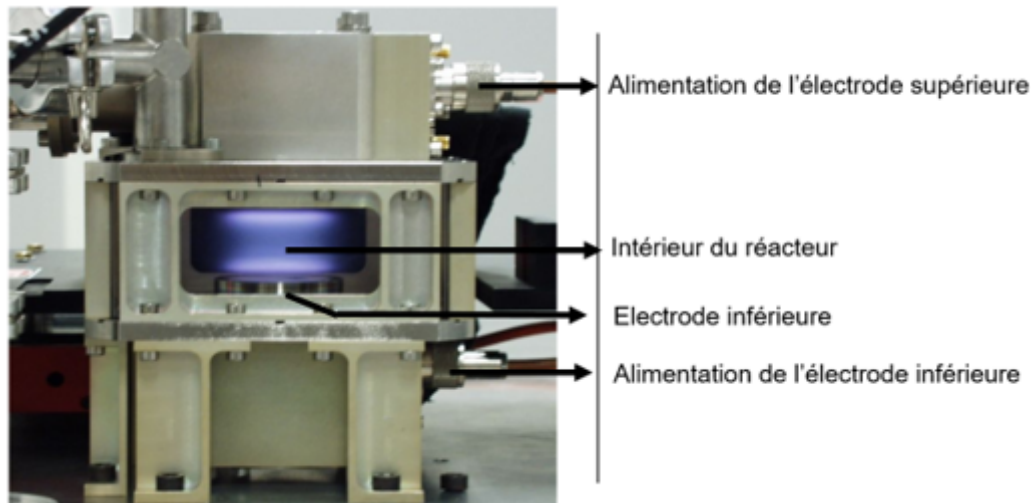


1 # Champ électrique dans un réacteur 716b-1603581

Au laboratoire du GREMI d'Orléans, on utilise un réacteur assimilable à deux électrodes planes et parallèles afin de générer des plasmas de laboratoire.



L'électrode inférieure est chargée au potentiel $V_p = 400V$ et l'électrode supérieure au potentiel $-V_p$. Le milieu entre les électrodes est encore assimilable à du vide.

1) Montrer que le potentiel électrostatique $V(M)$ entre les électrodes doit vérifier l'équation $\Delta V=0$ (appelée équation de Laplace)

La physique impose dans un milieu vide de charge et en régime stationnaire :

$$\begin{aligned} \text{div} \vec{E} &= 0 \\ \vec{E} &= -\vec{\text{grad}}(V) \end{aligned}$$

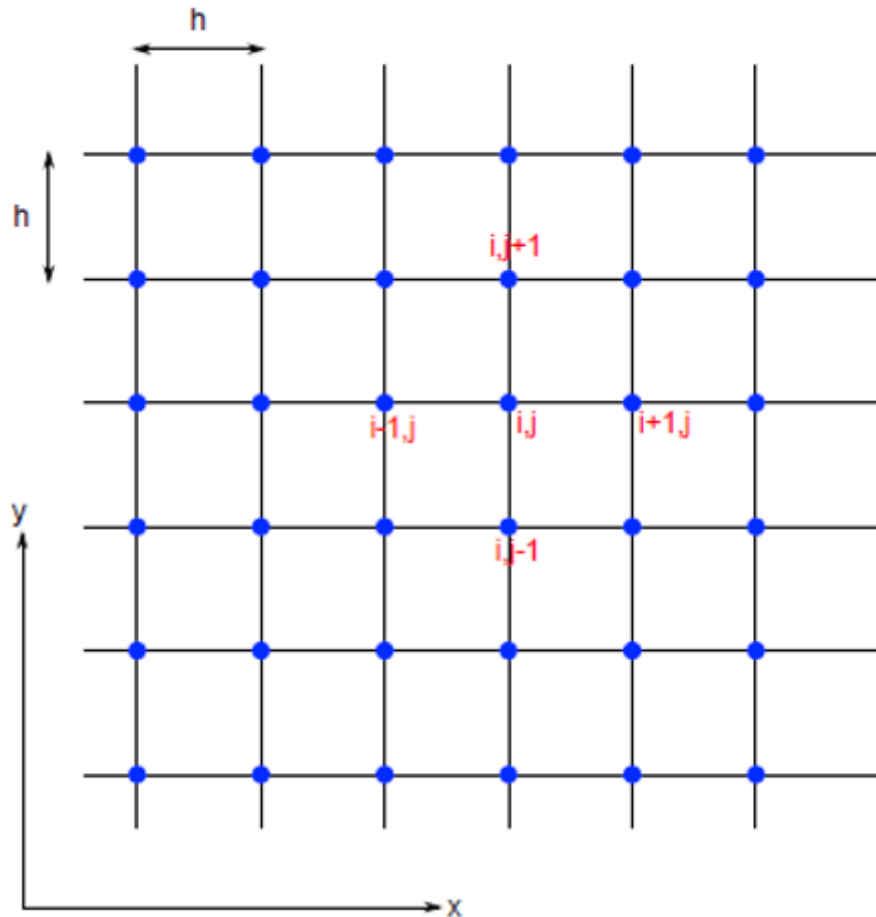
Soit :

