# 3 Multistep, Predictor-Corrector, and Implicit methods

In this section, we will introduce methods that may be as accurate as high-order Runge-Kutta methods but will require fewer function evaluations.

We will also introduce implicit methods, whose significance will become clearer in a later section.

3.1 Idea behind multistep methods

The figure on the right illustrates the (familiar) fact that if you know $y'(x_i)$, i.e. the *slope* of $y(x)$, then you can compute a first-order accurate approximation $Y_{i+1}^{1\text{st order}}$ to the solution y_{i+1}.

Likewise, if you know the slope and the *curvature* of your solution at a given point, you can compute a second-order accurate approximation, $Y_{i+1}^{2\text{nd order}}$, to the solution at the next step.
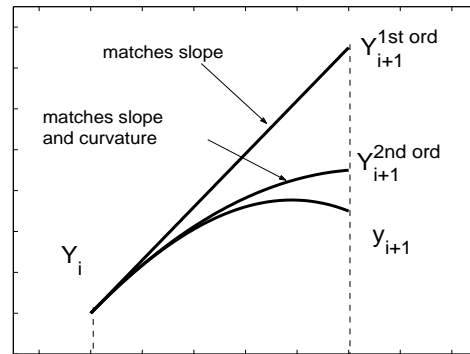
Now, recall that curvature is proportional to y''. This motivates the following.

Question: How can we find approximation to y_i'' using already computed values Y_{i-k}, $k = 0, 1, 2, \ldots$?

Answer: Note that

$$y_i'' \approx \frac{y_i' - y_{i-1}'}{h} = \frac{f_i - f_{i-1}}{h}. \tag{3.1}$$

Here and below we will use the notation f_i in two slightly different ways:

$$f_i \equiv f(x_i, y_i) \qquad \text{or} \qquad f_i \equiv f(x_i, Y_i) \tag{3.2}$$

whenever this does not cause any confusion.

Continuing with Eq. (3.1), we can state it more specifically by writing

$$y_i'' = \frac{y_i' - y_{i-1}'}{h} + O(h) = \frac{f_i - f_{i-1}}{h} + O(h), \tag{3.3}$$

where we will compute the $O(h)$ term later. For now, we use (3.3) to approximate y_{i+1} as follows:

$$
\begin{aligned}
y_{i+1} = y(x_i + h) = \quad & y_i + h y_i' + \frac{h^2}{2} y_i'' + O(h^3) \\
= \quad & y_i + h f_i + \frac{h^2}{2} \left(\frac{f_i - f_{i-1}}{h} + O(h) \right) + O(h^3) \\
= \quad & y_i + h \left(\frac{3}{2} f_i - \frac{1}{2} f_{i-1} \right) + O(h^3).
\end{aligned} \tag{3.4}
$$

Remark 1: To start the corresponding finite-difference method, i.e.

$$Y_{i+1} = Y_i + h \left(\frac{3}{2} f_i - \frac{1}{2} f_{i-1} \right) \tag{3.5}$$

(now we use f_i as $f(x_i, Y_i)$), one needs *two* initial points of the solution, Y_0 and Y_1. These can be computed, e.g., by the simple Euler method; this is discussed in more detail in Section 3.4.

<u>Remark 2</u>: Equation (3.4) becomes *exact* rather than approximate if $y(x) = p_2(x) \equiv ax^2 + bx + c$ is a second-degree polynomial in x. Indeed, in such a case,

$$y_i' = 2ax_i + b, \qquad \text{and} \qquad y_i'' = 2a = \frac{y_i' - y_{i-1}'}{h}\,; \qquad (3.6)$$

(note the exact equality in the last formula). We will use this remark later on.

Method (3.5) is of the second order. If we want to obtain a third-order method along the same lines, we need to use the third derivative of the solution:

$$y_i''' = \frac{y_i' - 2y_{i-1}' + y_{i-2}'}{h^2} + O(h) \qquad (3.7)$$

(you will be asked to verify this equation in one of the homework problems). Then we proceed as in Eq. (3.4), namely:

$$y_{i+1} = y_i + hy_i' + \frac{h^2}{2}y_i'' + \frac{h^3}{6}y_i''' + O(h^4)\,. \qquad (3.8)$$

If you now try to substitute the expression on the r.h.s. of (3.3) for y_i'', you notice that you actually need an expression for the $O(h)$-term there that would have accuracy of $O(h^2)$. Here is the corresponding calculation:

$$
\begin{aligned}
\frac{y_i' - y_{i-1}'}{h} &= \frac{y'(x_i) - y'(x_{i-1})}{h} \\
&= \frac{y_i' - \left[y_i' - hy_i'' + \frac{h^2}{2}y_i''' + O(h^3)\right]}{h} \qquad (3.9) \\
&= y_i'' - \frac{h}{2}y_i''' + O(h^2),
\end{aligned}
$$

whence

$$y_i'' = \frac{y_i' - y_{i-1}'}{h} + \frac{h}{2}y_i''' + O(h^2)\,. \qquad (3.10)$$

To complete the derivation of the third-order finite-difference method, we substitute Eqs. (3.10), (3.7), and $y_i' = f_i$ etc. into Eq. (3.8). The result is:

$$Y_{i+1} = Y_i + \frac{h}{12}[23f_i - 16f_{i-1} + 5f_{i-2}]\,; \qquad (3.11)$$

the local truncation error of this method is $O(h^4)$. Method (3.11) is called the 3rd-order *Adams–Bashforth* method.

