IVP $x^2 y' + 2xy - x + 1 = 0 \qquad y(1) = 0$

Exact Soln

$$y' + \frac{2}{x} y = \frac{x-1}{x^2}$$

general form of Linear 1st-order ODE

$$y' + p(x)y = g(x)$$

with I.F. $g(x) = e^{\int p(x)\,dx}$

$$\therefore g(x) = e^{\int \frac{2}{x}dx} = e^{2\ln x} = e^{\ln x^2}$$

or $g(x) = x^2$

Now multiply original ODE by x^2

$$\underbrace{x^2 y' + 2xy}_{\text{exact differential}} \overset{\text{ok}}{=} \frac{d}{dx}(x^2 y) = x - 1$$

now integrate both sides of the eqn

$$\int d(x^2 y) = \int (x-1)\,dx$$

$$x^2 y = \frac{x^2}{2} - x + c$$

or $$y(x) = \frac{1}{2} - \frac{1}{x} + \frac{c}{x^2}$$ general soln

To find the unique soln, apply the given I.C.

$$y(1) = 0 = \frac{1}{2} - 1 + c \qquad \therefore c = \frac{1}{2}$$

$$\therefore y(x) = \frac{1}{2} - \frac{1}{x} + \frac{1}{2x^2}$$ unique soln

Numerical Soln evaluate at discrete point x_i

$$y'\big|_i + \frac{2}{x_i} y_i = \frac{x_i - 1}{x_i^2}$$

This is just the Euler method

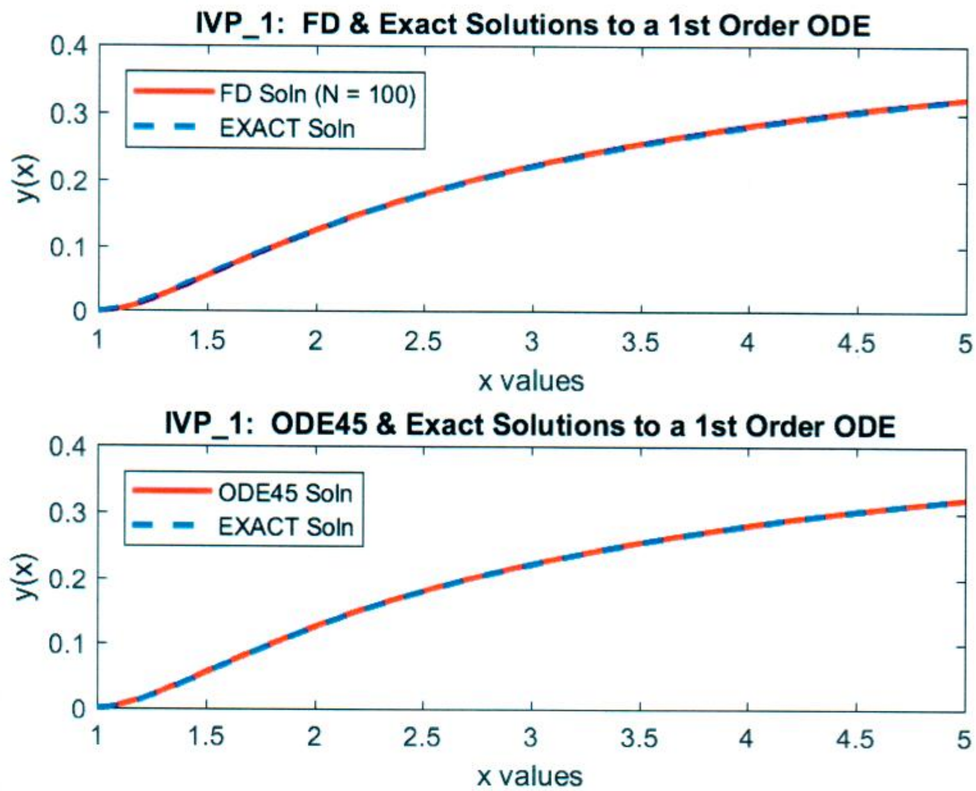
Using forward 1st-order approx for y'_i, we have