$$\Delta V = 0$$

On ramène ce problème à deux dimensions dans le plan P vertical médiateur du réacteur carré de côté $d=20\text{cm}$ et on suppose $V(x,y)$. Dans ces conditions, l'équation de Laplace devient :

$$\frac{\partial^2 V(x,y)}{\partial x^2} + \frac{\partial^2 V(x,y)}{\partial y^2} = 0$$

On va utiliser un maillage de P, de pas $h=d/(N-1)$ avec $N \times N$ le nombre de points du maillage (discrétisation identique dans les deux directions de l'espace). Un point M est alors repéré par $[x_i = ih, y_j = jh]$ et sera identifié plus simplement par le couple $[i, j]$.



Le potentiel en M est alors noté $V(M) = V[x_i, y_j] = V[i, j]$.

2) En utilisant la formule de différence finie centrée pour approximer les dérivées secondes, montrer que :

$$V[i, j] = \frac{(V[i + 1, j] + V[i - 1, j] + V[i, j + 1] + V[i, j - 1])}{4} \text{ (Équation 1)}$$

La solution (unique) de ce problème doit aussi vérifier les conditions aux limites imposées au potentiel par l'expérimentateur (conditions de Dirichlet). Ces conditions imposées au potentiel sont :

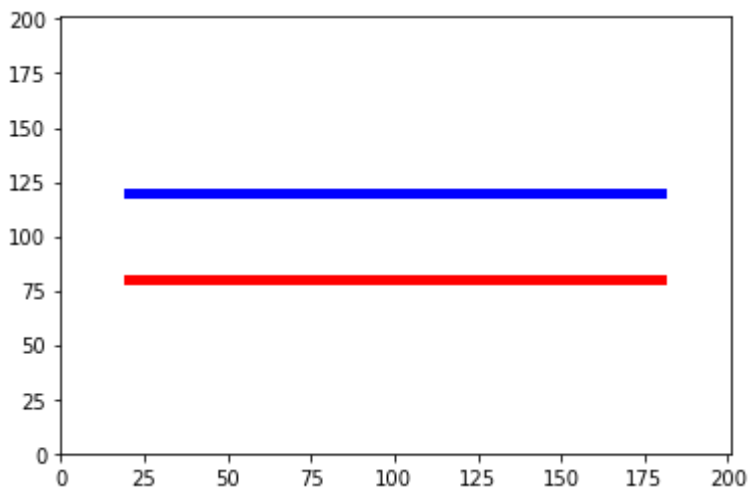
- $\pm V_p$ aux électrodes
- $V=0$ sur les parois du réacteur Les lignes de codes ci-dessous initialisent le programme :

In [6]:

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 N=201#nombre de lignes
4 M=201#nombre de colonnes
5 Vp=400#potentiel en Volt
6 V1=np.zeros((N,M))
7 V1[80,20:180]=Vp
8 V1[120,20:180]=-Vp
9 plt.plot([20,180], [80,80], 'r-', lw = 5) # plaque inf
10 plt.plot([20,180], [120,120], 'b-', lw = 5) # plaque sup
11 plt.axis([0,201,0,201]) # [x_min,x_max,y_min,y_max]
12 plt.show()
13

```



Pour résoudre ce problème, on va utiliser une méthode itérative (méthode de Jacobi). A partir du tableau initial V_1 , on calcule une nouvelle valeur du potentiel pour tous les points à l'aide de l'équation 1 en maintenant les conditions aux limites.

Le processus est ensuite répété jusqu'à obtenir des valeurs de potentiels stables. Au bout de k itérations, on a un potentiel $V_k[i,j]$ et le calcul est stoppé à l'aide d'un critère de convergence :

$$\varepsilon = \max |V_k[i,j] - V_{(k-1)}[i,j]| < \varepsilon_c$$

Où ε_c est un seuil de convergence.

On donne ci-dessous, le programme python permettant de réaliser cette méthode itérative et utilisant deux boucles « for ».

In []:

