3-Day Scientific Computing Virtual Workshop

September 10-12, 2020 09:00 AM - 12:00PM

Tulin Kaman
Department of Mathematical Sciences
E-mail: tkaman@uark.edu

Day 1-2, 09/10-11/2020

SIAM/AWM Student Chapters at U of A

(Biweekly) Meeting, Wednesdays starting from September 9th, @4:00PM

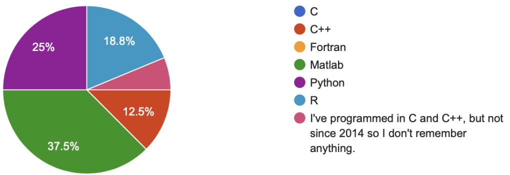
- UofA SIAM Student Chapters, since Fall 2018
  https://kaman.uark.edu/siam
- UofA AWM Student Chapters, since Summer 2019
  https://kaman.uark.edu/awm
- The participants of 3-day workshop are eligible to claim free SIAM membership. Complete the online application https://my.siam.org.
  - Free membership in two SIAM Activity Groups (SIAG)
  - Subscription to SIAM News
  - Subscription to SIAM Review
  - Free student members get electronic access only
  - Subscription to SIAM Unwrapped
  - 30% discount on all SIAM books

Agenda

- Day 1: Presentation of problems and algorithms for solving Ordinary Differential Equations
- Day 2: Systems of Differential Equations; Problem assignments
- Day 3: Presentation of solutions by students

Prize(iPad 6th generation)

1. Earn points each day by attending, participating and completing assignments.
2. Implement the algorithms in your preferred language C/C++/Fortran or MATLAB/Python.



Ordinary Differential Equations(ODE)

- A differential equation is an equation involving one or more derivatives of an unknown function with respect to a single independent variable.

$$f^{(p)}(x) + \cdots + f^{''}(x) + f'(x) + f(x) = 0$$

- A differential equation has order p if p is the maximum order of differentiation.
- Describe the evolution of many phenomena in many fields.
  Examples:
  1. Thermodynamics
  2. Population dynamics
  3. Baseball trajectory
  4. Electrical circuits

References: Scientific Computing with MATLAB and Octave, By Alfio Quarteroni, Fausto Saleri, Paola Gervasio
Programming Projects in C for Students of Engineering, Science, and Mathematics, by Rouben Rostamian.

- Focus on the first order differential equations, since pth order equations can be reduced to a system of p equations of order 1.
- An ODE admits an infinite number of solutions. To find the unique solution, we impose a condition.
- Consider the Cauchy problem: Find $y : I \subset \mathbb{R} \to \mathbb{R}$ such that

$$\begin{cases} y'(t) = f(t, y(t)) \quad \forall t \in I, \\ y(t_0) = y_0 \end{cases}$$

I: interval

f: function

y': derivative of y with respect to t

t_0: a point of I

y_0: initial data.

Assume that the function $f(t,y)$ is

1. continuous with respect to both its argument
2. Lipschitz-continuous with respect to its second argument, that is, there exists a positive constant L (Lipschitz constant) such that

$$|f(t,y_1) - f(t,y_2)| \leq L|y_1 - y_2|, \forall t \in I, \forall y_1, y_2 \in \mathbb{R}$$

Then the solution $y = y(t)$ of the Cauchy problem exists, is unique and belongs to $C^1(I)$.

- Explicit solutions are available only for very special of ODEs.
- Some cases: the solution is available only in implicit form.

$$y' = \frac{y-t}{y+t} \rightarrow \frac{1}{2} ln(t^2 + y^2) + arctan\frac{y}{t} = C$$

- Some cases: the solution is not even representable in implicit form, can only be expressed through a series expansion. $y' = e^{-t^2}$.

Numerical methods

approximate the solution of **every** family of ordinary equations for which solutions do exist.