Similarly, one can derive higher-order Adams–Bashforth methods. For example, the 4th-order Adams–Bashforth method is

$$Y_{i+1} = Y_i + \frac{h}{24}[55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}]\,. \qquad (3.12)$$

Methods like (3.5), (3.11), and (3.12) are called *multistep* methods. To start a multistep method, one requires more than one initial point of the solution (in the examples considered above, the number of required initial points equals the order of the method).

Comparison of multistep and Runge–Kutta methods

The advantage of multistep over single-step RK methods of the same accuracy is that the multistep methods require only one function evaluation per step, while, e.g., the cRK method requires 4, and the RK–Fehlberg method, 6, function evaluations.

The disadvantages of multistep methods are primarily related to the difficulties that one would face by making them use a varying step size. In contrast, for single-step RK methods, this is a straightforward procedure, as we learned in Lecture 2. Below I list some of those difficulties for multistep methods.

1. Using a variable step size requires either an interpolation of the numerical solution from a variable- onto a constant-step size grid, or using rather complicated coefficients involving ratios of step sizes, say h_i/h_{i-1}, instead of the constants like 3/2, 1/2 in formulae (3.5) etc.

2. If function $f(x, y)$ changes abruptly at $x = x_{\text{change}}$, as in the problems in Homework # 2 involving an opening parachute, then a RK method would require only one step to adjust to the new values of f beyond x_{change}. In contrast, a k-step method would take k steps to "recover". That is, over those k steps taken past the point x_{change} it would be using the information about the "pre-x_{change}" values of f, which are irrelevant for what is happening beyond x_{change}.[2]

3. In Sec. 3.6 we will learn about error control in so-called predictor–corrector methods (see Sec. 3.5), which is based on a key fact examplified by Eqs. (3.35), (3.36), and the pair of equations below them. Namely, the local truncation errors of the predictor and corrector equations must be proportional to each other. However, the derivation of this fact, part of which is given in Sec. 3.8, hinges on the assumption that all steps have equal size.

Despite the aforementioned difficulties, multistep methods with a variable step size have been constructed and studied, although their theory goes far beyond the material covered in this course.

3.2 An alternative way to derive formulae for multistep methods

Recall that the 2nd-order Adams–Bashforth method (3.5) was *exact* on solutions $y(x)$ that are 2nd-degree polynomials: $y(x) = p_2(x)$ (see Remark 2 after Eq. (3.4)). Similarly, one expects that the 3rd-order Adams–Bashforth method should be exact for $y(x) = p_3(x)$. We will now use this observation to derive the formula for this method, Eq. (3.11), in a different manner than in Sec. 3.1.

To begin, we take, according to the above observation, $f(x, y) = y'(x) = (p_3(x))' = p_2(x)$, i.e. a 2nd-degree polynomial in x. We now integrate the differential equation $y' = f(x, y)$ from x_i to x_{i+1} and obtain:

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y(x))dx. \tag{3.13}$$

Let us approximate the integral by a quadrature formula, as follows:

$$\int_{x_i}^{x_{i+1}} f(x, y(x))dx \approx h(b_0 f_i + b_1 f_{i-1} + b_2 f_{i-2}) \tag{3.14}$$

and *require* that the above equation hold *exactly*, rather than approximately, for any $f(x, y(x)) = p_2(x)$. This is equivalent to requiring that (3.14) hold exactly for $f = 1$, $f = x$, and $f = x^2$. Without loss of generality[3], one can set $x_i = 0$ and then rewrite Eq. (3.14) for the above three

---

[2]This reason was suggested by Mr. Jacob Wahlen–Strothman, who took this course in 2012.

[3]In a homework problem, you will be asked to show this.

forms of f:

$$
\begin{aligned}
\text{for } f = 1: && \int_0^h 1 \, dx &= h &= h \left(b_0 \cdot 1 + b_1 \cdot 1 + b_2 \cdot 1 \right) \\
\text{for } f = x: && \int_0^h x \, dx &= \tfrac{1}{2} h^2 &= h \left(b_0 \cdot 0 + b_1 \cdot (-h) + b_2 \cdot (-2h) \right) \\
\text{for } f = x^2: && \int_0^h x^2 \, dx &= \tfrac{1}{3} h^3 &= h \left(b_0 \cdot 0 + b_1 \cdot (-h)^2 + b_2 \cdot (-2h)^2 \right).
\end{aligned}
\tag{3.15}
$$

Equations (3.15) constitute a linear system of 3 equations for 3 unknowns b_0, b_1, and b_2. Solving it, we obtain

$$
b_0 = \frac{23}{12}, \qquad b_1 = -\frac{16}{12}, \qquad b_2 = \frac{5}{12},
$$

which in combination with Eq. (3.14) yields the same method as (3.11). Methods of higher order can be obtained similarly.

