```
In [1]: %matplotlib notebook
        %pylab
```

```
Using matplotlib backend: nbAgg
Populating the interactive namespace from numpy and matplotlib
```

Poisson's Equation

Poisson's equation is the name we usually give to the higher dimensional version of the 1d elliptic problem we discussed earlier. A *constant coefficient* Poisson problem (named after French mathematician [Siméon Denis Poisson (https://en.wikipedia.org/wiki/Sim%C3%A9on_Denis_Poisson)](https://en.wikipedia.org/wiki/Sim%C3%A9on_Denis_Poisson), 1781-1840), is given by

$$\nabla^2 u = f(\mathbf{x}), \qquad \text{where} \quad \mathbf{x} \in \Omega$$

where Ω may be a two- or three- dimensional domain. Examples of Ω include finite domains (e.g. unit disk, unit sphere, rectangular regions, or any other closed two or three dimensional region), semi-infinite domains (e.g. half-planes) or infinite domains (all of $\mathcal{R}^2$ or $\mathcal{R}^3$).

On the boundary of the domain, we can impose *Dirichlet* conditions,

$$u(\mathbf{x}) = g(\mathbf{x}), \qquad x \in \partial\Omega,$$

Neumann conditions

$$\frac{\partial u(\mathbf{x})}{\partial n} = g(\mathbf{x}), \qquad x \in \partial\Omega,$$

or *Robin* (mixed) conditions.

$$au(\mathbf{x}) + b\frac{\partial u(\mathbf{x})}{\partial n} = g(\mathbf{x}), \qquad x \in \partial\Omega,$$

For infinite domains, we will typically impose conditions that do now allow the solution to grow at infinity.

The Poisson problem is actually a generalization of *Laplace*'s equation, given by

$$\nabla^2 u = 0$$

subject to Dirichlet, Neumann or mixed boundary conditions.

A *variable coefficient* Poisson problem is given by

$$\nabla \cdot \beta(\mathbf{x})\nabla u = f$$

where $\beta(\mathbf{x})$ is a prescribed function. Written in this form, the Poisson problem can be viewed as expressing conditions on the divergence of a flux, where the flux is given by

$$\mathbf{F} = \beta(\mathbf{x})\nabla u$$

Both the constant coefficient and the variable coefficient problems appear frequently in models of many physical phenomena, including computational fluid dynamics (the "Pressure Poisson equation", potential flow), electrostatics, and magnetohydrodynamics (MHD).

Distributions in higher dimensions

We can extend our one-dimensional delta function $\delta(x)$ to higher dimensions. To do this, we will use a tensor product definition

$$\delta(\mathbf{x}) = \delta(x)\delta(y), \qquad \mathbf{x} \in \mathcal{R}^2$$

or

$$\delta(\mathbf{x}) = \delta(x)\delta(y)\delta(z), \qquad \mathbf{x} \in \mathcal{R}^3$$

Where it is convenient, we may write $\delta(x, y)$, or $\delta(x, y, z)$ for these higher dimensional distributions. These higher dimensional distributions will have the properties analogous to their one dimensional counterparts, namely

$$\int_{\mathcal{R}^d} \delta(\mathbf{x}) \; d\mathbf{x} = 1, \qquad d = 2, 3$$

and

$$\int_{\mathcal{R}^d} f(\mathbf{y}) \; \delta(\mathbf{y} - \mathbf{x}) \; d\mathbf{y} = f(\mathbf{x})$$

Fundamental solutions in 3d

As a basic building block to a wide variety of Poisson problems, we will seek a Green's function solution to the fundamental problem

$$\nabla^2 u = \delta(\mathbf{x})$$

To start, we will consider the problem in $\mathbf{R}^3$. Since the delta distribution is spherically symmetric, we can re-write the fundamental problem in terms of a radial coordinate r

$$\frac{\partial^2 u}{\partial r^2} + \frac{2}{r}\frac{\partial u}{\partial r} = \delta(r)$$

