Math.316

Set III

Solutions

MATLAB

Section 2.4, p. 119

Use Euler's method to find the desired solution values in 27, 28. Start with stepsize $h = 0.1$ and then use successively smaller stepsizes until successive approximate solution values at $x = 2$ agree rounded off to 2 decimal places.

Problem 27

$$y' = x^2 + y^2 - 1, y(0) = 0, y(2) = ?$$

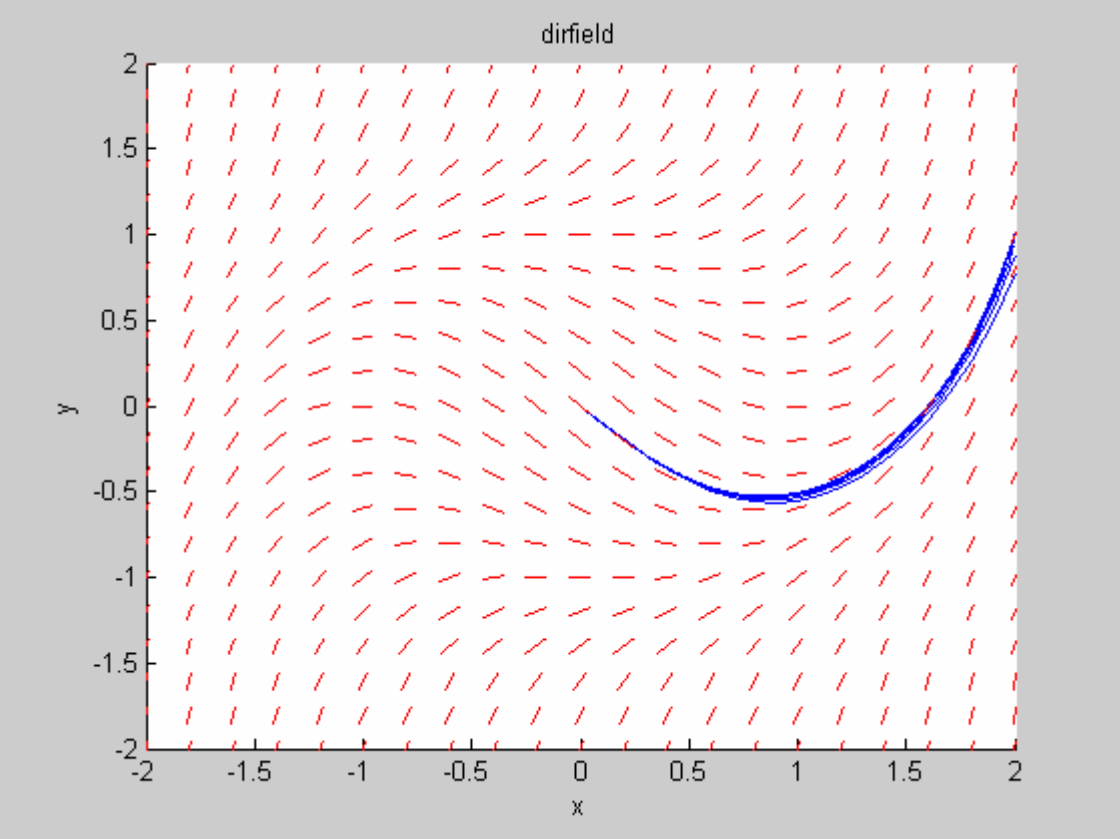


Fig. 1: Direction field and plot of solution for h =.1, .05, .025, .0125, .00625, …, .0007813

Stepsize	Y(2)	abserr
1.000e-001	7.772e-001	1.035e-001
5.000e-002	8.808e-001	5.879e-002
2.500e-002	9.396e-001	3.159e-002
1.250e-002	9.712e-001	1.642e-002
6.250e-003	9.876e-001	8.374e-003
3.125e-003	9.960e-001	4.230e-003
1.563e-003	1.000e+000	2.126e-003
7.813e-004	1.002e+000	1.066e-003

We see that the solution at x=2 becomes accurate to 2 decimals when h = .1/64 = .001563

Problem 28

$$y' = x + \frac{1}{2}y^2, y(-2) = 0, y(2) = ?$$

Here h starts as .1 and is successively halved to .05, .025, ….001563 where Y(2) attains the value 1.461 which agrees to the solution found with one half that stepsize (i.e. with h = .001563) to two decimals.

