Numerical Methods for ODEs

Euler
Augmented Euler
• 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
  • dirfield(@(x,y) f(x,y), X1, Y1);

• function yp = f(x,y)
  • yp = x+y;

•
An example of Euler's method

\[ y' = x + y, \]
\[ y(-2) = 1.2 \]
\[ n = 5, 10, 20, 40, 80, 160, 320 \]
Different step size: varying answers

- Notice how decreasing the stepsize in the previous example produces smoother solution curves. These keep changing as the stepsize decreases, but do approach a limit for very small (approx $4 \times 10^{-4}$) stepsize $h$. The rate of convergence of Euler’s method is linear (i.e. the error decreases proportionally to $h$).
Improved Euler

- function \([X, Y] = \text{impeuler}(x, y, x1, n)\)
- \(h = (x1 - x)/n;\)
- \(X = x;\)
- \(Y = y;\)
- for \(i = 1:n\)
  - \(k1 = f(x, y);\)
  - \(k2 = f(x+h, y+h\cdot k1);\)
  - \(k = (k1 + k2)/2;\)
  - \(x = x + h;\)
  - \(y = y + h\cdot k;\)
- \(X = [X; x];\)
- \(Y = [Y; y];\)
- end

```matlab
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);

function yp = f(x,y)
    yp = x+y;
```
Same calculation using Euler’s improved method

Notice how the curves are much closer together, even for 10 point resolution.