It is easy to verify that the solution to the homogeneous problem is given by

$$u = \frac{C_0}{r} + C_1.$$

Imposing $u(\infty) = 0$, we set $C_1 = 0$. To evaluate C_0, we integrate our equation over a sphere of radius ϵ centered at the origin to get

$$\int_{S_\epsilon} \nabla^2 u(\mathbf{x}) d\mathbf{x} = \int_{S_\epsilon} \delta(\mathbf{x}) d\mathbf{x} = 1$$

Using Gauss's Divergence Theorem, we can compute the left hand side as

$$\int_{S_\epsilon} \nabla \cdot \nabla u(\mathbf{x})\ d\mathbf{x} = \int_{\partial S_\epsilon} \frac{\partial u}{\partial r}\ dA = \cdots = -4\pi C_0 = 1.$$

From this, we get that $C_0 = -\frac{1}{4\pi}$ and the fundamental solution is given by

$$u(r) = -\frac{1}{4\pi r}$$

If the source is located at $\mathbf{y}$, then the general solution to

$$\nabla^2 u = \delta(\mathbf{x} - \mathbf{y})$$

is given by

$$G_0(\mathbf{x}, \mathbf{y}) = -\frac{1}{4\pi|\mathbf{x} - \mathbf{y}|}, \qquad \mathbf{x}, \mathbf{y} \in \mathcal{R}^3$$

Since this is a solution in the infinite domain, we will call this a *fundamental solution*.

Fundamental solutions in 2d

Following essentially the same steps as we did for the 3d case, we can solve the fundamental problem

$$\nabla^2 u = \delta(r), \qquad r = \sqrt{x^2 + y^2}$$

to get

$$u(r) = c_0 \log(r) + C_1$$

It is no longer possible to use $u(\infty) = 0$, but we can require $u(1) = 0$ to elimnate C_1. Then, applying Gauss's Divergence Theorem, we get

$$u(r) = \frac{1}{2\pi}\log(r)$$

or more generally

$$G_0(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi}\log(|\mathbf{x} - \mathbf{y}|)$$

Observations/Questions

- The fundamental solution in 2d diverges, but the solution in 3d does not.
- Unlike the 1d Green's function, the two and three dimensional Green's functions are singular at $\mathbf{x} = \mathbf{y}$.
- Are there higher dimensional analogs of the three conditions that we required for linear operators in one dimension?
- What is the 1d analog to the infinite domain solution in 1d?

Solving Poisson's problem using a Green's function

Using the fundamental solutions found above, we can now solve a problem in the infinite domain, given by

$$\nabla^2 u = f, \qquad \mathbf{x} \in \mathcal{R}^2 \quad \text{or} \quad \mathbf{x} \in \mathcal{R}^3$$

The solution can be written in integral form as

$$u(\mathbf{x}) = \int_{\mathcal{R}^d} f(\mathbf{y})\; G_0(\mathbf{x}, \mathbf{y})\; d\mathbf{y}$$

Boundary conditions

The solutions we found above are all "free space Green's functions", meaning that they were defined in infinite domains. But most practical problems of interest are defined in finite domains with prescribed conditions on the boundary of the domain.

In fact, we already solved a problem on the finite domain when we solve the 1d problem

$$u''(x) = f$$

on $[0, 1]$, subject to $u(0) = a$ and $u(b) = 1$. The Green's function solution

$$G(x, y) = \begin{cases} (y-1)x & 0 \le x \le y \\ y(x-1) & y \le x \le 1 \end{cases}$$

satisfied homogeneous boundary conditions $G(0, y) = G(1, y) = 0$. In order to satisfy the boundary conditions on $u(x)$, our final solution consisted of a "homogeneous" part plus some additional boundary terms

$$u(x) = a(1-x) + bx + \int_0^1 G(x, y) f(y)\, dy$$

where the additional boundary terms are $a(1-x) + bx$.

