ME 704

Computational Methods in Thermal and Fluids Engineering

(Ordinary Differential Equations-2)

Kannan Iyer
Kiyer@iitb.ac.in

Department of Mechanical Engineering
Indian Institute of Technology, Bombay

Runge-Kutta Methods

- From two function evaluations, we can go to several functional evaluation methods which improve accuracy.

$$y_{n+1} = y_n + h\sum_{i=1}^{r}\gamma_i k_i, \quad with \quad \sum_{i=1}^{r}\gamma_i = 1$$

Note that k_is are the slopes or $f(x_i, y_i)$ evaluated at several is

- The values of (x_i, y_i) are chosen appropriately

Runge-Kutta Methods (cont'd)

- In general

$$k_1 = f(x_n, y_n)$$

$$k_i = f\left((x_n + h\alpha_i), (y_n + h\sum_{j=1}^{i-1}\beta_{i,j}k_{j-1})\right)$$

$$y_{n+1} = y_n + h\sum_{i=1}^{r}\gamma_i k_i, \quad with \quad \sum_{i=1}^{r}\gamma_i = 1$$

- The coefficients α, β and γ are obtained using Taylor Series

Runge-Kutta Fourth Order Method

$$\begin{array}{c|cccc}
\alpha_i & & \beta_{i,j} & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
\gamma_i & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6}
\end{array}$$

$$k_1 = f(x_n, y_n)$$
$$k_2 = f\left(x_n + 0.5h, y_n + h(0.5k_1)\right)$$
$$k_3 = f\left(x_n + 0.5h, y_n + h(0.5k_2)\right)$$
$$k_4 = f\left(x_n + h, y_n + hk_3\right)$$
$$y^{n+1} = y^n + h/6\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

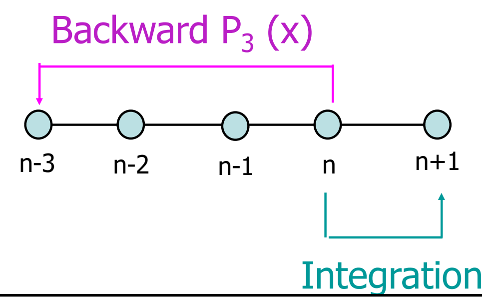
- A large variety of methods upto sixth order global accuracy are available (Refer Numerical Solution of ODE by M.K. Jain, Wiley Eastern, 1987)

Multi-step Methods

- Till now we have marched one-step at a time which involves typically n functional evaluations for the n^{th} order method
  - The question is whether we can have the same order of accuracy using fewer functional evaluations?
  - Multi-step methods precisely accomplish this

Adams Method

$$\frac{dy}{dx} = f(x,y) \quad \Rightarrow \int_{y_n}^{y_{n+1}} dy = \int_{x_n}^{x_{n+1}} f(x,y)dx$$



Adams Method (Cont'd)

$$P_3(x) = f(x_n) + s\nabla f(x_n) + \frac{(s)(s+1)}{2!}\nabla^2 f(x_n) + \frac{(s)(s+1)(s+2)}{3!}\nabla^3 f(x_n) + sC4h^4 f^{IV}(\xi)$$

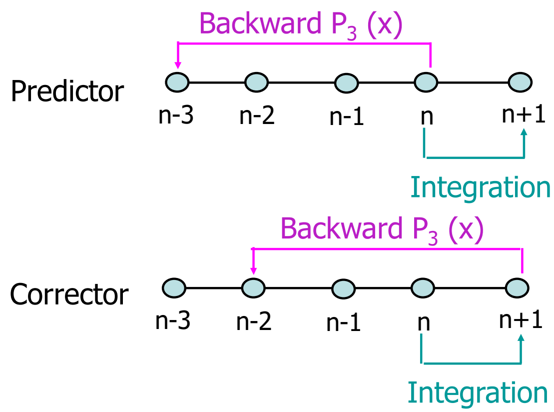
$$y_{n+1} - y_n = \int_0^1 h \begin{pmatrix} f(x_n) + s\nabla f(x_n) + \dfrac{(s)(s+1)}{2!}\nabla^2 f(x_n) \\ + \dfrac{(s)(s+1)(s+2)}{3!}\nabla^3 f(x_n) + sC4h^4 f^{IV}(\xi) \end{pmatrix} ds$$