```

1  """itération sans vectorisation"""
2  V2=V1.copy() #pour éviter les effets de bords !!!essentiel pour la 1e comparai
3  epsilon =0.01#seuil de convergence
4  ecart = 1#initialisation
5  while ecart>epsilon:
6      for i in range (1,N-1):
7          for j in range (1,N-1):
8              V2[i,j]=0.25*(V1[i-1,j]+V1[i+1,j]+V1[i,j-1]+V1[i,j+1])
9              V2[80,20:180]=Vp #on impose à chaque itération les conditions aux
10             V2[120,20:180]=-Vp
11         ecart = np.max(abs(V2[:, :]-V1[:, :]))
12         V1=V2.copy() #permet de comparer V entre deux itérations
13

```

Le programme précédent se termine au bout de 858s (sur mon ordinateur) après 5547 itérations !!!

Python est un langage interprété qui n'est pas mis en valeur avec des programmes utilisant des boucles for imbriquées pour calculer des tableaux.

3) Réécrire le programme précédent en proposant une version vectorisée (sans les deux boucles for) permettant d'obtenir :

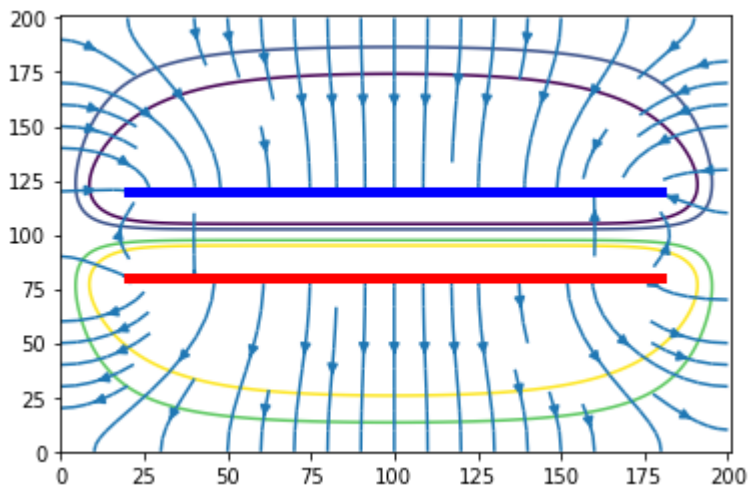
- plus rapidement le tableau numpy V2.
- le tracé de quelques équipotentiels
- le tracé de quelques lignes de champ électrique

In [13]:

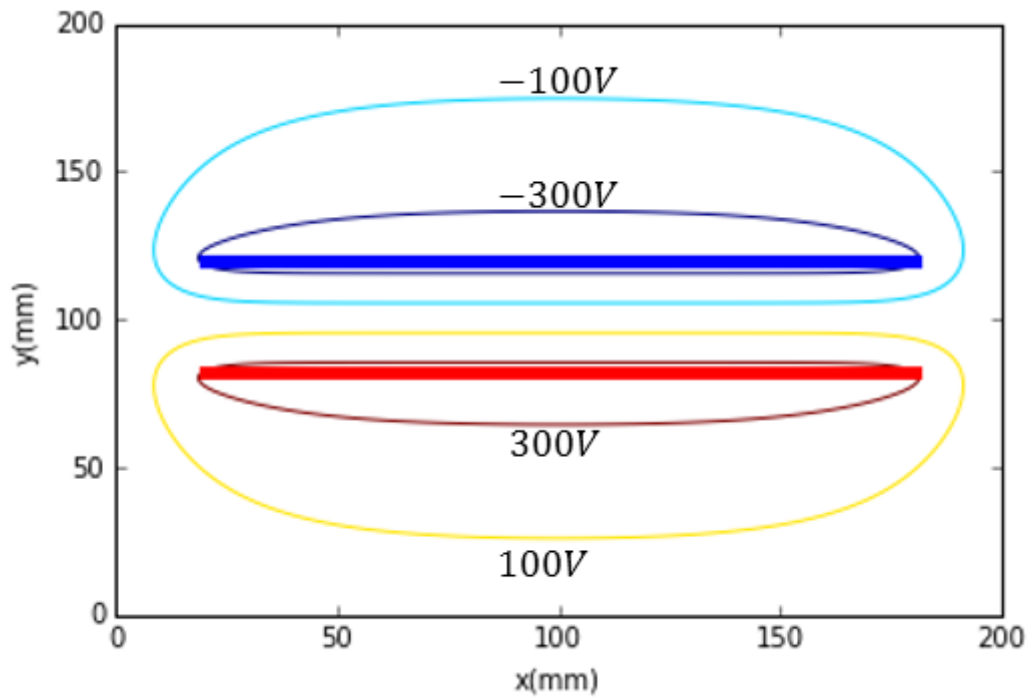
```

