Solution of Initial Value Problems for ODEs

1. Write a general computer code to solve systems of up to five coupled first order initial value problems using the Heun and Newton Iteration trapezoidal methods. These are to be combined in the same algorithm in such a way that Heun's method is automatically employed to provide the initial guess at each time step for the Newton Iteration required by fully implicit trapezoidal integration. Otherwise, the time stepping is done only with the explicit Heun's method, and no iterations are needed. The complete pseudo-language algorithm is attached.

Test this code by solving the following problem.

$$\frac{d^2u}{dt^2} + 2\frac{du}{dt} + 4u = 0 \qquad t \in [0,6]$$

$$u(0) = 2, \quad \frac{du}{dt}(0) = 0$$

Use step sizes h = 0.1, 0.05 and 0.025. Solve the problem first with the explicit Heun's method, and then with implicit Newton Iteration integration for each value of h. Employ a convergence tolerance $\varepsilon = 10^{-6}$ for the Newton Iterations of the trapezoidal method. The exact solution to this problem is

$$u(t) = 2e^{-t}\left(\cos(\sqrt{3}\,t) + \frac{1}{\sqrt{3}}\sin(\sqrt{3}t)\right)$$

Make a table of results of convergence tests (based on the exact solution) at t = 1, 2, 3, 4, 5 & 6. This table should include the exact solution value, the computed solution, the error and the error ratios from successive step sizes for each of the required values of t. Discuss how these results compare with theory. Also discuss what factors should influence the choice of convergence tolerance ε for the Newton iterations, and whether the specified value given above is appropriate.

2. Solve the following problem using only the Newton Iteration trapezoidal method.

$$\frac{d^3u}{dt^3} + u\frac{d^2u}{dt^2} - \left(\frac{du}{dt}\right)^2 + 2u = 10\sin 6t \qquad t \in [0,2]$$

$$u(0) = \frac{du}{dt}(0) = \frac{d^2u}{dt^2}(0) = 0$$

Employ step sizes h = 0.1, 0.05, 0.01 and an iteration convergence tolerance $\varepsilon = 10^{-6}$. Consider the h = 0.01 solution to be "exact" in order to carry out convergence tests between the h = 0.1 and h = 0.05 solutions at t = 0.5, 1.0, 1.5 and 2.0. Make a table, similar to the table in Prob. 1, summarizing results of the convergence tests.

Simultaneous ODE's

$\Delta u = u_{calc} - u_{exact}$

Given multiple differential equations, f_i where $f_i = \frac{du_i}{dt}$

error $\Delta u(0.05/0.1)$
error $\Delta u(0.025/0.05)$

$u_{calc} \rightarrow u_i$

Use the trapezoidal rule to integrate

$$u_i = u_{old,i} + \frac{h}{2}(fold_i + f_i) \quad (1)$$

where $uold_i$ is the u_i at the beginning of the time step

h is the time step

$fold_i = f_i|_{told}$ is the derivative at the beginning of time step

f_i is the derivative at the end of the time step

Since u_i & f_i depend on each other, we must iterate

When u_i & f_i have been correctly evaluated, eqn (1) becomes

$$0 = u_i - h/2 f_i - (uold_i + h/2 fold_i) \quad (1a)$$

However, when u_i & f_i are not known, define a function, F_i where

$$F_i = u_i - h/2 f_i - (uold_i + h/2 fold_i) \quad (2)$$

Note that when u_i & f_i converge, $F_i \rightarrow 0$

The Taylor-Series expansion for a function of multiple variables,

$$F_i(\bar{u}_1, \bar{u}_2 \ldots) = F_i(u_1, u_2 \ldots) + \frac{\partial F_i}{\partial u_1}(\bar{u}_1 - u_1) + \frac{\partial F_i}{\partial u_2}(\bar{u}_2 - u_2) \ldots$$

where $\bar{u}_i$ are the converged solutions

u_i are the approximate solutions

$\Delta u_i = \bar{u}_i - u_i$

$F_i(\bar{u}_1, \bar{u}_2 \ldots \bar{u}_{neqns}) = 0$

$$\therefore \sum_{j=1}^{neqns} \frac{\partial F_i}{\partial u_j} \Delta u_j = -F_i(u_1, u_2, \ldots u_{neqns}) \text{ for } i = 1 \rightarrow neqns \quad (3)$$

Simultaneous ODE's (page 2/2)

Eqn (3) in matrix form:

$$\begin{bmatrix} \frac{\partial F_1}{\partial u_1} & \frac{\partial F_1}{\partial u_2} \cdots & \frac{\partial F_1}{\partial u_{neqns}} \\ \frac{\partial F_2}{\partial u_1} & \frac{\partial F_2}{\partial u_2} \cdots & \frac{\partial F_2}{\partial u_{neqns}} \\ \vdots & & \\ \frac{\partial F_{neqns}}{\partial u_1} & \frac{\partial F_{neqns}}{\partial u_2} \cdots & \frac{\partial F_{neqns}}{\partial u_{neqns}} \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \vdots \\ \Delta u_{neqns} \end{bmatrix} = \begin{bmatrix} -F_1 \\ -F_2 \\ \vdots \\ -F_{neqns} \end{bmatrix}$$

or $[J(F)_{ij}][\Delta u_j] = [-F_i]$ - evaluate Jacobian, $J(F)_{ij}$ & F_i based on best values of u_i & f_i
- use Gauss Elim. to solve for Δu_i
- update u_i where $u_i^{(m)} = u_i^{(m-1)} + \Delta u_i^{(m)}$

where (m) is the iteration number

step 6

To calculate the Jacobian of (2)

$$\frac{\partial}{\partial u_j}\left[F_i = u_i - h/2\, f_i - (uold_i + h/2\, fold_i)\right]$$

$$\frac{\partial F_i}{\partial u_j} = J(F)_{ij} = \frac{\partial u_i}{\partial u_j} - h/2 \frac{\partial f_i}{\partial u_j}$$

Note: $uold_i$ & $fold_i$ don't change as u_i changes

where $\frac{\partial u_i}{\partial u_j} = 1$ when $i=j$; $= 0$ when $i \neq j$ $\Big\} = \delta_{ij}$

step 7

$\therefore J(F)_{ij} = \delta_{ij} - h/2\, J(f)_{ij}$ (4)

- load $[J(f)_{ij}]$ into the coefficient matrix $[A_{ij}]$
- load $[-F_i]$ into the $A_{i,neqns+1}$ column of the A-array
- update coefficient matrix using eqn (4) $A_{ij} = \delta_{ij} - \frac{h}{2} A_{ij}$

  i.e. $J(F)_{ij} = \delta_{ij} - h/2\, J(f)_{ij}$
- solve for Δu_j using Gaussian Elimination
- update $u_i^{(m)} = u_i^{(m-1)} + \Delta u_i^{(m)}$
- if $\Delta u_i < 10^{-6}$ => converged ; if not, repeat iteration