KUTAISI INTERNATIONAL UNIVERSITY

Numerical Programming

Ramaz Botchorishvili

Kutaisi International University

AP#8

AP#8:

Problem 8.1

1. Find a system of ODEs describing problems from biology, finances, engineering etc.
2. Select linear multistep method(s) or RK method(s)
3. Select adaptive step size control procedure
4. Solve the system of ODEs using selected numerical method(s) with adaptive step size control
5. Perform numerical experiments and visualize

Tasks and sub-problems to consider

- Using methods and models from previous APs are accepted.
- Post your model and method in the Teams chat.
- Remember the system should contain more than one equations.
- Describe your model in written, give reference to the source.
- Set up numerical experiments and test the method. Give detailed

AP#8

Important Notice

- AP assigned 0 points if:
  - a model problem (ODEs, image or video etc.) provided twice by students. Make sure, your model is different from models given by others.
  - submitted results are not reproducible.
  - student cannot apply his own code for the input data provided by TA or instructor.
  - AP is submitted without written explanation of methods and approaches used.
- Submission deadline: 8 days after the date of AP publication.

Adaptive stepsize control

- Embedded Runge-Kutta Methods:

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error
  - Use local trucation error for step size calculation

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error
  - Use local trucation error for step size calculation
  - Examples: Fehlberg method, Dormand-Prince methods.

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error
  - Use local trucation error for step size calculation
  - Examples: Fehlberg method, Dormand-Prince methods.
- Step-halving Techniques:

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error
  - Use local trucation error for step size calculation
  - Examples: Fehlberg method, Dormand-Prince methods.
- Step-halving Techniques:
  - find solution using stepsize h and $h/2$

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error
  - Use local trucation error for step size calculation
  - Examples: Fehlberg method, Dormand-Prince methods.
- Step-halving Techniques:
  - find solution using stepsize h and $h/2$
  - Estimate the local error

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error
  - Use local trucation error for step size calculation
  - Examples: Fehlberg method, Dormand-Prince methods.
- Step-halving Techniques:
  - find solution using stepsize h and $h/2$
  - Estimate the local error
  - Use local error for step size calculation

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error
  - Use local trucation error for step size calculation
  - Examples: Fehlberg method, Dormand-Prince methods.
- Step-halving Techniques:
  - find solution using stepsize h and $h/2$
  - Estimate the local error
  - Use local error for step size calculation
  - update solution using new step size

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error
  - Use local trucation error for step size calculation
  - Examples: Fehlberg method, Dormand-Prince methods.
- Step-halving Techniques:
  - find solution using stepsize h and $h/2$
  - Estimate the local error
  - Use local error for step size calculation
  - update solution using new step size
- General approach for the step size update:

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error
  - Use local trucation error for step size calculation
  - Examples: Fehlberg method, Dormand-Prince methods.
- Step-halving Techniques:
  - find solution using stepsize h and $h/2$
  - Estimate the local error
  - Use local error for step size calculation
  - update solution using new step size
- General approach for the step size update:
  - if the error is too large, reduce the step size;

Adaptive stepsize control

- Embedded Runge-Kutta Methods:
  - Compute two solutions of different orders simultaneously
  - Estimate of the local truncation error
  - Use local trucation error for step size calculation
  - Examples: Fehlberg method, Dormand-Prince methods.
- Step-halving Techniques:
  - find solution using stepsize h and $h/2$
  - Estimate the local error
  - Use local error for step size calculation
  - update solution using new step size
- General approach for the step size update:
  - if the error is too large, reduce the step size;
  - if the error is too small, increase the step size.