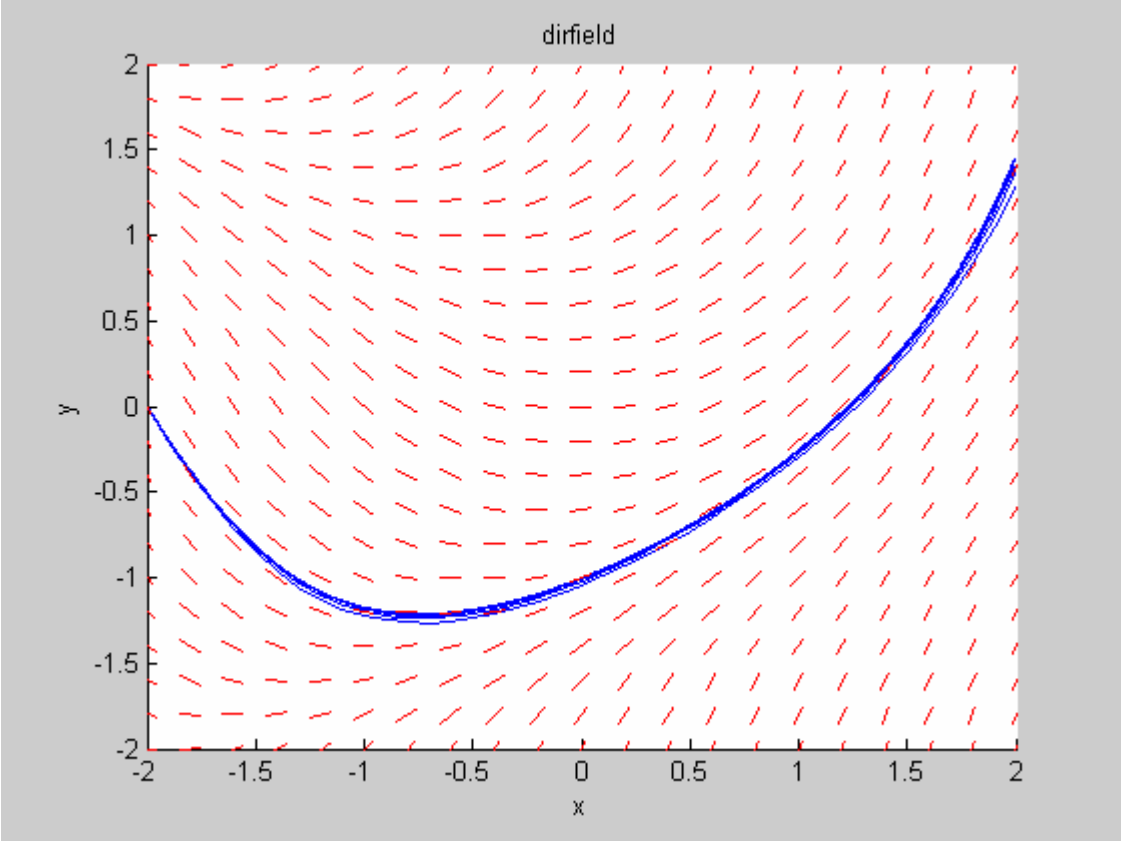


Fig.2 Same as in fig.1 for problem 28.

