Numerical Solutions to Differential Equations

Lecture Notes #22 — Nonlinear Equations

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Spring 2015

Outline

1. Nonlinear Boundary Value Problems
   - An Alternative Model For Beam-Bending
   - Model Problem: General 2nd Order Nonlinear BVP
2. Do Well-Behaved Solutions Exist?!?
   - Finite-Time Blowup...
   - Existence and Uniqueness
3. ODE $\rightsquigarrow$ Nonlinear Algebraic System
   - Finite Differencing
   - Nonlinear System, and Uniqueness Condition
   - The Return of Newton's Method...
4. Example
   - Code
   - Numerical Results

So far we have exclusively looked at Linear BVPs.

We will now consider some non-linear problems.

Burden-Faires [p. 672] suggests that

$$\frac{\mathbf{1}}{\sqrt[3]{1+w'(x)}}w''(x) = \frac{S}{EI}w(x) + \frac{qx}{2EI}(x-L), \quad w(0) = w(L) = 0$$

is a more appropriate equation for the deflection of a supported beam subject to uniform loading.

Note that the original forth order beam equation has been integrated twice to give a second order equation.

Nonlinear BVPs, II

- Quite a few of the linear models we use are simplifications of more accurate nonlinear models.
- Usually the linear model is valid in a limited regime (*e.g.* small deflection of the beam), whereas the non-linear models have larger regimes of validity.
- Since closed-form solutions for non-linear equations are hard to find, finding numerical solutions seem like a good idea...

General Second Order Non-linear BVP

- We are going to look at the general second order nonlinear BVP

$$y''(x) = f(x, y(x), y'(x)), \quad x \in [a, b], \quad y(a) = y_a \quad y(b) = y_b$$

- We are going to apply our trusted finite difference methods to this problem.
- In this setting we get a **non-linear system of algebraic equations.**
- In order to solve this system we need an iterative process.

Even a very benign-looking non-linear ODE can produce solutions which “blow up” (reach infinite values). Consider the initial value problem:

$$y'(t) = t^2, \quad y(0) = y_0 > 0$$

which has the solution

$$y(t) = \frac{1}{\frac{1}{y_0} - t}$$

and

$$\lim_{t \to \frac{1}{y_0}} y(t) = \infty$$

We need some **restrictions** in order to guarantee the existence of a unique solution...

Existence and Uniqueness of the Solution

We are studying

$$y''(x) = f(x, y(x), y'(x)), \quad x \in [a, b], \quad y(a) = y_a \quad y(b) = y_b$$

If we assume:

[1] f and the partial derivatives f_y and $f_{y'}$ are continuous on

$$D = \{(x, y, y') : \, a \le x \le b, \, -\infty < y < \infty, \, -\infty < y' < \infty\}$$

[2] $f_y(x, y, y') \ge \delta > 0$ on D.

[3] Constants k and L exist, with the properties

$$k = \max_{(x,y,y') \in D} |f_y(x, y, y')| \quad \text{and} \quad L = \max_{(x,y,y') \in D} |f_{y'}(x, y, y')$$

Then the **existence** of a unique solution **is guaranteed**.

Constructing the Nonlinear Algebraic System

We subdivide the interval $[a, b]$ into $(N-1)$ subintervals:

$$x_n = a + (n-1)h, \quad n = 1, 2, \ldots, N \qquad h = \frac{(b-a)}{(N-1)}$$

We apply second-order centered differences and get

$$\frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} = f\left(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h} - \frac{h^2}{2} y'''(\eta_n)\right) + \frac{h^2}{12} y^{(4)}(\xi_n)$$

where $\eta_n, \xi_n \in (x_{n-1}, x_{n+1})$.

The error terms make the assumption that $y(x) \in C^4[a, b]$.

The finite difference method is the result of dropping the error terms, and adding the boundary conditions

$$y_1 = y_a, \quad y_N = y_b$$

The System

$$
\begin{cases}
y_1 & = & y_a \\
y_3 - 2y_2 + y_1 & = & h^2 f\left(x_2, y_2, \frac{y_3 - y_1}{2h}\right) \\
y_4 - 2y_3 + y_2 & = & h^2 f\left(x_3, y_3, \frac{y_4 - y_2}{2h}\right) \\
 & \vdots & \\
y_N - 2y_{N-1} + y_{N-2} & = & h^2 f\left(x_{N-1}, y_{N-1}, \frac{y_N - y_{N-2}}{2h}\right) \\
y_N & = & y_b
\end{cases}
$$

This system has a **unique solution** provided $\mathbf{h} < \mathbf{2}/\mathbf{L}$.

[Keller, H.B., *Numerical Methods for Two-Point Boundary-Value Problems*, Blaisdell, Waltham, MA 1968].

Solving the System

We vaguely remember talking about using Newton's method for systems in the context of Implicit Linear Multistep Methods for Stiff ODEs (lecture 12)...

Define $\tilde{\mathbf{y}} = \{y_1, y_2, \ldots, y_N\}^T$, and write the vector equation:

$$F(\tilde{\mathbf{y}}) = \begin{bmatrix} y_1 & - & y_a \\ y_3 - 2y_2 + y_1 & - & h^2 f\left(x_2, y_2, \frac{y_3 - y_1}{2h}\right) \\ y_4 - 2y_3 + y_2 & - & h^2 f\left(x_3, y_3, \frac{y_4 - y_2}{2h}\right) \\ & \vdots & \\ y_N - 2y_{N-1} + y_{N-2} & - & h^2 f\left(x_{N-1}, y_{N-1}, \frac{y_N - y_{N-2}}{2h}\right) \\ y_N & - & y_b \end{bmatrix} = \tilde{\mathbf{0}}$$

Newton's method applied to $F(\tilde{\mathbf{y}}) = \tilde{\mathbf{0}}$ is

$$\tilde{\mathbf{y}}^{n+1} = \tilde{\mathbf{y}}^{n} - [J(\tilde{\mathbf{y}})]^{-1} F(\tilde{\mathbf{y}}),$$

where $J(\tilde{\mathbf{y}})$ is the Jacobian of $F(\tilde{\mathbf{y}})$:

$$J_{ij}(\tilde{\mathbf{y}}) = \frac{\partial F_i(\tilde{\mathbf{y}})}{\partial y_j}, \quad i, j = 1, 2, \ldots, N.$$

The first and last row of F are very simple:

$$F_{\{1,N\}}(\tilde{\mathbf{y}}) = \left[\begin{array}{ccc} y_1 & - & y_a \\ y_N & - & y_b \end{array} \right] \Rightarrow J_{1,1} = J_{N,N} = 1.$$

The remaining entries on the first and last (Nth) rows are zero.

Newton's Method: A General Row of the Jacobian

For $n = 2, 3, \ldots, (N-1)$ we have the non-linear equation

$$F_n(\tilde{\mathbf{y}}) = y_{n+1} - 2y_n + y_{n-1} - h^2 f\left(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}\right)$$

Hence

$$\begin{aligned} J_{n,(n-1)}(\tilde{\mathbf{y}}) &= 1 + \frac{h}{2} f_{y_n'}\left(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}\right) \\ J_{n,n}(\tilde{\mathbf{y}}) &= -2 - h^2 f_{y_n}\left(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}\right) \\ J_{n,(n+1)}(\tilde{\mathbf{y}}) &= 1 - \frac{h}{2} f_{y_n'}\left(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}\right) \end{aligned}$$

Since J is tridiagonal, the Newton iteration is not that expensive.