Connection to Laplace's equation:

- The function $u(x) = a(1-x) + bx$ is a solution to the 1d Laplace's equation

  $$u''(x) = 0$$

  subject to $u(0) = a$, $u(1) = b$. As such, this solution is a harmonic function.
- The solution to the boundary value problem $u''(x) = f$, subject to $u(0) = a$, $u(1) = b$ can be written as the sum of the solution to Laplace's equation with the given boundary data plus the solution to the Poisson problem with homogeneous boundary data.

The above observations motivate us to seek a general approach for solving the potential problem $\nabla^2 u = f$ on a finite domain $\Omega \subset \mathcal{R}^d$ subject to boundary conditions on $\partial\Omega$. To do this, we need two ingredients

1. A solution to Laplace's equation satifying inhomogeneous boundary conditions on $\partial\Omega$.
2. A Green's function solution satisfying homogeneous boundary conditions on $\partial\Omega$,

We can consider general boundary conditions of Dirichlet type, Neumann type, or mixed type.

Laplace's equation

We consider the problem

$$\nabla^2 u = 0, \qquad \mathbf{x} \in \Omega$$

subject to $u(\mathbf{x}) = g(\mathbf{x})$ on $\partial\Omega$.

Solution to Laplace's equation in 1d

We solved this in 1d by using the Fundamental Theorem of Calculus, and imposing boundary conditions to obtain two unknown constants of integration. Another approach, and one that generalizes more readily to higher dimensions is to use Green's Second Identity. Here, we state the Identity for the interval $[0, 1]$.

Given two twice-differential functions u and v on on $[0, 1]$, Green's Second Identity states

$$\int_0^1 (u(y)v''(y) - v(y)u''(y))dy = u(y)v'(y)\big|_{y=1} - v(y)u'(y)\big|_{y=0}$$

Substituting in the Green's function $v = G(x, y)$ satisfying homogeneous boundary conditions on $[0, 1]$, we get

$$\int_0^1 (u(y)G''(x, y) - G(y)u''(y))dy = u(y)\frac{\partial G(x, y)}{\partial y}\bigg|_{y=0}^{y=1} - G(x, y)u'(y)\big|_{y=0}^{y=1}$$

which, because $u''(x) = 0$, and $G(x, y)$ is a fundamental solution satisfying homogeneous boundary conditions, reduces to

$$\int_0^1 u(y)\delta(x - y)dy = u(y)\frac{\partial G(x, y)}{\partial y}\bigg|_{y=0}^{y=1} = u(1)\frac{\partial G(x, 1)}{\partial y} - u(0)\frac{\partial G(x, 0)}{\partial y} = bx - a(x - 1) = a(1 - x) + b$$

General solution to the Dirichlet problem

The above idea extends naturally to higher dimensions. We assume that we have a Green's function $G(\mathbf{x}, \mathbf{y})$ that satisfies homogeneous boundary conditions on $\partial\Omega$.

Green's Second Identity can be stated as

$$\int_\Omega (u\nabla^2 v - v\nabla^2 u)dV = \int_{\partial\Omega} \left(u\frac{\partial v}{\partial n} - v\frac{\partial u}{\partial n}\right)\, dA$$

Substituting $v = G(\mathbf{x}, \mathbf{y})$ the above reduces to

$$u(\mathbf{x}) = \int_{\partial\Omega} g\frac{\partial G}{\partial n}\, dA$$

where $u(\mathbf{x}) = g(\mathbf{x})$ on $\partial\Omega$.

General solution to the Neumann problem

Assuming we have a Green's function that satisfies homogeneous Neumann data on $\partial\Omega$, a solution to the Neumann problem is given by

$$u(\mathbf{x}) = \int_{\partial\Omega} g(\mathbf{x})G(\mathbf{x}, \mathbf{y})\, dA$$

where $u_n(\mathbf{x}) = g(\mathbf{x})$ on $\partial\Omega$.

Next : Finding Green's functions for finite domains

- Method of images for simple geometry
- Integral equations solutions for more general geometries.

Some clarifying terminology

- **Fundamental solution** is a solution in the infinite domain, with singular source.

