Queens College, CUNY, Department of Computer Science

Numerical Methods

CSCI 361 / 761

Spring 2018

Instructor: Dr. Sateesh Mane

April 29, 2018

18 Lecture 18

Numerical solution of systems of ordinary differential equations

- In this lecture we continue the study of **initial value problems**.
- In this lecture we present a **semi-implicit** integration algorithm.

18.1 Basic notation

- We repeat the basic definitions from the previous lecture.
- Let the system of coupled differential equations be

$$\frac{d\boldsymbol{y}}{dx} = \boldsymbol{f}(x, \boldsymbol{y})\,. \tag{18.1.1}$$

- There are m unknown variables y_j, $j = 1, \dots, m$.
- The starting point is $x = x_0$, and the initial value $\boldsymbol{y}_0$ is given.
- The above is called the **Cauchy problem** or **initial value problem.**
- Our interest is to integrate eq. (18.1.1) numerically, using steps h_i, so $x_{i+1} = x_i + h_i$.
- The steps h_i need not be of equal size.
- We define $\boldsymbol{y}_i = \boldsymbol{y}(x_i)$.
- We employ the notation $\boldsymbol{y}_i^{\mathrm{ex}} = \boldsymbol{y}^{\mathrm{ex}}(x_i)$ to denote the exact solution.
- We employ the notation $\boldsymbol{y}_i^{\mathrm{num}} = \boldsymbol{y}^{\mathrm{num}}(x_i)$ to denote the numerical solution.

18.2 Semi-implicit method

- The system of equations is given by eq. (18.1.1).
- At the step i we employ a finite difference to approximate the derivative, which yields the implicit equation

$$\boldsymbol{y}^{\rm imp}_{i+1} = \boldsymbol{y}^{\rm imp}_{i} + h_i \boldsymbol{f}(x_{i+1}, \boldsymbol{y}^{\rm imp}_{i+1}) \,. \tag{18.2.1}$$

- We employ a multivariate Newton–Raphson algorithm to solve eq. (18.2.1).
- For the first iteration, we write

$$\boldsymbol{f}(x_{i+1}, \boldsymbol{y}^{\rm imp}_{i+1}) = \boldsymbol{f}(x_i, \boldsymbol{y}^{\rm imp}_{i}) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}} \cdot (\boldsymbol{y}^{\rm imp}_{i+1} - \boldsymbol{y}^{\rm imp}_{i}) \,. \tag{18.2.2}$$

- The matrix $\partial \boldsymbol{f}/\partial \boldsymbol{y}$ is the **Jacobian matrix** J

$$J_{jk} = \frac{\partial f_j}{\partial y_k} \qquad\qquad (1 \le j,k \le m) \,. \tag{18.2.3}$$

- We substitute into eq. (18.2.1) to obtain

$$\boldsymbol{y}^{\rm imp}_{i+1} - \boldsymbol{y}^{\rm imp}_{i} = h_i \Big[\boldsymbol{f}(x_i, \boldsymbol{y}^{\rm imp}_{i}) + J\big(\boldsymbol{y}^{\rm imp}_{i+1} - \boldsymbol{y}^{\rm imp}_{i}\big) \Big] \,. \tag{18.2.4}$$

- Rearranging the terms in eq. (18.2.4) yields a matrix equation in n variables (I is the unit $m \times m$ matrix)

$$\big(I - J\big)\boldsymbol{y}^{\rm imp}_{i+1} = \big(I - J\big)\boldsymbol{y}^{\rm imp}_{i} + h_i \, \boldsymbol{f}(x_i, \boldsymbol{y}^{\rm imp}_{i}) \,. \tag{18.2.5}$$

- The formula in eq. (18.2.5) is called a **semi-implicit method.**
- We stop the Newton–Raphson iteration after only one iteration.
- To compute $\boldsymbol{y}^{\rm imp}_{i+1}$, eq. (18.2.5) must be solved using a matrix equation solver.
  1. We can write the formal solution of eq. (18.2.5) as

  $$\boldsymbol{y}^{\rm imp}_{i+1} = \boldsymbol{y}^{\rm imp}_{i} + h_i \left(I - J\right)^{-1} \boldsymbol{f}(x_i, \boldsymbol{y}^{\rm imp}_{i}) \,. \tag{18.2.6}$$

  2. Note that eq. (18.2.5) is simply a one-iteration approximation to solve eq. (18.2.1).
  3. The semi-implicit algorithm is not guaranteed to be computationally stable.
  4. The basic hope in the semi-implicit algorithm is that if the magnitude of the stepsize $|h_i|$ is small enough, using only one Newton–Rahpson iteration will be satisfactory.