The Butcher Factor

Robert M. Corless

April 2025

Western University, Canada

A version of this talk was given at woRK 2024 in Auckland,
celebrating the 91st birthday of John C. Butcher

Maple Transactions

Maple Transactions, now in its fifth year

a "Diamond" class open access journal with no page charges

Now listed by DBLP

mapletransactions.org

Interpolational polynomials

"...a theorem of great antiquity...the simple theorems of polynomial interpolation upon which much practical numerical analysis rests."
—Philip J. Davis, Interpolation and Approximation
quoted page 290 in Hairer & Wanner II

- Although the idea is very old, new things come up from time to time (see e.g. Nick Trefethen's lovely book *Approximation Theory and Approximation Practice*)
- A useful technique was introduced in a paper of John Butcher's published in 1967, namely "A Multistep Generalization of Runge-Kutta Methods With Four or Five Stages." I call this technique *the Butcher factor*.
- We (JCB, RMC, LGV, and Azar Shakoori) used the Butcher factor in our 2010 paper "Polynomial Algebra for Birkhoff Interpolants," but I think it deserves wider attention.

The same princess can choose many different sets of clothes

The most basic kind of interpolation: given discrete data as a finite number of (x,y) pairs, say $[(-1,-1),(-1/2,1),(1/2,-1),(1,1)]$ find a polynomial of minimal degree, expressed in the monomial basis, i.e. of the form $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$, which *interpolates* the data (i.e. fits the data exactly). For this data such a polynomial is $y = 4x^3 - 3x$, which you can find by solving the Vandermonde system

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & -\frac{1}{2} & \frac{1}{4} & -\frac{1}{8} \\ 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} . \tag{1}$$

We don't have to use the monomial basis

Alternative expressions for that particular polynomial include $y = T_3(x)$ where $T_n(x) = \cos n\theta$ with $x = \cos\theta$, the Chebyshev polynomial of degree n. The set $\{1, x, x^2, \ldots, x^n\}$ is a basis for polynomials of grade* n, but so is the set $\{T_0(x), T_1(x), \ldots, T_n(x)\}$.

We may, of course, express each element of one polynomial basis in terms of the other, via the change-of-basis matrix.

$$\begin{bmatrix} T_0(x) \\ T_1(x) \\ T_2(x) \\ T_3(x) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 2 & 0 \\ 0 & -3 & 0 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} \tag{2}$$

The different polynomial bases are therefore mathematically equivalent. However, they do not necessarily have the same numerical conditioning.

* the word "grade" means "degree at most." This is a convention widely used in papers on polynomial eigenvalue problems.

The Lagrange basis

Another well-known basis is the *Lagrange basis* $\{\ell_k(x)\}_{k=0}^n$ (which depends on the data). Here we have

$$\begin{aligned}
\ell_0(x) &= \beta_0(x+1/2)(x-1/2)(x-1) \\
\ell_1(x) &= \beta_1(x+1)(x-1/2)(x-1) \\
\ell_2(x) &= \beta_2(x+1)(x+1/2)(x-1) \\
\ell_3(x) &= \beta_3(x+1)(x+1/2)(x-1/2)\,,
\end{aligned} \tag{3}$$

where the *barycentric weights* β_k are chosen so that $\ell_k(x_j) = [j = k]$ (this is the "Iverson notation," which means 1 if the entry in brackets is true, and zero otherwise). Then the example data is fit by

$$y = -1 \cdot \ell_0(x) + 1 \cdot \ell_1(x) - 1 \cdot \ell_2(x) + 1 \cdot \ell_3(x)\,. \tag{4}$$

Expanding this out will change this to the Chebyshev or monomial basis and we get $T_3(x) = 4x^3 - 3x$ as before.