1. Subdivide the integration interval $I = [t_0, T]$, $T < \infty$, into N_h intervals of length $h = (T - t_0)/N_h$.
   h: discretization step or time step or steplength...
2. At each node $t_n = t_0 + nh, \quad (n = 1, \cdots, N_h)$, we seek the unknown value u_n which approximates $y_n = y(t_n)$.
3. The set of values $\{u_0 = y_0, u_1, \cdots, u_{N_h}\}$ **numerical solution**.

Approximation of function derivatives I

- Consider a function $f : [a, b] \to \mathbb{R}$ continuously differentiable in $[a, b]$.
- Expand f in a Taylor series; assume $f \in C^2((a, b))$,
  1. $$f(\bar{x} + h) = f(\bar{x}) + hf'(\bar{x}) + \frac{h^2}{2} f''(\xi),$$
     where ξ is a point in interval $(\bar{x}, \bar{x} + h)$. Therefore
     $$(\delta_+ f)(\bar{x}) = f'(x) + \frac{h}{2} f''(\xi),$$
  2. $$f(\bar{x} - h) = f(\bar{x}) - hf'(\bar{x}) + \frac{h^2}{2} f''(\eta),$$
     where η is a point in interval $(\bar{x} - h, \bar{x})$. Therefore
     $$(\delta_- f)(\bar{x}) = f'(\bar{x}) + \frac{h}{2} f''(\eta),$$

Approximation of function derivatives II

- For h sufficiently small and positive, we can assume that the quantity

  1. $$(\delta_+ f)(\bar{x}) = \frac{f(\bar{x}+h) - f(\bar{x})}{h}$$

     a first-order approximation of $f'(\bar{x})$, **forward finite difference**.

  2. $$(\delta_- f)(\bar{x}) = \frac{f(\bar{x}) - f(\bar{x}-h)}{h}$$

     a first-order approximation of $f'(\bar{x})$, **backward finite difference**.

  3. $$(\delta f)(\bar{x}) = \frac{f(\bar{x}+h) - f(\bar{x}-h)}{2h}$$

     a second-order approximation of $f'(\bar{x})$, **centered finite difference**.

Approximation of function derivatives III

Exercise: Assume $f \in C^3((a,b))$, expand $f(\bar{x}+h)$ and $f(\bar{x}-h)$ at the third order around $\bar{x}$ and sum up the two expressions, what do you obtain $f'(x) - (\delta f)(\bar{x})$?

$$
\begin{gathered}
f'(x) - (\delta f)(\bar{x}) = -\frac{h^2}{12}\left[f'''(\xi_-) + f'''(\xi_+)\right] \\
\textit{where } \xi_- \in (\bar{x}-h, \bar{x}), \qquad \xi_+ \in (\bar{x}, \bar{x}+h)
\end{gathered}
$$

Euler Methods

- Consider the differential equation $y'(t) = f(t, y(t))$ at every node t_n.
- Generate the numerical solution u_{n+1} at the node t_{n+1}
  1. **Forward Euler Method**: Approximate $y'(t_n)$ by forward finite difference
  $$\boxed{u_{n+1} = u_n + hf_n, \quad n = 0, \cdots N_h - 1}$$
  2. **Backward Euler Method**: Approximate $y'(t_{n+1})$ by backward finite difference
  $$\boxed{u_{n+1} = u_n + hf_{n+1}, \quad n = 0, \cdots N_h - 1}$$
- Both methods are **one_step method**.

Population dynamics I

Example

Consider a population of bacteria in a confined environment in which no more than B elements can coexist. Assume that, at the initial time, the number of individuals is equal to $y_b \ll B$ and the growth rate of the bacteria is a positive constant C. In this case the rate of change of the population is proportional to the number of existing bacteria, under the restriction that the total number cannot exceed B. This is expressed by the differential equation

$$\frac{dy}{dt} = Cy\left(1 - \frac{y}{B}\right),$$

whose solution $y = y(t)$ denotes the number of bacteria at time t.

Population dynamics II

Assume that two populations y_1 and. y_2 be in competition, then

$$\frac{dy_1}{dt} = C_1 y_1 \left(1 - b_1 y_1 - d_2 y_2\right),$$
$$\frac{dy_2}{dt} = -C_2 y_2 \left(1 - b_2 y_2 - d_1 y_1\right),$$

C_1, C_2: growth rates of the two populations
d_1, d_2: coefficients that govern the type of interaction between the two populations
b_1, b_2: coefficients related to the available quantity of nutrients
We will revisit this problem in Day2.

Explicit/Implicit Euler Methods

$$\frac{dy}{dt} = Cy\left(1 - \frac{y}{B}\right),$$

- In the forward Euler method u_{n+1} depends on the value u_n previously computed, called (*Explicit Euler method*)

$$u_{n+1} = u_n + hCu_n(1 - u_n/B)$$

- In the backward Euler method u_{n+1} depends on itself through the value f_{n+1}. (*Implicit Euler method*)

$$u_{n+1} = u_n + hCu_{n+1}(1 - u_{n+1}/B)$$

Convergence analysis I

- A numerical method is **convergent** if

$$\forall n = 0, \cdots N_h, \qquad |y_n - u_n| \leq C(h)$$

$C(h)$: inifinitesimal w.r.t. to h when $h \to 0$.

- If $C(h) = \mathcal{O}(h^p)$, there exists a positive constant c such that $C(h) \leq ch^p$ and p maximum integer that holds the inequality, the method converges with **order p**.
- Verify the forward Euler method converges!

$$e_n = y_n - u_n = (y_n - u_n^*) + (u_n^* - u_n)$$

where $u_n^* = y_{n-1} + hf(t_{n-1}, y_{n-1})$ the numerical solution at time t_n starting from the exact solution at time t_{n-1}.

Convergence analysis II

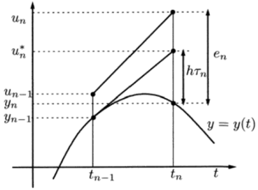


Figure: Geometrical representation

$$y_n - u_n^* = \frac{h^2}{2} y''(\xi_n)$$

- The local truncation error of the forward Euler method: $\tau_n(h) = (y_n - u_n^*)/h$.
- The global truncation error: $\tau(h) = \max_{n=0,\cdots N_h} |\tau_n(h)|$.
- For forward Euler: $\tau(h) = Mh/2$ where $M = \max_{t\in[t_0,T]} |y''(t)|$.
- The method is consistent $\lim_{h\to 0} \tau(h) = 0$.

Order of convergence

- The numerical process is an approximation of the mathematical model obtained as a function of a discretization parameter h.
- If the absolute or relative error is bounded as a function of h

$$e \leq ch^p$$

the method converges with order p.
- The errors $e_n \leq ch_n^p$ relative to the discretization parameter h_n, the order of convergence is estimated by

$$p_n = log(e_n/e_{i-1})/log(h_n/h_{i-1})$$

1. **Forward Euler Method**:

$$\boxed{u_{n+1} = u_n + hf_n, \quad n = 0, \cdots N_h - 1}$$

2. **Backward Euler Method**:

$$\boxed{u_{n+1} = u_n + hf_{n+1}, \quad n = 0, \cdots N_h - 1}$$

3. **The Crank-Nicolson method**: combine forward and backward Euler methods

$$\boxed{u_{n+1} = u_n + \frac{h}{2}[f_n + f_{n+1}], \quad n = 0, \cdots N_h - 1}$$

Hands-on session (Day-1)

