Appendix I

Multistep Methods

In this appendix, we provide a brief introduction to the derivation of linear multistep methods and analyze the conditions on the coefficients that are necessary and sufficient to guarantee convergence of order P.

Among the seven basic examples in Chapter 5, one was a two-step method, the leapfrog method. Multistep methods potentially obtain more accurate approximations from fewer costly evaluations of a vector field if prior evaluations and values of the solution are stored for later use. With each additional stored value comes the need for an additional value for initialization not provided by the analytical problem. Each of these carries an additional degree of potential instability and adds to the cost of changing the step-size. In spite of this, some multistep methods have desirable absolute stability properties. We will also describe some relationships between the accuracy and stability of these methods.

Recall that we are considering methods for approximating solutions of the IVP

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(t_o) = \mathbf{y}_o, \quad t \in [t_o, t_o + T], \quad \mathbf{y} \in \mathbf{R}^D, \tag{I.1}$$

satisfying a Lipschitz condition in some norm on $\mathbf{R}^D$,

$$||\mathbf{f}(t, \mathbf{y}_1) - \mathbf{f}(t, \mathbf{y}_2)|| \leq L||\mathbf{y}_1 - \mathbf{y}_2||. \tag{I.2}$$

For simplicity of exposition, we will henceforth use notation for the scalar case, but unless otherwise noted, generalizations to the case of systems are typically straightforward. In scalar notation, linear

m-step methods take the form

$$y_{n+1} = \sum_{j=0}^{m-1} a_j y_{n-j} + h \sum_{j=-1}^{m-1} b_j y'_{n-j}, \tag{I.3}$$

for all integers n satisfying $0 \le nh \le T$. Here, $t_0 = t_o$, $t_{n+1} = t_n + h$, $y'_j = f(t_j, y_j)$, and $y_0, \ldots, y_{m-1}$ are initial values obtained by companion methods discussed below. When the meaning is unambiguous, we will leave the dependence of y_n on h implicit.

There are several strategies that may be used to obtain families of linear m-step methods with higher-order accuracy. The first such family we will consider is the m-step *backward difference formula* methods, BDFm. These methods are derived by replacing the derivative on the left of $y'(t_{n+1}) = f(t_{n+1}, y(t_{n+1}))$ with the approximation obtained by differentiating the polynomial $p_m(t)$ of degree m that interpolates $y(t)$ at $t_{n+1}, t_n, \ldots, t_{n+1-m}$ and then discretizing. For $m = 1$, since

$$p_1(t) = y(t_{n+1}) + (t - t_{n+1}) \frac{y(t_{n+1}) - y(t_n)}{h},$$

we discretize

$$\frac{y(t_{n+1}) - y(t_n)}{h} = f(t_{n+1}, y(t_{n+1}))$$

and find that BDF1 is the Backward Euler Method,

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

For $m = 2$, iterative interpolation, described in Appendix J, yields

$$p_2(t) = y(t_{n+1}) + (t - t_{n+1})[\frac{y(t_{n+1}) - y(t_n)}{h} + (t - t_n)\frac{y(t_{n+1}) - 2y(t_n) + y(t_{n-1})}{2h^2}]$$

and

$$p'_2(t_{n+1}) = \frac{3y(t_{n+1}) - 4y(t_n) + y(t_{n-1})}{2h}.$$

Discretizing, we find that BDF2 is given by

$$y_{n+1} = \frac{4}{3} y_n - \frac{1}{3} y_{n-1} + \frac{2h}{3} f(t_{n+1}, y_{n+1}).$$

The Adams family of methods arises when we approximate the integral on the right of $y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} y'(s)\, ds$ with

$$\int_{t_n}^{t_{n+1}} P_m^{A\cdot}(s)\, ds.$$

where $P^{A\cdot}$ interpolates $y'(s)$ at a prescribed set of time-steps, and then discretize. For the explicit *Adams-Bashforth* Methods, ABm, $P_m^{AB}(s)$ is the polynomial of degree m that interpolates $y'(s)$ at $t_n, \ldots, t_{n+1-m}$. For the implicit *Adams-Moulton* Methods, AMm, $P_m^{AM}(s)$ is the polynomial of degree $m+1$ that interpolates $y'(s)$ at $t_{n+1}, t_n, \ldots, t_{n+1-m}$.

For $m = 1$, $P_1^{AB}(s) = y'(t_n)$, and we find that AB1 is Euler's Method. Moreover,

$$P_1^{AM}(s) = y'(t_{n+1}) + (s - t_{n+1})\frac{y'(t_{n+1}) - y'(t_n)}{h}$$

and

$$\int_{t_n}^{t_{n+1}} P_1^{AM}(s)\, ds = h\frac{y'(t_{n+1}) + y'(t_n)}{2},$$

so we find that AM1 is the trapezoidal method. For $m = 2$,

$$P_2^{AB}(s) = y'(t_n) + (s - t_n)\frac{y'(t_n) - y'(t_{n-1})}{h}$$

and

$$\int_{t_n}^{t_{n+1}} P_2^{AB}(s)\, ds = h\frac{3y'(t_n) - y'(t_{n-1})}{2},$$

so AB2 is given by

$$y_{n+1} = y_n + h(\frac{3}{2}y'_n - \frac{1}{2}y'_{n-1}).$$

Again iterative interpolation yields

$$P_2^{AM}(s) = y'(t_{n+1}) + (s - t_{n+1})[\frac{y'(t_{n+1}) - y'(t_n)}{h} + (s - t_n)\frac{y'(t_{n+1}) - 2y'(t_n) + y'(t_{n-1})}{2h^2}].$$

Using

$$\int_{t_n}^{t_{n+1}} (s - t_{n+1})(s - t_n)\, ds = \int_0^h (u^2 - uh)\, du = -h^3/6$$

reduces this to

$$\int_{t_n}^{t_{n+1}} P_2^{AM}(s)\, ds = h[\frac{y'(t_{n+1}) + y'(t_n)}{2} - \frac{y'(t_{n+1}) - 2y'(t_n) + y'(t_{n-1})}{12}].$$

Discretizing, we find that AM2 is given by

$$y_{n+1} = y_n + h(\frac{5}{12}y'_p + \frac{8}{12}y'_n - \frac{1}{12}y'_{n-1}).$$

Another strategy for deriving multistep methods obtains the coefficients a_j, b_j as solutions of linear equations that guarantee the method is formally accurate of order P. These conditions are related to the order of accuracy of a convergent method by the *local truncation error* of the method, ϵ_n. This quantity measures by how much a solution of the differential equation fails to satisfy the difference equation, in the sense

$$y(t_{n+1}) = \sum_{j=0}^{m-1} a_j y(t_{n-j}) + h \sum_{j=-1}^{m-1} b_j y'(t_{n-j}) + \epsilon_n, \qquad \text{where } y'_j = f(t_j, y(t_j)). \tag{I.4}$$

In its top row, Table I.1 contains the first few terms of the Taylor expansion of the left-hand side of (I.4), $y(t_{n+1})$, about t_n, in powers of h. Below the line, the rows contain the Taylor expansions of the terms $y(t_n - jh)$ and $y'(t_n - jh)$ on the right of (I.4), where we have placed terms of the same order in the same column and set $q = m = 1$ for compactness of notation.