Vandermonde matrices

The change-of-basis matrix from the Lagrange basis is a (transposed) Vandermonde matrix:

$$\begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ (-1) & (-1/2) & (1/2) & (1) \\ (-1)^2 & (-1/2)^2 & (1/2)^2 & (1)^2 \\ (-1)^3 & (-1/2)^3 & (1/2)^3 & (1)^3 \end{bmatrix} \begin{bmatrix} \ell_0(x) \\ \ell_1(x) \\ \ell_2(x) \\ \ell_3(x) \end{bmatrix} \tag{5}$$

This is because expressing x^j in the Lagrange basis just gives $x_0^j\ell_0(x) + x_1^j\ell_1(x) + \cdots + x_n^j\ell_n(x)$: the coefficients are the values of the polynomial x^j at each of the nodes. As is well-known, the determinant of this matrix is the product of all the differences between the nodes, and hence is nonzero if the nodes are all distinct.

The matrix is usually very ill-conditioned if the dimension is large, however.

Bernstein bases

Put $B_k^n(x) = \binom{n}{k}(x+1)^k(1-x)^{n-k}/2^n$, for $0 \le k \le n$. This is the Bernstein basis on $[-1,1]$. [One usually sees the Bernstein basis on $[0,1]$, where it is $\binom{n}{k}x^k(1-x)^{n-k}$.] Each basis element is nonnegative on the interval. This is important for numerical conditioning.

Our example polynomial can be written as
$p = -1 \cdot B_0^3(x) + 5 \cdot B_1^3(x) - 5 \cdot B_2^3(x) + 1 \cdot B_3^3(x)$.

It's the same princess, just wearing different clothes.

Whole books have been written about interpolation

I want to tell you *all about it*. Just to make sure "everyone is on the same page." But that would take too long. But there are two facts that everyone should know about interpolation:

- Interpolation is most useful when the data is *very accurate*
- The interpolation process frequently will *not converge* as the number of nodes goes to infinity.

Nevertheless it's very useful, both in theory and in practice.

Forward error

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\theta) w(x) \tag{6}$$

where the node polynomial is $w(x) = \prod_{k=0}^{n}(x - x_k)$ for Lagrange interpolation (no matter what clothes the princess is wearing). If complex values are needed,

$$f(z) - p(z) = \frac{1}{2\pi i} \oint_C \frac{w(z) f(\zeta)}{w(\zeta)(\zeta - z)}\, dz\,. \tag{7}$$

Sadly, this is not the end of the "error" story: numerical stability and conditioning are both important (and here is where which clothes the princess wears begin to matter).

Polynomial bases and numerical condition

Avoid changing the polynomial basis you use, because the condition number of the change-of-basis matrix is usually exponential in the degree.

The Lagrange basis is frequently the best-conditioned*.

The barycentric forms of the Lagrange interpolational polynomial are efficient and perfectly (componentwise!) numerically stable.

Hermite interpolational bases, which use derivative data as well, are not as good, but they are not bad, when the confluencies are not large.

* Corless & Watt, 2004, "Bernstein bases are optimal, but, sometimes, Lagrange bases are better." NB Condition number for polynomial evaluation, not "matrix condition number." See also Carnicer, Khiar, & Peña, 2019

Example

Suppose the values of a polynomial p are known at the nodes $\tau = [-1, -1/2, 1/2, 1]$. Say they are $[\rho_0, \rho_1, \rho_2, \rho_3]$. Then, without converting to a monomial basis, the polynomial can be written as

$$p(z) = w(z)\left(-\frac{2\rho_0/3}{z+1} + \frac{4\rho_1/3}{z+1/2} - \frac{4\rho_2/3}{z-1/2} + \frac{2\rho_3/3}{z-1}\right) \tag{8}$$

where the *node polynomial* is $w(z) = (z+1)(z+1/2)(z-1/2)(z-1)$.

This is the Lagrange basis in another form: $\ell_j(z) = w(z)\beta_j/(z-x_j)$, and here the barycentric weights are $\pm 2/3$ or $\pm 4/3$.

Second barycentric form

Equivalently but sometimes better, the polynomial may be written as

$$p(z) = \frac{-\frac{2\rho_0/3}{z+1} + \frac{4\rho_1/3}{z+1/2} - \frac{4\rho_2/3}{z-1/2} + \frac{2\rho_3/3}{z-1}}{-\frac{2/3}{z+1} + \frac{4/3}{z+1/2} - \frac{4/3}{z-1/2} + \frac{2/3}{z-1}} \tag{9}$$

and a further improvement can be made by cancelling common factors in the barycentric weights in the numerator and denominator (this helps to avoid overflow and underflow for larger examples).

