

Queens College, CUNY, Department of Computer Science

Numerical Methods

CSCI 361 / 761

Fall 2017

Instructor: Dr. Sateesh Mane

© Sateesh R. Mane 2017

December 31, 2017

16 Lecture 16a

Numerical solution of systems of ordinary differential equations

- We display worked examples of **initial value problems using auxiliary variables.**

16.12 Worked example 1

16.12.1 Equation

- Consider the following ordinary differential equation

$$\frac{d^2 y}{dx^2} + 2x \frac{dy}{dx} + (1 - x^2)y = e^x. \quad (16.12.1)$$

- It is a second order ordinary differential equation. We introduce an auxiliary variable v via

$$v = \frac{dy}{dx}. \quad (16.12.2)$$

- We reexpress eq. (16.12.1) as a pair of coupled first order equations as follows:

$$\frac{dy}{dx} = v, \quad (16.12.3a)$$

$$\frac{dv}{dx} = -2xv - (1 - x^2)y + e^x. \quad (16.12.3b)$$

- We express this in the formal notation as follows. We define $\mathbf{y} = (y_1, y_2) = (y, v)$. Then

$$\begin{aligned} \frac{d\mathbf{y}}{dx} &= \mathbf{f}(x, \mathbf{y}), \\ \frac{d}{dx} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} &= \begin{pmatrix} f_1(x, y_1, y_2) \\ f_2(x, y_1, y_2) \end{pmatrix}. \\ \frac{d}{dx} \begin{pmatrix} y \\ v \end{pmatrix} &= \begin{pmatrix} v \\ -2xv - (1 - x^2)y + e^x \end{pmatrix}. \end{aligned} \quad (16.12.4)$$

- Therefore the right hand side functions are

$$f_1(x, y_1, y_2) = f_1(x, y, v) = v = y_2, \quad (16.12.5a)$$

$$\begin{aligned} f_2(x, y_1, y_2) &= f_2(x, y, v) \\ &= -2xv - (1 - x^2)y + e^x \\ &= -2xy_2 - (1 - x^2)y_1 + e^x. \end{aligned} \quad (16.12.5b)$$

16.12.2 C++ code

- In terms of C++ function calls, we have $n = 2$ and

```
int f(int n, double x, const std::vector<double> & y, std::vector<double> & g)
{
    // first component "f1(x,y,v) = v"
    g[0] = y[1];

    // second component "f2(x,y,v) = -2xy -(1-x^2)y + exp(x)"
    g[1] = -2.0*x*y[1] - (1.0 - x*x)*y[0] + exp(x);

    return 0;
}
```

All the integration schemes can be called with the following inputs:

```
// n = 2
// x = x_i
// h = step size
// vector (array) y_in = (y, v)_i
// vector (array) y_out = (y, v)_{i+1}

int Euler_explicit(int n, double x, double h,
                  std::vector<double> & y_in,
                  std::vector<double> & y_out);

int midpoint(int n, double x, double h,
            std::vector<double> & y_in,
            std::vector<double> & y_out);

int trapezoid(int n, double x, double h,
             std::vector<double> & y_in,
             std::vector<double> & y_out);

int RK4(int n, double x, double h,
       std::vector<double> & y_in,
       std::vector<double> & y_out);
```

16.12.3 Solution for $i = 0$, etc

- Let the initial conditions at $x_0 = 0$ be $y_0 = 1$ and $v_0 = y'_0 = -1$.
- We shall integrate eq. (16.12.4) using explicit Euler integration with a stepsize h .
- The formal equations are

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h\mathbf{f}(x, \mathbf{y}). \quad (16.12.6)$$

- In terms of y and v , the equations are

$$\begin{pmatrix} y \\ v \end{pmatrix}_{i+1} = \begin{pmatrix} y \\ v \end{pmatrix}_i + h \begin{pmatrix} v_i \\ -2x_i v_i - (1 - x_i^2)y_i + \exp(x_i) \end{pmatrix}. \quad (16.12.7)$$

- Step $i = 0$:

1. We have

$$y_1 = y_0 + hv_0 = 1 - h. \quad (16.12.8)$$

2. Next we have

$$\begin{aligned} v_1 &= v_0 + h(-2x_0 v_0 - (1 - x_0^2)y_0 + \exp(x_0)) \\ &= -1 + h(0 - (1 - 0) + \exp(0)) \\ &= -1. \end{aligned} \quad (16.12.9)$$

- Step $i = 1$:

1. We have

$$y_2 = y_1 + hv_1 = 1 - h - h = 1 - 2h. \quad (16.12.10)$$

2. Next we have

$$\begin{aligned} v_2 &= v_1 + h(-2x_1 v_1 - (1 - x_1^2)y_1 + \exp(x_1)) \\ &= -1 + h(2h - (1 - h^2)(1 - h) + \exp(h)) \\ &= -1 + h(-1 + 3h + h^2 - h^3 + \exp(h)) \\ &= -1 - h + 3h^2 + h^3 - h^4 + h \exp(h). \end{aligned} \quad (16.12.11)$$