Algebraic conditions that determine a bound on the order of ϵ_n are obtained by comparing the collective expansions of both sides. The terms in each column are multiples of $h^p y_n^{(p)}$. If we form a common denominator by multiplying the b terms by p/p, the right-hand

sides of order $p \geq 1$ have the form

$$\sum_{j=0}^{m-1} \frac{1}{p!}(-j)^p a_j h^p y_n^{(p)} + \sum_{j=-1}^{m-1} \frac{1}{p!} p(-j)^{p-1} b_j h^p y_n^{(p)},$$

so to make these terms equal, we must have

$$\sum_{j=0}^{m-1} (-j)^p a_j + \sum_{j=-1}^{m-1} p(-j)^{p-1} b_j = 1. \tag{I.5}$$

It can be shown using the linearity of the local truncation error with respect to solutions $y(t)$ and a basis of functions of the form $(t - t_n)^k$ that

$$\epsilon_n \leq C h^{P+1} \tag{I.6}$$

if and only if conditions (I.5) are satisfied for each order h^p, $p = 0, 1, \ldots, P$. The conditions (I.5) are equivalent to requiring that the numerical method is exact on polynomials of degree P, assuming that it is initialized with exact values.

Table I.1: Taylor expansions of $y(t_{n+1})$ and terms in multistep methods

	y_{n+1}	$=$	y_n	$+$	$y'_n h$	$+$	$\frac{1}{2}y''_n h^2$	$+$	$\frac{1}{6}y'''_n h^3 + \cdots$
	$b_{-1}hy'_{n+1}$	$=$			$b_{-1}y'_n h$	$+$	$b_{-1}y''_n h^2$	$+$	$\frac{1}{2}b_{-1}y'''_n h^3 + \cdots$
$+$	$a_0 y_n$	$=$	$a_0 y_n$						
$+$	$b_0 hy'_n$	$=$			$b_0 y'_n h$				
$+$	$a_1 y_{n-1}$	$=$	$a_1 y_n$	$-$	$a_1 y'_n h$	$+$	$\frac{1}{2}a_1 y''_n h^2$	$-$	$\frac{1}{6}a_1 y'''_n h^3 + \cdots$
$+$	$b_1 hy'_{n-1}$	$=$			$b_1 y'_n h$	$-$	$b_1 y''_n h^2$	$+$	$\frac{1}{2}b_1 y'''_n h^3 + \cdots$
$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$	
$+$	$a_q y_{n-q}$	$=$	$a_q y_n$	$-$	$a_q y'_n qh$	$+$	$\frac{1}{2}a_q y''_n q^2 h^2$	$-$	$\frac{1}{6}a_q y'''_{n-q} q^3 h^3 + \cdots$
$+$	$b_q hy'_{n-q}$	$=$		$+$	$b_q y'_n h$	$-$	$b_q y''_n qh^2$	$-$	$\frac{1}{2}b_q y'''_{n-q} q^2 h^3 + \cdots$

We say a method is *consistent* if conditions (I.5) are satisfied for $p = 0$ and $p = 1$, i.e., if

$$\sum_{j=0}^{m-1} a_j = 1 \quad \text{and} \quad \sum_{j=0}^{m-1} -ja_j + \sum_{j=-1}^{m-1} b_j = 1. \tag{I.7}$$

In this case we know that the method formally approximates the differential equation. This guarantees that the approximated equation is the one that we intended. The more subtle issue of convergence of a numerical method involves determining whether solutions of the approximating equation (in this case the multistep method) do indeed approximate solutions of the approximated equation as the discretization parameter tends to zero. The root condition for 0-stability discussed in Chapter 5 together with consistency are necessary and sufficient for a multistep method to be convergent. If, in addition, (I.5) is satisfied for all $p \le P$, then the convergence is with global order of accuracy P.

Four of our working example methods of Chapter 5 and three additional methods discussed above fit into the linear m-step framework with $m \le 2$. Table I.2 summarizes the nonzero coefficients defining these methods and identifies the value of P for which the matching conditions up to order P are satisfied, but not the conditions of order $P + 1$. For reference, the conditions for $p = 2, 3$, and 4 are

$$\sum_{j=0}^{m-1} j^2 a_j - \sum_{j=-1}^{m-1} 2jb_j = 1,$$

$$-\sum_{j=0}^{m-1} j^3 a_j + \sum_{j=-1}^{m-1} 3j^2 b_j = 1,$$

and

$$\sum_{j=0}^{m-1} j^4 a_j - \sum_{j=-1}^{m-1} 4j^3 b_j = 1,$$

respectively.

Table I.2: Coefficients and order of accuracy of example multistep methods

Method (P)	Coefficients
Euler (1)	$a_0 = b_0 = 1$
Backward Euler*(1)	$a_0 = 1,\ b_{-1} = 1$
Trapezoidal*(2)	$a_0 = 1,\ b_{-1} = b_0 = \frac{1}{2}$
Leapfrog (2)	$a_1 = 1,\ b_0 = 2$
BDF2*(2)	$a_0 = \frac{4}{3},\ a_1 = -\frac{1}{3},\ b_{-1} = \frac{2}{3}$
AB2(2)	$a_0 = 1,\ b_0 = \frac{3}{2},\ b_1 = -\frac{1}{2}$
AM2*(3)	$a_0 = 1,\ b_{-1} = \frac{5}{12},\ b_0 = \frac{8}{12},\ b_1 = -\frac{1}{12}$

*Implicit method

Since explicit linear m-step methods are determined by $2m$ coefficients and implicit linear m-step methods are determined by $2m+1$ coefficients, we can obtain multistep methods not belonging to the BDF or Adams families by requiring that a method satisfy as many of the matching conditions of linear equations as possible. The first $2m$ or $2m+1$ conditions, respectively, form nonsingular systems of linear equations in those coefficients whose solution maximizes the order of the local truncation error.

Any linear m-step method with $m = 1$ that satisfies the consistency conditions $a_0 = 1$ and $b_{-1} + b_0 = 1$ is among the family of θ-methods:

$$y_{n+1} = y_n + h((1-\theta)y'_n + \theta y'_{n+1}). \tag{I.8}$$

This family includes one explicit method, Euler's Method, for $\theta = 0$. Second-order accuracy requires $2b_{-1} = 1$, corresponding to the trapezoidal method with $\theta = \frac{1}{2}$. Since the order 3 condition $3b_{-1} = 1$ is not satisfied, the maximal order of an implicit method with $m = 1$ is 2, attained by the trapezoidal method. The θ-method family also includes the Backward Euler Method ($\theta = 1$). The restriction $\theta \in [0, 1]$ is not required for consistency, but since the amplification factor is

$$a(w) = \frac{(1 + (1-\theta)w)}{(1 - \theta w)},$$

it is standard to assure greater stability for λ in the left half-plane,

To obtain an explicit two-step method with local truncation error of order 4 in this way, we look for a method of the form

$$y_{n+1} = a_0 y_n + a_1 y_{n-1} + h(b_0 y'_n + b_1 y'_{n-1})$$

whose coefficients satisfy the four linear conditions $a_0 + a_1 = 1$, $-a_1 + b_0 + b_1 = 1$, $a_1 - 2b_1 = 1$, $-a_1 + 3b_1 = 1$. The method corresponding to the unique solution of this system is

$$y_{n+1} = -4y_n + 5y_{n-1} + h(4y'_n + 2y'_{n-1}). \tag{I.9}$$

We showed in Section 5.3 that (I.9) is not 0-stable. Another method that is consistent, but not 0-stable, is

$$y_{n+1} = 3y_n - 2y_{n-1} - hy'_n. \tag{I.9$'$}$$

We can confirm the instability by considering the roots of its characteristic polynomial for $w = 0$, $p_0(r) = \rho(r) = r^2 - 3r + 2 = (r-1)(r-2)$. Though this method does not satisfy the second-order accuracy conditions, keeping the same $a_0 = 3$, $a_1 = -2$ and modifying the derivative coefficients to $b_0 = \frac{1}{2}$ and $b_1 = -\frac{3}{2}$ yields a method that would be second-order accurate were it not for the same instability.

The connection between the truly unstable behavior of the method (I.10), $y_{n+1} = 3y_n - 2y_{n-1} - hy'_n$, and the roots of its characteristic polynomial for $w = 0$, $p_0(r) = \rho(r) = r^2 - 3r + 2 = (r-1)(r-2)$, is apparent. This also make it clear that we could extend the example to have higher-order truncation error while retaining the same unstable behavior by keeping the same $a_0 = 3$, $a_1 = -2$ but modifying the derivative coefficients to $b_0 = \frac{1}{2}$ and $b_1 = -\frac{3}{2}$.

To obtain an implicit two-step method with local truncation error of order 5 in this way, we look for a method of the form

$$y_{n+1} = a_0 y_n + a_1 y_{n-1} + h(b_{-1} y'_{n+1} + b_0 y'_n + b_1 y'_{n-1})$$

whose coefficients satisfy the five linear conditions $a_0 + a_1 = 1$, $-a_1 + b_{-1} + b_0 + b_1 = 1$, $a_1 + 2b_{-1} - 2b_1 = 1$, $-a_1 + 3b_{-1} + 3b_1 = 1$, $a_1 + 4b_{-1} - 4b_1 = 1$. The method corresponding to the unique solution

of this system is

$$y_{n+1} = y_{n-1} + 2h\left(\frac{1}{6}y'_{n+1} + \frac{4}{6}y'_n + \frac{1}{6}y'_{n-1}\right), \tag{I.10}$$

known as *Milne's corrector*. We can also interpret this as integrating quadratic interpolation of y' at t_{n+1}, t_n, t_{n-1} (the Simpson-parabolic rule) to approximate the integral in

$$y_{n+1} - y_{n-1} = \int_{t_{n-1}}^{t_{n+1}} y'(s)\, ds.$$

Additional families of methods may be obtained using approximations of the integral in

$$y_{n+1} - y_{n-j} = \int_{t_{n-j}}^{t_{n+1}} y'(s)\, ds$$

for larger values of j.

Seeking higher-order accuracy to improve efficiency does not assure convergence. It can actually hinder it by compromising 0-stability. This is the case even for the implicit methods, i.e., they do not always have stability properties that are superior to those of explicit methods. In fact, for $m > 6$, the backward difference methods, BDFm, are implicit methods with arbitrarily high formal accuracy, but they are not even 0-stable.

If a general multistep method is applied to the model problem $y' = \lambda y$ and we set $w = \lambda h$, it takes the form of a homogeneous linear difference equation

$$(1 - b_{-1}w)y_{n+1} = \sum_{j=0}^{m-1}(a_j + b_j w)y_{n-j}. \tag{I.11}$$

We call the polynomial

$$p_w(r) = (1 - b_{-1}w)r^m - \sum_{j=0}^{m-1}(a_j + b_j w)r^{m-(j+1)}$$

the *characteristic polynomial* of the multistep method (I.3). We also define $\rho(r)$ and $\sigma(r)$ by $p_w(r) = \rho(r) + w\sigma(r)$, so in particular,

$$\rho(r) = p_0(r) = r^m - \sum_{j=0}^{m-1} a_j r^{m-(j+1)}.$$

When $p_w(r)$ has distinct roots $r_j(w),\ j = 0, \dots, m-1$, the general solution of (I.11) is a linear combination

$$y_n = \sum_{j=0}^{m-1} c_j r_j^n. \tag{I.12}$$

If $p_w(r)$ has some multiple roots, we can index any set of them consecutively, $r_j(w) = \cdots = r_{j+s}(w)$, in which case we replace the corresponding terms in (I.12) by terms of the form $c_{j+k} n^k r_j^n,\ k = 0, \dots, s$.

As $w \to 0$, the roots of $r_j(w)$ approach corresponding roots of $\rho(r)$. We can use the fact that some root $r(w)$ must approximate $e^w = 1 + 1w$ as $w \to 0$ as another derivation of the consistency conditions (I.7). Since $e^0 = 1$ must be a root of p_0, $p_0(1) = 1 - \sum_j a_j = 0$, which is the zeroth-order consistency condition. Treating $r(w)$ as a curve defined implicitly by the relation $P(r, w) = p_w(r) = 0$ and differentiate implicitly with respect to w at $(r, w) = (1, 0)$, we obtain

$$-\sum_{j=-1}^{m-1} b_j + r'(w) \left(m - \sum_{j=0}^{m-1} a_j (m - (j+1)) \right) = 0.$$

Employing the zeroth-order consistency condition, factoring m from the second term, and setting $r'(w) = 1$ yields the first-order consistency condition of (I.7). This approach can be continued to any order. Alternatively, we may consider to what degree $r = e^w$ is a solution of the characteristic equation $\rho(r) + w\sigma(r) = 0$. The equations (I.5) for $p = 0, \dots, P$ are equivalent to $\rho(e^w) + w\sigma(e^w) = O(w^{P+1})$ as $w \to 0$. If we use $w = \ln(r)$ to write this in the form $\rho(r) + \ln(r)\sigma(r) = 0$, they are also equivalent to

$$\rho(r) + \ln(r)\sigma(r) = C|r-1|^{P+1} + O(|r-1|^{P+2}), \tag{I.13}$$

as $r \to 1$ (so $w \to 0$). It is convenient to expand $\ln(r)$ in powers of $u = r - 1$ near $u = 0$, in which case (I.13) becomes

$$\rho(1+u) + \ln(1+u)\sigma(1+u) = C|u|^{P+1} + O(|u|^{P+2}).$$

In terms of the coefficients a_j and b_j of the numerical method, and using $q = m - 1$ as before, this becomes

$$(1+u)^m - (a_0(1+u)^q + \cdots + a_q) - (u - \frac{u^2}{2} + \frac{u^3}{3} - \cdots) \times (b_{-1}(1+u)^m + b_0(1+u)^q + \cdots + b_q) = C|u|^{P+1} + O(|u|^{P+2}). \tag{I.14}$$

The condition that the coefficient of the u^p term on the left-hand side vanishes is equivalent to the order p matching condition we have given above.

The competition between accuracy and stability is explained in part by two results of Dahlquist that describe barriers to the order of accuracy of multistep methods that satisfy certain stability conditions. The first barrier gives the maximum order of a stable m-step method. Specifying $m - 1$ nonprincipal roots of $\rho(r)$ that satisfy the root condition is equivalent to specifying $m - 1$ real parameters that describe some combination of real roots and complex conjugate pairs. Along with $r_0 = 1$, these determine the real coefficients a_j through $\rho(r) = \Pi(r - r_j)$. Depending on whether the method is explicit or implicit, this leaves m or $m+1$ coefficients b_j with which to satisfy the accuracy conditions of order $p = 1, \ldots, P$. If the method is explicit, one would expect that this is possible through $P = m$, and through $P = m + 1$ if the method is implicit. We know that these are attainable from the examples of AB2 and AM2, stable two-step methods of order 2 and 3, respectively. In the explicit case, this bound turns out to be correct in general, and also in the implicit case if m is odd. However, if m is even, it is possible to satisfy one more additional equation, i.e., there are stable implicit m-step methods with order $m + 2$, but none higher. Milne's corrector satisfies the root condition, so it is 0-stable and convergent. But, for arbitrarily small w the magnitude of the root of $p_w(r)$ that approaches -1 as $w \to 0$ can exceed that of the principal root that approaches $+1$. Because of this, it lacks a desirable property called relative stability, but it is still

0-stable and convergent. It is suggestive that this method contains a form of the Simpson-parabolic integration method, an example of the Newton-Cotes quadrature methods based on an odd number of nodes. Due to symmetry, these quadrature methods attain an additional degree of accuracy over the number of nodes when the number of nodes is odd.

The second barrier refers to methods that are *A-stable*, which means that their region of absolute stability contains the entire left half-plane, i.e., all $w \in \mathbf{C}$ such that $\mathrm{Re}(w) \leq 0$. Dahlquist showed that any A-stable linear multistep method has order of accuracy less than or equal to 2. Because of the usefulness of methods with large regions of absolute stability, considerable effort has gone into finding higher-order A(α)-stable methods whose regions of absolute stability contain large wedges symmetric about the negative real axis in the left half-plane.

The analysis of propagation of errors for linear multistep methods involves issues arising from multiple initial values and modes of amplification that are not present in one-step methods. When we analyzed the error propagation of Euler's Method, we saw that the global error is bounded in terms of a sum of contributions arising from initial error and local truncation error, interacting with the amplification associated with the method. The portion of the bound resulting from the local truncation error has order one less than that of the local truncation error itself, while the portion resulting from the initialization error has the same order of the initialization error. The heuristic explanation is that the number of steps in which truncation errors are introduced grows in inverse proportion to the step size, contributing a factor of h^{-1}. Initialization errors may be amplified by some bounded constant, but they are introduced in a fixed number of steps that are independent of h. So the global order of accuracy of one-step and 0-stable linear multistep methods is at most one less than the order of the local truncation error. But initialization errors are only introduced in a fixed number of steps that is independent of h, so their contribution to the global error has the same magnitude as that of the initialization errors themselves. For the global order to be as small as possible, the initial values must also be one less than the

order of the local truncation error; any more accuracy is wasted. For one step methods, the initial value can be considered exact, since it is given in the IVP, though even this value may include experimental or computational errors. But for m-step methods with $m > 1$, we must use one-step methods to generate one or more additional values. Once we have a second initial value, we could also use a two-step method to generate a third, then a three-step method to generate a fourth, and so on. No matter how we choose to do this, it is just the order of the (local truncation) error of the initial values that limits the global error of the solution. For this reason, it is sufficient to initialize a method whose local truncation error has order $P + 1$ using a method whose local truncation error has order P. For example, the local truncation error of the leapfrog method has order 3. If $y_0 = y_o$, the exact initial value, and we use Euler's Method, whose local truncation error has order 2, to obtain y_1 from y_0, the resulting method has global order of accuracy 2. If we use the midpoint method or Heun's Method, whose local truncation errors both have order 3, the global order of accuracy of the resulting methods is still 2, no more accurate than if we use Euler's Method to initialize. But if we use a lower-order approximation, $y_1 = y_0$, a method whose local truncation error has order 1 and is not even consistent, the savings of one evaluation of f degrades the convergence of all subsequent steps to global order 1. As another example, the two-step implicit Adams-Moulton Method, AM2, has local truncation error of order 4. If we initialize it with the midpoint method or Heun's Method, we achieve the greatest possible global order of accuracy, 3. Initializing with RK4 will not improve this behavior, and initializing with Euler's Method degrades the order to 2. So the reason for including initial errors in the analysis of error propagation for one-step methods is clarified when we consider multistep methods.