These look ridiculous, but I tell you they're beautiful. Berrut & Trefethen 2004 and N.J. Higham 2004 proved them to be componentwise numerically stable: evaluation is fast, and robust.

Really??

Mathematically,

$$\begin{aligned}\lim_{z\to -1} p(z) &= \lim_{z\to -1} \frac{-\frac{2\rho_0/3}{z+1} + \frac{4\rho_1/3}{z+1/2} - \frac{4\rho_2/3}{z-1/2} + \frac{2\rho_3/3}{z-1}}{-\frac{2/3}{z+1} + \frac{4/3}{z+1/2} - \frac{4/3}{z-1/2} + \frac{2/3}{z-1}} \\ &= \lim_{z\to -1} \frac{-\frac{2\rho_0/3}{z+1} + \cdots}{-\frac{2/3}{z+1} + \cdots} \\ &= \rho_0 \end{aligned} \tag{10}$$

but it's a very interesting numerical miracle that this works out in floating-point arithmetic as well: you can take z to be *one unit in the last place* different to -1 and the *rounding errors cancel out*. You get the exact answer for a minutely perturbed polynomial.

Hermite Interpolation and Birkhoff Interpolation

If we are given data that includes consecutive values of the derivatives as well then the problem is called *Hermite Interpolation*. For example, if you are given (x_0,y_0), (x_0, y_0'), and (x_1,y_1) with $x_1 \neq x_0$, then you can fit a grade 2 polynomial to this data. The three given pieces of information determine the three coefficients of the interpolational polynomial.

Birkhoff interpolation happens when some of the data is missing. For example, if you are given (x_0,y_0), (x_0, y_0'') (but not y_0', which is missing), and (x_1,y_1), then you might be able to fit a unique grade 2 polynomial to this data, *or you might not*. Some Birkhoff problems can be solved uniquely, but others cannot. We say that a Birkhoff problem is "poised" if it can be solved uniquely. The example problem just given is poised, which you can check. If, instead of y_0'', the third derivative y_0''' was given, it would *not* be poised.

Partial Fractions and the Cauchy Integral Formula

Suppose the following function of the variable z (with parameter θ) is written in terms of partial fractions:

$$\begin{aligned}\frac{1}{(z-\theta)z^2(z-1)^2} =& \frac{1/\theta^2(\theta-1)^2}{z-\theta} \\ & - \frac{1/\theta}{z^2} - \frac{(1+2\theta)/\theta^2}{z} \\ & + \frac{(2\theta-3)/(\theta-1)^2}{z-1} + \frac{1/(1-\theta)}{(z-1)^2} \end{aligned} \tag{11}$$

continued

Then we can solve the interpolation problem by using a contour that encloses all poles, as follows. The following equation is valid for polynomials $p(z)$ of grade* 3:

$$0 = \frac{1}{2\pi i} \oint_C \frac{p(z)}{(z-\theta)z^2(z-1)^2}\, dz \tag{12}$$

using that partial fraction expansion together with the Cauchy Integral Formula

$$\frac{f^{(j)}(a)}{j!} = \frac{1}{2\pi i} \oint_C \frac{f(z)}{(z-a)^{j+1}}\, dz \tag{13}$$

as follows.

* reminder: the word "grade" means "degree at most."

continued

$$
\begin{aligned}
0 = & \frac{1}{\theta^2(\theta-1)^2} p(\theta) \\
& - \frac{1}{\theta} p'(0) - \frac{(1+2\theta)}{\theta^2} p(0) \\
& + \frac{(2\theta-3)}{(\theta-1)^2} p(1) + \frac{1}{(1-\theta)} p'(1) \, .
\end{aligned} \tag{14}
$$

Isolating the term containing $p(\theta)$ and multiplying by $\theta^2(\theta-1)^2$ gives the unique grade 3 Hermite interpolational polynomial with given values and derivatives at 0 and 1.