3.3   A more general form of multistep methods, with examples

The Adams–Bashforth methods above have the following common form:

$$
Y_{i+1} - Y_i = h \sum_{k=0}^{N} b_k f_{i-k} .
\tag{3.16}
$$

As has been shown in Sec. 3.2, the sum on the r.h.s. approximates

$$
\int_{x_i}^{x_{i+1}} f(x, y(x)) dx .
$$

Let us now consider multistep methods of a more general form:

$$
Y_{i+1} - \sum_{k=0}^{M} a_k Y_{i-k} = h \sum_{k=0}^{N} b_k f_{i-k} ,
\tag{3.17a}
$$

where

$$
\sum_{k=0}^{M} a_k = 1.
\tag{3.17b}
$$

The last condition intuitively makes sense since the sum on the l.h.s. of (3.17a) replaces the term $1 \cdot Y_i$ in (3.16). Note that if we rewrite the l.h.s. of (3.17a) as

$$
\sum_{k=0}^{M} a_k (Y_{i+1} - Y_{i-k}),
$$

where we have used (3.17b), then the r.h.s. of (3.17a) acquires an interpretation that is similar to that of the r.h.s. of (3.16). Namely, it approximates

$$
\sum_{k=0}^{M} a_k \int_{x_{i-k}}^{x_{i+1}} f(x, y(x)) dx .
$$

In the next Lecture, we will discover that many methods of the form (3.17) have a serious flaw in them, but for now let us consider two particular examples, focusing only on the accuracy of the following methods.

Simple center-difference (Leap-frog) method

Recall that

$$\frac{y_i - y_{i-1}}{h} = y_i' + O(h). \tag{3.18}$$

However[4],

$$\frac{y_{i+1} - y_{i-1}}{2h} = y_i' + O(h^2). \tag{3.19}$$

Thus, the l.h.s. of (3.19) provides a more accurate approximation to y_i' than does the l.h.s. of (3.18). So we use Eq. (3.19) to produce a 2nd-order method:

$$Y_{i+1} = Y_{i-1} + 2hf_i, \tag{3.20}$$

which is of the form (3.17). We need both Y_0 and Y_1 to start this method.

A divergent third-order method

(The term "divergent" will be explained in the next Lecture.)

Let us try to increase the order of method (3.20) from 2nd to 3rd by including extra terms into the scheme:

$$Y_{i+1} - (a_0 Y_i + a_1 Y_{i-1} + a_2 Y_{i-2}) = b_0 h f_i, \tag{3.21}$$

where we now require that the local truncation error of (3.21) be $O(h^4)$. We can follow the derivation found either in Sec. 3.1 (Taylor-series expansion) or Sec. 3.2 (requiring that (3.21) hold true for $y = p_3(x)$) to obtain the values of the coefficients a_0 through a_2, and b_0. The result is:

$$Y_{i+1} + \frac{3}{2}Y_i - 3Y_{i-1} + \frac{1}{2}Y_{i-2} = 3h f_i. \tag{3.22}$$

Supposedly, method (3.22) is more accurate than the Leap-frog method (3.20). However, we will show in the next Lecture that method (3.22) is *completely useless* for numerical computations.

3.4 Starting a multistep method

To start any of the single-step methods, considered in Lectures 1 and 2, one only needs to know the initial condition, $Y_0 = y_0$, at $x = x_0$. To start any multistep method, one needs to know the numerical solution at several points. For example, to start an Adams–Bashforth method of order m, one would need the values $Y_0, \dots, Y_{m-1}$ (see Eqs. (3.5), (3.11), and (3.12)). That is, to start an mth-order method, one needs to know the solution at the first m points. We will now address the following **question**:

Suppose that we want to start a multistep method of order m using the values $Y_1, \dots, Y_{m-1}$ that have been computed by a starting (single-step) method of order n. What should the order n of the starting method be so as **not to compromise the order m of the multistep method**?

First, it is clear that if $n \geq m$, then the local error made in the computation of $Y_1, \dots, Y_{m-1}$ and of the terms on the r.h.s. of (3.16) and (3.17) will be at least as small (in the order of

---

[4]You will be asked to verify this.

magnitude sense) as the local error of the multistep method. So, using a starting method whose order is *no less* than the order of the multistep method will not degrade the accuracy of the latter method. But is it possible to use a starting method with $n < m$ for the same end result?

We will now show, using method (3.16) as an example, that it is possible to take $n = m - 1$ (i.e., the starting method's order may be one less than the multistep method's order).[5]

The local truncation errors of Y_1 through Y_{m-1} are $O(h^{n+1})$. Then the error contributed to Y_m from the second term (i.e., from Y_i with $i = m - 1$) on the l.h.s. of (3.16), is $O(h^{n+1})$:

$$\text{error of l.h.s. of } (3.16) = O(h^{n+1}). \tag{3.23}$$

Next, if f_i through f_{i-N} on the r.h.s. were calculated using the exact solution $y(x)$, then the error of the r.h.s. would have been $\underline{O(h^{m+1})}$. Indeed, this error is just the local truncation error of method (3.16) that arises due to the approximation of $\int_{x_i}^{x_{i+1}} f(x, y(x))dx$ by $h \sum_{k=0}^{N} b_k f_{i-k}$. However, the f_{i-k}'s are calculated using values Y_1 through Y_{m-1} which themselves have been obtained with the error $O(h^{n+1})$ of the starting method. Then the error of each f_{i-k} is also $\underline{O(h^{n+1})}$.[6] Therefore, combining the two underlined errors in the text above, one has:

$$\text{error of r.h.s. of } (3.16) = O(h^{m+1}) + h \cdot O(h^{n+1}) = \max\{O(h^{n+2}), O(h^{m+1})\}. \tag{3.24}$$

Let us summarize. The error on the l.h.s. of (3.16), which occurs due to Y_i (with $i = m - 1$) being computed by an nth-order starting method, is $O(h^{n+1})$. The error on the r.h.s of (3.16) is given by (3.24). Therefore:

$$\text{combined error in } (3.16) = O(h^{n+1}) \text{ from l.h.s.} + \max\{O(h^{n+2}), O(h^{m+1})\} \text{ from r.h.s.} = O(h^{n+1}). \tag{3.25}$$

(recall that we are only interested in the situation where $n < m$).

