Boundary Value Problems
Mathematical Programming with Python

MATH 2604: Advanced Scientific Computing 4
Spring 2025
Monday/Wednesday/Friday, 1:00-1:50pm
Room A202 Langley Hall

https://people.sc.fsu.edu/∼jburkardt/classes/python_2025/bvp/bvp.pdf

Reaching from one end to the other.

A boundary value problem (BVP) is a special kind of problem in which we know the value of a function $u(x)$ at both ends of an interval $a \leq x \leq b$, and fill in the missing values of u by solving a second order differential equation.

For an initial value problem, we have all the data at a single starting point. Here, instead, we have the value $u(a)$ at one end, and $u(b)$ at the other. We need a new strategy that will start with this information and enable us to fill in the gaps.

One way to do this is to:

- select a sequence of n equally spaced points in the interval using `np.linspace()`;
- at each point, approximate the differential equation, using the finite difference method;
- solve all the equations at once using np.linalg.solve().

1 A discretized second derivative

We will need to be able to estimate the second derivative of the unknown solution $u(x)$. We are already familiar with an estimate for the first derivative, or slope, at the point x, using a nearby value at $x + h$:

$$\frac{du}{dx} \approx \frac{u(x+h) - u(x)}{h}$$

We can estimate the second derivative by taking differences of this first derivative estimate, using a nearby value at $x - h$, to arrive at

$$\frac{d^2u}{dx^2} \approx \frac{u(x-h) - 2u(x) + u(x+h)}{h^2}$$

This is the estimate we will use in order to construct a system of algebraic equations that are a model of the original differential equation.

2 A discretized boundary value problem

We start by choosing n, the number of equally spaced points x_i over our interval $[a, b]$. This results in a uniform spacing h between consecutive points

$$h = |x_{i+1} - x_i| = \frac{b-a}{n-1}$$

This spacing will control the accuracy of our results.

Corresponding to the point x_i, we need to come up with u_i, an approximation to the exact solution. The boundary conditions specify the first and last values of u. For the intermediate values, we have to use the differential equation. Assume that equation has the simple form, where the right hand side does not include any dependence on u:

$$\frac{d^2u}{dx^2} = f(x)$$

Then, at any x_i, for $0 < i < n-1$ at x_i, our discretized equation would be:

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = f(x_i)$$

We now have two equations from the boundary conditions, and $n-2$ equations from the differential equation. We want to use this information to solve for the n values of u. You should see that we have created a system of n linear equations in n unknowns, of the form $A * u = b$, so that we only have to call a linear equation solver for our answer.

3 Example: $u" = 3$

Consider the trivial BVP $u" = 3$, over the interval $0 \leq x \leq 2$, with boundary conditions $u(0) = 2$ and $u(2) = 4.5$. Let us use $n = 5$ points. Then $h = \frac{2.0-0.0}{5-1} = 0.5$. If we multiply the intermediate equations by $h^2 = 0.25$, then our linear equations become

```
u_0                                       = 2
u_0  - 2 * u_1  +    u_2                  = 3 * 0.25
           u_1  - 2 * u_2  +    u_3       = 3 * 0.25
                      u_2  - 2 * u_3  + u_4 = 3 * 0.25
                                         u_4 = 4.5
```

and we can rewrite this as a matrix-vector system $A * u = f$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad u = \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} \quad f = \begin{pmatrix} 2 \\ 0.75 \\ 0.75 \\ 0.75 \\ 4.5 \end{pmatrix}$$

Calling `np.linalg.solve()` and then plotting our approximation against the exact solution, we may be startled to see that we have gotten exact results:

X	U(X)	Uexact
0.	0.5	0.5
0.5	0.375	0.375
1.	1.	1.
1.5	2.375	2.375
2.	4.5	4.5

Since the exact solution is a quadratic polynomial, it turns out that our method is able to match exactly. Of course, we can't expect this kind of precision for most problems!

4 Example: $u" = -3u/(1 + x^2)^2$

Let's consider a problem in which what we think of as the right hand side of the differential equation includes a reference to the unknown function u.

$$\begin{aligned} \frac{d^2u}{dx^2} &= -\frac{3.0\,u}{(1.0 + x^2)^2} \\ u(0.0) &= 0.0 \\ u(1.0) &= \frac{1.0}{\sqrt{2.0}} \end{aligned}$$

Obviously, we can't put this information into f, the right hand side of the linear system we will be setting up. Luckily, u appears in a linear form (it's not square, or inside a function), so we will can move that expression to the left hand side of our equation:

$$\frac{d^2u}{dx^2} + \frac{3}{(1 + x^2)^2}\,u = 0$$

and modify our matrix A as well, so that our discretized equation becomes

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + \frac{3}{(1 + xi^2)^2} ui = 0$$

This means that, for $0 < i < n - 1$, our diagonal matrix entry is

```
A[i,i] = -2.0 + 3 / ( 1 + x[i]**2 )**2 * h**2
```

Where did the factor of h^2 come from? Recall that we simplified our linear system by multiplying through by h^2. If we do that here, then that factor has to multiply the new term as well.