When y_{n+1} is only defined implicitly, the ease with which we can determine its value from y_n (and previous values in the case of a multistep method) is significant from both practical and theoretical points of view. In the first place, a solution might not even exist for all values of $h > 0$. For a simple one-step method such as the Backward Euler Method, it can fail to have a solution even for the linear equation

$y' = \lambda y$, $y(0) = y_o$, where it reduces to $y_{n+1}(1 - \lambda h) = y_n$, which clearly has no solution if $\lambda h = 1$ and $y_n \neq 0$.

When $b_{-1} \neq 0$, (I.3) can be considered as a family of fixed-point equations $y_{n+1} = T(y_{n+1}, h)$ depending on the parameter h. If we let

$$y_{n+1}{}^* = \sum_{j=0}^{m-1} a_j y_{n-j}, \tag{I.15}$$

then $y_{n+1}{}^* = T(y_{n+1}{}^*, 0)$. Using the Lipschitz continuity of f with respect to y and the linearity of T with respect to h, we can show that for sufficiently small h, $T(\cdot, h)$ is a contraction that maps an interval I containing $y_{n+1}{}^*$ into itself. By the contraction mapping principle, for any $y_{n+1}{}^{(0)}$ in this interval, the iteration

$$y_{n+1}{}^{(k+1)} = T(y_{n+1}{}^{(k)}, h) \tag{I.16}$$

converges linearly to a unique fixed point y_{n+1}, satisfying $y_{n+1} = T(y_{n+1}, h)$, with rate

$$b_{-1}h|\frac{\partial f}{\partial y}(y_{n+1})| \leq |b_{-1}hL|. \tag{I.17}$$

The situation for implicit Runge-Kutta Methods is more involved, since each step requires the solution of a nonlinear system of equations, but the same principles can be extended to derive existence, smooth dependence, and a convergence rate proportional to h, when h is sufficiently small.

It is a key principle in the design, analysis, and implementation of predictor-corrector methods that the convergence rate of the fixed-point iteration (I.16) is proportional to h. If we perform a single step of an implicit method of order p by iterating (I.16) to convergence, the resulting $y_{n+1}{}^\infty$ has a local truncation error that behaves like $|y(t_{n+1}) - y_{n+1}{}^\infty| \approx Ch^{P+1}$ as $h \to 0$. If we only iterate (I.16) so far that $|y_{n+1}{}^{(k)} - y_{n+1}{}^\infty| \approx C'h^{P+1}$ as $h \to 0$ as well, then using $y_{n+1}{}^{(k)}$ instead of $y_{n+1}{}^\infty$ should still result in a method with the same order.

We may do this in a variety of ways, but it is common to initialize the iteration with an explicit method, called a *predictor*, whose order is the same or one less than that of the implicit method, P.

Each iteration of (I.16) is called a *corrector* step, and if our predictor has global order $P-1$, its local truncation error will behave like $|y(t_{n+1}) - {y_{n+1}}^{(0)}| \approx C_P h^P$. Since this dominates the local truncation error of the corrector, $|{y_{n+1}}^{\infty} - {y_{n+1}}^{(0)}| \approx C_P h^P$, it makes sense to perform one corrector iteration. Due to the h dependence of the rate of convergence, $|{y_{n+1}}^{\infty} - {y_{n+1}}^{(1)}| \approx C_C h^{P+1}$ and therefore $|y(t_{n+1}) - {y_{n+1}}^{(1)}| \approx C'_C h^{P+1}$, and further iterations do not increase the order of the local truncation error. If a predictor is already as accurate as the implicit method, $y(t_{n+1}) - {y_{n+1}}^{(0)} \approx C'_P h^{P+1}$, it would seem pointless to iterate, since one iteration provides no improvement in overall accuracy. At the opposite extreme, we could even initialize with the *constant method*, $y_{n+1} = y_n$, for which the local truncation error $|y(t_{n+1}) - {y_{n+1}}^{(0)}| \approx y'_n h$, and perform P corrector iterations.

We now consider two simple concrete examples. The implicit method we will use in the first example is the Backward Euler Method, and in the second example we will use the trapezoidal method. We will analyze both accuracy and stability for the model problem $y' = \lambda y$ in order to understand why it makes sense to correct to—or even beyond—the maximal achievable accuracy of the method. One reason is improvement in the region of absolute stability. The region of absolute stability of the explicit method corresponding to $y_{n+1} = {y_{n+1}}^{(0)}$ gets deformed step by step into that of the implicit method corresponding to $y_{n+1} = {y_{n+1}}^{\infty}$. A second reason is that the difference between ${y_{n+1}}^{(0)}$ and ${y_{n+1}}^{(1)}$, obtained from a corrector of the same order, can be used to estimate local errors with very little additional computation, and this can be used to adjust the step-size automatically and even change on the fly to an appropriate higher- or lower-order method.

For the purpose of analyzing the local truncation error in both examples, we let $y(t_{n+1})$ be the exact solution passing through (t_n, y_n), evaluated at t_{n+1}, so $y(t_{n+1}) = y_n e^{\lambda h} = \sum_{j=0}^{\infty} \frac{(\lambda h)^j}{j!}$. The iteration (I.16) corresponding to the Backward Euler Method is

$$ {y_{n+1}}^{(k+1)} = y_n + h f(t_{n+1}, {y_{n+1}}^{(k)}). $$

For the model problem $f(t, y) = \lambda y$, if we initialize the iteration with ${y_{n+1}}^{(0)} = y_n$ and perform no iterations, the local truncation

error behaves like $y'(t_n)h = \lambda h y_n$ as $h \to 0$. If we iterate once, ${y_{n+1}}^{(1)} = (1 + \lambda h)y_n$, and the result is no different than if we had applied one step of Euler's Method, which has local truncation error $y(t_{n+1}) - {y_{n+1}}^{(1)} \approx y''(t_n)\frac{h^2}{2} = \frac{(\lambda h)^2}{2}y_n$ as $h \to 0$. Another iteration gives ${y_{n+1}}^{(2)} = (1 + \lambda h + (\lambda h)^2)y_n$, and we may also think of this as using an Euler's Method predictor followed by one Backward Euler corrector step. The local truncation error $y(t_{n+1}) - {y_{n+1}}^{(2)} \approx -y''(t_n)\frac{h^2}{2} = -\frac{(\lambda h)^2}{2}y_n$. This is to be expected since after this iteration, ${y_{n+1}}^{(2)}$ is an $O(h^3)$ approximation of the approximation ${y_{n+1}}^{(\infty)} = (1 - \lambda h)^{-1}y_n = \sum_{j=0}^{\infty}(\lambda h)^j$ whose terms to order h^2 agree with those of ${y_{n+1}}^{(2)}$ above and only agree to order h with $y(t_{n+1})$. Therefore, ${y_{n+1}}^{(2)}$ shares the same error behavior as $y(t_{n+1}) - {y_{n+1}}^{(\infty)} \approx -y''(t_n)\frac{h^2}{2} = -\frac{(\lambda h)^2}{2}y_n$ as $h \to 0$. Better approximations of ${y_{n+1}}^{(\infty)}$ are not better approximations of $y(t_{n+1})$.

The benefits of these iterations are increased stability and error estimation. Euler's Method is never absolutely stable for $w = \lambda h$ on the imaginary axis, since $|1 + w| > 1$ for $w = ai$, $a \neq 0$. The amplification factor corresponding to ${y_{n+1}}^{(2)}$, $a(w) = 1 + w + w^2$, satisfies $|1+w+w^2| \leq 1$ for $w = ai$, $a \in [-1, 1]$. Also, we can subtract $y(t_{n+1}) - {y_{n+1}}^{(2)} \approx -y''(t_n)\frac{h^2}{2}$ from ${y_{n+1}}^{(2)} - {y_{n+1}}^{(1)} \approx y''(t_n)h^2$, to obtain an estimate of the local truncation error in terms of computed quantities, $y(t_{n+1}) - {y_{n+1}}^{(2)} \approx \frac{1}{2}({y_{n+1}}^{(2)} - {y_{n+1}}^{(1)})$ as $h \to 0$. If this error exceeds a certain bound, we may decide to reduce the step size, or we can use these quantities once more to increase the order of our method by canceling the leading terms in their errors (a process known as extrapolation). In this case, we can define a new predictor from their mean, ${y_{n+1}}^{0} = \frac{1}{2}({y_{n+1}}^{(2)} - {y_{n+1}}^{(1)})$, and expect its local truncation error to behave as h^3 as $h \to 0$. This is indeed the case, as this is just Heun's Method. In conjunction with an implicit method with second-order accuracy, we could then continue the process. For this reason, we briefly perform a similar analysis of a trapezoidal corrector.

The iteration (I.16) corresponding to the trapezoidal method is

$$ {y_{n+1}}^{(k+1)} = y_n + \frac{h}{2}(f(t_n, y_n) + f(t_{n+1}, {y_{n+1}}^{(k)})). $$

For the model problem $f(t,y) = \lambda y$, if we initialize the iteration with ${y_{n+1}}^{(0)} = y_n$ and perform no iterations, the local truncation error behaves like $y'(t_n)h = \lambda h y_n$ as $h \to 0$. If we iterate once, ${y_{n+1}}^{(1)} = (1+\lambda h)y_n$ and the result is *still* no different than if we had applied one step of Euler's Method, which has local truncation error $y(t_{n+1}) - {y_{n+1}}^{(1)} \approx y''(t_n)\frac{h^2}{2} = \frac{(\lambda h)^2}{2} y_n$ as $h \to 0$. But this time, another iteration gives ${y_{n+1}}^{(2)} = (1 + \lambda h + \frac{(\lambda h)^2}{2})y_n$, and we may also think of this as using an Euler's Method predictor followed by one trapezoidal corrector step, i.e., Heun's Method. The local truncation error $y(t_{n+1}) - {y_{n+1}}^{(2)} \approx -y''(t_n)\frac{h^3}{6} = \frac{(\lambda h)^3}{6} y_n$. Beyond the first iteration, ${y_{n+1}}^{(k)}$ attains the same accuracy of ${y_{n+1}}^{(\infty)}$; better approximations of ${y_{n+1}}^{(\infty)}$ are not better approximations of $y(t_{n+1})$. The asymptotic form of the third-order local error terms for Heun's Method, a Runge-Kutta Method, depends on the problem to which it is applied. For the model problem, the asymptotic form of the local error is identical to that of the trapezoidal method, so they cannot be used together even in this situation for local error estimation with the Milne device.

For any $m > 0$, the ABm predictor, AM$m-1$ corrector pair has good stability and convergence properties and does satisfy the common order requirements of the Milne device.

Using a predictor with one iteration of a corrector is sometimes denoted PC or PC^1, with n corrector iterations, PC^n, and corrector iterations to convergence, PC^∞. Redefining $\tilde{P} = PC^{n-1}$ to be a new predictor turns PC^n into $\tilde{P}C$. The evaluations of $f(t,y)$ in the definition of the method are sometimes denoted as a distinct step with the letter E, especially when expensive updates can be omitted and iterations can be usefully performed with prior evaluations. This is somewhat analogous to using a single Jacobian for multiple iterations of Newton's Method for the reduction of operations from N^3 to N^2. When all evaluations are performed, the method is denoted $PE(CE)^n$, though there are many variations possible. If absolute stability of the implicit method is essential for the particular problem at hand, e.g., when an A-stable method is required, more efficient approaches to computing y_{n+1} (e.g., Newton or quasi-Newton Methods) should be implemented for quadratic rather than linear convergence.

This highlights a perspective that the predictor-corrector idea is *not* only about finding the solution of the implicit stepping method—that may be found by more efficient means. Rather, it is about designing efficient intermediate explicit methods with enhanced stability and error estimation properties. The use of two methods of common order that share a substantial portion of their computational effort for automatic step-size control is not limited to explicit-implicit pairs. The same idea is used in one-step explicit methods, e.g., the Runge-Kutta-Fehlberg pair.

We conclude with some general remarks on the theoretical and practical effects of stability and instability for multistep methods. The growth of errors in numerical approximations obtained from linear multistep methods is governed by a difference inequality consisting of a homogeneous error amplification term correlated with the stability properties of the method and an inhomogeneous error-forcing term correlated with the local truncation error. The contribution from error amplification is governed by the same recurrence that determines the stability of the method and in particular by the behavior of the method on the model problem $y' = \lambda y$ where $\lambda = L$, the Lipschitz constant for f.

The most fundamental result in the theory of linear multistep methods is known as *Dahlquist's Equivalence Theorem.* This theorem relates the root condition and convergence as follows. Let (I.3) be a consistent m-step method applied to the well-posed IVP (I.1), with initial values $y_0, \ldots, y_{m-1}$ approaching y_o as $h \to 0$. The method is convergent if and only if $\rho(r)$ satisfies the root condition. Furthermore, if the local truncation error defined in (I.4) satisfies $|\epsilon_n| \leq C_T h^{P+1}$ and the initial errors $e_{j,h} = y_{j,h} - y(t_j),\ j = 0, \ldots, m-1$, satisfy

$$\max_{j=0,\ldots,m-1} |e_{j,h}| \leq C_I h^P,$$

then

$$\max_{0 \leq nh \leq T} |y_{n,h} - y(t_n)| \leq C_G h^P$$

as $h \to 0$. In other words, if the local truncation error has order $P+1$, the initial error only needs to have order P for the global convergence

to have order P. A proof that applies to an even more general class of methods may be found in [IK]. In the much greater generality of linear finite difference methods for partial differential equations, the fact that stability and consistency together are both necessary and sufficient for convergence is the content of the important Lax-Richtmyer Equivalence Theorem [LR].

There are several other more stringent conditions that have been developed to distinguish the behavior observed in convergent linear multistep methods. The *strong root condition* says that except for r_0, the roots of $\rho(r)$ are all inside the open unit disc, a condition that, as the name suggests, clearly implies the root condition. By continuity, for sufficiently small h, the nonprincipal roots $r_j(w)$ of $\rho(r) + w\sigma(r)$ will also have magnitude less than 1. If the coefficients of an m-step method with $m > 1$ satisfy $a_j = 0$ for $j < m - 1$, so $a_{m-1} = 1$ for consistency, it cannot satisfy the strong root condition, since the roots of $\rho(r) = r^m - 1$ are all of the form $r_j = e^{2\pi j/m}$, $j = 0, \ldots, m - 1$. This class includes the leapfrog method and Milne's corrector. Even if the strong root condition is not satisfied, we can require that for sufficiently small h, the parasitic roots of $p_w(r)$ have magnitudes less than or equal to the magnitude of the principal root, a condition called *relative stability*. In this case, parasitic roots can only grow exponentially when the principal root is growing faster exponentially, making them less of a concern. The term *weak stability* is used to describe a method that is stable but not relatively stable. Since the leapfrog method satisfies the root condition and we have shown that as $h \to 0$, the parasitic root of the leapfrog method has magnitude greater than its principal root, the leapfrog method is weakly stable and demonstrates that the root condition cannot imply relative stability. However, the continuity argument that shows that the strong root condition implies the root condition can be used just as easily to show that the strong root condition implies relative stability. By the observation above, the consistent two-step method $y_{n+1} = y_{n-1} + 2hy'_{n-1}$ cannot satisfy the strong root condition. But since its characteristic polynomial is

$$r^2 - (1 + 2w) = (r - \sqrt{1 + 2w})(r + \sqrt{1 + 2w}),$$

its principal and parasitic roots have the same magnitude and it is relatively stable. This shows that relative stability is strictly weaker than the strong root condition.

For Euler's Method, the Backward Euler Method, and the trapezoidal method, $\rho(r) = r - 1$. Since there are no nonprincipal roots, they satisfy the strong root condition, the root condition, and the relative stability condition by default. Both the explicit and the implicit m-step Adams Methods, ABm and AMm, are designed to have $\rho(r) = r^m - r^{m-1} = (r-1)r^{m-1}$, so that all parasitic roots are zero! These methods satisfy the strong root condition, as nicely as possible. For BDF2, $\rho(r) = r^2 - \frac{4}{3}r + \frac{1}{3} = (r-1)(r-\frac{1}{3})$ satisfies the strong root condition. For higher m, BDFm is designed to have order of accuracy m if the method is convergent. However, these methods are only 0-stable for $m \le 6$, so BDFm is not convergent for $m \ge 7$.

If we apply Milne's corrector, the implicit two-step method having maximal local truncation error, to the model problem $y' = \lambda y$, it takes the form

$$y_{n+1} = y_{n-1} + (\frac{w}{3}y_{n+1} + \frac{4w}{3}y_n + \frac{w}{3}y_{n-1}).$$

Solutions are linear combinations $y_n = c_+ r_+ + c_- r_-$ where $r_\pm$ are the roots of

$$p_w(r) = (1 - w/3)r^2 - (4w/3)r - (1 + w/3).$$

By setting $u = w/3$ and multiplying by $1/(1-u) = 1 + u + \cdots$, to first order in u, these roots satisfy

$$r^2 - 4u(1 + \cdots)r - (1 + 2u + \cdots) = 0$$

or

$$r_\pm = 2u \pm \sqrt{4u^2 + 1 + 2u}.$$

Using the binomial expansion $(1+2u)^{1/2} \approx 1 + u + \cdots$, to first order in u, $r_+ \approx 1 + 3u$ and $r_- \approx -1 + u$. The root $\approx 1 + 3u = 1 + \lambda h$ approximates the solution of the model problem $y' = \lambda y$. As $\lambda h \to 0$ in a way that u in a neighborhood of the negative real axis near the origin, the other root $\approx -1 + u$ has magnitude greater than 1, showing that Milne's Method corrector is not relatively stable. Like the leapfrog method, it satisfies the root condition, so it is stable, but

only weakly stable. So even a convergent implicit method can have worse stability properties than a convergent explicit method.

We can summarize the relationship among various types of stability for multistep methods and their consequences as follows:

$$\text{strong root condition} \Rightarrow \text{relative stability} \Rightarrow \text{root condition},$$

$$\text{absolute stability for } w = 0 \Rightarrow \text{root condition} \Leftrightarrow 0\text{-stability},$$

$$\text{consistency} + \text{root condition} \Leftrightarrow \text{convergence},$$

$$P\text{th-order formal accuracy} + \text{root condition} \Leftrightarrow P\text{th-order convergence}.$$

We conclude by cautioning the reader that many examples in Chapter 5 show that the behavior of a method is not determined by either the order of its local truncation error or by the rather loose bounds provided by 0-stability. 0-stability is certainly important, helping us to avoid nonconvergent methods, a caveat that cannot be overemphasized. Convergence and stability are a minimal but important requirement for a useful method. Along with the order of accuracy, the actual performance of a method is more closely correlated with its absolute stability with respect to modes present in the ODE.