Now, in order not to decrease the accuracy of the multistep method, this error must satisfy two criteria:

(i) It must have the same order of magnitude as the global error at the end of the computation, i.e., $O(h^m)$; and $\underline{\text{in addition,}}$

(ii) It may *propagate* to the next computed solution, i.e., to Y_{i+2}, but it *must not accumulate* at each step with other errors of the same magnitude. (See an explanation below.)

One can easily see that criterion (i) is indeed satisfied for $n + 1 = m$, i.e., when $n = m - 1$. As for criterion (ii), it is also satisfied, although at first sight that might seem strange. Indeed, we have seen in Lecture 1 that if a local error is $O(h^{n+1})$, then the global error should be $O(1/h) \cdot O(h^{n+1}) = O(h^n)$. However, this is not so for the situation considered here. To see why, it will suffice to repeat the above error estimates for Y_{i+1} with $i = m$. We do so in the next paragraph.

Since, as we have just shown, the error in Y_m is $O(h^{n+1})$, then the error on the l.h.s. of (3.16) with $i = m$ is given by (3.23). The error on the r.h.s. is given by (3.24), which follows from

---

[5]Unfortunately, I was unable to find any detailed published proof of this result, and so the derivation found below is my own. As such, it is subject to mistakes ⌣. However, a set of Matlab codes accompanying this Lecture where the 3rd-order Adams–Bashforth method (3.11) can be started using the modified Euler, Midpoint, or simple Euler method, shows that if not this derivation itself, than at least its result is probably correct.

[6]If the last statement is not clear, do not worry and just read on. More details about it are presented in the derivation of Eq. (3.32) in the next Section, and also in the Appendix.

exactly the same considerations as given before that equation. Therefore, the combined error in determining Y_{m+1} is nothing but the $O(h^{n+1})$ term in (3.25), which was originally generated by the starting method. Thus, this $O(h^{n+1})$ error *propagates* from one step to the next, but does *not* accumulate with other $O(h^{n+1})$ terms (simply because there are no such terms). Hence, it does *not* grow to become $O(h^n)$.

With a little more work, a similar derivation can be extended to methods of the more general form (3.17). (E.g., the Leap-frog method (3.20) is a particular representative of the latter case.)

Let us summarize. We have shown that in order to have a multistep method (3.16) of order m, the order n of the starting single-step method should be no lower than $(m-1)$. However, in Sec. 3.6 we will see that there is a reason why one may want to use a single-step method of order m (as opposed to $(m-1)$) to start a multistep method of order m.

3.5 Predictor–corrector methods: General form

Let us recall the Modified Euler method introduced in Lecture 1 and write it here using slightly different notations:

$$
\begin{aligned}
Y_{i+1}^p &= Y_i + hf_i \\
Y_{i+1}^c &= Y_i + \tfrac{1}{2}h\left(f_i + f(x_{i+1}, Y_{i+1}^p)\right) \\
Y_{i+1} &= Y_{i+1}^c \,.
\end{aligned}
\tag{3.26}
$$

We can interpret the above as follows: We first *predict* the new value of the solution Y_{i+1} by the first equation, and then *correct* it by the second equation. Methods of this kind are called *predictor–corrector* (P–C) methods.

Question: What is the optimal relation between the orders of the predictor and corrector equations?

Answer: The example of the Modified Euler method suggests that the order of the corrector should be one higher than that of the predictor. More precisely, the following theorem holds:

Theorem If the order of the corrector equation is n, then the order of the corresponding P–C method is also n, provided that the order of the predictor equation is no less than $n-1$.

Proof We will assume that the global error of the corrector equation by itself is $O(h^n)$ and the global error of the predictor equation by itself is $O(h^{n-1})$. Then we will prove that the global error of the combined P–C method is $O(h^n)$.

The general forms of the predictor and corector equations are, respectively:

$$
\text{Predictor:} \qquad Y_{i+1}^p = Y_{i-Q} + h\sum_{k=0}^{N} p_k f_{i-k}, \tag{3.27}
$$

$$
\text{Corrector:} \quad Y_{i+1}^c = Y_{i-D} + h\sum_{k=0}^{M} c_k f_{i-k} + hc_{-1}f(x_{i+1}, Y_{i+1}^p)\,. \tag{3.28}
$$

In the above two equations, Q, D, N, M are some integer nonnegative numbers. (One of the questions at the end of this Lecture asks you to represent Eq. (3.26) in the form (3.27), (3.28), i.e. to give values for Q, D, N, M and the coefficients p_k's and c_k's.)