$$\frac{y_{i+1} - y_i}{\Delta x} + \frac{2}{x_i} y_i = \frac{x_i - 1}{x_i^2}$$

implement this recursive eqn with $y_1 = y(x_1) = 0$

or $$y_{i+1} = y_i + \frac{2\Delta x}{x_i} y_i + \frac{(x_i - 1)\Delta x}{x_i^2}$$

see IVP_1.m


IVP_1: FD & Exact Solutions to a 1st Order ODE
FD Soln (N = 100)
EXACT Soln
y(x)
x values
IVP_1: ODE45 & Exact Solutions to a 1st Order ODE
ODE45 Soln
EXACT Soln
y(x)
x values

```
%
%   IVP_1.M   FD and ODE45 Solution of IVPs
%
%   This example solves the 1st order IVP :  y' = (x-1-2xy)/(x^2)   with   y(1) = 0
%   using a simple FD approach and with Matlab's built-in ode45 routine
%   and compares the numerical solutions to the exact soln:
%       y(x) = 0.5 - 1/x + 0.5/x^2  (obtained via integrating factor method)
%
%   File prepared by J. R. White, UMass-Lowell (Nov. 2017)
%

      clear all, close all,  nfig = 0;
%
%   initial setup
      xo = 1;   xf = 5;   yo = 0;   tol = .00001;
%
%   evaluate exact solution
      Nx = 81;   xe = linspace(xo,xf,Nx);
      ye = 0.5 - 1./xe + 0.5./xe.^2;
%
%   evaluate the FD solution (Euler Method) and plot/compare solns
      Nfd = input('Input number of intervals for FD calc: ');
      dx = (xf-xo)/Nfd;   xfd = xo:dx:xf;   yfd = zeros(size(xfd));
      yfd(1) = yo;
      for i = 1:Nfd
        yfd(i+1) = -2*dx*yfd(i)/xfd(i) + yfd(i) + (xfd(i)-1)*dx/xfd(i)/xfd(i);
      end
%
      nfig = nfig+1;  figure(nfig)
      subplot(2,1,1)
      plot(xfd,yfd,'r-',xe,ye,'b--','LineWidth',2)
      title('IVP\_1:  FD & Exact Solutions to a 1st Order ODE')
      xlabel('x values'),ylabel('y(x)'),grid
      legend(['FD Soln (N = ',num2str(Nfd),')'],'EXACT Soln','Location','NorthWest');
%
%   now evaluate numerical solution (using ODE45) and plot/compare solns
      options = odeset('RelTol',tol);
      fxy = @(x,y) (x-1-2*x*y)/(x^2) ;  % anonymous function for ODE routine
      [xn,yn] = ode45(fxy,[xo,xf],yo,options);
%
      subplot(2,1,2)
      plot(xn,yn,'r-',xe,ye,'b--','LineWidth',2)
      title('IVP\_1:  ODE45 & Exact Solutions to a 1st Order ODE')
      xlabel('x values'),ylabel('y(x)'),grid
      legend('ODE45 Soln','EXACT Soln','Location','NorthWest');
%
%   end of simulation
```

BVP $xy'' + 2y' + xy = 0$ with $y(\frac{\pi}{2}) = 1 = y_0$

$y(\pi) = 1 = y_f$

Exact Soln via Power Series Soln Method

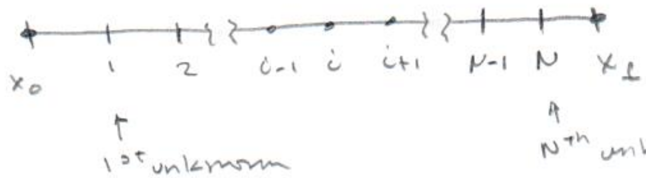
$$y(x) = \frac{\pi}{2x}(\sin x - 2\cos x)$$

← use this to compare to FD soln

Note: Power Series method is tedious, but quite useful for solving linear variable-coefficient ODEs