The table of errors (as estimated by the difference of a solution at x=2 from that found using a stepsize of twice the size) follows:

Stepsize	Y(2)	abserr
1.000e-001	1.290e+000	8.034e-002
5.000e-002	1.370e+000	4.468e-002
2.500e-002	1.415e+000	2.366e-002
1.250e-002	1.439e+000	1.219e-002
6.250e-003	1.451e+000	6.187e-003
3.125e-003	1.457e+000	3.117e-003
1.563e-003	1.460e+000	1.565e-003

Driver scripts and functions used

```
%script file hw3.m---driver for problem 2.4.27
h = .1
          disp(sprintf('-------------------------------------------'))
          disp(sprintf('  Stepsize    ------  Y(2)  ------  abserr   '))
          disp(sprintf('-------------------------------------------'))
          n = 2/h;
     [X,Y] = euler(0,0,2,n); abserr = 1;
         Yold = Y(n+1);
while abserr >.002
     h = h/2;
     n = 2/h;
     [X,Y] = euler(0,0,2,n);
     abserr = abs(Yold - Y(n+1));
     disp(sprintf('   %4.3e      %4.3e    %4.3e    ',2*h,Yold, abserr))
     Yold = Y(n+1);
end
%script file hw3.m---driver for problem 2.4.28
h = .1
          disp(sprintf('-------------------------------------------'))
          disp(sprintf('  Stepsize    ------  Y(2)  ------  abserr   '))
          disp(sprintf('-------------------------------------------'))
          n = 4/h;
     [X,Y] = euler(-2,0,2,n); abserr = 1;
     Yold = Y(n+1);
while abserr >.001
     h = h/2;
     n = 4/h;
     [X,Y] = euler(-2,0,2,n);
     abserr = abs(Yold - Y(n+1));
     disp(sprintf('   %4.3e      %4.3e    %4.3e    ',2*h,Yold, abserr))
     Yold = Y(n+1);
end
function [X, Y] = euler(x,y,x1,n)
  h = (x1-x)/n;
  X = x;
  Y = y;
  for i = 1:n
      k = f(x,y);
      x = x + h;
      y = y + h*k;
      X = [X; x];
      Y = [Y; y];
  end
  hold on
  plot(X,Y)
  axis([-2 2 -2 2])
  X1 = linspace(-2,2,21); Y1 = linspace(-2,2,21);
  hold on
  df(@f,X1,Y1,'x','y','dirfield','r',0);

  %expects function as "f(x,y)", alter accordingly
  function yp = f27(x,y)
    yp = x.^2+y.^2-1;
     function yp = f28(x,y)
    yp = x + .5*y.^2;
```

Section 2.5, p. 129

Use the improved Euler method to find the desired solution values in 27, 28. Start with stepsize $h = 0.1$ and then use successively smaller stepsizes until successive approximate solution values at $x = 2$ agree rounded off to 2 decimal places.