As we have done in previous derivations, let us assume that all computed values Y_{i-k}, $k = 0, 1, 2, \ldots$ coincide with the exact solution at the coresponding points: $Y_{i-k} = y_{i-k}$. Then

we can use the identity

$$y_{i+1} = y_{i-Q} + (y_{i+1} - y_{i-Q}) \stackrel{\text{see } (3.13)}{=} y_{i-Q} + \int_{x_{i-Q}}^{x_{i+1}} y'(x)dx = Y_{i-Q} + \int_{x_{i-Q}}^{x_{i+1}} f(x, y(x))dx$$

and rewrite Eq. (3.27) as:

$$Y_{i+1}^p = Y_{i-Q} + \int_{x_{i-Q}}^{x_{i+1}} f(x, y(x))dx + \left(h\sum_{k=0}^{N} p_k f_{i-k} - \int_{x_{i-Q}}^{x_{i+1}} f(x, y(x))dx \right) \quad \Rightarrow \quad Y_{i+1}^p = y_{i+1} + E_P .$$
$$(3.29)$$

Here E_P is the error made by replacing the exact integral

$$\int_{x_{i-Q}}^{x_{i+1}} f(x, y(x))dx$$

by the linear combination of f_{i-k}'s, found on the r.h.s. of (3.27). Since, by the condition of the Theorem, the global error of the predictor equation is $O(h^{n-1})$, then the local truncation error E_P has the order of $O(h^{(n-1)+1}) = O(h^n)$.

Similarly, Eq. (3.28) is rewritten as

$$Y_{i+1}^c = y_{i+1} + E_C + hc_{-1}\left(f(x_{i+1}, Y_{i+1}^p) - f(x_{i+1}, y_{i+1}) \right) . \qquad (3.30)$$

Here E_C is the error obtained by replacing the exact integral

$$\int_{x_{i-D}}^{x_{i+1}} f(x, y(x))dx$$

by the quadrature formula

$$h\sum_{k=-1}^{M} c_k f_{i-k}$$

(note that the lower limit of the summation is different from that in (3.28)!). The last term on the r.h.s. of (3.30) occurs because, unlike all previously computed Y_{i-k}'s, the $Y_{i+1}^p \neq y_{i+1}$.

To complete the proof,[7] we need to show that $Y_{i+1}^c - y_{i+1} = O(h^{n+1})$ in (3.30). By the condition of the Theorem, the corrector equation has order n, and hence the local truncation error $E_C = O(h^{n+1})$. Then all that remains to be estimated is the last term on the r.h.s. of (3.30). To that end, we recall that f satisfies the Lipschitz condition with respect to y, whence

$$|f(x_{i+1}, Y_{i+1}^p) - f(x_{i+1}, y_{i+1})| \leq L|Y_{i+1}^p - y_{i+1}| = L\,|E_P|, \qquad (3.31)$$

where L is the Lipschitz constant. Combining Eqs. (3.30) and (3.31) and using the triangle inequality (1.5), we finally obtain

$$|Y_{i+1}^c - y_{i+1}| \leq |E_C| + hL|E_P| = O(h^{n+1}) + h \cdot O(h^n) = O(h^{n+1}), \qquad (3.32)$$

which proves that the P–C method has the local truncation error of order $n + 1$, and hence is the nth-order method. ***q.e.d.***

---

[7]At this point, you have probably forgotten what we are proving. Pause, re-read the Theorem's statement, and then come back to finish the reading.

We now present two P–C pairs that in applications are sometimes preferred[8] over the Modified Euler method. The first pair is:

$$\begin{aligned}
\text{Predictor}: \quad Y_{i+1}^{p} &= Y_i + \tfrac{1}{2}h(3f_i - f_{i-1}) \\
\text{Corrector}: \quad Y_{i+1}^{c} &= Y_i + \tfrac{1}{2}h\left(f_i + f_{i+1}^{p}\right),
\end{aligned} \tag{3.33}$$

where $f_{i+1}^{p} = f(x_{i+1}, Y_{i+1}^{p})$. The order of the P–C method (3.33) is two.

The other pair is:

$$\begin{aligned}
&\text{Predictor:} \quad \text{4th-order Adams–Bashforth} \\
&Y_{i+1}^{p} = Y_i + \tfrac{1}{24}h(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}). \\
&\text{Corrector:} \quad \text{4th-order Adams–Moulton} \\
&Y_{i+1}^{c} = Y_i + \tfrac{1}{24}h(9f_{i+1}^{p} + 19f_i - 5f_{i-1} + f_{i-2}),
\end{aligned} \tag{3.34}$$

This P–C method as a whole has the same name as its corrector equation: the 4th-order Adams–Moulton.

3.6   Predictor–corrector methods: Error monitoring

An observation one can make from Eqs. (3.33) is that *both* the predictor and corrector equations have the order two (i.e. the local truncation errors of $O(h^3)$). In view of the Theorem of the previous subsection, this may seem to be unnecessary. Indeed, the contribution of the predictor's local truncation error is $h \cdot O(h^3) = O(h^4)$ (see Eq. (3.32)), while the local truncation error of the corrector equation (which determines that of the entire P–C method) is only $O(h^3)$. There is, however, an important consideration because of which method (3.33) may be preferred over the Modified Euler. Namely, one can monitor the error size in (3.33), whereas the Modified Euler does not give its user such a capability. Below we explain this statement in detail. A similar treatment can be applied to the Adams–Moulton method (3.34).

