Finite Difference Answer: Summer Semester 2016

May 16, 2016

Background: A typical one-dimensional boundary value problem might be written as

$$-\frac{d}{dx}a(x)\frac{du(x)}{dx} + c(x)u(x) = f(x) \text{ for } 0 < x < 1$$
$$u(0) = 0; u(1) = 0$$

where $a(x), c(x), f(x)$ are given functions. In order to approximate such a problem with finite differences, it is necessary to rewrite the equation as

$$-a(x)\frac{d^2u(x)}{dx} - \frac{da(x)}{dx}\frac{du(x)}{dx} + c(x)u(x) = f(x)$$

Assuming we use n equally spaced nodes for our finite difference scheme, then at each interior node, we can replace $\frac{d^2u(x)}{dx^2}$ and $\frac{du(x)}{dx}$ by centered difference approximations, involving the left and right neighbor nodes, while evaluating $a(x), \frac{da(x)}{dx}, c(x), f(x)$ at the node. The solutions at the first and last nodes are zero, from the boundary conditions.

If we use n equally spaced nodes in $[0, 1]$, then the mesh size is $h = \frac{1}{n-1}$.

If u_i^h is the approximate solution at node x_i and u is the exact solution, then the l2 error will be

$$e(h) = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(u_i^h - u(x_i))^2}$$

Given solutions computed at h_1 and a smaller mesh size h_2, we say the observed l2 convergence rate is

$$\text{rate} = \log(e(h1)/e(h2))/\log(h1/h2)$$

A: Suppose we have a boundary value problem for which:

- $a(x) = 1 + x^2$;

- $c(x) = 2$;
- $f(x) = (3x + x^2 + 5x^3 + x^4)e^x$;
- $u(x) = (x - x^2)e^x$.

Write a program which is given n, the number of nodes, and computes the solution u^h and the l2 error $e(h)$. Include a copy of your program with your answers.

```
function bvp_rate ( )

%*****************************************************************************80
%
%% BVP_RATE computes convergence rates for a finite difference BVP solution.
%
  n = 8;
  factor = 2.0;

  e = zeros ( 5, 1 );

  for i = 1 : 5
    e(i) = bvp_solve ( n + 1 );
    n = n * factor;
  end

  fprintf ( 1, '\n' );
  fprintf ( 1, '   I               H        l2 error\n' );
  fprintf ( 1, '\n' );
  n = 8;
  factor = 2.0;
  for i = 1 : 5
    h = 1.0 / n;
    fprintf ( 1, '  %2d  %14.6f  %14.6g\n', i, h, e(i) );
    n = n * factor;
  end

  fprintf ( 1, '\n' );
  fprintf ( 1, '  Convergence rates log(e(i)/e(i+1)) / log(h(i)/h(i+1)):\n' );
  fprintf ( 1, '\n' );
  for i = 1 : 4
    fprintf ( 1, '  %2d/%2d  %14.6g\n', i, i+1, log ( e(i) / e(i+1) ) / log ( factor ) )
  end

  return
end
function e = bvp_solve ( n )
```

```
%*****************************************************************************80
%
%% BVP_SOLVE solves a boundary value problem with a specified number of nodes.
%
  x1 = 0.0;
  x2 = 1.0;
%
%  Set up X, and force it to be a column vector.
%
  x = ( linspace ( x1, x2, n ) )';
%
%  Make room for the matrix and right hand side.
%
  A = zeros ( n, n );
  rhs = zeros ( n, 1 );
%
%  The first equation is the left boundary condition, U(X(1)) = 0.0;
%
  A(1,1) = 1.0;
  rhs(1) = 0.0;
%
%  Coefficients and right hand sides of equations 2 through N-1.
%
  for i = 2 : n - 1

    xm  = x(i);

    am  = 1.0 + xm * xm;
    apm = 2.0 * xm;
    cm  = 2.0;
    fm  = ( 3.0 * xm + xm^2 + 5.0 * xm^3 + xm^4 ) * exp ( xm );

    A(i,i-1) = - 2.0 * am / ( x(i) - x(i-1) ) / ( x(i+1) - x(i-1) ) ...
      + apm / ( x(i+1) - x(i-1) );

    A(i,i) = + 2.0 * am / ( x(i) - x(i-1) ) / ( x(i+1) - x(i) ) ...
      + cm;

    A(i,i+1) = - 2.0 * am / ( x(i+1) - x(i) ) / ( x(i+1) - x(i-1) ) ...
      - apm / ( x(i+1) - x(i-1) );

    rhs(i)   = fm;

  end
%
```

```
%  The last equation is the right boundary condition, U(X(N)) = 0.0;
%
  A(n,n) = 1.0;
  rhs(n) = 0.0;
%
%  Solve the linear system.
%
  u = A \ rhs;

  uexact = x .* ( 1 - x ) .* exp ( x );
%
%  Print the solution.
%
  fprintf ( 1, '\n' );
  fprintf ( 1, '     I    X         U         Uexact    Error\n' );
  fprintf ( 1, '\n' );
  for i = 1 : n
    fprintf ( 1, '  %4d  %8f  %8f  %8f  %8e\n', ...
      i, x(i), u(i), uexact(i), abs ( u(i) - uexact(i) ) );
  end

  e = norm ( u - uexact ) / n;

  return
end
```

B: Using the value $n = 9$, compute the solution to the problem, and print a table of $i, u_i^h, u(x_i)$.

I	X	U	Uexact	Error
1	0.000000	-0.000000	0.000000	5.551115e-17
2	0.125000	0.122182	0.123938	1.756392e-03
3	0.250000	0.237485	0.240755	3.269790e-03
4	0.375000	0.336576	0.341014	4.437438e-03
5	0.500000	0.407045	0.412180	5.135071e-03
6	0.625000	0.432652	0.437870	5.217972e-03
7	0.750000	0.392417	0.396938	4.520477e-03
8	0.875000	0.259524	0.262377	2.853463e-03
9	1.000000	0.000000	0.000000	0.000000e+00

C: Compute a sequence of five approximate solutions, starting with $n = 9$ and reducing the mesh size by half each time.

- C1: Print a table of your mesh sizes h and l2 errors $e(h)$;

```
     I              H          l2 error
```

```
1        0.125000       0.00119485
2        0.062500       0.00022436
3        0.031250      4.08948e-05
4        0.015625      7.34185e-06
5        0.007812      1.30799e-06
```

- C2: Compute and print 4 convergence rates by comparing successive pairs of l2 errors.

```
Convergence rates log(e(i)/e(i+1)) / log(h(i)/h(i+1)):

 1/ 2        2.41294
 2/ 3        2.45583
 3/ 4         2.4777
 4/ 5        2.48879
```