Explicitly

$$\begin{aligned} p(\theta) = {} & (2\theta + 1)\,(\theta - 1)^2\, p(0) \\ & + \theta^2\,(3 - 2\theta)\, p(1) \\ & + \theta\,(\theta - 1)^2\, D(p)(0) \\ & + (\theta - 1)\,\theta^2 D(p)(1)\ . \end{aligned} \tag{15}$$

[It's better numerically and for efficiency *not* to write these this way, when the grade is much higher. For cubic Hermite, it doesn't matter.] This technique is usable by hand, but also works well in a computer algebra system such as Maple where one can compute residues easily.

"Takes all the fun out of it." — G. V. Parkinson

The general case

General Hermite interpolational polynomials can be constructed using the node polynomial $w(z) = \prod_{i=0}^{n}(z-\tau_i)^{s_i}$ by expanding

$$\frac{1}{w(z)} = \sum_{i=0}^{n}\sum_{j=0}^{s_i-1}\frac{\beta_{i,j}}{(z-\tau_i)^{j+1}} \tag{16}$$

which gives the "generalized barycentric weights" $\beta_{i,j}$. These can be further used to construct differentiation matrices* for polynomials given by the Hermite interpolational data on these nodes with the given confluencies s_j. That is, the data are local Taylor series for $f(z) = \rho_{i,0} + \rho_{i,1}(z-\tau_i) + \rho_{i,2}(z-\tau_i)^2 + \cdots \rho_{i,s_i-1}(z-\tau_i)^{s_i-1}$ known at each node, with known coefficients $\rho_{i,j} = f^{(j)}(\tau_i)/j!$ incorporating the factorials.

*Amiraslani, RMC, and Gunasingham, Differentiation Matrices for Univariate Polynomials 2020 (link goes to 2018 arXiv version)

Barycentric Hermite Interpolational Polynomials

The first and second forms are

$$p(z) = w(z) \sum_{i=0}^{n} \sum_{j=0}^{s_i-1} \sum_{k=0}^{j} \frac{\beta_{i,j}\rho_{i,k}}{(z-\tau_i)^{j+1-k}} \tag{17}$$

and

$$p(z) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{s_i-1} \sum_{k=0}^{j} \frac{\beta_{i,j}\rho_{i,k}}{(z-\tau_i)^{j+1-k}}}{\sum_{i=0}^{n} \sum_{j=0}^{s_i-1} \frac{\beta_{i,j}}{(z-\tau_i)^{j+1}}} \tag{18}$$

Both forms are quite numerically stable, unless the confluencies get too large. The second form allows scaling the $\beta_{i,j}$ to reduce the impact of overflow or underflow.

An example

Suppose our nodes are (again) $[-1,-1/2,1/2,1]$ and the values are $[1,-1,1,-1]$ and we also insist that the derivatives are zero at each node. The first barycentric form of the interpolant is

$$
\begin{aligned}
&(1+x)^2\left(x+\frac{1}{2}\right)^2\left(x-\frac{1}{2}\right)^2(x-1)^2 \\
&\cdot\left(\frac{76}{27\,(1+x)}+\frac{4}{9\,(1+x)^2}+\frac{32}{27\left(x+\frac{1}{2}\right)}-\frac{16}{9\left(x+\frac{1}{2}\right)^2}\right. \\
&\left.+\frac{32}{27\left(x-\frac{1}{2}\right)}+\frac{16}{9\left(x-\frac{1}{2}\right)^2}+\frac{76}{27\,(x-1)}-\frac{4}{9\,(x-1)^2}\right)
\end{aligned}
\tag{19}
$$