The key fact is that the local truncation errors of the predictor and correction equations (3.33) are *proportional to each other* in the leading order:

$$\begin{aligned}
y_{i+1} - Y_{i+1}^{p} &= \tfrac{5}{12}h^3 y_i''' + O(h^4), & (3.35) \\
y_{i+1} - Y_{i+1}^{c} &= -\tfrac{1}{12}h^3 y_i''' + O(h^4). & (3.36)
\end{aligned}$$

For the reader's information, the analogues of the above estimates for the Adams–Moulton method (3.34) are:

$$\begin{aligned}
&\text{4th-order Adams–Moulton method:} \\
&y_{i+1} - Y_{i+1}^{p} = \frac{251}{720}h^5 y_i^{(5)} + O(h^6), \\
&y_{i+1} - Y_{i+1}^{c} = -\frac{19}{720}h^5 y_i^{(5)} + O(h^6).
\end{aligned}$$

We derive (3.36) in Appendix 1 to this Lecture, while the derivation of (3.35) is left as an exercise. Here we only note that the **derivation of (3.36) hinges upon the fact that** $y_i - Y_i^{p} = O(h^3)$, **which is guaranteed by (3.35). Otherwise, i.e. if** $y_i - Y_i^{p} = O(h^2)$, **as in the predictor for the Modified Euler method, the term on the r.h.s. of (3.36) would not have had such a simple form.** See Remark 2 below Eq. (3.40) for details.

---

[8]In the next Section we will explain why this is so.

We will now explain how (3.35) and (3.36) can be used together to monitor the error of the P–C method (3.33). From (3.36) we obtain the error of the corrector equation:

$$|\epsilon^c_{i+1}| \approx \frac{1}{12} h^3 |y'''_i| . \tag{3.37}$$

On the other hand, from Eqs. (3.35) and (3.36) together, we have

$$|Y^p_{i+1} - Y^c_{i+1}| \approx \left(\frac{5}{12} + \frac{1}{12} \right) h^3 |y'''_i|. \tag{3.38}$$

Thus, from (3.37) and (3.38) one can estimate the error via the difference of the predicted and corrected values of the solution:

$$|\epsilon^c_{i+1}| \approx \frac{1}{6} |Y^p_{i+1} - Y^c_{i+1}| . \tag{3.39}$$

Moreover, Eqs. (3.35) and (3.36) can also be used to obtain a higher-order method than (3.33), because they imply that

$$y_{i+1} = \frac{1}{6} \left(Y^p_{i+1} + 5 Y^c_{i+1} \right) + O(h^4) .$$

Hence

$$Y_{i+1} = \frac{1}{6} \left(Y^p_{i+1} + 5 Y^c_{i+1} \right) \tag{3.40}$$

produces a more accurate approximation to the solution than either Y^p_{i+1} or Y^c_{i+1} alone. (Note a similarity with the Romberg extrapolation described in Lecture 1.)

Thus, Eqs. (3.33), (3.39), and (3.40) can be used to program a P–C method that allows the user to monitor the error. Namely:

1. Compute Y^p_i and Y^c_i from (3.33). Compute the improved solution from (3.40).
2. Estimate the error of Y^c_{i+1} using (3.39).

The above procedure produces a 3rd-order-accurate solution (3.40) while monitoring the error size of the associated 2nd-order method (3.33). This is analogous to the situation for the adaptive RK methods described in Lecture 2, which computed an nth-order accurate solution while controlling the error of a related method of the lower order, $(n-1)$.

<u>Remark 1</u> As we have just discussed, using the predictor and corrector equations of the same order has the advantage of allowing one to monitor the error. However, it may have a disadvantage of making such schemes less *stable* compared to schemes where the predictor's order is one less than that of the corrector. (We will study the concept of stability in the next Lecture.) Thus, choosing a particular P–C pair may depend on the application, and a significant body of research has been devoted to this issue.

<u>Remark 2</u> Suppose we plan to use a P–C method with the predictor and corrector equations having the same order, say m, so as to monitor the error, as described above. If the predictor equation comes from a multistep method of the form (3.16) (as, e.g., in (3.33) or (3.34)), then we need to re-examine the question addressed in Sec. 3.4. Namely, what order starting method should we use for the predictor equation to be able to monitor the error?   In Section 3.4 we showed that a starting method of order $(m-1)$ would suffice to make the predictor method have order m. More precisely, the global error of the predictor method is $O(h^m)$ in this case. However, the *local truncation error* of such a predictor method is also $O(h^m)$ and *not* $O(h^{m+1})$! [9]

---

[9]This error made by the starting method would propagate (but not accumulate) to the last computed value of the numerical solution, as we showed in Sec. 3.4.

For example, if we start the 2nd-order predictor method in (3.33) by the simple Euler method, then the local truncation error of $O(h^2)$ that the simple Euler made in computing Y_1^p, will propagate up to Y_i for any i. Such an error will invalidate the derivation of the local truncation error of the corrector equation, found in Appendix 1 (Sec. 3.8). Thus, we conclude: **If you want to be able to monitor the error in a P–C method where both the predictor and corrector equations are multistep methods of order m, you need to start the predictor equation by an <u>mth-order</u> starting method.**

<u>Remark 3</u> At the end of Sec. 3.1 we listed disadvantages of using multistep methods with a variable step size. Let us reiterate here one of them. Namely, the r.h.s.'s of (3.35) and (3.36) are not simply proportional to y''' but contain other terms if the step size is varied. Therefore, the error monitoring for PC methods with a variable step size is a complicated (if feasible at all) task.

Appendix 2 (Sec. 3.9) discusses a repeated use of the corrector equation within a given PC method for the goal of increasing the overall accuracy of the PC method. This Appendix can be skipped on the first reading.

To summarize on the P–C methods:

1. The P–C methods may provide both high accuracy and the capability of error monitoring, all at a potentially lower computational cost than RK-Fehlberg or RK-Merson methods. For example, the Adams–Moulton method (3.34) has the error of the same (fourth) order as the aforementioned RK methods, while requiring $k + 1$ function evaluations, where k is the number of times one has to iterate the corrector equation. If $k < 4$, then Adam-Moulton requires fewer function evaluations than either RK-Merson or RK-Fehlberg.

