FDPoisson: Usage Instructions

Vishnu Raghuraman

April 2021

1 One-Dimensional Case

The code deals with charge densities of the form

$$\rho(x) = \sum_{i}^{n} c_i x^i \tag{1}$$

where n is the degree of the polynomial. This has been chosen this way because any (smooth) function can be expressed in terms of a polynomial with suitable chosen coefficients.

1.1 Input File

The typical input file for the 1D case will look like this

```
Dimension 1  # Choose from 1,2
NX 30
Xmin -1
Xmax 1
BCmin 1.0
BCmax 1.0
FxType 1
Degree 0
PCoeff 1
```

Let's look at the input parameters.

1. Dimension - determines whether to do a 1D or 2D calculation.
2. NX - number of grid points
3. Xmin - lower bound for x
4. Xmax - upper bound for x
5. BCmin - value of potential at lower bound
6. BCmax - value of potential at upper bound
7. FxType - choose between 0 and 1. 0 means no charge density - FDPoisson will solve Laplace equation. 1 means polynomial charge density
8. Degree - if FxType is 1, then till parameter will be needed to specify the degree of the polynomial
9. PCoeff - the coefficients of the polynomial, starting from lowest to highest, separated by —. For example, if we want to input $x^2 + 2x + 1$, then the input parameters will be

```
Degree 2
PCoeff 1|2|1
```

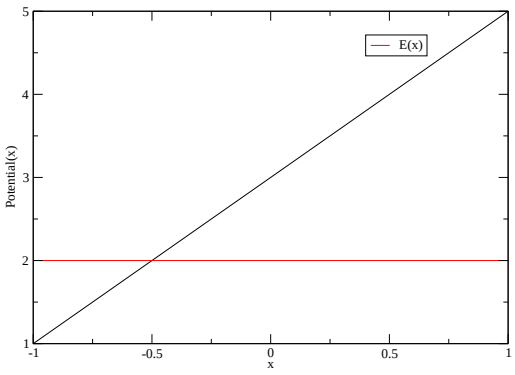

Figure 1: Solution of Laplace Equation from FDPoisson

1.2 Running FDPoisson

Suppose your input file is called ifile. We can run the calculation by calling

```
path_to_executable/fdpoisson ifile
```

If you are in the sample folder of the repo, then the executable can be called by

```
../bin/fdpoisson ifile
```

You should see the following print to screen

```
FDPoisson :            1  Dimensional Calculation
Potential calculated succesfully and stored in file: potential
```

The file called potential will contain a list of Potential and Electric Field values at all grid points. This can then be graphed using the user's desired graphing software. Let's look at some examples

1.3 Example 1: Laplace Equation

Consider the following input file

```
Dimension 1  # Choose from 1,2
NX 5
Xmin -1
Xmax 1
BCmin 1.0
BCmax 5.0
FxType 0
```

This should give the Laplace solution (straight line) with 1 and 5 as the boundary conditions, resulting in the following potential file

```
                x                  Phi(x)                E(x)
 -1.0000000000000000  1.0000000000000000  1.9999999999999996
 -0.5000000000000000  1.9999999999999998  1.9999999999999996
  0.0000000000000000  2.9999999999999996  1.9999999999999998
  0.5000000000000000  3.9999999999999996  2.0000000000000004
  1.0000000000000000  5.0000000000000000  2.0000000000000018
```

and on plotting, we can see the straight line (along with the constant field) in Figure 1

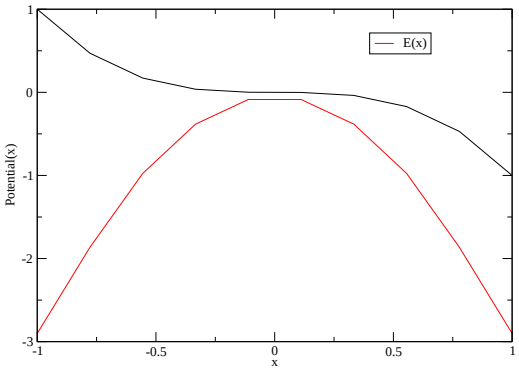

Figure 2: Solution for $\rho = 6x, \phi(-1) = 1, \phi(1) = 1$

1.4 Example 2: Linear charge

Consider same domain as previous case, but now

$$\rho(x) = 6x, \phi(-1) = 1, \phi(1) = -1 \tag{2}$$

We can analytically solve this to get $\phi(x) = x^3$. We can solve this using FDPoisson by using the following input file

```
Dimension 1  # Choose from 1,2
NX 10
Xmin -1
Xmax 1
BCmin 1.0
BCmax -1.0
FxType 1
Degree 1
PCoeff 0|6
```

resuting in Figure 2.

2 Two Dimensional Case

In 2D, we can deal with two types of (polynomial) charge densities. The first is

$$\rho(x, y) = \sum_{i}^{n} c_i x^i y^{n-i} \tag{3}$$

or we can have a product of polynomials

$$\rho(x, y) = \left(\sum_{i}^{n} c_i x^i \right) \left(\sum_{j}^{m} d_j y^j \right) \tag{4}$$

2.1 Input File

The 2D case has more parameters. A typical 2D input will look like this

```
Dimension 2  # Choose from 1,2,3
NX 50
NY 50
```

```
Xmin -1
Xmax 1
Ymin -1
Ymax 1
BCX1Degree 0
BCX1Coeff 1
BCX2Degree 0
BCX2Coeff 5
BCY1Degree 1
BCY1Coeff 3|2
BCY2Degree 1
BCY2Coeff 3|2
FxType 2
XDegree 2
XPCoeff -1|0|1
YDegree 2
YPCoeff -1|0|1
```

1. NX,NY - number of grid points along x and y respectively
2. Xmin, Xmax, Ymin, Ymax - specifying the domain
3. BCX1Degree, BCX1Coeff - these are the degree and coefficient parameters associated with $f_1(x)$ the boundary function at $y = y_{min}$. The format is same as before.
4. BCX2Degree, BCX2Coeff - these are the degree and coefficient parameters associated with $f_2(x)$ the boundary function at $y = y_{max}$. The format is same as before.
5. BCY1Degree, BCY1Coeff - these are the degree and coefficient parameters associated with $g_1(y)$ the boundary function at $x = x_{min}$. The format is same as before.
6. BCY2Degree, BCY2Coeff - these are the degree and coefficient parameters associated with $g_2(y)$ the boundary function at $x = x_{max}$. The format is same as before.
7. FxType - type of charge density function. It can be 0,1,2. 0 means Laplace equation (no charge density). 1 means polynomial of the form $x^i y^{n-i}$ and 2 refers to product of polynomials.
   - If FxType is 1, we have parameters XYDegree and XYPCoeff for the degree and coefficients
   - If FxType is 2, we have parameters XDegree and XPCoeff for the x-polynomial and parameters YDegree and YPCoeff for the y-polynomial. The code will then take a product of the two polynomials.

2.2 Running FDPoisson

The procedure for running FDPoisson for 2D calculations is the same, except that the stdout will look like this

```
FDPoisson :            2  Dimensional Calculation
Potential calculated succesfully and stored in file: potential
```

The output file (potential) will contain the value of potential, E_x and E_y at each (x, y) point.

2.3 Example: Laplace Equation

Consider $\rho = 0$ and $-1 < x < 1$, $-1 < y < 1$. Let the BCs be

$$f_1(x) = 1, f_2(x) = 5, g_1(y) = g_2(y) = 2y + 3 \tag{5}$$

This is the input file to get the result

```
Dimension 2  # Choose from 1,2,3
NX 50
NY 50
Xmin -1
Xmax 1
Ymin -1
Ymax 1
BCX1Degree 0
BCX1Coeff 1
BCX2Degree 0
BCX2Coeff 5
BCY1Degree 1
BCY1Coeff 3|2
BCY2Degree 1
BCY2Coeff 3|2
FxType 0
```

The output potential, Ex and Ey can be plotted both as a 2D color plot and a 3D contour plot (shown in next page)

3 Plotting

In addition to the main code, there are also some python scripts in the util folder that will produce plots of the potential and the field. The user can directly use these scripts, modify them to change the look of the plots, or ignore these and use some other plotting software. To generate plots for the 1D calculation, use

```
python 1d_plotter.py path_to_potential_file
```

if the potential folder is located in ../sample/potential, then this would give us

```
python 1d_plotter.py ../sample/potential
```

The program will generate two pdf files (potential_1dplot.pdf, Ex_1dplot.pdf). To get 2D color plots, we can use

```
python 2d_plotter_contour.py ../sample/potential nx ny
```

This will create three pdf files (potential_2dplot.pdf, Ex_2dplot.pdf, Ey_2dplot.pdf). Similarly, if we want to generate surface plots, we can use

```
python 2d_plotter_surface.py ../sample/potential nx ny
```

This will generate three pdf files (potential_surface.pdf, Ex_surface.pdf, Ey_surface.pdf)

Note that these scripts require numpy and matplotlib.