Problem 27

$$y' = x^2 + y^2 - 1, y(0) = 0, y(2) = ?$$

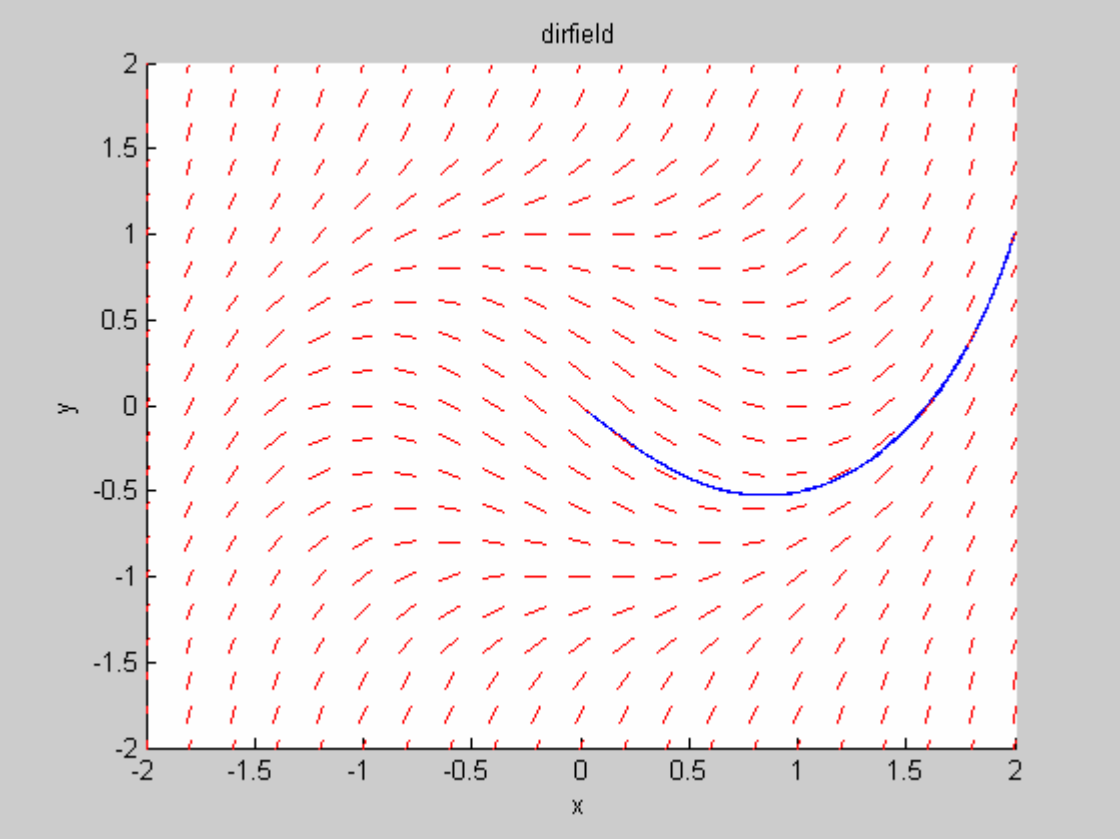
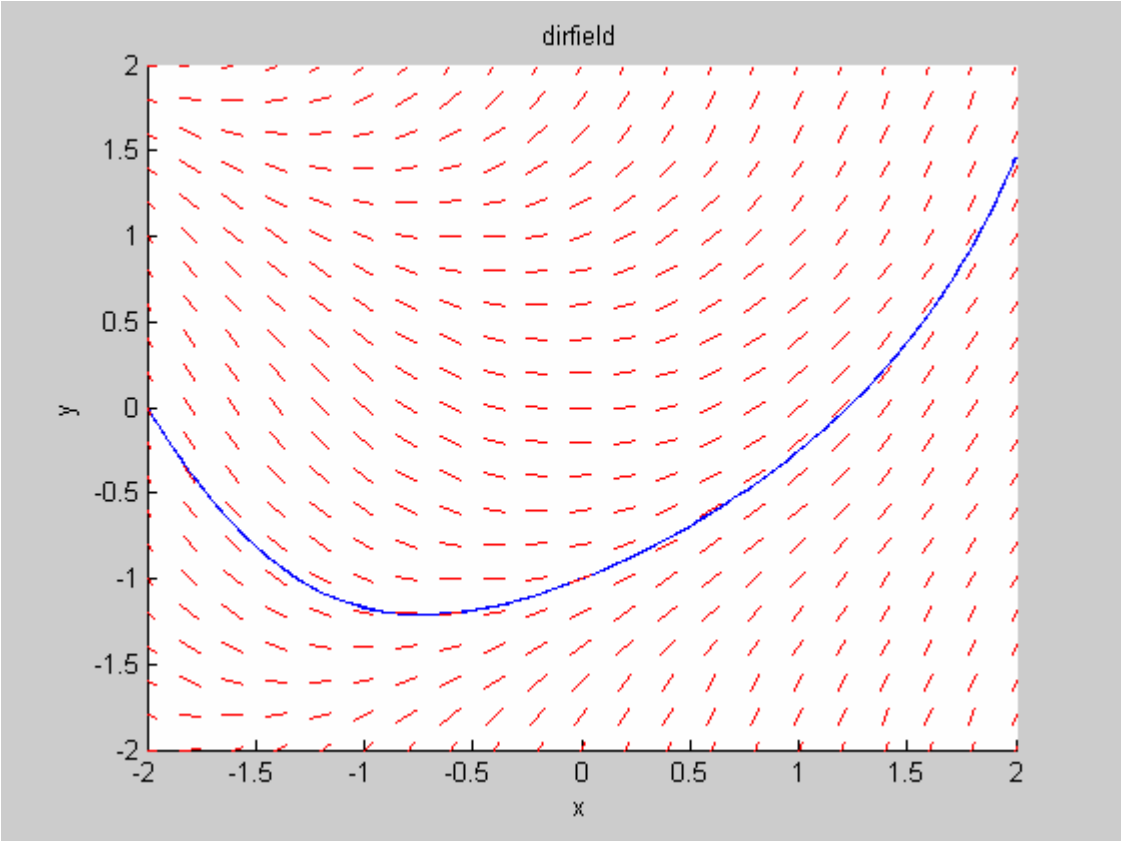