- Step $i = 2$:

1. We have

$$\begin{aligned} y_3 &= y_2 + hv_2 = 1 - 2h + h(-1 - h + 3h^2 + h^3 - h^4 + h \exp(h)) \\ &= 1 - 3h - h^2 + 3h^3 + h^4 - h^5 + h^2 \exp(h). \end{aligned} \quad (16.12.12)$$

2. Next we have

$$\begin{aligned} v_3 &= v_2 + h(-2x_2 v_2 - (1 - x_2^2)y_2 + \exp(x_2)) \\ &= v_2 + h(-4hv_2 - (1 - 4h^2)y_2 + \exp(2h)) \\ &= \text{ugh}. \end{aligned} \quad (16.12.13)$$

16.13 Worked example 2

16.13.1 Equation

- Consider the following ordinary differential equation

$$\frac{d^2 y}{dx^2} + y = 0. \quad (16.13.1)$$

- This is a simple equation. We know the general solution is

$$y(x) = c_1 \cos(x) + c_2 \sin(x). \quad (16.13.2)$$

- It is a second order ordinary differential equation. We introduce an auxiliary variable v via

$$v = \frac{dy}{dx}. \quad (16.13.3)$$

- We reexpress eq. (16.13.1) as a pair of coupled first order equations as follows:

$$\frac{dy}{dx} = v, \quad (16.13.4a)$$

$$\frac{dv}{dx} = -y. \quad (16.13.4b)$$

- We express this in the formal notation as follows. We define $\mathbf{y} = (y_1, y_2) = (y, v)$. Then

$$\begin{aligned} \frac{d\mathbf{y}}{dx} &= \mathbf{f}(x, \mathbf{y}), \\ \frac{d}{dx} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} &= \begin{pmatrix} f_1(x, y_1, y_2) \\ f_2(x, y_1, y_2) \end{pmatrix}. \\ \frac{d}{dx} \begin{pmatrix} y \\ v \end{pmatrix} &= \begin{pmatrix} v \\ -y \end{pmatrix}. \end{aligned} \quad (16.13.5)$$

- Therefore the right hand side functions are

$$f_1(x, y_1, y_2) = f_1(x, y, v) = v = y_2, \quad (16.13.6a)$$

$$f_2(x, y_1, y_2) = f_2(x, y, v) = -y = -y_1. \quad (16.13.6b)$$

16.13.2 C++ code

- In terms of C++ function calls, we have $n = 2$ and

```
int f(int n, double x, const std::vector<double> & y, std::vector<double> & g)
{
    // first component "f1(x,y,v) = v"
    g[0] = y[1];

    // second component "f2(x,y,v) = -y"
    g[1] = -y[1];

    return 0;
}
```

All the integration schemes can be called with the following inputs:

```
// n = 2
// x = x_i
// h = step size
// vector (array) y_in  = (y, v)_i
// vector (array) y_out = (y, v)_{i+1}

int Euler_explicit(int n, double x, double h,
                  std::vector<double> & y_in,
                  std::vector<double> & y_out);

int midpoint(int n, double x, double h,
            std::vector<double> & y_in,
            std::vector<double> & y_out);

int trapezoid(int n, double x, double h,
             std::vector<double> & y_in,
             std::vector<double> & y_out);

int RK4(int n, double x, double h,
       std::vector<double> & y_in,
       std::vector<double> & y_out);
```

16.13.3 Solution for $i = 0$, etc

- Let the initial conditions at $x_0 = 0$ be $y_0 = 1$ and $v_0 = y'_0 = 0$.
- Then we know the exact solution is

$$y_{\text{exact}}(x) = \cos(x). \quad (16.13.7)$$

- We shall integrate eq. (16.13.5) using explicit Euler integration with a stepsize h .
- The formal equations are

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h\mathbf{f}(x, \mathbf{y}). \quad (16.13.8)$$

- In terms of y and v , the equations are

$$\begin{pmatrix} y \\ v \end{pmatrix}_{i+1} = \begin{pmatrix} y \\ v \end{pmatrix}_i + h \begin{pmatrix} v_i \\ -y_i \end{pmatrix}. \quad (16.13.9)$$

- Step $i = 0$:

1. We have

$$y_1 = y_0 + hv_0 = 1 - 0 = 1. \quad (16.13.10)$$

2. Next we have

$$v_1 = v_0 - hy_0 = 0 - h = -h. \quad (16.13.11)$$

- Step $i = 1$:

1. We have

$$y_2 = y_1 + hv_1 = 1 + h(-h) = 1 - h^2. \quad (16.13.12)$$

2. Next we have

$$v_2 = v_1 - hy_1 = -h - h = -2h. \quad (16.13.13)$$

- Step $i = 2$:

1. We have

$$y_3 = y_2 + hv_2 = 1 - h^2 + h(-2h) = 1 - 3h^2. \quad (16.13.14)$$

2. Next we have

$$v_3 = v_2 - hy_2 = -2h - h(1 - h^2) = -3h + h^3. \quad (16.13.15)$$

- Step $i = 3$:

1. We have

$$y_4 = y_3 + hv_3 = 1 - 3h^2 + h(-3h + h^3) = 1 - 6h^2 + h^4. \quad (16.13.16)$$

2. Next we have

$$v_4 = v_3 - hy_3 = -3h + h^3 - h(1 - 3h^2) = -4h + 4h^3. \quad (16.13.17)$$

- We see that the explicit Euler method is not very accurate.