  Example

  $$u''(x) = \delta(\mathbf{x} - \mathbf{y}), \qquad -\infty < x < \infty$$

  We can denote this solution $F(x, y)$. This is also sometimes referred to as the "free space Green's function".

- **Green's function** is a solution to a problem with a singluar source term that satisfies homogeneous boundary conditions on the boundary of a domain.

  Example

  $$u''(x) = \delta(\mathbf{x} - \mathbf{y}), \qquad 0 \le x \le 1$$

  subject to $u(\mathbf{x}) = g(x)$ on . This solution is denoted $G(x, y)$, where $G(0, y) = G(1, y) = 0$.

So far, we have discussed

- Fundamental solutions for the elliptic problem in 2d and 3d problems

$$\begin{aligned}
\text{in 1d} \qquad & F(\mathbf{x}, \mathbf{y}) = R(x - y), & \mathbf{x} \in \mathcal{R} \\
\text{in 2d} \qquad & F(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi}\log(|\mathbf{x} - \mathbf{y}|), & \mathbf{x} \in \mathcal{R}^2 \\
\text{in 3d} \qquad & F(\mathbf{x}, \mathbf{y}) = -\frac{1}{4\pi|\mathbf{x} - \mathbf{y}|}, & \mathbf{x} \in \mathcal{R}^3
\end{aligned}$$

- Green's function solutions

  So far, we have only discussed the 1d Green's function on the domain $[0, 1]$. The goal now is to discuss more general Green's function solutions for finite domains in higher dimensions.

Fundamental solution in 1d revisited

We obtained the Green's function for the 1d elliptic pro

$$u''(x) = \delta(\mathbf{x} - \mathbf{y}), \qquad 0 \le x \le 1$$

subject to $u(0) = u(1) = 0$ by integrating twice and imposing the homogeneous boundary conditions.

$$u''(x) = \delta(x - y), \qquad u(0) = u(1) = 0$$

Integrating once, and using our heuristic $H'(x) = \delta(x)$, we get

$$u''(x) = H(x - y) + C_0$$

where c_0 is a constant of integration. Integrating a second time, we get

$$u(x) = R(x - y) + C_0 x + C_1$$

where $R(x)$ is the "ramp" function, defined as the anti-derivative of $H(x)$ and given by

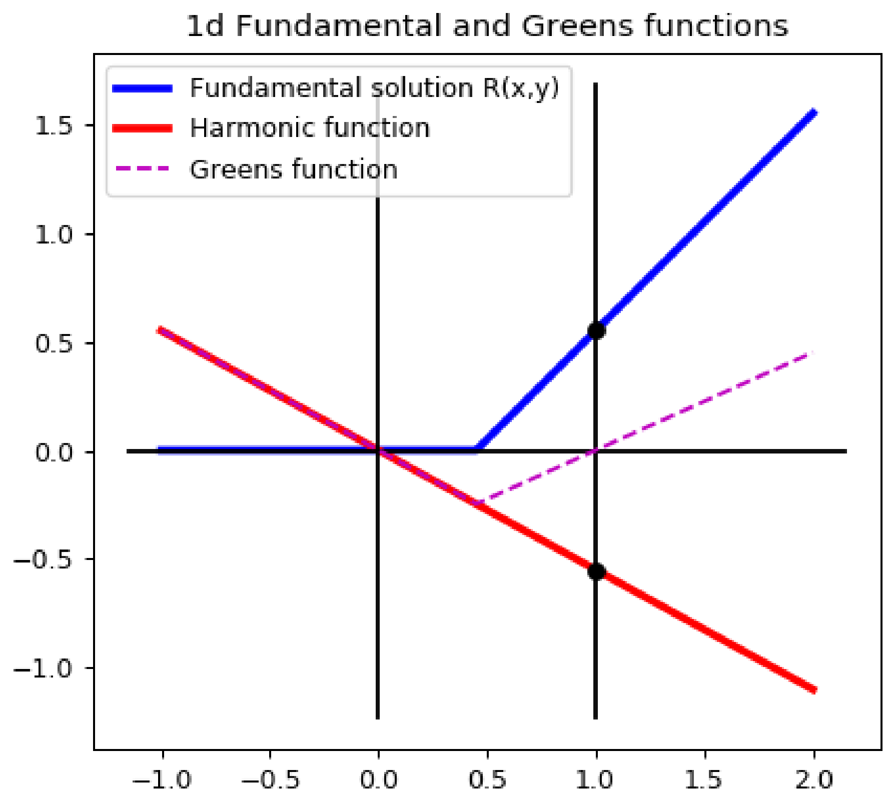
$$R(x) = \begin{cases} 0 & x < 0 \\ x & x \ge 0 \end{cases}$$