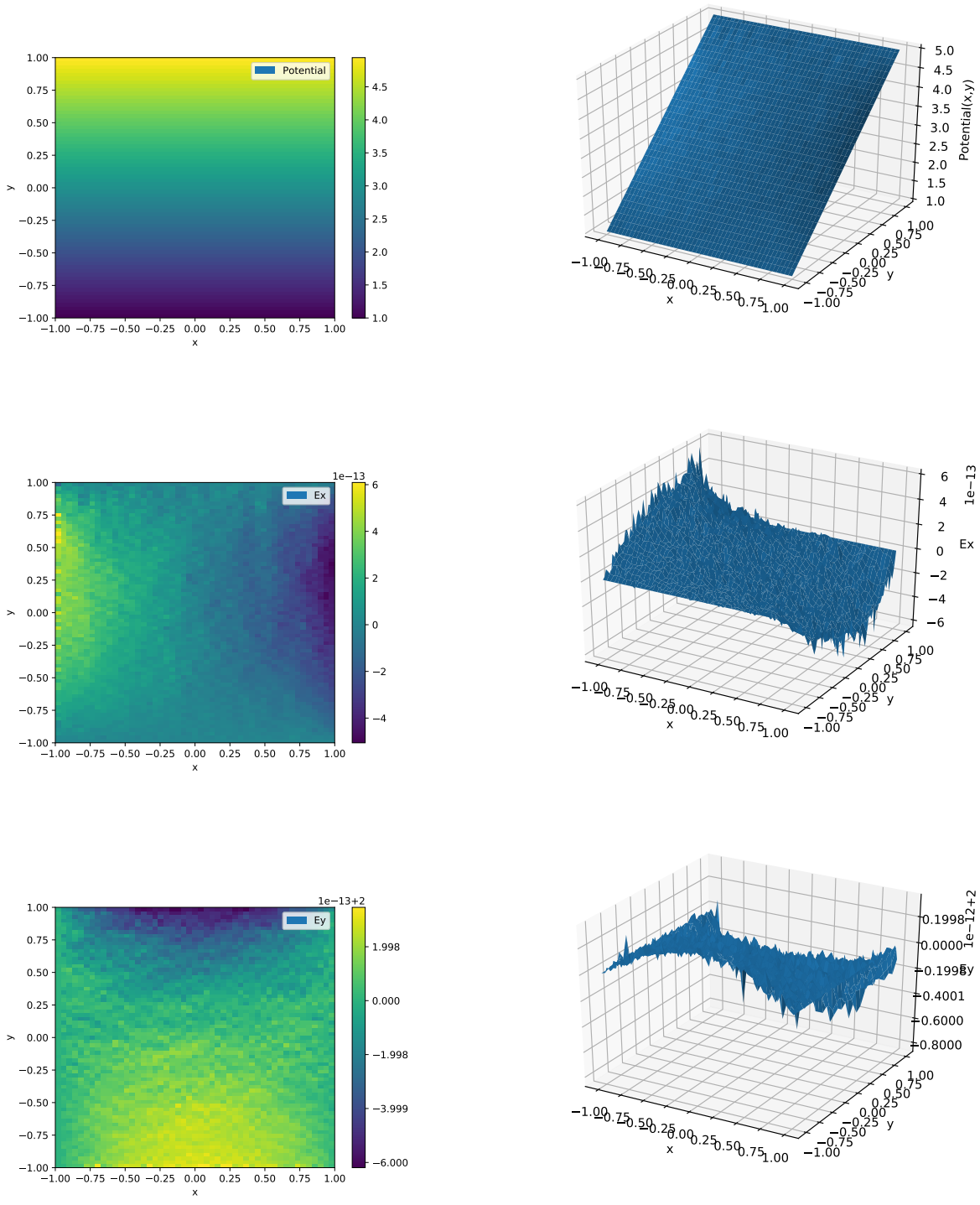

Figure 3: Caption