MA 214: Introduction to numerical analysis

Lecture 33

Shripad M. Garge.
IIT Bombay

(shripad@math.iitb.ac.in)

2021-2022

Runge-Kutta Methods of Order Two

This is a difference method to solve the initial-value problem

$$\frac{dy}{dt} = f(t, y), \quad a \leqslant t \leqslant b, \qquad y(a) = \alpha.$$

It gives $w_0 = \alpha$ and

$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2} f(t_i, w_i)\right)$$

for $i = 0, \ldots, N-1$. This was obtained by equating $T(t, y)$ with $a_1 f(t + \alpha_1, y + \beta_1)$ and then solving for a_1, α_1 and β_1.

The term $a_1 f(t + \alpha_1, y + \beta_1)$ has three terms and they were all needed to get the match with T.

So a more complicated form is required to satisfy the conditions for any of the higher-order Taylor methods.

Runge-Kutta Methods of Order Two

If we use the form $a_1 f(t,y) + a_2 f(t+\alpha_2, y+\delta_2 f(t,y))$ containing 4 parameters to approximate

$$T(t,y) = f(t,y) + \frac{h}{2} f'(t,y) + \frac{h^2}{6} f''(t,y)$$

then there is insufficient flexibility to match the term

$$\frac{h^2}{6}\left[\frac{\partial f}{\partial y}(t,y)\right]^2 f(t,y)$$

resulting from the expansion of $(h^2/6)f''(t,y)$.

Consequently, the best that can be obtained from the 4-parameter form are methods with $O(h^2)$ local truncation error.

Runge-Kutta Methods of Order Two

Since four parameters are available, it gives a flexibility in their choice, so a number of $O(h^2)$ methods can be derived.

One of the most important ones is the Modified Euler method, which corresponds to choosing $a_1 = a_2 = 1/2$ and $\alpha_2 = \delta_2 = h$.

It has the following difference-equation form: $w_0 = \alpha$ and

$$w_{i+1} = w_i + \frac{h}{2}[f(t_i, w_i) + f(t_{i+1}, w_i + hf(t_i, w_i))]$$

This method differs slightly from the Euler method, hence the name.

Let us do an example.

An example

We apply the modified Euler method with $N = 10$, $h = 0.2$, $t_i = 0.2i$ and $w_0 = 0.5$ to approximate the solution to our usual example

$$y' = y - t^2 + 1, \quad 0 \leqslant t \leqslant 2, \qquad y(0) = 0.5.$$

The method gives $w_0 = 0.5$ and

$$w_{i+1} = w_i + 1.22w_i - 0.0088i^2 - 0.008i + 0.216$$

for $i = 0, \ldots, 9$.

The first two approximations are

$$w_1 = 0.826 \quad \text{and} \quad w_2 = 1.20692.$$

The table follows.

The example, continued

t_i	$y(t_i)$	Midpoint Method	Error	Modified Euler Method	Error
0.0	0.5000000	0.5000000	0	0.5000000	0
0.2	0.8292986	0.8280000	0.0012986	0.8260000	0.0032986
0.4	1.2140877	1.2113600	0.0027277	1.2069200	0.0071677
0.6	1.6489406	1.6446592	0.0042814	1.6372424	0.0116982
0.8	2.1272295	2.1212842	0.0059453	2.1102357	0.0169938
1.0	2.6408591	2.6331668	0.0076923	2.6176876	0.0231715
1.2	3.1799415	3.1704634	0.0094781	3.1495789	0.0303627
1.4	3.7324000	3.7211654	0.0112346	3.6936862	0.0387138
1.6	4.2834838	4.2706218	0.0128620	4.2350972	0.0483866
1.8	4.8151763	4.8009586	0.0142177	4.7556185	0.0595577
2.0	5.3054720	5.2903695	0.0151025	5.2330546	0.0724173

Now we will approximate

$$T(t,y) = f(t,y) + \frac{h}{2}f'(t,y) + \frac{h^2}{6}f''(t,y)$$

by $f(t+\alpha_1, y+\delta_1 f(t+\alpha_2, y+\delta_2 f(t,y)))$ with error $O(h^3)$.

The number of parameters involved is 4 and the derivation of these constants is a bit involved.

The most common method is Heun's method.

The difference-equation of Heun's method is

$$w_{i+1} = w_i + \frac{h}{4}\left[f(t_i,w_i) + 3f\left(t_i + \frac{2h}{3}, w_i + \frac{2h}{3}f\left(t_i + \frac{h}{3}f(t_i,w_i)\right)\right)\right].$$

Our usual example

We apply Heun's method with $N = 10$, $h = 0.2$, $t_i = 0.2i$ and $w_0 = 0.5$ to approximate the solution to our usual example

$$y' = y - t^2 + 1, \quad 0 \leqslant t \leqslant 2, \qquad y(0) = 0.5.$$

The method gives $w_0 = 0.5$ and

$$\begin{aligned} w_{i+1} &= w_i + (0.05)\Big[f(0.2i, w_i) + \\ &\qquad 3\, f\Big(0.2i + 1.\bar{3}, w_i + 1.\bar{3}\; f\big(0.2i + 0.0\bar{6}\; f(0.2i, w_i)\big)\Big)\Big]. \end{aligned}$$

The first two values are

$$w_1 = 0.829244 \quad \text{and} \quad w_2 = 1.2139750.$$

The table follows.

The example, continued

t_i	$y(t_i)$	Heun's Method	Error
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292444	0.0000542
0.4	1.2140877	1.2139750	0.0001127
0.6	1.6489406	1.6487659	0.0001747
0.8	2.1272295	2.1269905	0.0002390
1.0	2.6408591	2.6405555	0.0003035
1.2	3.1799415	3.1795763	0.0003653
1.4	3.7324000	3.7319803	0.0004197
1.6	4.2834838	4.2830230	0.0004608
1.8	4.8151763	4.8146966	0.0004797
2.0	5.3054720	5.3050072	0.0004648

Runge-Kutta of order four

This is the most common Runge-Kutta method. It is given by $w_0 = \alpha$ and $w_{i+1} = w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ where

$$k_1 = hf(t_i, w_i), \qquad k_2 = hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_1\right),$$

$$k_3 = hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_2\right) \text{ and } k_4 = hf(t_{i+1}, w_i + k_3).$$

This method has local truncation error $O(h^4)$, hence the order of this method is four, provided the solution $y(t)$ has five continuous derivatives.

The notations k_1, k_2, k_3, k_4 are introduced to eliminate the need for successive nesting in the second variable of $f(t, y)$. Let us do our usual example.

Our usual example

We consider the usual initial-value problem:

$$y' = y - t^2 + 1, \quad 0 \leqslant t \leqslant 2, \qquad y(0) = 0.5$$

and approximate its solution by Runge-Kutta of order four with $N = 10$, $h = 0.2$ and $t_i = 0.2i$.

We first compute k_1, k_2, k_3 and k_4:

$$\begin{aligned} k_1 &= 0.2f(0, 0.5) &&= 0.2(1.5) = 0.3, \\ k_2 &= 0.2f(0.1, 0.65) &&= 0.328, \\ k_3 &= 0.2f(0.1, 0.664) &&= 0.3308, \\ k_4 &= 0.2f(0.2, 0.8308) &&= 0.35816. \end{aligned}$$

We have

$$w_1 = 0.5 + \frac{1}{6}(0.3 + 2(0.328) + 2(0.3308) + 0.35816) = 0.8292933.$$

The example, continued

t_i	Exact $y_i = y(t_i)$	Runge-Kutta Order Four w_i	Error $\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292933	0.0000053
0.4	1.2140877	1.2140762	0.0000114
0.6	1.6489406	1.6489220	0.0000186
0.8	2.1272295	2.1272027	0.0000269
1.0	2.6408591	2.6408227	0.0000364
1.2	3.1799415	3.1798942	0.0000474
1.4	3.7324000	3.7323401	0.0000599
1.6	4.2834838	4.2834095	0.0000743
1.8	4.8151763	4.8150857	0.0000906
2.0	5.3054720	5.3053630	0.0001089

Computational Comparisons

The main computational effort in applying the Runge-Kutta methods is the evaluation of f.

In the second-order methods, the local truncation error is $O(h^2)$, and the cost is two function evaluations per step.

The Runge-Kutta method of order four requires 4 evaluations per step, and the local truncation error is $O(h^4)$.

The following table establishes the relationship between the number of evaluations per step and the order of the local truncation error.

Computational Comparisons

Evaluations per step	Best possible local truncation error
2	$O(h^2)$
3	$O(h^3)$
4	$O(h^4)$
$5 \leqslant n \leqslant 7$	$O(h^{n-1})$
$8 \leqslant n \leqslant 9$	$O(h^{n-2})$
$10 \leqslant n$	$O(h^{n-3})$

This table indicates why the methods of order less than five with smaller step size are used in preference to the higher-order methods using a larger step size.

Computational Comparisons

The Runge-Kutta method of order four requires four evaluations per step, whereas Euler's method requires only one evaluation.

Hence if the Runge-Kutta method of order four is to be superior it should give more accurate answers than Euler's method with one-fourth of the step size.

Similarly, if the Runge-Kutta method of order four is to be superior to the second-order Runge-Kutta methods, which require two evaluations per step, it should give more accuracy with step size h than a second-order method with step size $h/2$.

The following example illustrates the superiority of the Runge-Kutta fourth-order method by this measure for our usual initial-value problem.

Computational Comparisons

We consider the problem

$$y' = y - t^2 + 1, \quad 0 \leqslant t \leqslant 2, \qquad y(0) = 0.5.$$

We employ Euler's method with $h = 0.025$, the modified Euler method with $h = 0.05$, and the Runge-Kutta fourth-order method with $h = 0.1$.

These methods are compared at the common mesh points of these methods: 0.1, 0.2, 0.3, 0.4, and 0.5.

Each of these techniques requires 20 function evaluations to determine the values listed in the next table to approximate $y(0.5)$.

We see that the fourth-order method is clearly superior.

Computational Comparisons

t_i	Exact	Euler $h = 0.025$	Modified Euler $h = 0.05$	Runge-Kutta Order Four $h = 0.1$
0.0	0.5000000	0.5000000	0.5000000	0.5000000
0.1	0.6574145	0.6554982	0.6573085	0.6574144
0.2	0.8292986	0.8253385	0.8290778	0.8292983
0.3	1.0150706	1.0089334	1.0147254	1.0150701
0.4	1.2140877	1.2056345	1.2136079	1.2140869
0.5	1.4256394	1.4147264	1.4250141	1.4256384

MA 214: Introduction to numerical analysis
Lecture 34

Shripad M. Garge.
IIT Bombay

(shripad@math.iitb.ac.in)

2021-2022

Error control in Runge-Kutta methods

In lecture 25, we studied the adaptive quadrature methods to approximate definite integrals.

These methods adapt the number and position of the nodes used in the approximation to keep the truncation error within a specific bound, in spite of an increase in the number of calculations.

The problem of approximating the value of a definite integral and that of approximating the solution to an initial-value problem are similar.

It is not surprising, then, that there are adaptive methods for approximating the solutions to initial-value problems and that these methods are not only efficient, but also incorporate the control of error.

Error control in Runge-Kutta methods

Any one-step method for approximating the solution, $y(t)$, of the initial-value problem

$$y' = f(t, y), \quad a \leqslant t \leqslant b, \qquad y(a) = \alpha$$

can be expressed in the form

$$w_{i+1} = w_i + h_i \phi(t_i, w_i, h_i)$$

for $i = 0, \ldots, N-1$ and some function ϕ.

An ideal such method will have the property that, given $\epsilon > 0$, a minimal number of mesh points could be used to ensure that the global error, $|y(t_i) - w_i|$, remains less than ϵ for every i.

In such a method, it will be natural to not have equally-spaced nodes.

Error control in Runge-Kutta methods

We now examine techniques used to control the error of a difference-equation method in an efficient manner by the appropriate choice of mesh points.

By using methods of differing order we can predict the local truncation error and, using this prediction, choose a step size that will keep it and the global error in check. We illustrate the technique by comparing two approximation techniques:

The first is obtained from an n-th order Taylor method of the form

$$y(t_{i+1}) = y(t_i) + h\phi(t_i, y(t_i), h) + O(h^{n+1})$$

and the second is obtained from a Taylor method of order $(n+1)$ of the form

$$y(t_{i+1}) = y(t_i) + h\tilde{\phi}(t_i, y(t_i), h) + O(h^{n+2}).$$

Error control in Runge-Kutta methods

The difference methods are given by $w_0 = \alpha = \tilde{w}_0$ and for $i \geqslant 0$

$$w_{i+1} = w_i + h\phi(t_i, w_i, h) \quad \text{and} \quad \tilde{w}_{i+1} = \tilde{w}_i + h\tilde{\phi}(t_i, \tilde{w}_i, h)$$

with respective local truncation errors: $\tau_{i+1}(h) = O(h^n)$ and $\tilde{\tau}_{i+1}(h) = O(h^{n+1})$.

We first make the assumption that $w_i \approx y(ti) \approx \tilde{w}_i$ and generate the approximations w_{i+1} and $\tilde{w}_{i+1}$ to $y(t_{i+1})$ with the same h. Then

$$\begin{aligned} \tau_{i+1}(h) &= \frac{y(t_{i+1}) - y(t_i)}{h} - \phi(t_i, y(t_i), h) \\ &\approx \frac{y(t_{i+1}) - w_i}{h} - \phi(t_i, w_i, h) \\ &= \frac{1}{h}\big(y(t_{i+1}) - w_{i+1}\big) \end{aligned}$$

Error control in Runge-Kutta methods

Hence

$$\tau_{i+1}(h) \approx \frac{1}{h}\big(y(t_{i+1}) - w_{i+1}\big)$$

and similarly

$$\tilde{\tau}_{i+1}(h) \approx \frac{1}{h}\big(y(t_{i+1}) - \tilde{w}_{i+1}\big).$$

Therefore

$$\tau_{i+1}(h) - \tilde{\tau}_{i+1}(h) \approx \frac{1}{h}\big(\tilde{w}_{i+1} - w_{i+1}\big)$$

The orders of these errors are $O(h^n)$ and $O(h^{n+1})$, respectively. Hence the significant part of $\tau_{i+1}(h)$ must come from

$$\frac{1}{h}\big(\tilde{w}_{i+1} - w_{i+1}\big).$$

After estimating the local truncation error, we now proceed to adjust the step size to keep it within a specified bound.

Error control in Runge-Kutta methods

Since $\tau_{i+1}(h) = O(h^n)$, we assume that there exists a K such that $\tau_{i+1}(h) \approx Kh^n$.

Then the local truncation error produced by applying the n-th order method with a new step size qh can be estimated using the original approximations:

$$\tau_{i+1}(qh) \approx K(qh)^n = q^n(Kh^n) \approx q^n\tau_{i+1}(h) \approx \frac{q^n}{h}\left(\tilde{w}_{i+1} - w_{i+1}\right).$$

To bound $\tau_{i+1}(qh)$ by ϵ we choose q such that

$$\frac{q^n}{h}\left|\tilde{w}_{i+1} - w_{i+1}\right| \approx \left|\tau_{i+1}(qh)\right| \leqslant \epsilon$$

that is,

$$q \leqslant \left(\frac{\epsilon h}{\left|\tilde{w}_{i+1} - w_{i+1}\right|}\right)^{1/n}.$$

Runge-Kutta-Fehlberg Method

One popular technique that uses the above inequality for error control is the Runge-Kutta-Fehlberg method.

This technique uses a Runge-Kutta method with local truncation error of order five,

$$\tilde{w}_{i+1} = \tilde{w}_i + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6$$

to estimate the local error in a Runge-Kutta method of order four given by

$$w_{i+1} = w_i + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5.$$

We give the coefficient equations on the next slide.

We have

$$
\begin{aligned}
k_1 &= hf(t_i, w_i) \\
k_2 &= hf\left(t_i + \frac{h}{4}, w_i + \frac{1}{4}k_1\right) \\
k_3 &= hf\left(t_i + \frac{3h}{8}, w_i + \frac{3}{32}k_1 + \frac{9}{32}k_2\right) \\
k_4 &= hf\left(t_i + \frac{12h}{13}, w_i + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right) \\
k_5 &= hf\left(t_i + h, w_i + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right) \\
k_6 &= hf\left(t_i + \frac{h}{2}, w_i - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right).
\end{aligned}
$$

Runge-Kutta-Fehlberg Method

An advantage to this method is that only six evaluations of f are required per step.

Arbitrary Runge-Kutta methods of orders four and five used together require at least four evaluations of f for the fourth-order method and an additional six for the fifth-order method, for a total of at least ten function evaluations.

So the Runge-Kutta-Fehlberg method has at least a 40% decrease in the number of function evaluations over the use of a pair of arbitrary fourth- and fifth-order methods.

In the error-control theory, an initial value of h at the i-th step is used to find the first values of w_{i+1} and $\tilde{w}_{i+1}$, which leads to the determination of q for that step, and then the calculations are repeated.

Runge-Kutta-Fehlberg Method

This procedure requires twice the number of function evaluations per step as without the error control.

In practice, the value of q to be used is chosen somewhat differently in order to make the increased function-evaluation cost worthwhile.

The value of q determined at the i-th step is used for two purposes:

- When $q < 1$: to reject the initial choice of h at the i-th step and repeat the calculations using qh, and
- When $q \geqslant 1$: to accept the computed value at the i-th step using the step size h, but change the step size to qh for the $(i+1)$-st step.

Since a penalty of function evaluations must be paid if the steps are repeated, q tends to be chosen conservatively.

An example

A common choice of q for the Runge-Kutta-Fehlberg method with $n = 4$ is

$$q = \left(\frac{\epsilon h}{2|\tilde{w}_{i+1} - w_{i+1}|} \right)^{1/4} = (0.84) \left(\frac{\epsilon h}{|\tilde{w}_{i=1} - w_{i+1}|} \right)^{1/4}.$$

We now use this method with a tolerance $\epsilon = 10^{-5}$, a maximum step size $h_{\max} = 0.25$, and a minimum step size $h_{\min} = 0.01$ to approximate the solution to our usual initial-value problem

$$y' = y - t^2 + 1, \;\; 0 \leqslant t \leqslant 2, \qquad y(0) = 0.5$$

and compare the results with the exact solution

$$y(t) = (t+1)^2 - 0.5e^t.$$

The initial condition gives $t_0 = 0$ and $w_0 = 0.5$.

The example, continued

Now, we compute the coefficients $k_1, \ldots, k_6$ using $h_{\max} = 0.25$:

$$k_1 = hf(t_0, w_0) = 0.25(0.5 - 0^2 + 1) = 0.375$$

$$k_2 = 0.3974609, \quad k_3 = 0.4095383, \quad k_4 = 0.4584971,$$

$$k_5 = 0.4658452, \quad \text{and} \quad k_6 = 0.4204789.$$

We now compute the approximations to $y(0.25)$:

$$\begin{aligned}
\tilde{w}_1 &= \tilde{w}_0 + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6 \\
&= 0.5 + \frac{16}{135}0.375 + \frac{6656}{12825}0.4095383 + \frac{28561}{56430}0.4584971 \\
&\quad -\frac{9}{50}0.4658452 + \frac{2}{55}0.4204789 \\
&= 0.9204870.
\end{aligned}$$

The example, continued

Similarly,

$$\begin{aligned} w_1 &= w_0 + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5 \\ &= 0.5 + \frac{25}{216}0.375 + \frac{1408}{2565}0.4095383 + \frac{2197}{4104}0.4584971 \\ &\quad -\frac{1}{5}0.4658452 \\ &= 0.9204886. \end{aligned}$$

Then

$$\tau_1(h) \approx \frac{1}{h}|\tilde{w}_1 - w_1| = 0.00000621388$$

and

$$q = (0.84)\left(\frac{\epsilon h}{|\tilde{w}_{i=1} - w_{i+1}|}\right)^{1/4} = 0.9461033291.$$

The example, continued

Since $q \approx 1$ we accept the approximation 0.9204886 for $y(0.25)$ but we should adjust the step size for the next iteration to

$$h = 0.9461033291(0.25) \approx 0.2365258.$$

However, only the leading 5 digits of this result would be expected to be accurate because $\tau_1(h)$ has only about 5 digits of accuracy.

Because we are effectively subtracting the nearly equal numbers w_i and $\tilde{w}_i$ when we compute $\tau_1(h)$, there is a good likelihood of round-off error.

This is an additional reason for being conservative when computing q.

The table of these values can not be displayed in a slide, it has 8 columns and 10 rows.

Why did we discuss this method

The point is that there is a close connection between the problem of approximating the value of a definite integral and that of approximating the solution to an initial-value problem.

Therefore, there are analogues of the adaptive methods for solving the initial-value problems.

However, these methods become very complicated very soon if the computations are to be done by hand, that is, on a calculator than by a programme.

As such, our interest in these methods is only for the academic purpose and not from the computational purposes.

MA 214: Introduction to numerical analysis

Lecture 35

Shripad M. Garge.
IIT Bombay

(shripad@math.iitb.ac.in)

2021-2022

Multi-step methods

The methods discussed to this point in the theme are called one-step methods because the approximation for the mesh point t_{i+1} involves information from only one of the previous mesh points, t_i.

Although these methods might use function evaluation information at points between t_i and t_{i+1}, they do not retain that information for direct use in future approximations.

All the information used by these methods is obtained within the subinterval over which the solution is being approximated.

The approximate solution is available at each of the mesh points $t_0, t_1, \ldots, t_i$ before the approximation at t_{i+1} is obtained, and the error $|w_j - y(t_j)|$ tends to increase with j, so it seems reasonable to develop methods that use these more accurate previous data when approximating the solution at t_{i+1}.

Multi-step methods

Methods using the approximation at more than one previous mesh points to determine the approximation at the next point are called multi-step methods.

An *m*-step method for solving the initial-value problem

$$y' = f(t, y), \quad a \leqslant t \leqslant b, \qquad y(a) = \alpha$$

is the difference method

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m}$$

$$+h\big[b_m f(t_{i+1}, w_{i+1}) + b_{m-1} f(t_i, w_i) + \cdots + b_0 f(t_{i+1-m}, w_{i+1-m})\big]$$

for $i = m-1, \ldots, N-1$, $h = (b-a)/N$, constants a_i, b_j and initial values $w_0 = \alpha$, $w_1 = \alpha_1$, $\ldots$, $w_{m-1} = \alpha_{m-1}$.

Multi-step methods

When $b_m = 0$, the method is called explicit or open, because the difference equation then gives w_{i+1} explicitly in terms of previously determined values.

When $b_m \neq 0$, the method is called implicit or closed, because then w_{i+1} occurs on both sides of the difference equation, so w_{i+1} is specified only implicitly.

The implicit methods are generally more accurate than the explicit methods, but to apply an implicit method directly, we must solve the implicit equation for w_{i+1}. This is not always possible, and even when it can be done, the solution for w_{i+1} may not be unique.

The starting values $\alpha_1, \ldots, \alpha_{m-1}$ must be specified, generally by assuming $w_0 = \alpha$ and generating the remaining values by either a Runge-Kutta or a Taylor method.

Adams-Bashforth technique: This is an *explicit* four-step method with the difference equation

$$\begin{aligned} w_{i+1} &= w_i + \frac{h}{24}[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) \\ &\quad -9f(t_{i-3}, w_{i-3})] \end{aligned}$$

Adams-Moulton technique: This is an *implicit* three-step method with the difference equation

$$\begin{aligned} w_{i+1} &= w_i + \frac{h}{24}[9f(t_{i+1}, w_{i+1}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) \\ &\quad +f(t_{i-2}, w_{i-2})] \end{aligned}$$

An example

We have been playing with the initial-value problem

$$y' = y - t^2 + 1, \;\; 0 \leqslant t \leqslant 2, \qquad y(0) = 0.5$$

throughout this theme.

We apply the Runge-Kutta method of order 4 to it with $h = 0.2$ and compute $w_0 = 0.5$, $w_1 = 0.8292933$, $w_2 = 1.2140762$ and $w_3 = 1.6489220$.

We use these as starting values for the Adams-Bashforth method to compute approximations for $y(0.8)$ and $y(1.0)$.

We will later compare them with those obtained by the above Runge-Kutta method of order four.

The example, continued

We get

$$\begin{aligned} w_4 &= w_3 + \frac{0.2}{24}\big[55f(0.6, w_3) - 59f(0.4, w_2) + 37f(0.2, w_1) \\ &\quad -9f(0, w_0)\big) \\ &= 2.1272892 \end{aligned}$$

and $w_5 = 2.6410533$.

The errors then are

$y(t_i)$	R-K	error	A-B	error
2.1272295	2.1272027	2.69×10^{-5}	2.1272295	5.97×10^{-5}
2.6408591	2.6408227	3.64×10^{-5}	2.6410533	1.94×10^{-4}

Local truncation error

If $y(t)$ is the solution to the initial-value problem $y' = f(t, y)$, $a \leqslant t \leqslant b$, $y(a) = \alpha$ and

$$\begin{aligned} w_{i+1} &= a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\ &\quad + h\Big[b_m f(t_{i+1}, w_{i+1}) + b_{m-1} f(t_i, w_i) + \ldots \\ &\quad + b_0 f(t_{i+1-m}, w_{i+1-m})\Big] \end{aligned}$$

is a multi-step method, then its local truncation error is

$$\begin{aligned} \tau_{i+1}(h) &= \frac{y_{i+1} - a_{m-1}y_i - a_{m-2}y_{i-1} - \cdots - a_0 y_{i+1-m}}{h} \\ &\quad - \Big[b_m f(t_{i+1}, y_{i+1}) + b_{m-1} f(t_i, y_i) + \ldots \\ &\quad + b_0 f(t_{i+1-m}, y_{i+1-m})\Big]. \end{aligned}$$

Predictor-corrector method

The implicit methods generally give better results than the explicit method of the same order.

But the implicit methods have the inherent weakness of first having to convert the method algebraically to an explicit representation for w_{i+1}. This procedure is not always possible.

We could use Newton's method or the secant method to approximate w_{i+1}, but this complicates the procedure considerably.

In practice, implicit multi-step methods are not used as described above. Rather, they are used to improve approximations obtained by explicit methods.

The combination of an explicit method to predict and an implicit method to improve the prediction is called a predictor-corrector method.

Predictor-corrector method

Consider the following fourth-order method for solving an initial-value problem.

The first step is to calculate the starting values w_0, w_1, w_2 and w_3 for the four-step explicit Adams-Bashforth method. To do this, we use the Runge-Kutta method of order four.

The next step is to calculate an approximation, w_{4p}, to $y(t_4)$ using the explicit Adams-Bashforth method as predictor:

$$w_{4p} = w_3 + \frac{h}{24}[55f(t_3, w_3) - 59f(t_2, w_2) + 37f(t_1, w_1) - 9f(t_0, w_0)].$$

This approximation is improved by inserting w_{4p} in the right side of the three-step implicit Adams-Moulton method and using that method as a corrector.

This gives

$$w_4 = w_3 + \frac{h}{24}\left[9f(t_4, w_{4p}) + 19f(t_3, w_3) - 5f(t_2, w_2) + f(t_1, w_1)\right].$$

The only new function evaluation required in this procedure is $f(t_4, w_{4p})$ in the corrector equation; all the other values of f have been calculated for earlier approximations.

The value w_4 is then used as the approximation to $y(t_4)$, and the technique of using the Adams-Bashforth method as a predictor and the Adams-Moulton method as a corrector is repeated to find w_{5p} and w_5, the initial and final approximations to $y(t_5)$.

This process is continued until we obtain an approximation to $y(t_N) = y(b)$.

Predictor-corrector method

Improved approximations to $y(t_{i+1})$ might be obtained by iterating the Adams-Moulton formula, but these converge to the approximation given by the implicit formula rather than to the solution $y(t_{i+1})$.

Hence it is usually more efficient to use a reduction in the step size if improved accuracy is needed.

Example: Apply the Adams fourth-order predictor-corrector method with $h = 0.2$ and starting values from the Runge-Kutta fourth order method to our standard initial-value problem $y' = y - t^2 + 1$, $0 \leqslant t \leqslant 2$, $y(0) = 0.5$.

The initial values are: $w_0 = 0.5$, $w_1 = 0.8292933$, $w_2 = 1.2140762$ and $w_3 = 1.6489220$.

The fourth-order Adams-Bashforth method gives $w_{4p} = 2.1272892$.

The example, continued

We now use w_{4p} as the predictor of the approximation to $y(0.8)$ and determine the corrected value w_4 from the implicit Adams-Moulton method. This gives

$$\begin{aligned} w_4 &= w_3 + \frac{0.2}{24}\big(9f(0.8, w_{4p}) + 19f(0.6, w_3) - 5f(0.4, w_2) \\ &\quad + f(0.2, w_1)\big) \\ &= 2.1272056. \end{aligned}$$

The predicted and corrected value for approximating $y(1.0)$ are similarly computed as

$$w_{5p} = 2.6409314 \quad \text{and} \quad w_5 = 2.6408286.$$

The example, continued

The values computed using only the explicit Adams-Bashford method were found to be inferior to those computed by the Runge-Kutta method of order 4.

Now we have managed to do better than the Runge-Kutta method.

$y(t_i)$	R-K	error	P-C	error
2.1272295	2.1272027	2.69×10^{-5}	2.1272056	2.39×10^{-5}
2.6408591	2.6408227	3.64×10^{-5}	2.6408286	3.05×10^{-5}

We give the table of the remaining values in the following slide:

The example, continued

t_i	$y_i = y(t_i)$	w_i	Error $\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292933	0.0000053
0.4	1.2140877	1.2140762	0.0000114
0.6	1.6489406	1.6489220	0.0000186
0.8	2.1272295	2.1272056	0.0000239
1.0	2.6408591	2.6408286	0.0000305
1.2	3.1799415	3.1799026	0.0000389
1.4	3.7324000	3.7323505	0.0000495
1.6	4.2834838	4.2834208	0.0000630
1.8	4.8151763	4.8150964	0.0000799
2.0	5.3054720	5.3053707	0.0001013

MA 214: Introduction to numerical analysis
Lecture 36

Shripad M. Garge.
IIT Bombay

(shripad@math.iitb.ac.in)

2021-2022

Winding up

A number of methods have been presented in this theme for approximating the solution to an initial-value problem.

Although numerous other techniques are available, we have chosen the methods described here because they generally satisfied three criteria:

- Their development is clear enough so that you can understand how and why they work.

- One or more of the methods will give satisfactory results for most of the problems that are encountered by students.

- Most of the more advanced and complex techniques are based on one or a combination of the procedures described here.

Consistency and convergence

We now discuss why these methods are expected to give satisfactory results when some similar methods do not.

A one-step difference-equation method with local truncation error $\tau_i(h)$ at the i-th step is said to be consistent with the differential equation it approximates if

$$\lim_{h \to 0} \max_{1 \leqslant i \leqslant N} |\tau_i(h)| = 0.$$

A consistent one-step method has the property that the difference equation for the method approaches the differential equation when the step size goes to zero. So the local truncation error of a consistent method approaches zero as the step size approaches zero.

Note that consistency is a local measure since, for each of the values $\tau_i(h)$, we are assuming that the approximation w_{i-1} and the exact solution $y(t_{i-1})$ are the same.

Consistency and convergence

A more realistic measure of analysing the effects of making h small is to determine the global effect of the method.

This is the maximum error of the method over the entire range of the approximation, assuming only that the method gives the exact result at the initial value.

A one-step difference-equation method is said to be convergent with respect to the differential equation it approximates if

$$\lim_{h \to 0} \max_{1 \leqslant i \leqslant N} |y(t_i) - w_i| = 0.$$

Consistency and convergence

Let us check the convergence for Euler's method. The global error in Euler's method is given by

$$\max_{1 \leqslant i \leqslant N} |y(t_i) - w_i| \leqslant \frac{Mh}{2L} |e^{L(b-a)} - 1|.$$

where M, L, a and b are constants.

Hence

$$\lim_{h \to 0} \max_{1 \leqslant i \leqslant N} |y(t_i) - w_i| \leqslant \lim_{h \to 0} \frac{Mh}{2L} |e^{L(b-a)} - 1| = 0.$$

Thus Euler's method is convergent.

Stability

The other problem that exists when using difference methods is a consequence of not using exact results.

In practice, neither the initial conditions nor the arithmetic that is subsequently performed is represented exactly because of the round-off error associated with finite-digit arithmetic.

We saw earlier that this consideration can lead to difficulties even for the convergent Euler's method.

To analyze this situation, at least partially, we will try to determine which methods are stable, in the sense that small changes or perturbations in the initial conditions produce correspondingly small changes in the subsequent approximations.

Suppose the initial-value problem

$$y = f(t, y), \quad a \leqslant t \leqslant b, \qquad y(a) = \alpha$$

is approximated by a one-step difference method in the form

$$w_0 = \alpha, \quad w_{i+1} = w_i + h\phi(t_i, w_i, h).$$

Suppose also that a number $h_0 > 0$ exists and that $\phi(t, w, h)$ is continuous and satisfies a Lipschitz condition in the variable w with Lipschitz constant L on the set

$$D = \{(t, w, h) : a \leqslant t \leqslant b \text{ and } -\infty < w < \infty, 0 \leqslant h \leqslant h_0\}.$$

Then the following three properties hold:

(1) The method is stable.

(2) The difference method is convergent if and only if it is consistent, the consistency being equivalent to

$$\phi(t, y, 0) = f(t, y), \quad \text{for all } a \leqslant t \leqslant b.$$

(3) If a function τ exists and, for each $i = 1, 2, \ldots, N$, the local truncation error $\tau_i(h)$ satisfies $|\tau_i(h)| \leqslant \tau(h)$ whenever $0 \leqslant h \leqslant h_0$ then

$$|y(t_i) - w_i| \leqslant \frac{\tau(h)}{L} e^{L(t_i - a)}.$$

The third part is about controlling the global error of a method by controlling its local truncation error.

It implies that when the local truncation error has the rate of convergence $O(h^n)$, the global error will have the same rate of convergence.

Modified Euler method

The Modified Euler method is given by $w_0 = \alpha$,

$$w_{i+1} = w_i + \frac{h}{2}\Big[f(t_i, w_i) + f\Big(t_{i+1}, w_i + hf(t_i, w_i)\Big)\Big]$$

for $i = 0, 1, \ldots, N-1$.

We verify that this method is stable by showing that it satisfies the hypothesis of the above Theorem.

Here $\phi(t, w, h) = \frac{1}{2}\Big[f(t, w) + f\Big(t, w + hf(t, w)\Big)\Big]$ and hence

$$\begin{aligned}\phi(t, w_1, h) - \phi(t, w_2, h) &= \frac{1}{2}f(t, w_1) + \frac{1}{2}f\Big(t + h, w_1 + hf(t, w_1)\Big) \\ &\quad -\frac{1}{2}f(t, w_2) - \frac{1}{2}f\Big(t + h, w_2 + hf(t, w_2)\Big)\end{aligned}$$

Modified Euler method

Since f satisfies Lipschitz condition with Lipschitz constant L

$$\begin{aligned}
|\phi(t, w_1, h) - \phi(t, w_2, h)| &\leqslant \frac{1}{2}L|w_1 - w_2| \\
&\quad + \frac{1}{2}L|w_1 + hf(t, w_1) - w_2 - hf(t, w_2)| \\
&\leqslant L|w_1 - w_2| + \frac{1}{2}L|hf(t, w_1) - hf(t, w_2)| \\
&\leqslant L|w_1 - w_2| + \frac{1}{2}hL^2|w_1 - w_2| \\
&= \left(L + \frac{1}{2}hL^2\right)|w_1 - w_2|.
\end{aligned}$$

Thus ϕ satisfies a Lipschitz condition with the costant $L + \frac{1}{2}hL^2$ on

$$D = \left\{(t, w, h) : a \leqslant t \leqslant b \text{ and } -\infty < w < \infty, 0 \leqslant h \leqslant h_0\right\}$$

for any $h_0 > 0$.

Modified Euler method

Finally, the continuity of f on $[a, b] \times \mathbb{R}$ implies the continuity of ϕ on $[a, b] \times \mathbb{R} \times [0, h_0]$.

Thus, by the above theorem, this method is stable.

Since

$$\phi(t, w, 0) = \frac{1}{2} f(t, w) + \frac{1}{2} f\big(t, w + 0 f(t, w)\big) = f(t, w)$$

we get that this method is consistent.

The above theorem then implies that the method is also convergent.

Moreover, we have seen that for this method the local truncation error is $O(h^2)$ so, by the third part of the theorem, the convergence of the Modified Euler method is also $O(h^2)$.

Multi-step methods

For multi-step methods, the problems involved with consistency, convergence, and stability are compounded because of the number of approximations involved at each step.

In the one-step methods, the approximation w_{i+1} depends directly only on the previous approximation w_i, whereas the multistep methods use at least two of the previous approximations, and the usual methods that are employed involve even more.

The convergence of a multi-step method is defined in the same way as for the one-step methods, based on the global error, $|y(t_i) - w_i|$, but the consistency and stability are phrased in a different way.

We will not go in the details but will only say here that both the Adams-Bashforth method and the Adams-Moulton method are stable.

The fifth theme is complete

This completes the fifth theme of our course:

Ordinary Differential Equations.

From our next lecture, we will begin the next (and essentially the last) theme:

Numerical Linear Algebra.

MA 214: Introduction to numerical analysis
Lecture 37

Shripad M. Garge.
IIT Bombay

(shripad@math.iitb.ac.in)

2021-2022

We begin our final theme

Numerical Linear Algebra.

In linear algebra, one needs to do several computations, right from solving a homogeneous linear system of equations to computing eigenvalues and eigenvectors of matrices.

There are many methods in this area and we will be able to do justice to only a few.

System of linear equations

To begin with, we interest ourselves with the homogeneous system of linear equations. Such a system is given in the form

$$A \cdot x = b$$

where A is an $n \times n$ real matrix, $b = [b_1, b_2, \ldots, b_n]^t$ is the given $n \times 1$ vector and $x = [x_1, x_2, \ldots, x_n]^t$ is the unknown vector. We want to solve for the vector x.

If the matrix A is invertible, then there is a unique solution,

$$x = A^{-1} b,$$

however, if A is not invertible then either the solution does not exist or there are infinitely many solutions.

System of linear equations

Consider the systems:

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \\ 6 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ 6 \end{pmatrix}.$$

The first system has infinitely many solutions, indeed each $(x_1, x_2, 2)$ with $x_1 + x_2 = 2$ is a solution to it.

Whereas the second system has no solution. This can be seen by observing $x_1 + x_2 + x_3 = 2x_1 + 2x_2 + x_3 = 4$ and hence $x_1 + x_2 = 0$. Putting this back in the first equation gives $x_3 = 4$, and putting it in the third equation gives $x_3 = 3$.

This is a contradiction! The invertibility is therefore necessary.

In the invertible case, we have Cramer's rule:

$$x_j = \frac{\det A_j}{\det A}$$

where A_j is the $n \times n$ matrix obtained by replacing the j-th column of A by the vector b. This method, however, involves way too many operations. If A is an upper triangular matrix,

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix}$$

then the system can be solved by an easier method, first solving for x_n, then for x_{n-1} and so on.

System of linear equations

This is a much simpler method and so it is desirable to transform every matrix to an upper triangular matrix.

This is indeed possible, and our old friend Carl Friedrich Gauß will help us do it.

There is a method called the Gauß elimination method, GEM for short, which we shall soon see.

This also leads to an interesting decomposition $A = LU$ of any given matrix A as a product of a lower triangular matrix L with an upper triangular matrix U. This is called an LU-decomposition.

We will see more of it later.

System of linear equations

We use three operations to simplify the given linear system of equations $E_1 - E_n$:

1. Equation E_i can be multiplied by any nonzero constant λ with the resulting equation used in place of E_i. This operation is denoted $(\lambda E_i) \to (E_i)$.
2. Equation E_j can be multiplied by any constant λ and added to equation E_i with the resulting equation used in place of E_i. This operation is denoted $(E_i + \lambda E_j) \to (E_i)$.
3. Equations E_i and E_j can be transposed in order. This operation is denoted $(E_i) \leftrightarrow (E_j)$.

By a sequence of these operations, a linear system will be systematically transformed into a new linear system that is more easily solved and has the same solutions.

An example

Let us consider an example:

$$\begin{array}{lrrrrcr} E_1: & x_1 + & x_2 & & + 3x_4 & = & 4, \\ E_2: & 2x_1 + & x_2 - & x_3 + & x_4 & = & 1, \\ E_3: & 3x_1 - & x_2 - & x_3 + & 2x_4 & = & -3, \\ E_4: & -x_1 + & 2x_2 + & 3x_3 - & x_4 & = & 4. \end{array}$$

We will solve it for x_1, x_2, x_3 and x_4.

We first use equation E_1 to eliminate the unknown x_1 from equations E_2, E_3, and E_4 by performing

$(E_2 - 2E_1) \to (E_2)$,

$(E_3 - 3E_1) \to (E_3)$ and

$(E_4 + E_1) \to (E_4)$.

The resulting system is as follows:

$$\begin{aligned}
E_1: \quad & x_1 + x_2 + 3x_4 = 4, \\
E_2: \quad & -x_2 - x_3 - 5x_4 = -7, \\
E_3: \quad & -4x_2 - x_3 - 7x_4 = -15, \\
E_4: \quad & 3x_2 + 3x_3 + 2x_4 = 8.
\end{aligned}$$

In the new system, E_2 is used to eliminate the unknown x_2 from E_3 and E_4 by performing $(E_3 - 4E_2) \to (E_3)$ and $(E_4 + 3E_2) \to (E_4)$. This results in

$$\begin{aligned}
E_1: \quad & x_1 + x_2 + 3x_4 = 4, \\
E_2: \quad & -x_2 - x_3 - 5x_4 = -7, \\
E_3: \quad & +3x_3 + 13x_4 = 13, \\
E_4: \quad & -13x_4 = -13.
\end{aligned}$$

The system of equations is now in triangular (or reduced) form and can be solved for the unknowns by a backward-substitution process.

Since E_4 implies $x_4 = 1$, we can solve E_3 for x_3 to give

$$x_3 = \frac{1}{3}(13 - 13x_4) = \frac{1}{3}(13 - 13) = 0.$$

Continuing, E_2 gives

$$x_2 = -(-7 + 5x_4 + x_3) = -(-7 + 5 + 0) = 2,$$

and E_1 gives

$$x_1 = 4 - 3x_4 - x_2 = 4 - 3 - 2 = -1.$$

The solution to system is, $x_1 = -1$, $x_2 = 2$, $x_3 = 0$ and $x_4 = 1$.

Gauß elimination method

We apply Gaußian elimination method to the linear system

$$\begin{aligned} E_1: &\quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ E_2: &\quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ &\vdots \qquad\qquad\qquad \vdots \qquad\qquad \vdots \\ E_n: &\quad a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n. \end{aligned}$$

as follows.

We first form the augmented matrix

$$\tilde{A} = [A : b].$$

If $a_{11} \neq 0$ then we perform $(E_j - (a_{j1}/a_{11})E_1) \to (E_j)$ for each $j = 2, 3, \ldots, n$ to eliminate x_1 in each of these rows.

Gauß elimination method

Although the entries in rows $2, 3, \ldots, n$ are expected to change, for ease of notation we again denote the entry in the i-th row and the j-th column by a_{ij}.

With this in mind, we follow a sequential procedure for $i = 2, 3, \ldots, n-1$ and perform the operation

$$(E_j - (a_{ji}/a_{ii})E_i) \rightarrow (E_j)$$

for each $j = i+1, i+2, \ldots, n$, provided $a_{ii} \neq 0$.

This eliminates x_i in each row below the i-th row for all values of $i = 1, 2, \ldots, n-1$. The resulting augmented matrix has the form

$$\tilde{A} = [A : b]$$

where the matrix A is now upper triangular.

The system now looks

$$\begin{aligned}
E_1: \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\
E_2: \quad & a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\
\vdots \quad & \ddots \quad \vdots \\
E_n: \quad & a_{nn}x_n = b_n.
\end{aligned}$$

so backward substitution can be performed.

Solving E_n for x_n gives $x_n = \dfrac{b_n}{a_{nn}}$. Then we solve for x_{n-1}, then for x_{n-2} and so on using

$$\begin{aligned}
x_i &= \frac{b_i - a_{i,n}x_n - a_{i,n-1}x_{n-1} - \cdots - a_{i,i+1}x_{i+1}}{a_{ii}} \\
&= \frac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}.
\end{aligned}$$