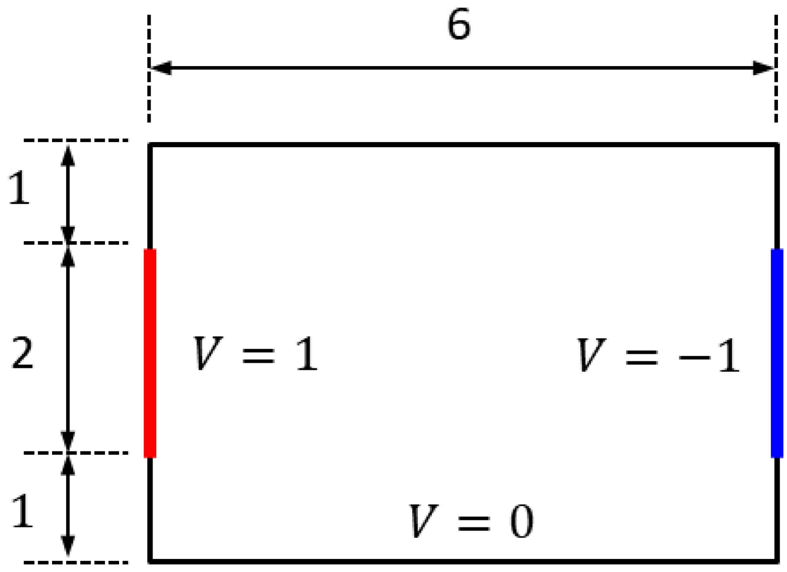
Question 1. Find the electric potential inside a 2D rectangle. There are two electrodes with V=+1 and -1, respectively. All the other parts of the boundary are at ground (V=0). It is a 2D problem.



In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
# Get some parameters
Nx=int(30)  # num. meshes in x
xmax=float(6)        # Max x
Ny=int(100)  # num. meshes in y
ymax=float(4)        # Max y
imax=int(100)   # max step to iterate
tol=float(1e-5)   # error tolerence. something like 1e-5
```

In [3]:

```
# location of eletric rode
ylow = 1
yhigh = 3
```

In [4]:

```
# Setup arrays and parameters.
dx=xmax/Nx    # mesh size in x
dy=ymax/Ny    # mesh size in y
x=np.arange(0,Nx+1)  # x-array
y=np.arange(0,Ny+1)  # y-array
X,Y=np.meshgrid(x,y) # Create a 2D mesh grid
u=0.0*X              # u-array, the potential
resid=0.0*X          # residue
```

In [5]:

```
# Jacobi spectral radius
rjac=(np.cos(np.pi/Nx) + (dx/dy)**2*np.cos(np.pi/Ny))/(1+(dx/dy)**2)
```

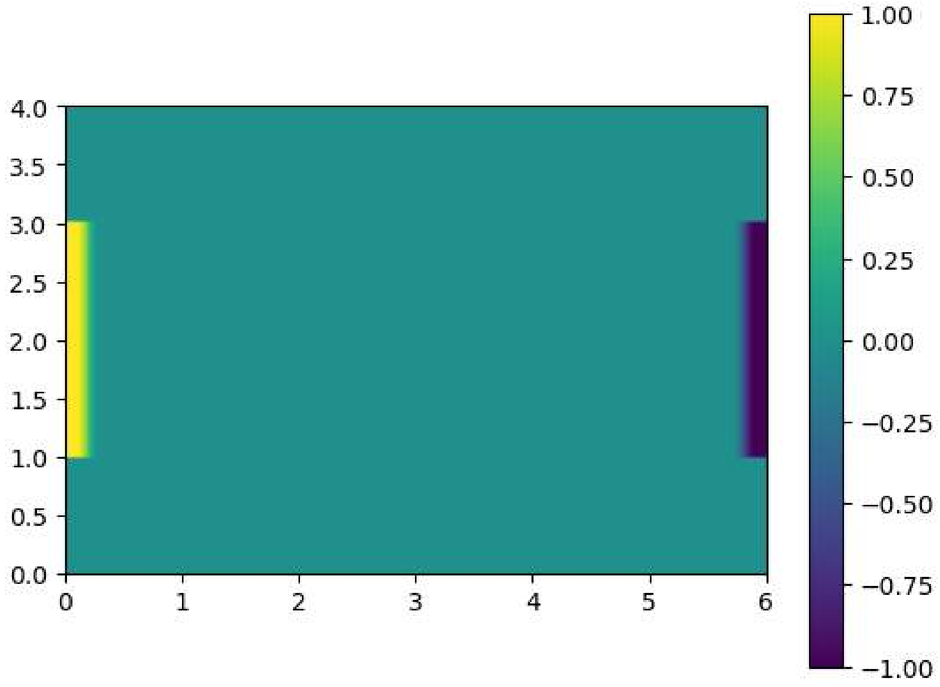
Set the boundary condition

In [6]:

