MS6021 Scientific Computing

Lecture/Work package #5

+

Assignment #5 (25%; due end-Week 13; recommended end-Week 12)

TOPICS:

A. Revision of Quadrature

B. Revision of ODEs

C. Parabolic PDEs (NEW)

A. Revision of Quadrature (see Part II of lecture 3)

Assignment #5, PART A (8%)

- Q1: Analytically evaluate the integral

$$\mathrm{I}(a) = \int_0^1 x^a (1 + 5x^4)\, dx \quad \text{for } a > -1$$

- Evaluate this integral numerically
  - Write a **Python** function trapezoidal (similar to your MatLab function of Ass.#3) and apply to the above integral with N=10, 100, 10000
  - Q2: Compare your numerical integrals I^{num} with the analytical result by completing the values $|I - I^{num}|$ in the table below.
  - Q3: What can one conclude from these results?? (You may plot $x^a(1 + 5x^4)$ for these 3 cases to understand/explain the observed behaviour.)
  - Q4: Next, write a **Python** function tailored (see Lecture #3 for the description of the method) and complete the corresponding values $|I - I^{num}|$ in the table below.

	a = 1.2	a = 0.3	a = -0.9
trapezoidal, N=10			
trapezoidal, N=100			
trapezoidal, N=10000			
tailored, N=10			
tailored, N=100			
tailored, N=10000			

B. Revision of ODEs (see Part I of lecture 4)

Assignment #5, PART B (8%)

- Implement, in Python, the explicit Euler method and the predictor-corrector method for Problem #1.
- Using the exact solution of Problem #1, compute the values max|error| and complete the Table:

	Prob#1, N=500	Prob#1, N=1000	Prob#1, N=2000
Explicit Euler			
Predictor-Corrector			

C. Parabolic PDEs

- Consider the simplest parabolic problem (for the heat equation):

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \qquad for \;\; 0 < x < 1, \qquad t > 0$$

$$u(x,0) = \sin(\pi x^2), \qquad u(0,t) = u(1,t) = 0$$

Here $u(x,t)$ is the unknown function for x in (0,1) and t>0,
with $\frac{\partial}{\partial t}$ and $\frac{\partial}{\partial x}$ denoting partial derivatives w.r.t. t and x respectively.

- This equation can be discretised in 2 steps:
  - Step 1: discretize in space as in Ass.#1, dividing the space interval (0,1) into N equal subintervals….
  - Step 2: now solve as a system of (N-1) ODEs for $U_i(t)$, i=2,3,…,N-1,e.g., using the explicit Euler method.

- Then one arrives at the discrete problem with time step τ and h=1/N:

$$\frac{U_n^{k+1} - U_n^{k}}{\tau} = \frac{U_{n+1}^{k} - 2U_n^{k} + U_{n-1}^{k}}{h^2}, \qquad k = 1,2,\dots K,$$

$$n = 1,\dots,N-1,$$

where U_n^{k} is the computed solution associated with x_n=nh, and t^k= τk.
Initial and boundary conditions: U_n^{0} = $\sin(\pi x_n^{2})$, and U_0^{k}= U_N^{k}=0.

C. Parabolic PDEs (continued)

Assignment #5, PART C (9%)

- Q1: Implement the above method with the following parameters and make a conclusion whether the resulting method is STABLE/UNSTABLE in each case by completing the following table: K is such that the final time is 1,

	$\tau = h^2/4$	$\tau = h^2/2$	$\tau = h^2$	$\tau = 2h^2$	$\tau = 4h^2$
N=20					
N=40					
N=200					

- Q2: Based on the above results, make a recommendation on the choice of τ for this method.

- You may start your code as follows:

```
from numpy import *
import matplotlib.pyplot as plt
N = 20
K = ...
h = 1./N
tau = 1./K
x=linspace(0,1,N+1)
y=sin(x*pi)
plt.plot(x, y, '-o')
plt.hold(True)
```

Concluding Remarks: What we did NOT consider? – A lot!

- Finite Element Methods (FEM) :
  - a powerful and flexible class of methods for the numerical solution of PDEs
  - easily deal with complicated domains and local solution singularities
- Hyperbolic PDEs (transport equations, conservation laws)
- Stochastic differential equations
- Data … (fitting…)

- Object-Oriented programming