement par une force linéaire avec $\alpha = 0.22$ kg/s
- Un modèle plus précis pourrait considérer deux phases : une première phase où la vitesse est faible puis un second modèle de frottement pour des vitesses plus importante. Il faudrait pour cela multiplier les expériences, pour vérifier si celles-ci sont reproductibles...