Converting to the monomial basis, this is

$$
\frac{x\left(16x^6-24x^4+x^2+5\right)}{2}\,. \tag{20}
$$

Its plot

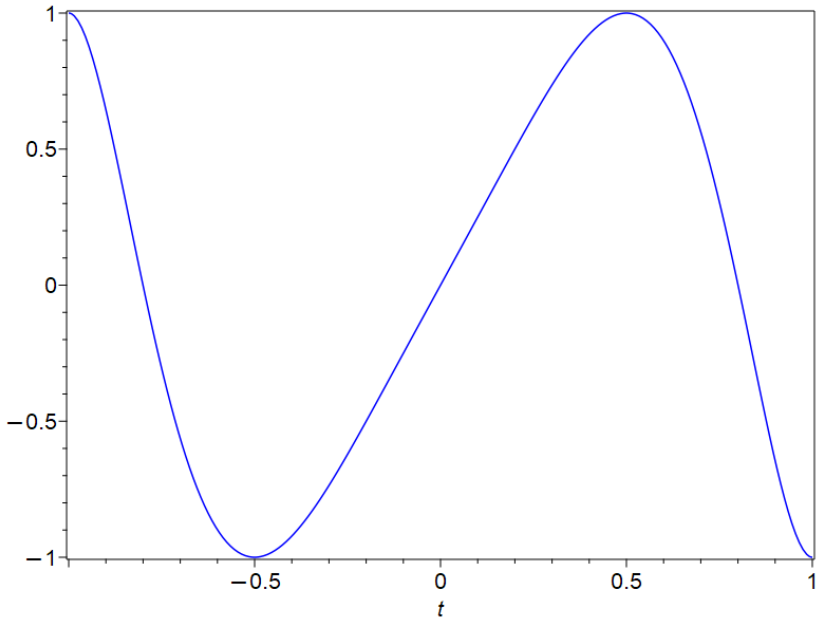

Figure 1: Hermite interpolating polynomial to the given data

Birkhoff interpolation

If some of the Hermite interpolational data is *missing*, then we have what is called a *Birkhoff* interpolation problem. Not all such are uniquely solvable.

Example 1: Asking for a degree two polynomial that satisfies $p(0) = p_0$, $p'(0) = d_0$, and $p'(1) = d_1$ has a unique solution.

Enter the Butcher Factor

If we insert a polynomial factor $B(z)$ in the numerator, of degree* m, then we can choose it so as to force m residues to be zero. Then the answer will not depend on the derivatives that correspond to those residues. Fix such a choice for $B(z)$. Then

$$0 = \frac{1}{2\pi i} \oint_C \frac{B(z)p(z)}{(z-\theta)w(z)}\, dz \tag{21}$$

where, say, the node polynomial $w(z)$ is $\prod_{k=0}^{n}(z-\tau_k)^{s_k}$. If $d = -1 + \sum_{k=0}^{n} s_k$ then the above formula will be valid for all polynomials $p(z)$ of grade $d - m$.

* We need degree and not just grade, to know the accuracy/order.

An example with a Butcher factor

Put $B(z) = z - b$. Then

$$\begin{aligned}\frac{B(z)}{(z-\theta)z^2(z-1)^2} =& \frac{b}{\theta\, z^2} + \frac{b-1}{(-1+\theta)\,(z-1)^2} \\ &+ \frac{-2b\theta + 3b + \theta - 2}{(z-1)\,(-1+\theta)^2} + \frac{\theta - b}{(z-\theta)\,\theta^2\,(-1+\theta)^2} \\ &+ \frac{2b\theta + b - \theta}{\theta^2 z} \end{aligned} \tag{22}$$

Continued

Setting $-2b\theta + 3b + \theta - 2 = 0$ forces the residue of $1/(z-1)$ to be zero, and so the interpolant will not depend on the value of $p(1)$. Carrying through the algebra gives

$$p(\theta) = p(0) + \frac{1}{2}\theta(2-\theta)p'(0) + \frac{1}{2}\theta^2 p'(1)\,. \tag{23}$$

We may verify that $p(0) = p(0)$, $p'(0) = p'(0)$, and $p'(1) = p'(1)$. This also tells us that $p(1) = p(0) + p'(0)/2 + p'(1)/2$, which "fills in" the missing data point.