Adams Method (Cont'd)

x	f	∇f	$\nabla^2 f$	$\nabla^3 f$
x_o	f_0	--	--	--
x_1	f_1	(f_1-f_0)	--	--
x_2	f_2	(f_2-f_1)	$(f_2-2f_1+f_0)$	--
x_3	f_3	(f_3-f_2)	$(f_3-2f_2+f_1)$	$(f_3-3f_2+3f_1-f_0)$

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) + \frac{251}{720}h^5 y^V(\xi)$$

Adams-Moulton Method



Adams-Moulton Method (Cont'd)

Predictor

$$y^P_{n+1} = y_n + \frac{h}{24}\left(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}\right) + \frac{251}{720}h^5 y^V(\xi)$$

Corrector

$$y^C_{n+1} = y_n + \frac{h}{24}\left(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}\right) - \frac{19}{720}h^5 y^V(\xi)$$

A-M with Error Correction

$$\bar{y}_{n+1} = y^P_{n+1} + \frac{251}{720}h^5 y^V(\xi)$$

$$\bar{y}_{n+1} = y^C_{n+1} - \frac{19}{720}h^5 y^V(\xi)$$

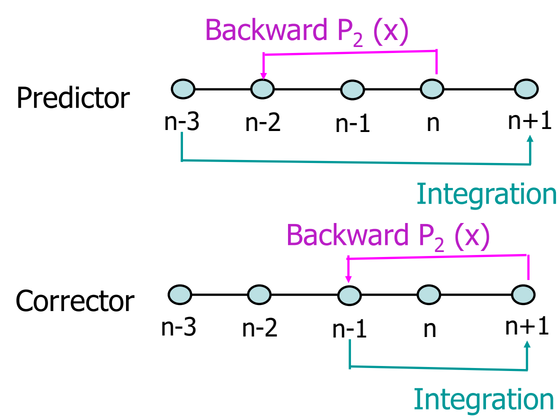
$$\therefore y^C_{n+1} - y^P_{n+1} = h^5 y^V(\xi)\left(\frac{19+251}{720}\right)$$

$$\therefore \left(y^C_{n+1} - y^P_{n+1}\right) = h^5 y^V(\xi)\left(\frac{270}{720} X \frac{19}{19}\right)$$

$$\therefore \left(y^C_{n+1} - y^P_{n+1}\right)\frac{19}{270} = h^5 y^V(\xi)\left(\frac{19}{720}\right)$$

$$y^{C,Mop}_{n+1} = y^C_{n+1} - \left(\frac{19}{270}\right)\left(y^C_{n+1} - y^P_{n+1}\right)$$

Milne's Method



Milne's Method (Cont'd)

Predictor

$$y_{n+1}^{P} = y_{n-3} + \frac{4h}{3}\left(2f_n - f_{n-1} + 2f_{n-2}\right) + \frac{28}{90}h^5 y^V(\xi)$$

Corrector

$$y_{n+1}^{C} = y_{n-1} + \frac{h}{3}\left(f_{n+1} + 4f_n + f_{n-1}\right) - \frac{1}{90}h^5 y^V(\xi)$$

General Comments

- ❑ Multi-step methods are efficient as they can give higher order accuracy with just one function evaluation
- ❑ Milne's method had simple coefficients, but has stability issues
- ❑ Adam's Moulton is the most preferred among multi-step methods
- ❑ Error estimated with the method can be used to correct and mop the error
- ❑ However, explicit methods like R-K methods can be used to adapt and control errors

Method for Error Control

- ❑ We have seen earlier that for stability of the algorithms, step size has to be controlled
- ❑ Establishing stability limits for higher order methods is laborious.
- ❑ These have been done, but rarely are they applied as many times the accuracy overrides stability
- ❑ Usually error control is established by choosing adaptive methods which chooses h automatically.
- ❑ R-K methods are most suited for adaptive algorithms as these are one-step methods

Error Control (Cont'd)

- ❑ If the magnitude of tolerable error is known, then the step size can be reduced till the estimated error is smaller than the acceptable error.
- ❑ If during later part of the computation, the error is too small, then the step size can be doubled.