1. Consider the Cauchy problem

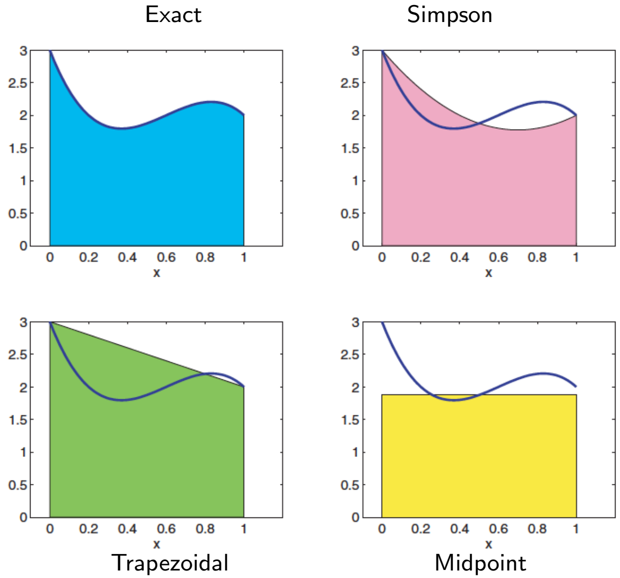
$$\begin{cases} y'(t) = \cos(2y(t)) & \forall t \in (0,1], \\ y(t_0) = 0 \end{cases}$$

The exact (analytic) solution is

$$y(t) = \frac{1}{2}\arcsin((e^{4t} - 1)/(e^{4t} + 1)).$$

2. Solve it by forward Euler method, backward Euler method and Crank-Nicolson using $h = 1/2, 1/4, 1/16, \cdots 1/512$.
3. Compute the absolute errors at the point $t = 1$ and store them in fe (forward Euler), be (backward Euler) and cn(Crank-Nicolson).
4. Show that the order of convergence 1, 1, and 2 for forward, backward Euler methods and Crank-Nicolson respectively.

Basic trapezoidal, midpoint and Simpson rules



Numerical Integration

Basic quadrature rules

$$I_f = \int_a^b f(x)dx \approx \sum_{j=0}^{n} a_j f(x_j)$$

- **Midpoint rule**

$$M = hf(\frac{a+b}{2})$$

- **Trapezoidal rule**

$$T = \frac{h}{2}(f(a) + f(b))$$

- **Simpson rule**

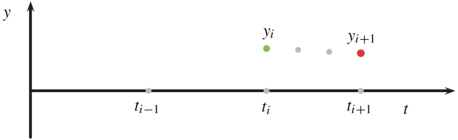
$$S = \frac{h}{6}\left(f(a) + 4f(\frac{a+b}{2}) + f(b)\right) = \frac{2}{3}M + \frac{1}{3}T$$

High order methods

- Euler method is only first order accurate.
- To obtain higher order of accuracy, Runge-Kutta (RK) methods and multistep methods
- RK method is one-step (from t_n to t_{n+1}) method in which repeated function evaluation are used to achieve a higher order.

$$u_{n+1} = u_n + h \sum_{i=1}^{s} b_n K_n$$

where $K_n = f(t_n + c_n h, u_n + h \sum_{j=1}^{s} a_{ij} K_j)$



RK-4 methods

- Integrate from t_n to t_{n+1}, the ODE

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt$$

- Numerical integration: Simpson quadrature rule four stages, explicit Simpson method

$$\begin{aligned}
K_1 &= f_n, \\
K_2 &= f(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_1), \\
K_3 &= f(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_2), \\
K_4 &= f(t_{n+1}, u_n + hK_3),
\end{aligned}$$

$$\boxed{u_{n+1} = u_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)}$$

Multistep Methods

- Achieve a high order of accuracy by involving the values u_n, u_{n-1}, u_{n-p} determining u_{n+1}.
- Three-step (p=2), third-order (explicit) Adams-Bashforth formula (AB3)

$$u_{n+1} = u_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2})$$

- Three-step, fourth-order (implicit) Adams-Moulton formula (AM4)

$$u_{n+1} = u_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2})$$

- Two-step, second-order (implicit) backward difference formula (BDF2)

$$u_{n+1} = \frac{4}{3}u_n - \frac{1}{3}u_{n-1} + \frac{2h}{3}f_{n+1}$$

- Three-step, third-order (implicit) backward difference formula (BDF3)

$$u_{n+1} = \frac{18}{11}u_n - \frac{9}{11}u_{n-1} + \frac{2}{11}u_{n-2} + \frac{6h}{11}f_{n+1}$$

RK method

- Integrate from t_n to t_{n+1}, the ODE

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt$$

- Numerical integration: Trapezoidal quadrature rule

$$u_{n+1} = u_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right)$$

  Implicit trapezoidal method
  (solving nonlinear equation, expensive and complicate)

- Approximate y_{n+1} by $u^*_{n+1} = u_n + hf(t_n, y_n)$, plug into the implicit trapezoidal formula

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_{n+1}, u^*_{n+1})\right)$$

  The explicit step (predictor), the implicit step (corrector). Heun method (Improved Euler method), RK2: use the first-order (explicit) forward Euler method to initialize Crank-Nicolson method.