Example

Nonlinear Beam Bending

Segment #1

```
% Nonlinear BVP

% Examples
%
% y'' = -f(x,y,y')
% y(1) = y(2) = 0

a = 1; ya = 0;
b = 2; yb = 0;

TOL  = 10^(-8);
N    = 128;
h    = (b-a)/(N-1);
x    = (a:h:b).';
y    = ya+(yb-ya)/(b-a)*(x-a);
y(1) = ya;
y(N) = yb;
```

Example

Nonlinear Beam Bending

Segment #2

```
ex = input('Run example #');
switch ex
 case 1
  f    = @(x,y,yp) ( -exp(-x.*y)-sin(yp) );
  f_y  = @(x,y,yp) ( x .* exp(-x.*y) );
  f_yp = @(x,y,yp) ( -cos(yp) );
 case 2
  f    = @(x,y,yp) ( -exp(-x.*y)-sin(10*yp) );
  f_y  = @(x,y,yp) ( x .* exp(-x.*y) );
  f_yp = @(x,y,yp) ( -10*cos(10*yp) );
 case 3
  f    = @(x,y,yp) ( -exp(-x.^4.*y)-sin(10*yp) );
  f_y  = @(x,y,yp) ( x.^4 .* exp(-x.^4.*y) );
  f_yp = @(x,y,yp) ( -10*cos(10*yp) );
 case 4
  f    = @(x,y,yp) ( -exp(-x.^4./(1+y))-sin(10*yp) );
  f_y  = @(x,y,yp) ( -x.^4 ./ (1+y).^2 .* exp(-x.^4./(1+y)) );
  f_yp = @(x,y,yp) ( -10*cos(10*yp) );
```

Example

Nonlinear Beam Bending — Segment #3

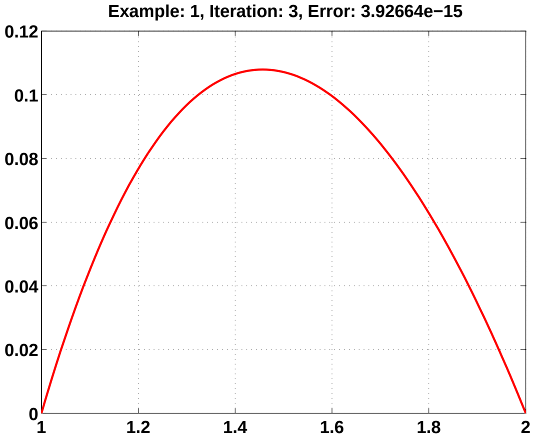
```
 case 5
  f    = @(x,y,yp) ( cos(6*pi*x)-exp(-x.^4./(1+y))-sin(10*yp) );
  f_y  = @(x,y,yp) ( -x.^4 ./ (1+y).^2 .* exp(-x.^4./(1+y)) );
  f_yp = @(x,y,yp) ( -10*cos(10*yp) );
end

ERR = TOL*2;
it = 0;
while( ERR > TOL )
  yp    = [0; (y(3:N)-y(1:(N-2)))/(2*h); 0];
  ypp   = [0; y(1:(N-2)) - 2*y(2:(N-1)) + y(3:N); 0];
  F     = ypp - h*h * f(x,y,yp);
  F(1) = 0;
  F(N) = 0;
  J     = zeros(N,N);
  J(1,1) = 1;
  J(N,N) = 1;
```