We can write the solution to the Green's function as

$$u(x) = R(x - y) + (y - 1)x$$

This can be rewritten as the piecewise linear function $G(x, y)$ which we have defined earlier. But now, we want to view this as a *fundamental solution* plus a *harmonic function* designed to satisfy the boundary conditions. A fundamental solution for this problem is the *ramp* function $F(x, y) = R(x - y)$, and the harmonic function is $u_h(x) = (y - 1)x$.

In [2]:
```
figure(1)
clf()

ax = -1
bx = 2

x = linspace(ax,bx,500)

def R(x):
    return where(x > 0, x,0)

def h(x,y):
    return (y-1)*x

def G(x,y):
    # return where(x < y, (y-1)*x,0) +  where(y <= x, y*(x-1),0)
    return R(x-y) + (y-1)*x


y = 0.45

plot(x,R(x-y),'b',label='Fundamental solution R(x,y)',linewidth=3)

plot(x,h(x,y),'r',label='Harmonic function',linewidth=3)

xmin,xmax = xlim()
ymin,ymax = ylim()


plot([xmin,xmax],[0,0],'k')
plot([0,0],[ymin,ymax],'k')
plot([1,1],[ymin,ymax],'k')

plot(x,G(x,y),'m--',label='Green''s function')



plot(1,1-y,'k.',markersize=12)
plot(1,y-1,'k.',markersize=12)

gca().set_aspect('equal')

title('1d Fundamental and Green''s functions')

legend()
```



Out[2]: <matplotlib.legend.Legend at 0x116615790>

Higher dimensions

In higher dimensions, we can apply a similiar idea and try to find harmonic functions we which can add to fundamental solutions so that the solution on the boundary of our domain of interest is canceled on the boundary.

Dirichlet problem in the upper half plane (2d)

Suppose we want to find the Green's function for the upper half plane $y \geq 0$. The Green's function in this case is given by

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi}\log(|\mathbf{x} - \mathbf{y}|) - \frac{1}{2\pi}\log(|\mathbf{x} - \bar{\mathbf{y}}|), \qquad \mathbf{x}, \mathbf{y} \in \mathcal{R}^2$$

where $\bar{\mathbf{y}}$ is the reflection of the point $\mathbf{y}$ about the x-axis.

The first term is the fundamental solution (or free space Green's function) and the second term is harmonic in the upper half plane, since the singularity is located in the lower half plane.

Dirichlet problem in the upper half space (3d)

We can extend the same idea to the upper half space $z \geq 0$ in three dimensions to get

$$G(\mathbf{x}, \mathbf{y}) = -\frac{1}{4\pi}\left(\frac{1}{|\mathbf{x} - \mathbf{y}|} - \frac{1}{|\mathbf{x} - \bar{\mathbf{y}}|}\right) \qquad \mathbf{x}, \mathbf{y}, \bar{\mathbf{y}} \in \mathcal{R}^2$$

Neumann problem in the upper half plane (2d)

How can we use the same idea for the Neumann problem in 2d?