16.13.4 Solution using midpoint method

- Let us integrate eq. (16.13.5) using the midpoint method with a stepsize h .
- In terms of y and v , the equations are

$$\begin{pmatrix} y \\ v \end{pmatrix}_{i+1} = \begin{pmatrix} y \\ v \end{pmatrix}_i + h \begin{pmatrix} v_i \\ -y_i \end{pmatrix}. \quad (16.13.18)$$

- However, now we require the values intermediate points.

1. First we have

$$\mathbf{g}_0 \sim \begin{pmatrix} g_{y,0} \\ g_{v,0} \end{pmatrix}_i = \begin{pmatrix} v_i \\ -y_i \end{pmatrix}. \quad (16.13.19)$$

2. Next we have

$$\mathbf{g}_1 \sim \begin{pmatrix} g_{y,1} \\ g_{v,1} \end{pmatrix}_i = \begin{pmatrix} f_1(x_i + \frac{1}{2}h, y_i + \frac{h}{2}(g_{y,0})_i, v_i + \frac{h}{2}(g_{v,0})_i) \\ f_2(x_i + \frac{1}{2}h, y_i + \frac{h}{2}(g_{y,0})_i, v_i + \frac{h}{2}(g_{v,0})_i) \end{pmatrix} = \begin{pmatrix} v_i - \frac{h}{2}y_i \\ -y_i - \frac{h}{2}v_i \end{pmatrix}. \quad (16.13.20)$$

- Because of too many subscripts, for clarity let us define, at the step i ,

$$\begin{pmatrix} G_y \\ G_v \end{pmatrix}_i = \begin{pmatrix} v_i - \frac{h}{2}y_i \\ -y_i - \frac{h}{2}v_i \end{pmatrix}. \quad (16.13.21)$$

- Step $i = 0$:

1. We have

$$(G_y)_0 = v_0 - \frac{h}{2}y_0 = 0 - \frac{h}{2}. \quad (16.13.22)$$

2. Next

$$(G_v)_0 = -y_0 - \frac{h}{2}v_0 = -1. \quad (16.13.23)$$

3. Then

$$\begin{aligned} y_1 &= y_0 + h(G_y)_0 = 1 - \frac{h^2}{2}, \\ v_1 &= v_0 + h(G_v)_0 = 0 - h = -h. \end{aligned} \quad (16.13.24)$$

- Step $i = 1$:

1. We have

$$(G_y)_1 = v_1 - \frac{h}{2}y_1 = -h - \frac{h}{2}\left(1 - \frac{h^2}{2}\right) = -\frac{3h}{2} + \frac{h^3}{4}. \quad (16.13.25)$$

2. Next

$$(G_v)_1 = -y_1 - \frac{h}{2}v_1 = -1 + \frac{h^2}{2} + \frac{h^2}{2} = -1 + h^2. \quad (16.13.26)$$

3. Then

$$\begin{aligned}
y_2 &= y_1 + h(G_y)_1 = 1 - \frac{h^2}{2} + h\left(-\frac{3h}{2} + \frac{h^3}{4}\right) \\
&= 1 - 2h^2 + \frac{h^4}{4}, \\
v_2 &= v_1 + h(G_v)_1 = -h + h(-1 + h^2) \\
&= -2h + h^3.
\end{aligned} \tag{16.13.27}$$

• Step $i = 2$:

1. We have

$$(G_y)_2 = v_2 - \frac{h}{2} y_2 = -2h + h^3 - \frac{h}{2} \left(1 - 2h^2 + \frac{h^4}{4}\right) = -\frac{5h}{2} + 2h^3 - \frac{h^5}{8}. \tag{16.13.28}$$

2. Next

$$(G_v)_2 = -y_2 - \frac{h}{2} v_2 = -1 + 2h^2 - \frac{h^4}{4} + h^2 - \frac{h^4}{2} = -1 + 3h^2 - \frac{3h^4}{4}. \tag{16.13.29}$$

3. Then

$$\begin{aligned}
y_3 &= y_2 + h(G_y)_2 = 1 - 2h^2 + \frac{h^4}{4} + h\left(-\frac{5h}{2} + 2h^3 - \frac{h^5}{8}\right) \\
&= 1 - \frac{9h^2}{2} + \frac{9h^4}{4} - \frac{h^6}{8}, \\
v_3 &= v_2 + h(G_v)_2 = -2h + h^3 + h\left(-1 + 3h^2 - \frac{3h^4}{4}\right) \\
&= -3h + 4h^3 - \frac{3h^5}{4}.
\end{aligned} \tag{16.13.30}$$

• This is more accurate than the explicit Euler method. It matches the exact solution to $O(h^2)$.