1  """itération avec vectorisation"""
2  V2=V1.copy() #pour éviter les effets de bords !!!essentiel pour la 1e comparai
3  epsilon =0.01#seuil de convergence
4  ecart = 1#initialisation
5  while ecart>epsilon:
6      V2[1:-1,1:-1]=0.25*(V1[0:-2,1:-1]+V1[2:,1:-1]+V1[1:-1,0:-2]+V1[1:-1,2:])
7      V2[80,20:180]=Vp #on impose à chaque itération les conditions aux limites
8      V2[120,20:180]=-Vp
9      ecart = np.max(abs(V2[:, :]-V1[:, :]))
10     V1=V2.copy() #permet de comparer V entre deux itérations
11 x=np.arange(0,M,1)#nbre de colonne
12 y=np.arange(0,N,1)#nbre de ligne
13 X,Y=np.meshgrid(x,y)
14 scale=[-100,-50,50,100]
15 plt.contour(X,Y,V2,scale)
16 Ey,Ex=np.gradient(V2)
17 Ex,Ey=-Ex,-Ey
18 plt.streamplot(X,Y,Ex,Ey,density=0.7)
19 plt.plot([20,180], [80,80], 'r-', lw = 5) # plaque inf
20 plt.plot([20,180], [120,120], 'b-', lw = 5) # plaque sup
21 plt.axis([0,201,0,201]) # [x_min,x_max,y_min,y_max]
22 plt.show()

```



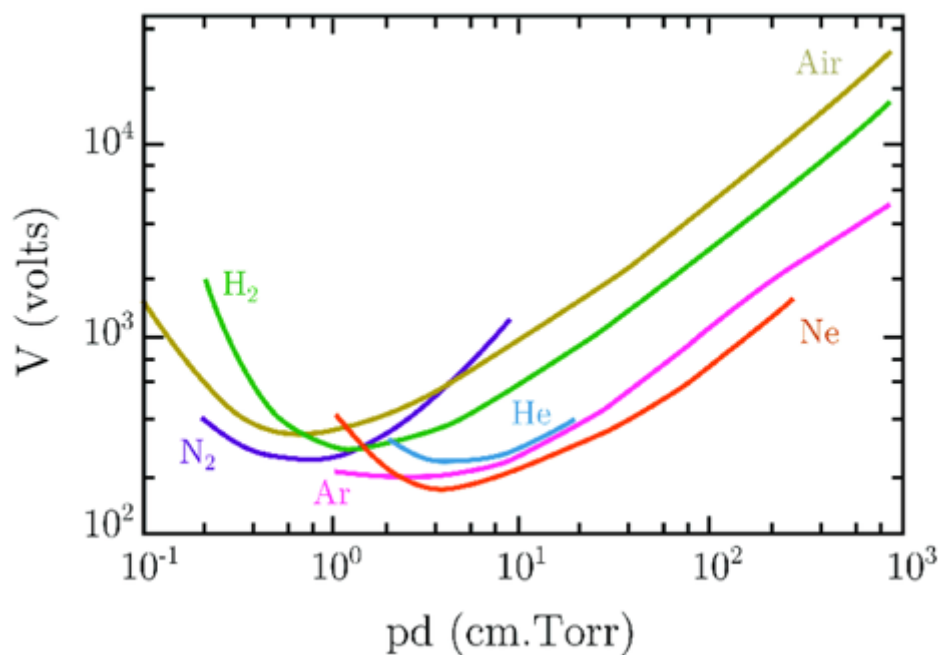
A l'issue de la simulation (en 3s avec vectorisation !!!!), il est possible d'obtenir le tracé de quelques équipotentiels dans le réacteur. Le pas qui a été choisi pour la simulation est $h=1\text{mm}$:



4) Peut-on négliger les effets de bords dans la région inter-électrode (assimilable à du vide) et supposer le champ uniforme ? Justifier.

Au coeur de la région inter-électrode, on a des équipotentiels régulièrement espacés : c'est le signe d'un champ électrique uniforme. En revanche, la situation est plus complexe en dehors de cette région.

On donne ci-dessous le graphe représentant la loi de Paschen donnant la tension V à appliquer entre deux électrodes distantes de $d(\text{cm})$ en fonction du produit pd où p est la pression du gaz en Torr ($1\text{Torr}=133\text{Pa}$)



5) A partir de quelle valeur de pression en Pa n'est-il plus possible d'ioniser un gaz d'argon ?

Donc : $pd=70\text{cm.Torr}$

soit une pression maximale de $p_{\text{max}} = (70 \times 133) / 20 \approx 465 \text{ Pa}$

In []:

1	
---	--