INTRODUCTION TO BOUNDARY VALUE PROBLEMS

David F. Gleich · CS 314 · Purdue University

November 21, 2016

1 A BVP

In a two-point boundary value problem, we look for a function that satisfies a certain property, like

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

where x ranges between $[0, 1]$ and $u(0) = \alpha$ and $u(1) = \beta$.

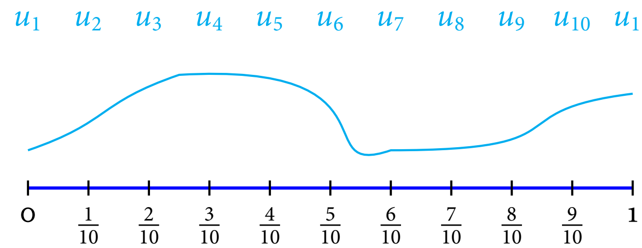
This is different from a standard ODE because we have TWO constraints, one at $u(0)$ and one at $u(1)$.

In this intro, we'll always assume that $u(x)$ is a function that is defined where x is between $[0, 1]$. In general, these can get more complicated, but that's for the future!

2 SOLUTION STRATEGY

The way we look for solutions of a BVP is to represent the function u via a regular grid of points. Then we can approximate the derivative terms via finite-difference approximations. This will turn into a *linear system* that we have to solve (which you now know how to solve!)

3 GRID REPRESENTATIONS



$u_i = u((i-1)h)$ this is the value of the function at $x_i = (i-1)h$

this is how matlab represents a function!

```
# We use a grid representation of the function u(x) on [0,1]
# so we divide [0,1] up into N+1 increments of 1/N each.

N = 10
xgrid = collect(0:1/N:1)

# This means that u is a vector with n = N+1 elements,
# but two of them are fixed at alpha and beta.

n = length(xgrid)

example_u = zeros(n);
example_u[1] = 1        # suppose alpha = 1
example_u[end] = 1/2    # suppose beta = 1/2

example_u[2:end-1] = 1.06-1*(xgrid[2:end-1]-0.25).^2;

plot(xgrid, example_u)
```

4 AN EXAMPLE

Let's try and solve

$$u''(x) = -2$$

with $u(0) = 1, u(1) = 1/2$. Let's represent u at exactly the points above, so $u_1 = u(0), u_2 = u(0.1), \ldots, u_{10} = u(0.9), u_{11} = u(1)$.

This means that we know that $u_1 = 1$ and $u_{N+1} = 1/2$.

Now, we also know that $u''(x) = -2$. But, we can approximate:

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

if $h = 0.1$ and $x = 0.1$, then we have:

$$u''(0.1) \approx \frac{u_1 - 2u_2 + u_3}{h^2}.$$

In order to solve the BVP, we need:

$$u''(0.1) = -2 \Leftrightarrow \frac{u_1 - 2u_2 + u_3}{0.1^2} = -2.$$

So we have one equation that defines a relationship between these three equations.

Let's repeat!

$$u''(0.2) = -2 \Leftrightarrow \frac{u_2 - 2u_3 + u_4}{0.1^2} = -2.$$

$$u''(0.3) = -2 \Leftrightarrow \frac{u_3 - 2u_4 + u_5}{0.1^2} = -2.$$

$$u''(0.4) = -2 \Leftrightarrow \frac{u_4 - 2u_5 + u_6}{0.1^2} = -2.$$

$$u''(0.5) = -2 \Leftrightarrow \frac{u_5 - 2u_6 + u_7}{0.1^2} = -2.$$

$$u''(0.6) = -2 \Leftrightarrow \frac{u_6 - 2u_7 + u_8}{0.1^2} = -2.$$

$$u''(0.7) = -2 \Leftrightarrow \frac{u_7 - 2u_8 + u_9}{0.1^2} = -2.$$

$$u''(0.8) = -2 \Leftrightarrow \frac{u_8 - 2u_9 + u_{10}}{0.1^2} = -2.$$

$$u''(0.9) = -2 \Leftrightarrow \frac{u_9 - 2u_{10} + u_{11}}{0.1^2} = -2.$$

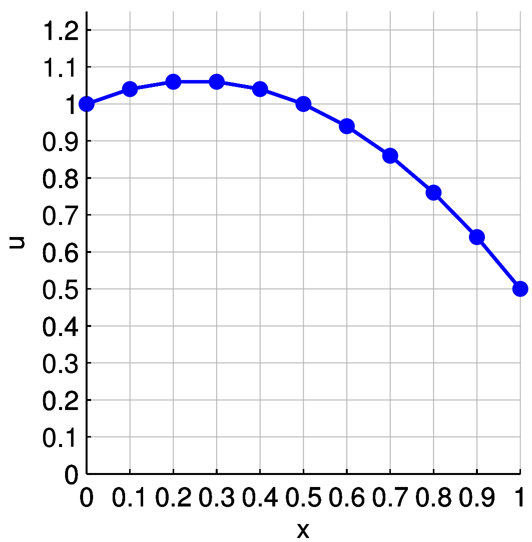
We also know that $u_1 = 1$ and $u_{11} = 1/2$. Thus we have 11 equations and 11 unknowns!
It's a linear system