The right hand side of our differential equation is 0, so entries $0 < i < n - 1$ if the vector f will be zero. It remains to set the boundary conditions:

```
f[0] = 0
f[n-1] = 1 / np.sqrt ( 2.0 )
```

And now we can solve our linear system. If we only use $n = 11$ points, we get a reasonable approximation to the exact soltuion, which is $u(x) = \frac{1}{\sqrt{1+x^2}}$

X	U(X)	Uexact
0.	0.	0.
0.1	0.0997531	0.09950372]
0.2	0.19657257	0.19611614]
0.3	0.28793977	0.28734789]
0.4	0.37203638	0.37139068]
0.5	0.44783848	0.4472136]
0.6	0.51504208	0.51449576]
0.7	0.57389183	0.57346234]
0.8	0.62498665	0.62469505]
0.9	0.66911031	0.66896473]
1.	0.70710678	0.70710678]]

5 Exercise: The humps function

Let's consider a more complicated right hand side, involving the humps function:

$$\text{humps}(x) = \frac{1.0}{((x-0.3)^2+0.01)} + \frac{1.0}{((x-0.9)^2+0.04)} - 6.0$$

We can imagine that a boundary value problem with this as its solution will be more challenging for our methods. If we work out the second derivative:

$$\text{humps"}(x) = -2.0 * \frac{(x-0.3)}{((x-0.3)^2+0.01)^2} - 2.0 * \frac{(x-0.9)}{((x-0.9)^2+0.04)^2}$$

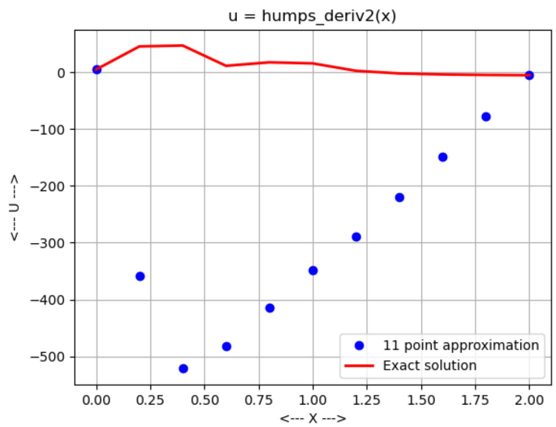
we can set up the boundary value problem:

$$\begin{aligned} \frac{d^2u}{dx^2} &= \text{humps"}(x) \\ u(0.0) &= \text{humps}(0.0) \\ u(2.0) &= \text{humps}(2.0) \end{aligned}$$

over the interval $0 \le x \le 2$. Since we know the exact solution, we can monitor the accuracy of our estimate. The calculation for $n = 11$ doesn't look good, so we try $n = 21$ next, getting the following table:

X	U(X)	Uexact
0.00000000e+00	5.17647059e+00	5.17647059e+00
2.00000000e-01	-3.58695046e+02	4.58867925e+01
4.00000000e-01	-5.21798143e+02	4.74482759e+01
6.00000000e-01	-4.82572321e+02	1.16923077e+01
8.00000000e-01	-4.14171442e+02	1.78461538e+01
1.00000000e+00	-3.48802334e+02	1.60000000e+01
1.20000000e+00	-2.88898825e+02	2.91181989e+00
1.40000000e+00	-2.20269134e+02	-1.73205201e+00
1.60000000e+00	-1.49151038e+02	-3.52497225e+00
1.80000000e+00	-7.71821298e+01	-4.38105154e+00
2.00000000e+00	-4.85517241e+00	-4.85517241e+00

If we plot our result versus the true solution, we can see that we are computing what seems like nonsense.



Compare 11 point approximation (blue dots) to exact (red curve).

However, this is a tough problem, and we are using an approximation that we expect will improve as the discretization value h decreases. If we continue with, say $n = 41$ and $n = 81$ we may hope to get results that we can believe.

But you need to ask yourself, if you didn't know the exact solution in advance, how could you judge that your approximation is reasonably close to the right values?

6 Exercise: Right hand side includes u' and u

Consider the following boundary value problem over the interval $\frac{0.5}{\pi} \leq x \leq \frac{3}{\pi}$:

$$u'' = -\frac{2}{x}u' - \frac{1}{x^4}u$$
$$u(0.5/\pi) = 0$$
$$u(3/\pi) = \frac{\sqrt{3}}{2}$$

The exact solution is

$$u(x) = \sin(\frac{1}{x})$$

The right hand side involves u' and u. Since both of these terms are linear in u, we can bring them to the left hand side. We need to replace u' by a difference quotient, and in this case, we will prefer to use a symmetric formula:

$$u'(x_i) \approx \frac{u_{i+1} - u_{i-1}}{2h}$$

You will need to be very careful in order to correctly define your matrix A. Once again, the right hand side vector f will be zero except for the first and last terms, which will be set by the boundary values.

This is a hard problem to approximate well, but if you experiment with the value of n, you should get reasonable results. One extra factor that will hurt you is that the approximation to u' has a lower degree of accuracy, that is, $O(h)$ rather than $O(h^2)$, so the improvement as you decrease h will be slower.

This is a hard problem to compute correctly! You have to think carefully about how you approximate each term, and how it is then stored in the linear system. As a clue, here is a plot of my results using $n = 41$: