Solving Ordinary Differential equations

Our starting point is the following ODE:

$$\left.\begin{array}{rcl} y'(t) & = & f(t, y(t)), \quad a \leq t \leq b \\ y(a) & = & \eta. \end{array}\right\} \tag{1}$$

Why do we need numerical methods for solving such an "innocuous" differential equation? Because problems of this form are often difficult, if not impossible, to solve analitically. This is specially the case when the equation is nonlinear.

Solving Ordinary Differential equations

Although we will concentrate on first-order scalar equations, the numerical methods are easily extended to cover first-order systems of equations. Furthermore, it is easy to convert higher order scalar initial value problems to first-order systems. Therefore, we are not restricted to first order equations. In fact, the methods can be used as part of numerical algorithms to find solutions of evolutionary **partial** differential equations. Additionally, we will consider also methods for solving **Boundary Value Problems**.

Solving Ordinary Differential equations

Outline of Study of ODE's:

1. Single-Step Methods for I.V. Problems

   a. Euler.
   b. Trapezoidal.
   c. Taylor methods.
   d. General Runge-Kutta
   e. Adaptive step-size control.

2. Stiff ODEs
3. Multi-Step Methods for I.V. Problems.
4. Boundary Value Problems

   1. Shooting Method
   2. Finite Difference Method
   3. Finite element Method

Single-Step Methods for I.V. Problems

Discretisation
The central idea behind numerical methods is that of discretisation. That is we partition the continuous interval $[a, b]$ by a discrete set of $N + 1$ points:

$$a = t_0 < t_1 < t_2 < ... < t_{N-1} < t_N = b.$$

The parameters

$$h_n = t_{n+1} - t_n \,, \quad n = 0, 1, ..., N - 1 \tag{2}$$

are called the step-sizes. We will be often be interested in using an equally spaced partition where

$$h_n = h = \frac{(b - a)}{N}, \;\; n = 0, 1, ..., N - 1.$$

Single-Step Methods for I.V. Problems

We will let y_n denote the numerical approximation to the exact solution $y(t_n)$. A numerical solution of (1) consists of a set of discrete approximations $\{y_n\}_{n=0}^{N}$. A numerical method is a difference equation involving a number of consecutive approximations

$$y_j, \;\; j = 0, 1, ..., k$$

from which we sequentially compute the sequence

$$y_{k+n}, \;\; n = 1, 2, ..., N.$$

Single-Step Methods for I.V. Problems

The derivation of a number of numerical methods begins by integrating (1) between t_n and t_{n+1}. This gives

$$\begin{aligned}
\int_{t_n}^{t_{n+1}} \frac{dy}{dt}\,dt &= \int_{t_n}^{t_{n+1}} f(t,y)\,dt \\
\Rightarrow y(t_{n+1}) - y(t_n) &= \int_{t_n}^{t_{n+1}} f(t,y)\,dt.
\end{aligned}$$

Now, if we make the approximation

$$f(t,y) \approx f(t_n, y(t_n)), \quad t \in (t_n, t_{n+1})$$

then

$$y(t_{n+1}) - y(t_n) \approx \int_{t_n}^{t_{n+1}} f(t_n, y(t_n))\,dt = (t_{n+1} - t_n) f(t_n, y(t_n)).$$

Therefore

$$y(t_{n+1}) \approx y(t_n) + (t_{n+1} - t_n) f(t_n, y(t_n)).$$

Single-Step Methods for I.V. Problems

This suggest the numerical method

$$\boxed{y_{n+1} = y_n + (t_{n+1} - t_n) f(t_n, y(t_n)), \quad n = 0, 1, ..., N-1 \qquad (3)}$$

which is called the forward or explicit Euler method. Note that from the initial condition $y_0 = \eta$ we can explicitly calculate y_1 by applying (3). This in turn allows us to calculate y_2 and then y_3 and so on.
A geometrical interpretation of the forward Euler method is that instead of following the possibly curved solution trajectory passing through (t_n, y_n), the method actually follows a straight line trajectory which has slope $f(t_n, y_n)$.
The forward Euler method is, of course, an approximate method which will only be exact in the trivial case of a linear solution in t. But we hope that the method will be closer to the exact solution as the step-size h is taken smaller. This is a neccesary condition for any reasonable numerical method.

Single-Step Methods for I.V. Problems

Definition

We will say that a numerical method is convergent when for all IVP (1) with solution sufficiently differentiable, the following condition applies

$$\lim_{h\to 0}\left(\max_{1\leq n\leq N}||y_n - y(t_n)||\right) = 0.$$

being $y_0 = y(t_0)$.
We will say that the order of convergence of the method is p, if as h is taken smaller (ie, with N large enough), then

$$\left(\max_{1\leq n\leq N}||y_n - y(t_n)||\right) = \mathcal{O}(h^p)\,,\;\; Nh = \text{constante}$$

ie, if there $\exists C \geq 0$ such that $\max_{1\leq n\leq N}||y_n - y(t_n)|| \leq C|h|^p$ for N large enough.

Single-Step Methods for I.V. Problems

Theorem

For all IVP (1) with f continuous and satisfying a Lipschitz condition on D, the forward Euler method is convergent and its order of convergence is 1. The error of the Euler method can be bound as follows

$$||y(t_n) - y_n|| \leq \frac{C}{2L}(e^{(t_n-a)L} - 1)h\,,\ 0 \leq n \leq N$$

being $y_0 = y(t_0)$, $C = \max_{x\in[a,b]} ||y''(x)||$ and L a Lipschitz constant.

Single-Step Methods for I.V. Problems

An obvious question is whether we can easily improve upon the forward Euler method. Remembering that Euler method replaces $f(t, y(t))$ by the slope $f(t_n, y_n)$, it seems likely that an improved approximation would be the average of the slopes at t_n and t_{n+1}. That is

$$y(t_{n+1}) - y(t_n) \approx (t_{n+1} - t_n)\frac{1}{2}\left(f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))\right) .$$

This suggest the following numerical method:

$$\boxed{y_{n+1} = y_n + \frac{h_n}{2}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right) . \qquad (4)}$$

This method is called the trapezoidal method and differs from the forward Euler method in an important way:

Single-Step Methods for I.V. Problems

At the $(n+1)$st step we have to solve the (generally no linear) equation

$$g(y_{n+1}) \equiv y_{n+1} - y_n - \frac{h_n}{2}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right) = 0 \qquad (5)$$

to determine y_{n+1}. Therefore y_{n+1} is defined implicitly and for that reason the trapezoidal method is an example of an implicit method. The forward Euler method, on the other hand, is an example of an explicit method.

Single-Step Methods for I.V. Problems

For the trapezoidal method, the following convergence theorem can be stablished

Theorem

For all IVP (1) satisfying a Lipschitz condition, the trapezoidal method is convergent and for $hL < 2$ (being L a Lipschitz constant) the error can be bound by

$$|e_n| \leq \frac{Ch^2}{L} \exp\left(\frac{L(t_n - a)}{1 - \frac{hL}{2}}\right),$$

where $|y'''| < C$.
Therefore, the trapezoidal method is convergent with order 2.

Solving Ordinary Differential equations

Taylor methods can be used to build explicit methods with higher order of convergence than Euler's method. The main difficult of these methods is the computation of the derivatives of $f(t, y)$.
The idea behind these methods is simple: considering the Taylor series of $y(t_{n+1})$ up to a certain order. For example, by truncating up to order h^2 we will have the Taylor method of order 2.
Let us build this method. Consider:

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + \frac{h^3}{6}y'''(\zeta_n)$$

We will take up to order h^2. For computing $y'(t_n)$ and $y''(t_n)$ we will use the differential equation:

$$\begin{array}{l} y'(t) = f(t, y(t)) \\ y''(t) = f_t(t, y(t)) + f_y(t, y(t))y'(t) = f_t(t, y(t)) + f_y(t, y(t))f(t, y(t)) \end{array}$$

Therefore we have the method

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2}(f_t(t_n, y_n) + f_y(t, y_n)f(t_n, y_n))$$

Solving Ordinary Differential equations

By defining as the local truncation error the term which is neglected for building the method, we see that in our method this error is $T_n = \frac{h^3}{6} y'''(\zeta_n)$, which satisfies $||T_n|| \leq Ch^3$. This means that the method is order 2.

Error Analysis

1. Truncation Error (Truncating Taylor Series) [Large step size $\Longrightarrow$ large errors]
2. Rounding Error (Machine precision) [very small step size $\Longrightarrow$ roundoff errors]

Two kinds of Truncation of Error:

1. **Local** - error within one step due to application of method
2. **Propagation** - error due to previous local errors.

Global Truncation Error = Local + Propagation

Generally if local truncation error is $\mathcal{O}(h^{(n+1)})$ then, the Global truncation error is $\mathcal{O}(h^{(n)})$.

Solving Ordinary Differential equations

General Single-step methods:

Before introducing the Runge-Kutta methods, we will briefly describe the general set-up for Single-step methods. In general, these methods can be written as:

$$y_{n+1} = y_n + h\Phi(x_n, y_n, y_{n+1}, h) \tag{1}$$

where Φ is related to f and it is called the Step Function. When Φ depends on y_{n+1} the method is implicit and in other case the method is explicit.

Single-Step Methods for I.V. Problems

The examples of single-step methods considered so far are:

1. Forward Euler, $\Phi(x_n, y_n, y_{n+1}, h) = f(x_n, y_n)$ (explicit).
2. Trapezoidal, $\Phi(x_n, y_n, y_{n+1}, h) = (f(x_n, y_n) + f(x_{n+1}, y_{n+1}))/2$ (implicit).
3. Taylor of order 2, $\Phi(x_n, y_n, y_{n+1}, h) = f(t_n, y_n) + \frac{h}{2}(f_t(t_n, y_n) + f_y(t, y_n)f(t_n, y_n)$ (explicit).

Single-Step Methods for I.V. Problems

For Single-step methods

Definition

We define the local truncation error, T_n, as:

$$T_n(h) = y(t_{n+1}) - y(t_n) - h\Phi(t_n, y(t_n), y(t_{n+1}), h) \tag{2}$$

Single-Step Methods for I.V. Problems

Runge-Kutta methods are very popular methods which allow us to obtain high-order methods avoiding the computation of the derivatives of the Taylor methods.
Let's see how to build an explicit Runge-Kutta method of order 2:

$$\begin{array}{l} y_{n+1} = y_n + h\Phi(t_n, y_n, h) \\ \Phi(t_n, y_n, h) = ak_1 + bf(t_n + \alpha h, y_n + \beta hk_1)\,,\ \ k_1 = f(t_n, y_n) \end{array} \qquad (3)$$

The constants of the Step function will be determine by imposing that the method behaves like the Taylor method up to order 2. The Taylor method of order 2 was:

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2}(f_t(t_n, y_n) + f_y(t, y_n)f(t_n, y_n))$$

Then, in order to compare with the new method, we just need to expand $f(t_n + \alpha h, y_n + \beta hk_1)$ up to order 1:

$$\begin{aligned} f(t_n + \alpha h, y_n + \beta hk_1) &= f(t_n, y_n) + f_t(t_n, y_n)\alpha h + f_y(t_n, y_n)\beta hk_1 + \mathcal{O}(h^2) \\ &= f(t_n, y_n) + f_t(t_n, y_n)\alpha h + f_y(t_n, y_n)f(t_n, y_n)\beta h + \mathcal{O}(h^2 \end{aligned}$$

Single-Step Methods for I.V. Problems

Going to Eq. (3), we have that up to order 2 the new method is:

$$y_{n+1} = y_n + h(a+b)f(t_n, y_n) + bh^2(\alpha f_t(t_n, y_n) + \beta f(t_n, y_n) f_y(t_n y_n))$$

Then, if the method must behave as the Taylor method up to order 2, we have the following equations for the parameters of the method:

$$a+b=1\,,\ \ b\alpha = b\beta = 1/2$$

The methods verifying these equations will be convergents with order of convergence 2.
Consider, for instance, the methods with $\alpha = \beta$. Then, $b = 1/2\alpha$ and $a = 1 - 1/2\alpha$. The methods with $\alpha = 1/2$ and $\alpha = 1$ are called the Modified Euler Method and the Heun Method, respectively.

Single-Step Methods for I.V. Problems

The Modified Euler Method ($\alpha = 1/2$) can be written as

$$\boxed{y_{n+1} = y_n + hf(t_n + \frac{h}{2}, y_n + \frac{h}{2} f_n)\,. \qquad (4)}$$

and using the notation

$$k_1 = f(t_n, y_n)\,, \quad k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2} k_1)\,,$$

we have that

$$y_{n+1} = y_n + hk_2\,.$$

Single-Step Methods for I.V. Problems

The Heun Method ($\alpha = 1$), can be written as

$$\boxed{y_{n+1} = y_n + \frac{h}{2}\left(f(t_n, y(t_n)) + f(t_{n+1}, y_n + hf(t_n, y_n))\right) . \qquad (5)}$$

If

$$k_1 = f(t_n, y_n)\,, \quad k_2 = f(t_n + h, y_n + hk_1)\,,$$

then, we have:

$$y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2)\,.$$

Single-Step Methods for I.V. Problems

The previous notation can be generalized

$$k_1 = f(t_n, y_n), \;\; k_2 = f(t_n + c_2 h, y_n + a_{21} h k_1)\,,$$

and

$$y_{n+1} = y_n + h(b_1 k_1 + b_2 k_2)\,,$$

where c_2, a_{21}, b_1 and b_2 are constants.
All these methods are members of a big family of methods which are called the Runge-Kutta Methods.

Single-Step Methods for I.V. Problems

The general Runge-Kutta *s*-Stages Method is

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i k_i \ ,$$

where

$$k_i = f\left(t_n + c_i h, y_n + h \sum_{j=1}^{s} a_{ij} k_j \right) \ , \quad i = 1, 2, ..., s.$$

A convenient way of displaying the coefficients of the Runge-Kutta Methods is the use of the Butcher tableaux:

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \end{array} = \begin{array}{c|cccccc} c_1 & a_{11} & a_{12} & a_{13} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & a_{23} & \dots & a_{2s} \\ c_3 & a_{31} & a_{32} & a_{33} & \dots & a_{3s} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_s & a_{s1} & a_{s2} & a_{s3} & \dots & a_{ss} \\ \hline & b_1 & b_2 & b_3 & \dots & b_s \end{array}$$

Single-Step Methods for I.V. Problems

Example

Find the Butcher tableaux of the following Runge-Kutta Method:

$$y_{n+1} = y_n + h\left(\frac{1}{4}k_1 + \frac{3}{4}k_2\right)$$

$$k_1 = f\left(t_n, y_n + h\left(\frac{1}{4}k_1 - \frac{1}{4}k_2\right)\right) \; ,$$

$$k_2 = f\left(t_n + \frac{2}{3}h, y_n + h\left(\frac{1}{4}k_1 + \frac{5}{12}k_2\right)\right) \; .$$

Sol.:

$$\begin{array}{c|cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

Single-Step Methods for I.V. Problems

If $p^*(s)$ is the maximum order of convergence which can be obtained by using an explicit Runge-Kutta s-stages method, then we have

$$\begin{aligned}
p^*(s) &= s, \quad s = 1, 2, 3, 4\\
p^*(5) &= 4\\
p^*(6) &= 5\\
p^*(7) &= 6\\
p^*(8) &= 6\\
p^*(9) &= 7\\
p^*(s) &\leq s - 2\,, \quad s = 10, 11, \ldots
\end{aligned}$$

This behaviour explains the popularity of the 4-stages Runge-Kutta Methods of order 4.

Single-Step Methods for I.V. Problems

Adaptive Step-size Control
Goal: with little additional effort estimate (bound) the magnitude of local truncation error at each step so that step size can be reduced/increased if local error increases/decreases.
Basic idea: 2. Use a matched pair of Runge-Kutta formulas of order p and $p+1$ that use common values of ki, and yield estimate or bound local truncation error.

Single-Step Methods for I.V. Problems

R-K Method of Order p:

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i$$

R-K method of Order $p+1$:

$$y_{n+1} = y_n + h\sum_{i=1}^{s} \tilde{b}_i k_i$$

These methods are called embedded Runge-Kutta Methods and can be expressed by using the following modified Butcher tableaux:

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \\ & \tilde{\boldsymbol{b}}^T \\ \hline & \boldsymbol{b}^T - \tilde{\boldsymbol{b}}^T \end{array}$$

Single-Step Methods for I.V. Problems

The local truncation error for the order-p method can be estimated as follows:

$$\tilde{T}_{n+1} = h\sum_{i=1}^{s}(b_i - \tilde{b}_i)k_i\,.$$

Then, the step-size can be adapted by imposing the following condition:

$$\tilde{T}_{n+1} < Tol,$$

being Tol the tolerance by step unit.
The “new” step size is then selected as follows:

$$h_{new} = \left(\frac{qTol}{|\tilde{T}_{n+1}|}\right)^{\frac{1}{p+1}} h\,. \tag{6}$$

where q is a factor with a value of $q \approx 0.8$.

Single-Step Methods for I.V. Problems

Example: The Runge-Kutta-Fehlberg (4,5) scheme has the following Butcher tableaux

$$
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\
\frac{3}{8} & \frac{3}{32} & \frac{9}{32} & 0 & 0 & 0 & 0 \\
\frac{12}{13} & \frac{1932}{2197} & -\frac{7200}{2197} & \frac{7296}{2197} & 0 & 0 & 0 \\
1 & \frac{439}{216} & -8 & \frac{3680}{513} & -\frac{845}{4104} & 0 & 0 \\
\frac{1}{2} & -\frac{8}{27} & 2 & -\frac{3544}{2565} & \frac{1859}{4104} & -\frac{11}{40} & 0 \\
\hline
 & \frac{25}{216} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & -\frac{1}{5} & 0 \\
 & \frac{16}{135} & 0 & \frac{6656}{12825} & \frac{28561}{56430} & -\frac{9}{50} & \frac{2}{55} \\
\hline
 & \frac{1}{360} & 0 & -\frac{128}{4275} & -\frac{2197}{75240} & \frac{1}{50} & \frac{2}{55}
\end{array}
$$

Single-Step Methods for I.V. Problems

Higher-Order ODEs and Systems of Equations
An $n-$th order ODE can be converged into a system of n coupled 1st-order ODEs. Systems of first order ODEs are solved just as one solves a single ODE.
Example:
Consider the 4th-order ODE

$$y'''' + a(x)y''' + b(x)y'' + c(x)y' + d(x)y = f(x)$$

By letting

$$y''' = v_3; \;\; y'' = v_2; \;\; \text{and } y' = v_1$$

this 4th-order ODE can be written as a system of four coupled 1st-order ODEs.

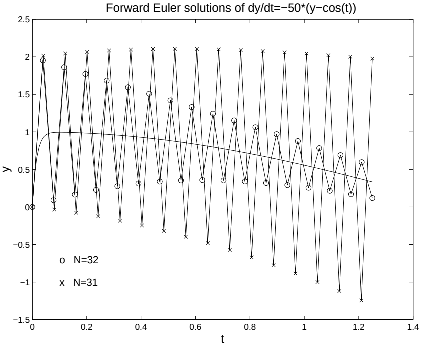
Single-Step Methods for I.V. Problems

Stiff Differential Equations
A stiff system of ODE's is one involving rapidly changing components together with slowly changing ones. In many cases, the rapidly varying components die away quickly, after which the solution is dominated by the slow ones.
Consider the following ODE:

$$y' = -50(y - \cos(t))\,, \;\; 0 \le t \le 1.25\,, \;\; y(0) = 0\,,$$

The figure shows the approximations obtained by using the Forward Euler method for $h = 1.25/31$ and $h = 1.25/32$.

Single-Step Methods for I.V. Problems



Single-Step Methods for I.V. Problems

In order to ensure the stability of the numerical solution, we have to choose an step size $h < 2/50$. This means that, for stiff problems, we would need a very small step to capture the behavior of the rapid transient and to preserve a stable and accurate solution, and this would then make it very laborious to compute the slowly evolving solution.

Euler's method is known as an explicit method because the derivative is taken at the known point i. An alternative is to use an implicit approach, in which the derivative is evaluated at a future step $i+1$. The simplest method of this type is the backward Euler method that yields

$$y_{n+1} = y_n + hf(y_{n+1}, t_{n+1}) \, .$$

In general, for this kind of problems, suitable methods are the BDF *backward differentiation formulae* methods. These methods have the general form:

$$\frac{1}{h} \sum_{j=1}^{k} \frac{1}{j} \nabla^j y_{n+1} = f_{n+1} \, .$$

Linear Multistep Methods for I.V. Problems

The linear multistep methods can be written in the general form

$$\boxed{\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j} \,, \qquad (1)}$$

where k is called the step number and without loss of generality we let $\alpha_k = 1$. Explicit methods are characterised by $\beta_k = 0$ and implicit methods have $\beta_k \neq 0$.

Linear Multistep Methods for I.V. Problems

The Adams Family
This familiy of methods is derived from the identity

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} f(t, y(t))dt\,. \qquad (2)$$

With a view to using previously computed values of y_n, we replace $f(t, y)$ by the polynomial of degree $k-1$ passing through the k points

$$(t_{n-k+1}, f_{n-k+1}),\ \ldots, (t_{n-1}, f_{n-1}), (t_n, f_n)\,.$$

Using a constant step size, this polynomial can be written (in the Newton backward difference form) as follows

$$P_{k-1}(r) = f_n + r\nabla f_n + \frac{r(r+1)}{2!}\nabla^2 f_n + \ldots + \frac{r(r+1)\ldots(r+k-2)}{(k-1)!}\nabla^{k-1} f_n\,.$$

donde $t = t_n + rh$, $r \in [-(k-1), 0]$.

Linear Multistep Methods for I.V. Problems

Then, the Adams-Bashforth method of k steps has the form

$$y_{n+1} - y_n = \int_{t_n}^{t_{n+1}} P_{k-1}(t)dt\,.$$

and by integrating the polynomial

$$y_{n+1} - y_n = \int_0^1 P_{k-1}(r)hdr = h\sum_{i=0}^{k-1} \gamma_i \nabla^i f_n = \sum_{i=0}^{k-1} \gamma_i \Delta^i f_{n-i} \qquad (3)$$

where $\gamma_i = (-1)^i \int_0^1 \begin{pmatrix} -r \\ i \end{pmatrix} dr$

$$\gamma_0 = \int_0^1 dr = 1,\;\; \gamma_1 = \int_0^1 rdr = \frac{1}{2},\;\; \gamma_2 = \int_0^1 \frac{r(r+1)}{2}dr = \frac{5}{2}\,,$$

$$\gamma_3 = \int_0^1 \frac{r(r+1)(r+2)}{6}dr = \frac{3}{8}\,,\;\; \dots$$

Linear Multistep Methods for I.V. Problems

The first Adams-Bashforth methods are then:

$$
\begin{array}{ll}
y_{n+1} = y_n + hf_n & \text{(AB1), Euler Method} \\
y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1}) & \text{(AB2)} \\
y_{n+1} = y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2}) & \text{(AB3)} \\
y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) & \text{(AB4)}
\end{array}
\tag{4}
$$

and so on.

Linear Multistep Methods for I.V. Problems

The Adams-Bashforth methods are **explicit methods**. **Implicit Adams methods** are derived by integrating the order k polynomial passing through the points

$$(t_{n+1}, f_{n+1}),\ (t_n, f_n),\ (t_{n-1}, f_{n-1})\dots, (t_{n-k+1}, f_{n-k+1})\,.$$

Note that we are considering the additional data point (t_{n+1}, f_{n+1}). The corresponding polynomial is

$$P_k(r) = f_{n+1} + r\nabla f_{n+1} + \frac{r(r+1)}{2!}\nabla^2 f_{n+1} + \dots + \frac{r(r+1)\dots(r+k-1)}{k!}\nabla^k f_{n+1}\,.$$

where $t = t_{n+1} + rh$, $r \in [-(k-1), 0]$.

Linear Multistep Methods for I.V. Problems

Then, for the implicit methods we will have

$$y_{n+1} - y_n = \int_{-1}^{0} P_k(r) h dr = h \sum_{i=0}^{k} \gamma_i' \nabla^i f_{n+1} = h \sum_{i=0}^{k} \gamma_i' \Delta^i f_{n+1-i} \,,$$

where

$$\gamma_0' = \int_{-1}^{0} dr = 1, \;\; \gamma_1' = \int_{-1}^{0} r dr = -\frac{1}{2}, \;\; \gamma_2' = \int_{-1}^{0} \frac{r(r+1)}{2} dr = -\frac{1}{12} \,,$$

$$\gamma_3' = \int_{-1}^{0} \frac{r(r+1)(r+2)}{6} dr = -\frac{1}{24} \,, \;\; \dots$$

Linear Multistep Methods for I.V. Problems

The implicit Adams methods are called Adams-Moulton methods (AM). The first AM methods are:

$$\begin{aligned}
y_{n+1} &= y_n + hf_{n+1} = y_n + hf(t_{n+1}y_{n+1}) && \text{(AM0), implicit Euler method}\\
y_{n+1} &= y_n + \frac{h}{2}(f_{n+1} + f_n) && \text{(AM1), trapezoidal method}\\
y_{n+1} &= y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1}) && \text{(AM2)}\\
y_{n+1} &= y_n + \frac{h}{24}(9f_{n+1} - 19f_n - 5f_{n-1} + f_{n-2}) && \text{(AM3)}
\end{aligned} \tag{5}$$

Note that, as typical for any implicit method, we have to solve a nonlinear equation.

Predictor-Corrector Methods for I.V. Problems

Consider an Adams-Moulton scheme with 2 steps (AM2), order 3. We start from the initial values y_0, y_1 (the value of y_1 can be obtained, for instance, by using a Taylor method of order 4) and we have to compute y_2. In order to compute this value we have to solve the, in general, nonlinear equation

$$y_2 = y_1 + \frac{h}{12}(5f(t_2, y_2) + 8f(t_1, y_1) - f(t_0, y_0))$$

So far we have solved this kind of equations by using, for instance, a fixed point method. As we already know, the fixed point method is a good choice if the initial value is near enough to the solution of the equation and h is small enough.

Predictor-Corrector Methods for I.V. Problems

In general, at each step $n+1$ we have to solve the nonlinear equation

$$y_{n+1} = G(y_{n+1}) = y_n - h\Phi(y_{n+1})\,.$$

In order to apply a fixed point method, a first estimation of the root of the equation would be very convenient. This estimation can be obtained by using, for instance, an Adams-Bashforth method.

Predictor-Corrector Methods for I.V. Problems

In this way, we will have the following **predictor-corrector scheme**: Starting form h and the initial values $(y_0, ..., y_{k-1})$, the following scheme is repeated at each step n (the values $f_n, f_{n-1}, ..., f_{n-k+1}$ are obtained in previous iterations):

P: Predict by using AB. For instance, by using AB2 and using the known values f_n, f_{n-1},

$$y_{n+1}^{(0)} = y_n + \frac{h}{2}(3f_n - f_{n-1})$$

.

E: Evaluate $f(t_{n+1}, y_{n+1}^{(0)})$

C: Correct by using the AM formula. For example, for AM2 we have:

$$y_{n+1} = y_n + \frac{h}{12}(5f(t_{n+1}, y_{n+1}^{(0)}) + 8f_n - f_{n-1})$$

E: Evaluate $f_{n+1} = f(t_{n+1}, y_{n+1})$ in order to be apply in subsequent steps.

Predictor-Corrector Methods for I.V. Problems

This algorithm is denoted simbolically by **PECE**. Another possibilities are: **PECECE** (2 fixed point iterations) or **PE(CE)**^{m} (m fixed point iterations).

Example

Consider the Forward Euler method as predictor and the AM of 2nd order as corrector.
The corresponding PECE algorithm reads: $y_n \to y_{n+1}^{(0)} \to y_{n+1}$, where

$$\text{Predictor: } y_{n+1}^{(0)} = y_n + hf(t_n, y_n)$$

$$\text{Corrector: } y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(0)})\right].$$

This is a RK Method of 2nd order!

Predictor-Corrector Methods for I.V. Problems

Example

Consider the ODE

$$y'(x) = -y(x) + 2\cos x, \quad y(0) = 1.$$

(1) Give the 2nd order Adams-Moulton iteration formula.

(2) Consider the Forward Euler Method as predictor and the Adams-Moulton scheme of 2nd order as corrector. Give the corresponding iteration formula.

(3) Use the previous algorithm to evaluate y_1 for $h = 0.1$.

Predictor-Corrector Methods for I.V. Problems

Sol:

∘ The 2nd order Adams-Moulton scheme reads

$$\begin{aligned} y_{k+1} &= y_k + \tfrac{h}{2}\left[f(x_k, y_k) + f(x_{k+1}, y_{k+1})\right] \\ &= y_k + \frac{h}{2}\left[2\cos x_k - y_k + 2\cos x_{k+1} - y_{k+1}\right] \end{aligned}$$

with $y_0 = y(0) = 1$. Therefore, the iteration formula is:

$$y_{k+1} = \frac{1 - h/2}{1 + h/2} y_k + \frac{h}{1 + h/2}\left[\cos x_k + \cos(x_k + h)\right],$$

$$k = 0, 1, 2, ...$$

Predictor-Corrector Methods for I.V. Problems

Sol (Cont.):

◦ The Predictor-Corrector formulas are

$$\begin{aligned} y_{k+1}^{(0)} &= y_k + f(x_k, y_k)h, \\ y_{k+1} &= y_k + \frac{h}{2}\left[f(x_k, y_k) + f(x_{k+1}, y_{k+1}^{(0)})\right]. \end{aligned}$$

Then, we have

$$\begin{aligned} y_{k+1} &= y_k + \frac{h}{2}[2\cos x_k - y_k + \\ &\quad 2\cos x_{k+1} - h(2\cos x_k - y_k) - y_k] \\ &= \left[1 - h + \frac{h^2}{2}\right] y_k + \\ &\quad h(\cos x_k + \cos(x_k + h)) - h^2 \cos x_k. \end{aligned}$$

Predictor-Corrector Methods for I.V. Problems

Sol (Cont.):

∘ By using the previous formulas, we obtain

$$y_1 \approx 1.0945.$$

Th exact solution at the point $x = x_1$ is $y(x_1) = (\cos x_1 + \sin x_1)|_{x=0.01} \approx 1.0948$. The approximate solution x_1 by using the Adams-Moulton method is $y_1|_{AM} \approx 1.0948$. Then, in this case, $y_1|_{AM}$ is more accurate than the corresponding solution of the PC scheme; however, the PC approximation is better than the approximation obtained with a single use of a predictor method.