Figure 3: same as in Fig. 1, but now solved by improved Euler method. No visible difference between different stepsize computations.

Stepsize	Y(2)	abserr
1.000e-001	1.011e+000	4.672e-003
5.000e-002	1.006e+000	1.293e-003

Problem 28

$$y' = x + \frac{1}{2}y^2, y(-2) = 0, y(2) = ?$$



Stepsize	Y(2)	abserr
1.000e-001	1.466e+000	2.131e-003
5.000e-002	1.464e+000	5.594e-004

Driver scripts and functions

```
%-------------------------------------------------------------------------
%script file hw3.m---driver for problem 2.5.27
    disp(sprintf('-----------------------------------------'))
    disp(sprintf('  Stepsize    ------  Y(2)  ------  abserr   '))
    disp(sprintf('-----------------------------------------'))
    h = .1; n = 2/h;
    clear X,Y;
    [X,Y] = impeuler(0,0,2,n); abserr = 1;
```

```
    Yold = Y(n+1);
while abserr >.002
    h = h/2;
    n = 2/h;
    clear X,Y;
    [X,Y] = impeuler(0,0,2,n);
    abserr = abs(Yold - Y(n+1));
    disp(sprintf('   %4.3e      %4.3e    %4.3e',2*h,Yold,abserr))
    Yold = Y(n+1);
end
%-------------------------------------------------------------------------
%script file hw3.m---driver for problem 2.5.28
h = .1
         disp(sprintf('-----------------------------------------------'))
         disp(sprintf('  Stepsize    ------  Y(2)  ------  abserr   '))
         disp(sprintf('-----------------------------------------------'))
         n = 4/h;
         clear X,Y;
    [X,Y] = impeuler(-2,0,2,n); abserr = 1;
    Yold = Y(n+1);
while abserr >.002
    h = h/2;
    n = 4/h;
    clear X,Y;
    [X,Y] = impeuler(-2,0,2,n);
    abserr = abs(Yold - Y(n+1));
    disp(sprintf('   %4.3e      %4.3e    %4.3e',2*h,Yold,abserr))
    Yold = Y(n+1);
end
```

```
%-------------------------------------------------------------------------
function [X, Y] = impeuler(x,y,x1,n)
  h = (x1-x)/n;
  X = x;
  Y = y;
  for i = 1:n
      k1 = f(x,y);
      k2 = f(x+h, y+h*k1);
      k = (k1 + k2)/2;
      x = x + h;
      y = y + h*k;
      X = [X; x];
      Y = [Y; y];
  end
  hold on
  plot(X,Y)
  axis([-2 2 -2 2])
  X1 = linspace(-2,2,21); Y1 = linspace(-2,2,21);
  hold on
  df(@f,X1,Y1,'x','y','dirfield','r',0);

  % program expects function as "f(x,y)", alter accordingly
  function yp = f27(x,y)
    yp = x.^2+y.^2-1;

  function yp = f28(x,y)
    yp = x + .5*y.^2;
```