Error Control

- We have seen that every numerical method has an associated error
- The errors are of two types
- The truncation error is associated with truncating the Taylor series to finite number of terms
- The roundoff error is associated with the limitied digits the computers work with
- The final error is a combination of both the errors.
- Estimation of truncation errors have been presented earlier and we shall visit them again
- The roundoff errors can amplify and have to be controlled by using stable methods
- Arriving at stability criterion can be laborious
- Further, the step size will vary from problem to problem and specifying time steps apriori is difficult

Error Control

- The multistep methods have a problem in adjusting the step size as the formulae are based on constant step sizes
- Though they are inferior, as they cannot adapt without interpolation, they are still used by many for constant step sizes.
- The predictor-corrector methods can estimate the error and this can be exploited.
- We had seen the Adams-Moulton method as

Predictor

$$y_{n+1}^P = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) + \frac{251}{720}h^5 y^V(\xi)$$

Corrector

$$y_{n+1}^C = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) - \frac{19}{720}h^5 y^V(\xi)$$

Error Control (Cont'd)

$$\bar{y}_{n+1} = y_{n+1}^P + \frac{251}{720}h^5 y^V(\xi)$$

$$\bar{y}_{n+1} = y_{n+1}^C - \frac{19}{720}h^5 y^V(\xi)$$

$$\therefore y_{n+1}^C - y_{n+1}^P = h^5 y^V(\xi)\left(\frac{19+251}{720}\right)$$

$$y_{n+1}^{C,Mop} = y_{n+1}^C + \left(\frac{19}{19+251}\right)\left(y_{n+1}^C - y_{n+1}^P\right)$$

Error Control (Cont'd)

- If the error is too high, then the step size is reduced
- However, once the step is reduced, the method has to be started all over again. Interpolation may be used for generating necessary steps for higher order methods
- The best approach is to use single step method like RK method and adapt accordingly.
- The most popular approach is to use RK-4 and computation is carried out twice
- Once a step of h is taken and then the same is repeated with two steps of h/2 and error is estimated as follows

$$y_{exact} = y_{N\text{-}h} + C\,h^5 \qquad (1)$$

$$y_{exact} = y_{N\text{-}0.5h} + 2C\,(h/2)^5 \qquad (2)$$

- Eq 1 – Eq 2 gives

$$0 = y_{N\text{-}h} - y_{N\text{-}0.5h} + C\,h^5(1-1/16)$$

Error Control (Cont'd)

$\Rightarrow y_{N-0.5h} - y_{N-h} = (15/16)C\,h^5$

$\Rightarrow (y_{N-0.5h} - y_{N-h})/15 = Ch^5/16$

- ❑ Thus the error is estimated and if this is less than tolerance/16, we can double step size
- ❑ If error is more than the tolerance, the step size shall be reduced by a factor of two
- ❑ Usually a factor of 1.5 to 2 is used as safety to prevent oscillation of the method. Thus the criterion for doubling is error < Tol/(16*safety)

Error Control (Cont'd)

- ❑ Often it is better to specify the tolerance on nomalized values of y
- ❑ The best way is to divide the error by y and specify a tolerance for this, say 1e-5
- ❑ This will have a problem if y crosses zero
- ❑ The alternative is to define y_{scale} as

$$y_{scale} = |y| + \left| h\frac{dy}{dx} \right|$$

- ❑ Since dy/dx = f(x,y) is the function value that would have been estimated, y_{scale} can be obtained