2. The adjustment of the step size in P–C methods is awkward (as it is in all multistep methods); see the end of Sec. 3.1 and Remark 3 above.

3. One may ask, why not just halve the step size of the Adams–Bashforth method (which would reduce the global error by a factor of $2^4 = 16$, i.e. a lot) and then use it alone without the Adams–Moulton corrector formula? The answer is this. First, one will then lose the ability to monitor the error. Second, the Adams–Bashforth may sometimes produce a numerical solution which has nothing to do with the exact solution, while the P–C Adams–Moulton's solution will stay close to the exact one. This issue will be discussed in detail in the next Lecture.

3.7   Implicit methods

We noted in Lecture 1 that the simple Euler method is analogous to the left Riemann sums when integrating the differential equation $y' = f(x)$.

The method analogous to the right Riemann sums is:

$$Y_{i+1} = Y_i + hf(x_{i+1}, Y_{i+1}). \qquad (3.41)$$

It is called the implicit Euler, or backward Euler, method. This is a first-order method: Its global error is $O(h)$ and the local truncation error is $O(h^2)$.

right Riemann sums

x_0 x_1 x_2 x_3
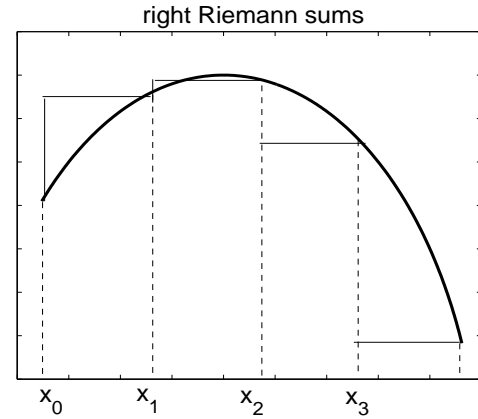
We note that if $f(x, y) = a(x)y + b(x)$, then the implicit equation (3.41) can be easily solved:

$$Y_{i+1} = \frac{Y_i + hb_{i+1}}{1 - ha_{i+1}}. \qquad (3.42)$$

However, for a general nonlinear $f(x, y)$, equation (3.41) cannot be solved exactly, and its solution then has to be found numerically, say, by the Newton-Raphson method.

Question: Why does one want to use the implicit Euler, which is so much harder to solve than the simple Euler method?

Answer: Implicit methods have *stability* properties that are much better than those of explicit methods (like the simple Euler). We will discuss this in the next Lecture.

Note that the last remark about Adams–Bashforth vs. Adams–Moulton, found at the end of the previous Section, is also related to the stability issue. Indeed, the Adams–Bashforth method (the first equation in (3.34)) is explicit, and thus according to the above, it should be not as stable as the Adams–Moulton method (the second equation in (3.34)), which is implicit if one treats Y_{i+1}^p in it as being approximately equal to Y_{i+1}^c.

Finally, we present equations for the Modified implicit Euler method:

$$Y_{i+1} = Y_i + \frac{h}{2} \left(f(x_i, Y_i) + f(x_{i+1}, Y_{i+1}) \right). \qquad (3.43)$$

This is a second-order method.

3.8 Appendix 1: Derivation of (3.36)

Here we derive the local truncation error of the corrector equation in the method (3.33). Assuming, as usual, that $Y_i = y_i$, and using $Y_{i+1}^p = y_{i+1} + O(h^3)$ (since the order of the predictor equation is two), one obtains from the corrector equation of (3.33):

$$
\begin{aligned}
Y_{i+1}^c &= & y_i + \tfrac{1}{2}h \left(y_i' + f(x_{i+1}, y_{i+1} + O(h^3)) \right) \\
&= & y_i + \tfrac{1}{2}h \left(y_i' + f(x_{i+1}, y_{i+1}) + O(h^3) \right) \\
&= & y_i + \tfrac{1}{2}h \left(y_i' + y_{i+1}' + O(h^3) \right) \\
&= & y_i + \tfrac{1}{2}h \left(y_i' + \left[y_i' + hy_i'' + \tfrac{1}{2}h^2 y_i''' + O(h^3) \right] + O(h^3) \right) \\
&= & y_i + hy_i' + \tfrac{1}{2}h^2 y_i'' + \tfrac{1}{4}h^3 y_i''' + O(h^4).
\end{aligned}
\qquad (3.44)
$$

On the other hand, for the exact solution we have the usual Taylor series expansion:

$$y_{i+1} = y_i + h_i' + \frac{1}{2}h^2 y_i'' + \frac{1}{6}h^3 y_i''' + O(h^4). \qquad (3.45)$$

Subtracting (3.44) from (3.45), we obtain

$$y_{i+1} - Y_{i+1}^c = -\frac{1}{12}h^3 y_i''' + O(h^4),$$

which is (3.36).

3.9 Appendix 2: Repeated application of the corrector equation in a PC method

Note that in a PC method, one can apply the corrector formula more than once. For example, for the method (3.33), we will then have:

$$
\begin{aligned}
Y_{i+1}^p &= Y_i + \tfrac{1}{2}h(3f_i - f_{i-1}) \\
Y_{i+1}^{c,1} &= Y_i + \tfrac{1}{2}h\left(f_i + f(x_{i+1}, Y_{i+1}^p)\right), \\
Y_{i+1}^{c,2} &= Y_i + \tfrac{1}{2}h\left(f_i + f(x_{i+1}, Y_{i+1}^{c,1})\right), \\
&\quad \text{etc.} \\
Y_{i+1} &= Y_{i+1}^{c,k}
\end{aligned}
\tag{3.46}
$$

Question: How many times should we apply the corrector equation?