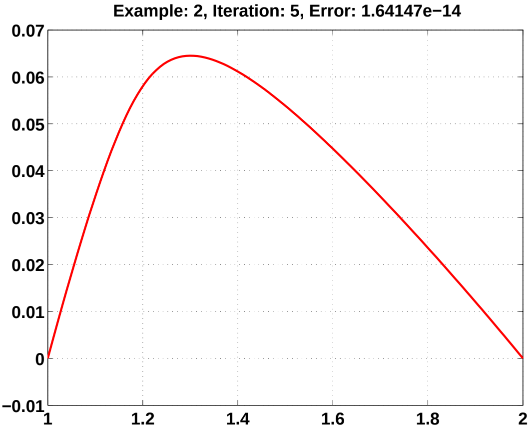
Example

Nonlinear Beam Bending — Segment #4

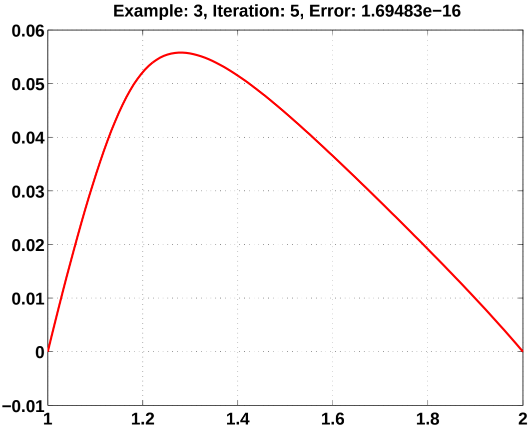
```
for n = 2:(N-1)
  J(n,n-1) =  1 + h/2 * f_yp( x(n), y(n), yp(n) );
  J(n,n)   = -2 - h*h * f_y(  x(n), y(n), yp(n) );
  J(n,n+1) =  1 - h/2 * f_yp( x(n), y(n), yp(n) );
end
deltaY = -J\F;
plot(x,y,'r-')
ERR     = norm(deltaY)
y       = y + deltaY;
grid on
title( sprintf('Example: %d, Iteration: %d, Error: %g',ex,it,ERR) )
it=1+it;
drawnow
pause(0.1)
end
```



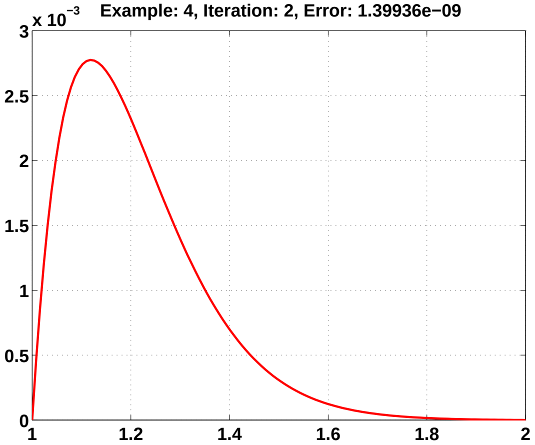
$$f(x, y, y') = -e^{xy} - \sin(y')$$



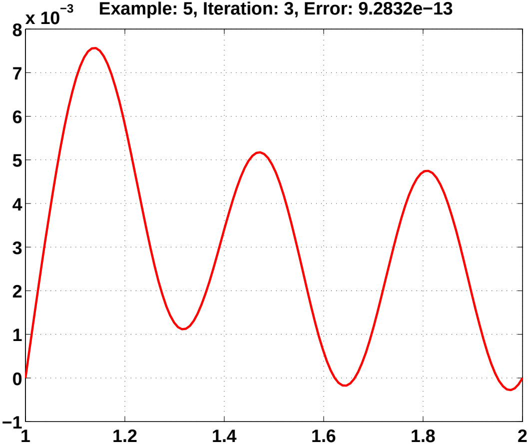
$$f(x, y, y') = -e^{xy} - \sin(10\, y')$$



$$f(x, y, y') = -e^{x^4 y} - \sin(10\, y')$$



$$f(x, y, y') = -e^{x^4/(1+y)} - \sin(10\, y')$$



$$f(x,y,y') = \cos(6\pi x) - e^{x^4/(1+y)} - \sin(10\,y')$$

Coming Up — A Different Point of View

A different approach to Boundary Value Problems:

The Rayleigh-Ritz Method / the **Finite Element Method.**

The Boundary Value Problem is reformulated as a problem of choosing, from the set of all sufficiently differentiable functions satisfying the boundary conditions, the function which minimizes a certain integral.

Also on the future menu:
- (*) Delay Differential Equations;
- (*) Spectral Methods for Boundary Value Problems.