```
# set the Eletric rode
#s = np.zeros(X.shape)
u[int(ylow/dy):int(yhigh/dy+1),0] = 1
u[int(ylow/dy):int(yhigh/dy+1),-1] = -1
# s[0:2, int(ylow/dy):int(yhigh/dy+1)] = 1
# s[-3:-1, int(ylow/dy):int(yhigh/dy+1)] = -1
#s=np.exp(-((X/Nx-0.5)**2+(Y/Ny-0.5)**2)/0.04)
```

In [7]:

```
#plot initial potential
extn=[0,xmax,0,ymax]
cs=plt.imshow(u,extent=extn); plt.colorbar(cs); plt.show()
```



In [9]:

```
# initial error
errf=sum(sum(np.fabs(u)))
err= errf  # initial error
```

In [10]:

```
err
```

Out[10]:

```
102.0
```

In [11]:

```
# Coefficients
a=1.0
b=1.0
c=(dx/dy)**2
d=(dx/dy)**2
e=-2.0-2.0*(dx/dy)**2
```

In [12]:

```
# initial overrelaxation factor
omega=1
```

In [13]:

```
i=0;
while err > (tol*errf) and i<imax:
      # odd calculation
    j=1; lsw=1
    while j < Nx:
        resid[lsw:-1:2, j] =  a*u[lsw:-1:2, j+1] + b*u[lsw:-1:2, j-1] ₩
                                    + c*u[lsw+1::2, j] + d*u[lsw-1:-2:2, j] ₩
                                    + e*u[lsw:-1:2, j]# + s[lsw:-1:2, j]  :delete charge distribution term
        u[lsw:-1:2, j] -= omega*resid[lsw:-1:2, j]/e
        err=sum(sum(np.fabs(resid)))
        j+=1
        lsw=3-lsw
    omega=1/(1-0.5*rjac**2) if i==1 else 1/(1-0.25*rjac**2*omega)

      # even calculation
    j=1;  lsw=2
    while j < Nx:
        resid[lsw:-1:2, j] =  a*u[lsw:-1:2, j+1] + b*u[lsw:-1:2, j-1] ₩
                                    + c*u[lsw+1::2, j] + d*u[lsw-1:-2:2, j] ₩
                                    + e*u[lsw:-1:2, j]# + s[lsw:-1:2, j]  delete the charge distribution term
        err += sum(sum(np.fabs(resid)))
        j+=1
        lsw=3-lsw
    omega = 1/(1-0.25*rjac**2*omega)
    i+=1
```

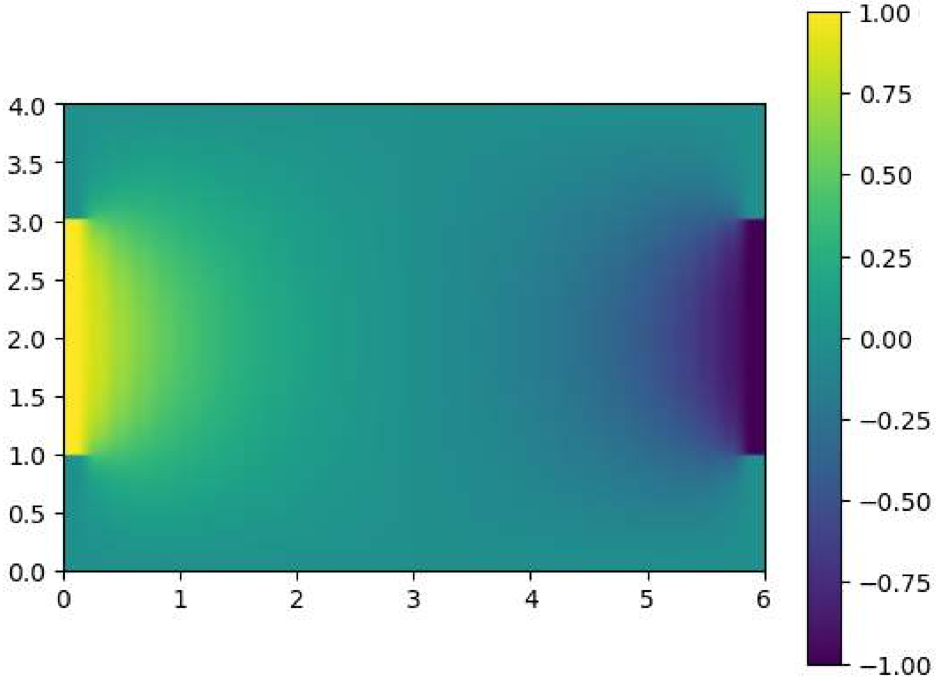
In [14]:

```
# print out run info.
print("Number of iteration=",i)
print("Relative Error=",err/errf)
```

```
Number of iteration= 100
Relative Error= 1.2095626386109868
```

In [15]:

```
# plot of the resulting potential
extn=[0,xmax,0,ymax]
cs=plt.imshow(u,extent=extn); plt.colorbar(cs);  plt.show()
```



In [ ]: