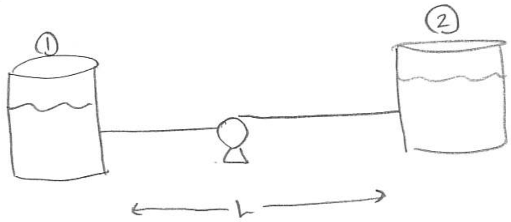
Example:

Suppose I have a pipeline from tank 1 to tank 2



$\Delta z = 4m$

$D = 1m$ (pipe diameter)

$L = 200\ m$

hydraullically smooth

* The pump performance curve is given by

$$h_p(Q) = h_{max} - BQ^2 \qquad h_{max} = 13m$$

$$B = 0.05\ m/Lpm^2 \qquad Lpm = \frac{\text{Liters}}{\text{meter}}$$

* You may assume:
  - a constant friction factor, $f = 0.002$
  - no minor losses (sudden contraction/expansion, bends, etc.)

* what is h_{sys}?

$$h_{sys} = \Delta H + h_L = \underbrace{\frac{\Delta P}{\rho g} + \frac{\Delta v^2}{2} + \Delta z}_{\Delta H} + \underbrace{\frac{v^2}{2g}\left(\frac{4Lf}{D}\right)}_{h_L}$$

- pick ① & ② at the top of the tank.

$\Delta P = 0$ (Patm for both)

$\Delta v = 0$ (even if D changed)

} normally, need to include expansion & contraction for pipe entrance/exit.

- let $v = \frac{4Q}{\pi D^2}$

$$h_{sys} = \Delta z + \left(\frac{4Q}{\pi D^2}\right)^2 \frac{1}{2g}\left(\frac{4Lf}{D}\right) = \Delta z + \frac{32}{\pi^2}\frac{Lf}{gD^5}Q^2$$

* Solve for the operating point:

$$h_{max} - BQ^2 = \Delta z + \frac{32}{\pi^2}\frac{Lf}{gD^5}Q^2$$

$$\left(B + \frac{32}{\pi}\frac{Lf}{gD^5}\right)Q^2 = h_{max} - \Delta z$$

$$Q = \sqrt{\frac{h_{max} - \Delta z}{B + \frac{32}{\pi}\frac{Lf}{gD^5}}}$$

Operating Point

$Q = 7.03\ Lpm$

$h = 10.53\ m$

Lecture 33 -- Example ¶

In [59]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve
```

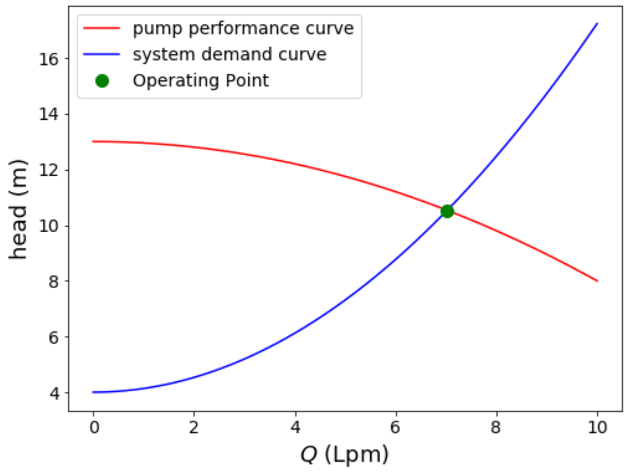
In [60]:
```
# constants (water)
g = 9.8 #m/s^2
rho = 1000 # kg/m^3
mu = 1e-3 # kg/m/s

# constants for the pump
hmax = 13 # m
B = 0.05 # m/Lpm^2

# constants for the system
dz = 4 # m
D = 1. # m
L = 200 # m
k = 0 # m (roughness)
```

Easy version (f is a constant)

In [84]:
```
f = 0.002 # dimless

def hp(Q):
    return hmax - B*Q**2

def hsys(Q):
    v = 4*Q/np.pi/D**2
    return dz + v**2/(2*g)*(4*L*f/D)

# solve for OP (by hand)
Q_OP1 = np.sqrt((hmax - dz)/(B + 32/np.pi**2*L*f/g/D**5))
h_OP1 = hp(Q_OP1)

print('---------------------')
print('Operating point')
print('Q_OP = ', np.round(Q_OP1,2), ' (Lpm)')
print('h_OP = ', np.round(h_OP1,2), ' (m)')
print('---------------------')
```

```
---------------------
Operating point
Q_OP =  7.03  (Lpm)
h_OP =  10.53  (m)
---------------------
```

In [85]:
```
# make a plot
Q_plt = np.linspace(1e-5, 10, 201) # Lpm

plt.rc('font', size=14)
plt.figure(figsize=(8,6))
plt.plot(Q_plt, hp(Q_plt), '-r', label='pump performance curve')
plt.plot(Q_plt, hsys(Q_plt), '-b', label='system demand curve')
plt.plot(Q_OP1, h_OP1, 'go', label='Operating Point', markersize=10)
plt.legend()
plt.xlabel('$Q$ (Lpm)', fontsize=18)
plt.ylabel('head (m)', fontsize=18)
plt.show()
```



Harder version (f is not a constant)

In [86]:
```
def hp(Q): # m
    return hmax - B*Q**2

def hsys(Q): # m
    v = 4*Q/np.pi/D**2
    Re = rho*v*D/mu
    f = ( 3.6 * np.log10(6.9/Re + (k/D*1/3.7)**(1.11)) )**(-2)
    return dz + v**2/(2*g)*(4*L*f/D)

def phi(Q):
    return hp(Q) - hsys(Q)

# solve for OP (via fsolve)
Q_guess = 7 # Lpm
Q_OP2 = fsolve(phi, Q_guess)[0]
h_OP2 = hp(Q_OP1)

print('----------------------')
print('Operating point')
print('Q_OP = ', np.round(Q_OP2,2), ' (Lpm)')
print('h_OP = ', np.round(h_OP2,2), ' (m)')
print('----------------------')
```

```
----------------------
Operating point
Q_OP =  6.94  (Lpm)
h_OP =  10.53  (m)
----------------------
```

In [87]:
```
# make a plot
Q_plt = np.linspace(1e-5, 10, 201) # Lpm

plt.rc('font', size=14)
plt.figure(figsize=(8,6))
plt.plot(Q_plt, hp(Q_plt), '-r', label='pump performance curve')
plt.plot(Q_plt, hsys(Q_plt), '-b', label='system demand curve')
plt.plot(Q_OP1, h_OP1, 'go', label='Operating Point', markersize=10)
plt.plot(Q_OP2, h_OP2, 'mo', label='Operating Point', markersize=10, markeredgewidth=3, fillstyle='none')
plt.legend()
plt.xlabel('$Q$ (Lpm)', fontsize=18)
plt.ylabel('head (m)', fontsize=18)
plt.show()
```