▷ **Exercise I–1.** Determine the regions of absolute stability for BDF2, AM2, and AM2. For implicit methods, it may be advantageous to consider the fact that if $r \neq 0$ is a root of $ar^2 + br + c$, then $s = \frac{1}{r}$ is a root of $cs^2 + bs + a$.

Appendix J

Iterative Interpolation and Its Error

In this appendix we give a brief review of iterative polynomial interpolation and corresponding error estimates used in the development and analysis of numerical methods for differential equations.

The unique polynomial of degree n,

$$p_{x_0,\dots,x_n}(x) = \sum_{j=0}^{n} a_j x^j, \tag{J.1}$$

that interpolates a function $f(x)$ at $n+1$ points,

$$p_{x_0,\dots,x_n}(x_i) = y_i = f(x_i), \quad 0 \le i \le n, \tag{J.2}$$

can be found by solving simultaneously the $(n+1) \times (n+1)$ linear system of equations for the $n+1$ unknown coefficients a_j given by (J.2). It can also be found using Lagrange polynomials

$$p_{x_0,\dots,x_n}(x) = \sum_{i=0}^{n} y_i L_{i,x_0,\dots,x_n}(x) \tag{J.3}$$

where

$$L_{i,x_0,\dots,x_n}(x) = \prod_{0\le j\le n, j\ne i} \frac{(x-x_j)}{(x_i-x_j)}. \tag{J.4}$$

Here, we develop $p_{x_0,\dots,x_n}(x)$ inductively, starting from $p_{x_0}(x) = y_0$ and letting

$$\begin{aligned} &p_{x_0,\dots,x_{j+1}}(x) \\ &\quad = p_{x_0,\dots,x_j}(x) + c_{j+1}(x-x_0)\cdots(x-x_j), \quad j = 0,\dots,n-1 \end{aligned} \tag{J.5}$$

(so that each successive term does not disturb the correctness of the prior interpolation) and defining c_{j+1} so that $p_{x_0,\dots,x_{j+1}}(x_{j+1}) = y_{j+1}$, i.e.,

$$c_{j+1} = \frac{y_{j+1} - p_{x_0,\dots,x_j}(x_{j+1})}{(x_{j+1} - x_0)} = f[x_0, \dots, x_{j+1}]. \tag{J.6}$$

Comparing (J.6) with (J.3), (J.4) gives an alternate explicit expression for $f[x_0, \dots, x_n]$, the leading coefficient of the polynomial of degree n that interpolates f at $x_0, \dots, x_n$:

$$f[x_0, \dots, x_n] = \sum_{i=0}^{n} \frac{f(x_i)}{\prod_{j \neq i}(x_i - x_j)} \tag{J.7}$$

from which follows the divided difference relation

$$f[x_0, \dots, x_n] = \frac{f[x_0, \dots, \hat{x_j}, \dots x_n] - f[x_0, \dots, \hat{x_i} \dots, x_n]}{x_i - x_j} \tag{J.8}$$

(where $\hat{}$ indicates omission).

For our purposes, we want to estimate $p_{x_0,\dots,x_n}(t) - f(t)$, and to do so, we simply treat t as the next point at which we wish to interpolate f in (J.5):

$$p_{x_0,\dots,x_n}(t) + f[x_0, \dots, x_n, t](t - x_0) \cdots (t - x_n) = f(t)$$

or

$$p_{x_0,\dots,x_n}(t) - f(t) = f[x_0, \dots, x_n, t](t - x_0) \cdots (t - x_n). \tag{J.9}$$

Finally, we estimate the coefficient $f[x_0, \dots, x_n, t]$ using several applications of Rolle's Theorem. Since $p_{x_0,\dots,x_n,t}(x) = f(x)$ or $p_{x_0,\dots,x_n,t}(x) - f(x) = 0$ at $n+2$ points $x_0, \dots, x_n, t$, Rolle's Theorem says that $p'_{x_0,\dots,x_n,t}(x) - f'(x) = 0$ at $n+1$ points, one in each open interval between consecutive distinct points of $x_0, \dots, x_n, t$. Repeating this argument, $p''_{x_0,\dots,x_n,t}(x) - f''(x) = 0$ at n points on the intervals between the points described in the previous stage, and repeating this $n-1$ more times, there is one point ξ in the interior of the minimal closed interval containing all of the original points $x_0, \dots, x_n, t$ at which

$$p^{(n+1)}_{x_0,\dots,x_n,t}(\xi) - f^{(n+1)}(\xi) = 0. \tag{J.10}$$

But because $f[x_0, \dots, x_n, t]$ is the leading coefficient of the polynomial $p_{x_0,\dots,x_n,t}(x)$ of degree $n+1$ that interpolates f at the $n+2$ points $x_0, \dots, x_n, t$, if we take $n+1$ derivatives, we are left with a constant, that leading coefficient times $(n+1)!$:

$$p^{(n+1)}_{x_0,\dots,x_n,t}(x) = (n+1)! f[x_0, \dots, x_n, t]. \tag{J.11}$$

Combining this with (J.10) gives

$$f[x_0, \dots, x_n, t] = \frac{f^{(n+1)}(\xi)}{(n+1)!} \tag{J.12}$$

where ξ in the interior of the minimal closed interval containing all of the original points $x_0, \dots, x_n, t$, and substituting into (J.9) yields the basic interpolation error estimate:

$$p_{x_0,\dots,x_n}(t) - f(t) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(t - x_0) \cdots (t - x_n). \tag{J.13}$$

For $n = 0$ this recovers the mean value theorem

$$\frac{f(t) - f(x_0)}{t - x_0} = f'(\xi) \tag{J.14}$$

for some $\xi \in (x_0, t)$.

Since many multistep methods involve simultaneous interpolation of y and y' at $t_n, \dots, t_{n-m+1}$, to treat these, we would want to have the corresponding estimates for osculatory interpolation that can be obtained by letting pairs of interpolation points coalesce. In the simplest cases, for two points, this process recovers the tangent line approximation and estimate

$$f[x_0] + f[x_0, x_0](x - x_0) = f(x_0) + f'(x_0)(x - x_0).$$

For four points, it recovers the cubic spline interpolation approximating a function and its derivative at two points.