FD Soln

geometry:



values of $y(x)$ are known at the end points

$$\Delta x = \frac{x_f - x_0}{\#\text{ of intervals}} = \frac{x_f - x_0}{N+1}$$

and $x = x_0 + \Delta x : \Delta x : x_f - \Delta x$

Now discretize ODE at each unknown x_i location

Interior Nodes $i = 2:N-1$ (use central approx)

$$x_i\left(\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2}\right) + 2\left(\frac{y_{i+1} - y_{i-1}}{2\Delta x}\right) + x_i y_i = 0$$

multiply by $\Delta x^2 / x_i$

$$y_{i+1} - 2y_i + y_{i-1} + \frac{\Delta x}{x_i}(y_{i+1} - y_{i-1}) + y_i \Delta x^2 = 0$$

collect terms

$$\underbrace{\left(1 - \frac{\Delta x}{x_i}\right)}_{A(i,i-1)} y_{i-1} + \underbrace{(\Delta x^2 - 2)}_{A(i,i)} y_i + \underbrace{\left(1 + \frac{\Delta x}{x_i}\right)}_{A(i,i+1)} y_{i+1} = \underbrace{0}_{b(i)}$$

↑ these are the non zero coeffs in row i of the matrix eqn $\underline{\underline{A}}\,\underline{y} = \underline{b}$ where $\underline{y}$ is the desired soln vector

Valid for $i = 2:N-1$

Left Boundary Node $i = 1$

→ since node 1 is internal to the geometry, the central approx for the derivatives is valid. Thus, we can use the same eqn as above with $y_{i-1} = y_0$ ← given BC

$$\therefore \underbrace{(\Delta x^2 - 2)}_{A(1,1)} y_1 + \underbrace{\left(1 + \frac{\Delta x}{x_1}\right)}_{A(1,2)} y_2 = \underbrace{-\left(1 - \frac{\Delta x}{x_1}\right) y_0}_{b(1)}$$

← nonzero coeffs for row 1

Right Boundary Node $i = N$

→ the same statement as above is true here, with

$y_{i+1} = y_{N+1} = y_f$ ← given BC.

$$\therefore \underbrace{\left(1 - \frac{\Delta x}{x_N}\right)}_{A(N,N-1)} y_{N-1} + \underbrace{(\Delta x^2 - 2)}_{A(N,N)} y_N = \underbrace{-\left(1 + \frac{\Delta x}{x_N}\right) y_{N+1}}_{b(N)}$$

nonzero coeffs for row N

→ Once the coeff matrices are constructed, then $\underline{y} = \underline{\underline{A}} \backslash \underline{b}$ ← in matlab

(soln vector)

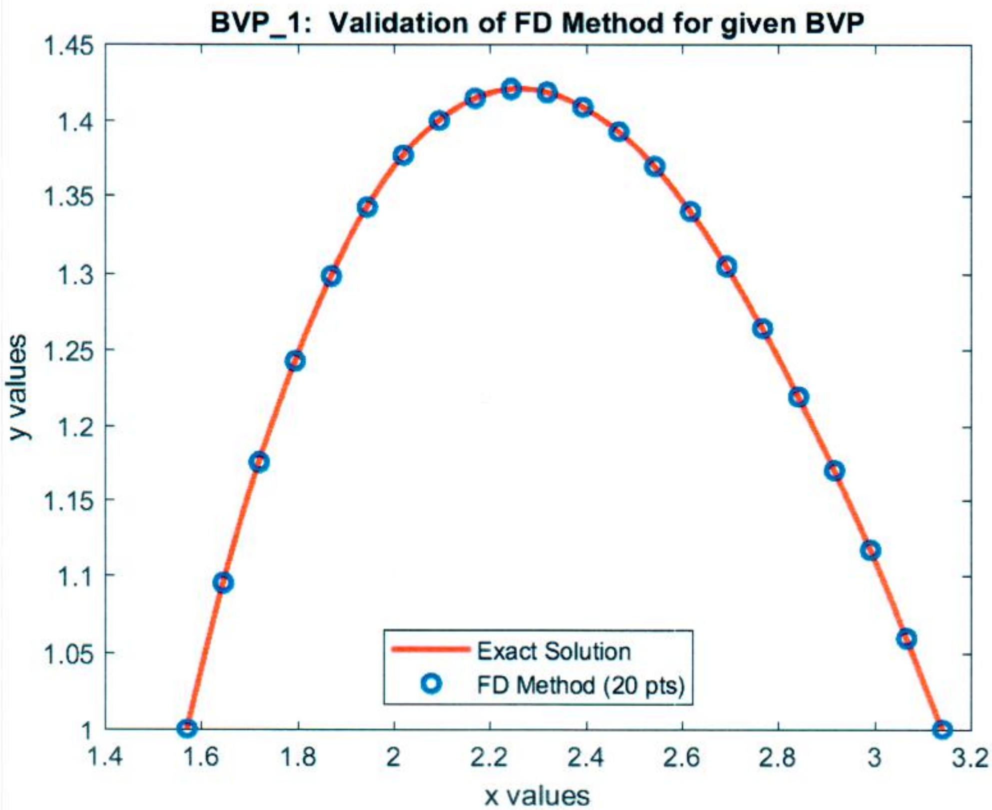
→ Finally, before plotting, add any known boundary information to the x and y vectors.