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 100 & -200 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & -200 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & -200 & 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & -200 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & -200 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & -200 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 & -200 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & -200 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & -200 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ -2 \\ -2 \\ -2 \\ -2 \\ -2 \\ -2 \\ -2 \\ -2 \\ 1/2 \end{bmatrix}.$$

```julia
N = 10
xgrid = collect(0:1/N:1)

alpha = 1
beta = 0.5
f = x -> -2.0
h = 1/N

A = zeros(N+1,N+1)
b = zeros(N+1,1)
for i=1:N+1
  if i == 1
    A[i,i] = 1
    b[i] = alpha
  elseif i==N+1
    A[i,i] = 1
    b[i] = beta
  else
    A[i,i-1] = 1/h^2
    A[i,i] = (-2)/h^2
    A[i,i+1] = 1/h^2
    b[i] = f(xgrid[i])
  end
end
u = A\b
plot(xgrid, u, marker=:circle)
```



5 A more complicated example

This example models a rod that we are heating. (See the book for the physics.)

$$\frac{\partial}{\partial x}\left[p(x)\frac{\partial u}{\partial x}\right] = f(x)$$

The function $p(x)$ gives the conductivity of the bar. The function $f(x)$ is the *negative* of the heat-input. So if $f(x) = 0$, then there is no heat coming in.

The way to interpret this equation is that we are looking for a function $u(x)$ that has the property that

$$p(x)u''(x) + p'(x)u'(x) = f(x).$$

But we need to know $u(0)$ and $u(1)$ too.

For this example, we'll use:

$$u(0) = -1, u(1) = 1, p(x) = 1 - 4(x - 1/2)^2, f(x) = -2\sin(\pi x).$$

We discretize the derivative as follows: Let $g(x) = \left[p(x)\frac{\partial u}{\partial x}\right](x)$. That is, $g(x)$ is the function we get from taking one derivative with respect to u and then multiplying by $p(x)$. Then our BVP is:

$$g'(x) = f(x).$$

if we take a centered difference formula with $h/2$, we get

$$g'(x) = f(x) \approx \frac{g(x + h/2) - g(x - h/2)}{h} = f(x).$$

Now, we need a formula for $g(x + h/2)$!. This we get from using another centered difference formula for $\partial u/\partial x$:

$$g(x + h/2) \approx p(x + h/2)\frac{u(x + h) - u(x)}{h}.$$

Hence, we have the full discretization

$$g(x) = \left[p(x)\frac{\partial u}{\partial x}\right](x) \Leftrightarrow g(x) = \frac{p(x + h/2)\frac{u(x+h)-u(x)}{h} - p(x - h/2)\frac{u(x)-u(x-h)}{h}}{h}$$

Or, for a grid of u_i values:

$$g(x_i) = \frac{p(x_i + h/2)\frac{u_{i+1}-u_i}{h} - p(x_i - h/2)\frac{u_i-u_{i-1}}{h}}{h}$$

This gives us a linear system that the following Julia code evaluates:

```julia
N = 10
xgrid = collect(0:1/N:1)

alpha = -1
beta = 1
f = x -> -2*sin(pi*x)   # we are heating a rod in the middle
p = x -> 1-4*(x-0.5)^2  # the rod becomes more like an insulator towards the end.
h = 1/N

A = zeros(N+1,N+1)
b = zeros(N+1,1)
for i=1:N+1
  if i == 1
    A[i,i] = 1
    b[i] = alpha
  elseif i==N+1
    A[i,i] = 1
    b[i] = beta
  else
    A[i,i-1] = p(xgrid[i] - h/2)/h^2
    A[i,i] = (-p(xgrid[i] + h/2)-p(xgrid[i] - h/2))/h^2
    A[i,i+1] = p(xgrid[i] + h/2)/h^2
    b[i] = f(xgrid[i])
  end
end
u = A\b
plot(xgrid, u)
```