Supplemental notes for Chapter 5

Backward differences

Suppose we want to interpolate values $f(t_{n-1})$, ... $f(t_{n-k})$ for some k.

Let us use the Newton interpolation polynomial; use notation $f_{n-j} = f(t_{n-j})$, $j = 1, ..k$:

$$P(t) \equiv f_{n-1} + [f_{n-1}, f_{n-2}](t - t_{n-1}) + ... + [f_{n-1}, ..., f_{n-k}](t - t_{n-1})...(t - t_{n-k+1}).$$

If the points are equispaced (i.e., $t_{n-j} = t_n - hj$, $j = 1, .., k$) , this can be simplified by introducing backward differences $\nabla^{(i)} f_j$:

$$\begin{aligned}
\nabla^0 f_j &\equiv f_j, \\
\nabla^1 f_j &\equiv f_j - f_{j-1}, \\
\nabla^2 f_j &\equiv \nabla^1 f_j - \nabla^1 f_{j-1}, \\
&... \\
\nabla^i f_j &\equiv \nabla^i f_j - \nabla^i f_{j-1}, \qquad \text{and so on.}
\end{aligned}$$

Then

$$P(t) \equiv f_{n-1} + \frac{\nabla^1 f_{n-1}}{1!h}(t - t_{n-1}) + \frac{\nabla^2 f_{n-1}}{2!h^2}(t - t_{n-1})(t - t_{n-2}) ... + \frac{\nabla^{k-1} f_{n-1}}{(k-1)!h^{k-1}}(t - t_{n-1})...(t - t_{n-k+1}).$$

Note the formula

$$\nabla^l f_j = \sum_{m=0}^{l} (-1)^m \begin{pmatrix} l \\ m \end{pmatrix} f_{j-m}, \tag{1}$$

where $\begin{pmatrix} l \\ m \end{pmatrix}$ is the binomial coefficients:

$$\begin{pmatrix} l \\ m \end{pmatrix} = \frac{l(l-1)(l-2)...(l-m+1)}{m!}, \qquad l \geq m; \qquad \begin{pmatrix} l \\ 0 \end{pmatrix} = \begin{pmatrix} l \\ l \end{pmatrix} = 1.$$

This formula can also be extended to non-integer numbers by replacing l with a real number s:

$$\begin{pmatrix} s \\ m \end{pmatrix} = \frac{s(s-1)(s-2)...(s-m+1)}{m!}.$$

Notice that for equispaced points the formula for $P(t)$ can be further simplified by a substitution $s = (t - t_{n-1})/h$. Then

$$\begin{aligned}
P(sh) &= f_{n-1} + \nabla^1 f_{n-1} s + \nabla^2 f_{n-1} \frac{s(s+1)}{2!} ... + \nabla^{k-1} f_{n-1} \frac{s(s+1)...(s+k-2)}{(k-1)!} && (2) \\
&= f_{n-1} + (-1)^1 \nabla^1 f_{n-1} \begin{pmatrix} -s \\ 1 \end{pmatrix} + (-1)^2 \nabla^2 f_{n-1} \begin{pmatrix} -s \\ 2 \end{pmatrix} ... + (-1)^{(k-1)} \nabla^{k-1} f_{n-1} \begin{pmatrix} -s \\ k-1 \end{pmatrix} \\
&= \sum_{i=0}^{k-1} (-1)^i \nabla^i f_{n-1} \begin{pmatrix} -s \\ i \end{pmatrix}. && (3)
\end{aligned}$$

Formula (2) is convenient for computations by hand. For example, for $k = 3$ we have

$$P(sh) = f_{n-1} + (f_{n-1} - f_{n-2})s + (f_{n-1} - 2f_{n-2} + f_{n-3})\frac{s(s+1)}{2!} + (f_{n-1} - 3f_{n-2} + 3f_{n-3} - f_{n-4})\frac{s(s+1)(s+2)}{3!}.$$

Let us combine formula (2) with (1) (replace l by i and j by $n-1$):

$$\begin{aligned} P(sh) &= \sum_{i=0}^{k-1}(-1)^i \left[\sum_{m=0}^{i}(-1)^m \binom{i}{m} f_{n-1-m}\right]\binom{-s}{i} \\ &= \sum_{m=0}^{k-1} f_{n-(m+1)}(-1)^m \sum_{i=m}^{k-1}(-1)^i \binom{i}{m}\binom{-s}{i}. \end{aligned}$$

In the above equation, please note the change in the summation indices when the summation order is interchanged. Let's replace $m+1$ by j and, correspondingly, m by $j-1$:

$$P(sh) = \sum_{j=1}^{k} f_{n-j}(-1)^{j-1} \sum_{i=j-1}^{k-1}(-1)^i \binom{i}{j-1}\binom{-s}{i}.$$

Explicit Adams-Bashforth methods

These methods are obtained by approximating the identity:

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

Now, define $f_{n-j} = f(t_{n-j}, y_{n-j})$, and instead of $f(t, y(t))$ integrate the interpolating polynomial $P(t)$:

$$\begin{aligned} \int_{t_{n-1}}^{t_n} f(t, y(t))dt \approx \int_{t_{n-1}}^{t_n} P(t)dt &= h\int_0^1 P(sh)ds \\ &= \sum_{j=1}^{k} f_{n-j}\left\{(-1)^{j-1}\sum_{i=j-1}^{k-1}\binom{i}{j-1}\left[(-1)^i\int_0^1\binom{-s}{i}ds\right]\right\}. \end{aligned}$$

Let us denote the expression in the brackets by γ_i and the expression in the curly braces by β_j:

$$\gamma_i = (-1)^i\int_0^1\binom{-s}{i}ds,$$

$$\beta_j = (-1)^{j-1}\sum_{i=j-1}^{k-1}\binom{i}{j-1}\gamma_i.$$

Then the explicit Adams-Bashforth methods are defined by the formula

$$y_n = y_{n-1} + \sum_{j=1}^{k} \beta_j f_{n-j},$$

with β_j defined above. This recovers formulas given in section 5.1.1, page 129.

Implicit backward differentiation methods (BDF)

These methods are obtained by interpolating approximate values of $y(t)$ (i.e. $y_{n-j} \approx y(t_{n-j}),\ \ j = 0, ..., k$) and by differentiating the interpolating polynomial at $t = t_n$. In other words if $Q(t)$ interpolates values y_{n-j} then $Q(t) \approx y(t)$ and

$$Q'(t_n) \approx y'(t_n) = f(t_n, y_n)$$

represents a non-linear equation for finding y_n. The derivative needs to be expressed explicitly through the values of y_{n-j}, $j = 0, ..., k$.

Let us use backward differences to write

$$Q(t) \equiv y_n + \frac{\nabla^1 y_n}{1!h}(t - t_n) + \frac{\nabla^2 y_n}{2!h^2}(t - t_n)(t - t_{n-1})... + \frac{\nabla^k y_n}{k!h^k}(t - t_n)...(t - t_{n-k+1}).$$

Differentiate in t, evaluate at t_n, notice that

$$\left\{\frac{d}{dt}[(t - t_n)(t - t_{n-1})...(t - t_{n-m})]\right\}\Big|_{t=t_n}$$
$$= \left\{(t - t_n)\frac{d}{dt}[(t - t_{n-1})...(t - t_{n-m})]\right\}\Big|_{t=t_n} + \{(t - t_{n-1})...(t - t_{n-m})]\}|_{t=t_n}$$
$$= (t_n - t_{n-1})...(t_n - t_{n-m}).$$

If the points are equispaced

$$(t_n - t_{n-1})...(t_n - t_{n-m}) = m!h^m.$$

Then we obtain

$$Q'(t_n) = \frac{\nabla^1 y_n}{1!h} + \frac{\nabla^2 y_n}{2!h^2}h... + \frac{\nabla^k y_n}{k!h^k}(k-1)!h^{k-1} = \frac{1}{h}\left(\nabla^1 y_n + \frac{\nabla^2 y_n}{2}... + \frac{\nabla^k y_n}{k}\right).$$

Now, we want to equate

$$Q'(t_n) = \frac{1}{h}\left(\nabla^1 y_n + \frac{\nabla^2 y_n}{2}... + \frac{\nabla^k y_n}{k}\right) = f(t_n, y_n)$$

thus obtaining

$$\sum_{i=1}^{k} \frac{1}{i}\nabla^i y_n = hf(t_n, y_n). \tag{4}$$

This defines an implicit k-step BDF method. To obtain explicit expression for the coefficients, use the formula

$$\nabla^i y_n = \sum_{m=0}^{i} (-1)^m \left(\begin{array}{c} i \\ m \end{array} \right) y_{n-m}. \tag{5}$$

If one substitutes (5) into (4) and interchanges the order of summations, then (4) can be re-written in the form

$$\sum_{m=0}^{k} \alpha_m y_{n-m} = h\beta_0 f(t_n, y_n),$$

where coeffficient β_0 is chosen in such a way that $\alpha_0 = 1$.