Here, since both end points are given BCs, we have

x = [xo x xf]
y = [yo y' yf]

in matlab

see implementation in BVP_1.m


BVP_1: Validation of FD Method for given BVP
1.45
1.4
1.35
1.3
1.25
1.2
1.15
1.1
1.05
1
y values
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
3.2
x values
Exact Solution
FD Method (20 pts)

```
%
%   BVP_1.M      Solution of specific BVP Using the Finite Difference Method
%                          Compared to the Analytical Solution
%
%   The BVP of interest here is:    xy'' + 2y' + xy = 0
%                     with  BCs:  y(pi/2) = 1    and    y(pi) = 1
%
%   This problem is solved using the FD Method and the solution is compared
%   to the analytical result (which was generated via the Power Series Method)
%   The Finite Difference (FD) method breaks the problem into N unknowns and solves
%   the resultant system of equations (need to treat BCs separately).
%
%   File prepared by J. R. White, UMass-Lowell (Nov. 2017)
%

     clear all,   close all,  nfig = 0;
%
%   domain limits and BCs
     xo = pi/2;    xf = pi;    yo = 1;    yf = 1;
%
%   Exact Solution from Power Series Method
     xe = linspace(xo,xf,200);
     ye = (pi./(2*xe)).*(sin(xe) - 2*cos(xe));
%
%   Finite Difference Method
     N = input('Input number of unknowns (N): ');
     dx = (xf-xo)/(N+1);   h = dx;  h2 = h*h;   x = xo+h:h:xf-h;
     A = zeros(N,N);   b = zeros(N,1);
%   central nodes
     for i = 2:N-1
       A(i,i-1) = x(i)-h;
       A(i,i) = x(i)*h2 - 2*x(i);
       A(i,i+1) = x(i)+h;
       b(i) = 0;
     end
%   left boundary
     A(1,1) = x(1)*h2 - 2*x(1);  A(1,2) = x(1)+h;   b(1) = -(x(1)-h)*yo;
%   right boundary
     A(N,N-1) = x(N)-h;  A(N,N) = x(N)*h2 - 2*x(N);  b(N) = -(x(N)+h)*yf;
%
     y = A\b;
%
%   add boundary points to solution for plotting
     yfd = [yo y' yf];   xfd = [xo x xf];
%
%   plot results from both methods
     nfig = nfig+1;   figure(nfig)
     plot(xe,ye,'r-',xfd,yfd,'bo','LineWidth',2)
     title('BVP\_1:  Validation of FD Method for given BVP')
     xlabel('x values'),ylabel('y values'),grid
     legend('Exact Solution',['FD Method (',num2str(N),' pts)'],'Location','South')
%
%   end of problem
```