RK-2 method

- Integrate from t_n to t_{n+1}, the ODE

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt$$

- Numerical integration: Midpoint quadrature rule

$$y_{n+1} = y_n + hf(t_{n+1/2}, y_{n+1/2})$$

Implicit midpoint method

$$t_{n+1/2} = \frac{t_n + t_{n+1}}{2} = t_n + h/2, \quad y_{n+1/2} = \frac{y_n + y_{n+1}}{2}$$

- Approximate $y_{n+1/2}$ yields two stage, explicit midpoint method

$$y_{n+1} = y_n + hf(t_{n+1/2}, Y), \quad Y = y_n + \frac{h}{2} f(t_n, y_n)$$

ODE in MATLAB

- **ode** followed by numbers and letters. The integration step varies in order to guarantee that the error remons below $RelTol = 10^{-3}$.
- **ode45** based on a pair of explicit RK methods (the Dormand-Prince pair)
- **ode23** based on a pair of explicit RK methods (the Bogacki and Shampine pair)
- **ode23tb** based on implicit RK whose first stage trapezoidal rule, second stage backward differentiation formula.
- **ode15s** multistep methods implementation
- **ode113** combined Adams-Bashforth-Moulton.

```
% y'=2t
tspan = [0 5];
y0 = 0;
[t,y] = ode23(@(t,y) 2*t, tspan, y0);
plot(t,y,'-o')
```

Systems of differential equations

$$\begin{cases} y_1'(t) = f_1(t, y_1, \cdots, y_m), \\ \vdots \\ y_m'(t) = f_m(t, y_1, \cdots, y_m), \end{cases}$$

Example

Population dynamics (P2.m)

$$\frac{dy_1}{dt} = C_1 y_1 \left(1 - b_1 y_1 - d_2 y_2\right),$$
$$\frac{dy_2}{dt} = -C_2 y_2 \left(1 - b_2 y_2 - d_1 y_1\right),$$

```
C1=1; C2=1; d1=1; d2=1; b1=0; b2=0;
[t,u] = feuler(@f,[0,10],[2 2],20000,C1,C2,d1,d2,b1,b2);
```

C_1, C_2: growth rates of the two populations
d_1, d_2: interaction between the two populations
b_1, b_2: available quantity of nutrients

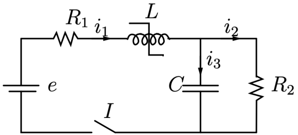
Example: Electrical circuits

$$\begin{cases} v'(t) = w(t), \\ w'(t) = -\frac{1}{LC}\left(\frac{L}{R} + RC\right) w(t) - \frac{2}{LC} v(t) + \frac{e}{LC}, \\ v(0) = w(0) = 0. \end{cases}$$

The system is obtained from the second-order differential equation

$$LC\frac{d^2v}{dt^2} + \left(\frac{L}{R_2} + R_1 C\right)\frac{dv}{dt} + \left(\frac{R_1}{R_2} + 1\right) v = e$$

Set $L = 0.1$ Henry (inductance), $C = 10^{-3}$ Farad (capacitance), $R = 10$ Ohm (resistance), and $e = 5$ Volt (voltage).



Day3-Presentation

10min presentations + 5min QAs

1. Begin with the presentation of a problem (ODE)
2. Introduce the algorithm(s) for solving it
3. Go through implementation of the algorithm. You can use your favorite programming language within C/C++/Fortran/Matlab/Python.
4. Questions & answers !