Filling in one missing data point

John Butcher pointed out that if only one data point is missing, then we could use a simpler method: compute the partial fraction expansion of just

$$\frac{1}{z^2(z-1)^2} = -\frac{2}{z-1} + \frac{1}{(z-1)^2} + \frac{1}{z^2} + \frac{2}{z}\,, \tag{24}$$

and then the Cauchy Integral formula requires that for all grade 2 polynomials

$$-2p(1) + p'(1) + p'(0) + 2p(0) = 0\,. \tag{25}$$

Solving this for $p(1)$ gives $p(0) + p'(0)/2 + p'(1)/2$ as before. Now we can just use the ordinary Hermite interpolational formula with this data. That's for cubic polynomials, but it works for quadratics, too.

Another example

Example 2: Suppose that the nodes are as before but now the data is [[1,0], [undefined,0],[1,0],[−1,0]]. This is the same as before, but now we are missing one data point, which makes it a Birkhoff problem.

Expanding the output of our program finds

$$16x^6 - 6x^5 - 30x^4 + \frac{25}{2}x^3 + 12x^2 - \frac{15}{2}x + 2\,. \tag{26}$$

What a difference a point makes

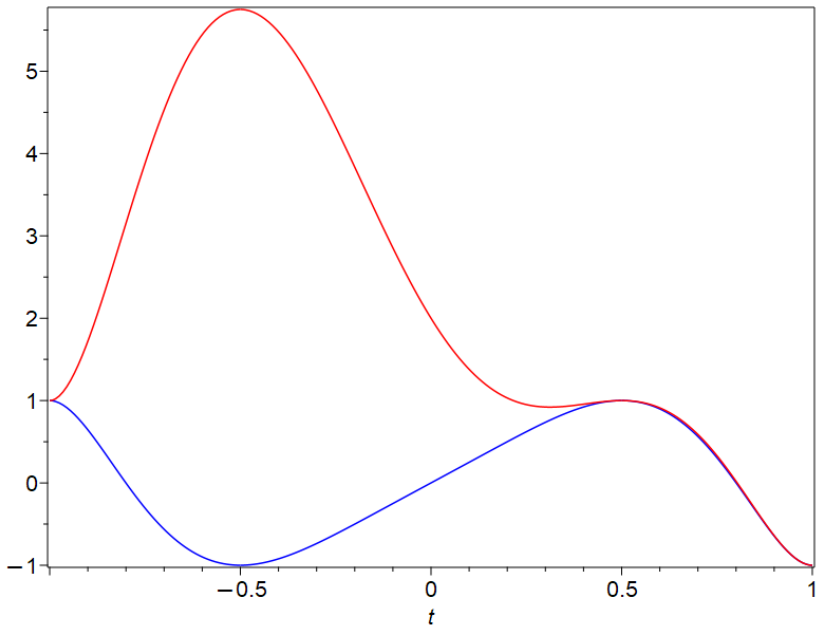

Figure 2: One missing bit of information

Butcher factors for Birkhoff interpolation

In our 2011 paper (previously linked) we used Butcher factors to solve some quite general Birkhoff interpolational problems. If the problem was poised, we were able to do so *except* on certain algebraic surfaces.

Another example: Setting many residues to zero

If $w(z) = \prod_{\ell=0}^{k} (z - \tau_\ell)^2$ and we want the residue of $B(z)/w(z)$ at $z = \tau_j$ for $1 \le j \le k$ to be zero, put $y = \prod_{\ell \ge 1 \& \ell \ne j} (z - \tau_\ell)^2$ and expand it in a local Taylor series at $z = \tau_j$, for instance by taking logarithms:

$$\ln y(z) = \sum_{\ell \ge 1 \ \& \ \ell \ne j} -2 \ln (z - \tau_\ell) + 2\pi i K \tag{27}$$

$$\frac{y'(z)}{y(z)} = -2 \sum_{\ell \ge 1 \ \& \ \ell \ne j} \frac{1}{z - \tau_\ell} \, . \tag{28}$$