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi}\log(|\mathbf{x} - \mathbf{y}|) + \frac{1}{2\pi}\log(|\mathbf{x} - \bar{\mathbf{y}}|), \qquad \mathbf{x}, \mathbf{y} \in \mathcal{R}^2$$

Neumann problem in the upper half space (3d)

How can we use the same idea for the Neumann problem in 3d?

$$G(\mathbf{x}, \mathbf{y}) = -\frac{1}{4\pi}\left(\frac{1}{|\mathbf{x} - \mathbf{y}|} + \frac{1}{|\mathbf{x} - \bar{\mathbf{y}}|}\right) \qquad \mathbf{x}, \mathbf{y}, \bar{\mathbf{y}} \in \mathcal{R}^2$$

Homework problem : What is the Green's function for the interior/exterior of the unit circle? The unit sphere?

Fundamental solution (freespace.png)

Harmonic function

Click this link : Harmonic function (harmonic.png)

Greens function in the upper half plane (G=0 on boundary)

Click this link : Greens function (greens_dirichlet.png)

Greens function in the upper half plane (dG/dn = 0 on boundary)

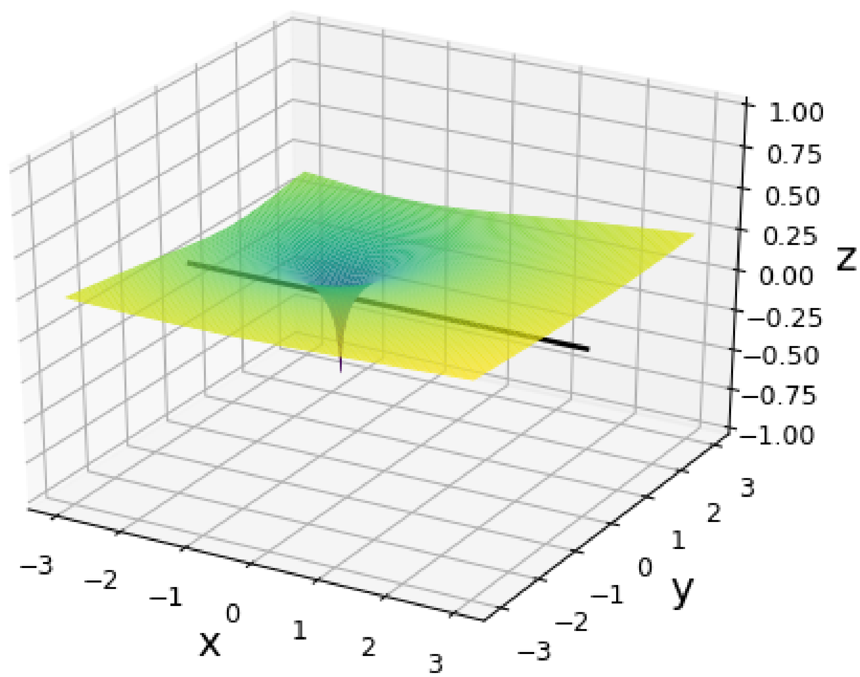
Click this link : Greens function (greens_neumann.png)

Greens function on the unit circle.

Click this link : Greens function (greens_circle.png)

The above plots were done using Matlab. Below is a Python code which also illustrates the 2d fundamental solution.

In [3]:
```
from mpl_toolkits import mplot3d
from matplotlib.colors import LightSource

figure(2)
clf()

ax = -3
bx = 3
ay = -3
by = 3
az = -1
bz = 1


ax3d = plt.axes(projection='3d')

x = linspace(ax,bx,500);
xm,ym = meshgrid(x,x)

xbar, ybar = (-1.3456, 1.1391)

PmQ = sqrt((xm-xbar)**2 + (ym-ybar)**2)

F = 1/(2*pi)*log(PmQ)     # Fundamental solution

ax3d.plot_surface(xm,ym,F,cmap='viridis', edgecolor='none',rstride=5, cs
tride=5,shade=True)

ax3d.set_zlim([az,bz])

ax3d.plot3D([ax,bx],[0,0],[0,0],'k',linewidth=2)

# ls = LightSource()

xlabel('x',fontsize=16)
ylabel('y',fontsize=16)
ax3d.set_zlabel('z',fontsize=16)


show()
```


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
z
−3
−2
−1
0
1
2
3
x
3
2
1
0
−1
−2
−3
y