We need to strike a compromise here. If we apply the corrector too many times, then we will waste computer time if each iteration of the corrector changes the solution by less than the truncation error of the method. On the other hand, we may have to apply it more than once in order to make the difference $|Y_{i+1}^{c,k} - Y_{i+1}^{c,k-1}|$ between the last two iterations much smaller than the truncation error of the corrector equation (since the latter error is basically the error of the method; see Eq. (3.36)).

Ideally, one would like to know the conditions under which it is sufficient to apply the corrector equation only <u>once</u>, so that no benefits would be gained by its successive applications. Below we derive such a sufficient condition for the method (3.33)/(3.46). For another P–C method, e.g., Adams–Moulton, an analogous condition can be derived along the same lines.

Suppose the maximum allowed global error of the solution is $\varepsilon_{\text{glob}}$. The allowed local truncation error is then about $\varepsilon_{\text{loc}} = h\varepsilon_{\text{glob}}$ (see Sec. 2.2). We impose two requirements:
(i)   The local truncation error of our solution should not exceed ε_{loc};
(ii) The difference $|Y_{i+1}^{c,2} - Y_{i+1}^{c,1}|$ should be *much smaller* than ε_{loc}.
Requirement (i) is necessary to satisfy in order to obtain the required accuracy of the numerical solution. Requirement (ii) is necessary to satisfy in order to use the corrector equation only once.

Requirement (i) along with Eq. (3.39) yields

$$\frac{1}{6}|Y_{i+1}^p - Y_{i+1}^{c,1}| < \varepsilon_{\text{loc}}, \qquad \Rightarrow \qquad |Y_{i+1}^p - Y_{i+1}^{c,1}| < 6\varepsilon_{\text{loc}}. \tag{3.47}$$

If at some x_i condition (3.47) does not hold, the step size needs to be reduced in accordance with the error's order $|Y_{i+1}^p - Y_{i+1}^{c,1}| = O(h^3)$.

Requirement (ii) implies:

$$
\begin{aligned}
|Y_{i+1}^{c,2} - Y_{i+1}^{c,1}| = \ & \left| \left[Y_i + \tfrac{1}{2}h \left(f_i + f(x_{i+1}, Y_{i+1}^{c,1}) \right) \right] - \right. \\
& \left. \left[Y_i + \tfrac{1}{2}h \left(f_i + f(x_{i+1}, Y_{i+1}^{p}) \right) \right] \right| \\
= \ & \tfrac{1}{2}h \left| f(x_{i+1}, Y_{i+1}^{c,1}) - f(x_{i+1}, Y_{i+1}^{p}) \right| \\
\overset{\text{Lipschitz}}{\leq} \ & \tfrac{1}{2}hL \left| Y_{i+1}^{c,1} - Y_{i+1}^{p} \right| \\
\overset{(3.47)}{\leq} \ & \tfrac{1}{2}hL \cdot 6\varepsilon_{\text{loc}} .
\end{aligned}
\tag{3.48}
$$

Thus, a sufficient condition for $|Y_{i+1}^{c,2} - Y_{i+1}^{c,1}| \ll \varepsilon_{\text{loc}}$ to hold is

$$
\frac{1}{2}hL \cdot 6\varepsilon_{\text{loc}} \ll \varepsilon_{\text{loc}}, \qquad \text{or} \qquad hL \ll \frac{1}{3} .
\tag{3.49}
$$

If condition (3.49) is satisfied, then a single application of the corrector equation is adequate. If, however, the step size is not small enough, we may require two iterations of the corrector. Then a second application of (3.48) would produce the condition:

$$
|Y_{i+1}^{c,3} - Y_{i+1}^{c,2}| \ll \varepsilon_{\text{loc}} \quad \Rightarrow
$$

$$
\left(\frac{1}{2}hL \right)^2 \cdot 6\varepsilon_{\text{loc}} \ll \varepsilon_{\text{loc}}, \qquad \text{or} \qquad hL \ll \sqrt{\frac{2}{3}} \approx 0.82 ,
\tag{3.50}
$$

which is less restrictive than (3.49).

3.10   Questions for self-assessment

1. Make sure you can reproduce the derivation of Eq. (3.4).

2. What is the idea behind the derivation of Eq. (3.5)?

3. Derive Eqs. (3.9) and (3.10).

4. Derive Eq. (3.11) as indicated in the text.

5. Describe two alternative ways to derive formulae for multistep methods.

6. Verify Eq. (3.19).

7. For a multistep method of order m, what should the order of the starting method be?

8. Convince yourself that method (3.26) is of the form (3.27) and (3.28).

9. What is the origin of the error E_P in Eq. (3.29)?

10. What is the origin of the error E_C in Eq. (3.30)?

11. How should the orders of the predictor and corrector equations be related? Why?

12. Is there a reason to use a predictor as accurate as the corrector?

13. What are the advantages and disadvantages of the P–C methods compared to the RK methods?

14. What is the reason one may want to use an implicit method?

15. What is the significance of Requirements (i) and (ii) found before Eq. (3.47)?

16. Make sure you can explain the derivations of (3.48) and (3.49).