Continued

So the local Taylor series for $y(z)$ is

$$y(z) = y(\tau_j) - 2y(\tau_j)\left(\sum_{\ell\geq 1\ \&\ \ell\neq j}\frac{1}{\tau_j-\tau_\ell}\right)(z-\tau_j) + O(z-\tau_j)^2\,. \quad (29)$$

Multiply by $B(z) = B(\tau_j) + B'(\tau_j)(z-\tau_j) + O(z-\tau_j)^2$ and we have, since $y(\tau_j)\neq 0$,

$$B'(\tau_j) - 2B(\tau_j)\left(\sum_{\ell\geq 1\ \&\ \ell\neq j}\frac{1}{\tau_j-\tau_\ell}\right) = 0\,. \quad (30)$$

as the required condition. Typically we will also require the residue at $z=\tau_0$ to be -1.

$$B'(\tau_0) - 2B(\tau_0)\left(\sum_{\ell\geq 1}\frac{1}{\tau_0-\tau_\ell}\right) = -1\,. \quad (31)$$

This gives $k+1$ equations, normally sufficient for a degree k Butcher factor.

Continuing

We find the vector of $B'(\tau_j)$ from the vector of $B(\tau_j)$ by the *Lagrange differentiation matrix* $\mathbf{D}$ on those nodes. We have $B' = \mathbf{D}B$. For example, if $\tau = [-1, -1/2, 1/2, 1]$ then the matrix is

$$\mathbf{D} = \begin{bmatrix} -\frac{19}{6} & 4 & -\frac{4}{3} & \frac{1}{2} \\ -1 & \frac{1}{3} & 1 & -\frac{1}{3} \\ \frac{1}{3} & -1 & -\frac{1}{3} & 1 \\ -\frac{1}{2} & \frac{4}{3} & -4 & \frac{19}{6} \end{bmatrix}. \tag{32}$$

A formula for generating these can be found in my 2013 book with Nic Fillion.

A linear system

$$\begin{bmatrix} -\frac{19}{6} - 2\sum & 4 & -\frac{4}{3} & \frac{1}{2} \\ -1 & \frac{1}{3} - 2\sum & 1 & -\frac{1}{3} \\ \frac{1}{3} & -1 & -\frac{1}{3} - 2\sum & 1 \\ -\frac{1}{2} & \frac{4}{3} & -4 & \frac{19}{6} - 2\sum \end{bmatrix} \begin{bmatrix} B(x_0) \\ B(x_1) \\ B(x_2) \\ B(x_3) \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{33}$$

So to find the values of B on the nodes (which will identify it) we must solve this linear system. The $\sum$'s on the diagonal are all different—I just didn't want to write them all out. They come from the equations we saw before, like

$$B'(\tau_j) - 2B(\tau_j) \left(\sum_{\ell \geq 1 \;\&\; \ell \neq j} \frac{1}{\tau_j - \tau_\ell} \right) = 0\,. \tag{34}$$

Some Maple code

In September 2009 I wrote a Maple procedure, `BHBIP.mpl`, to use this method. It can construct the Hermite–Birkhoff interpolational polynomial explicitly, or it can "fill in" the missing data.

Piers Lawrence wrote a Matlab version of the code.

Filling in missing data

As previously stated, if only one piece of data is missing, then we do not need a Butcher factor. Simply expanding $1/w(z)$ in partial fractions gives us (after contour integration) an equation for the missing piece of information.

More generally, if we are missing m pieces of information, we can use a Butcher factor of degree $m-1$ for each one.

Even better, if we wish to fill in *all* the missing data, we can do this all at once by inverting an $m-1$ by $m-1$ matrix*.

* This is nearly the only time I have ever explicitly used a matrix inverse in any of my codes. The reason is that each column of the inverse gives the coefficients of the Butcher factor of degree $m-1$ needed to identify one of the missing pieces of data.

A more complicated example

Suppose we know that $p(0) = y_0$, $p'(0) = d_0$, $p'(c_1) = f_1$, $p'(c_2) = f_2$, $p(1) = y_1$, and $p'(1) = d_1$. So the function values at $x = c_2$ and $x = c_1$ are missing. Then the fill-in technique needs only two one-by-one matrices to set the residues separately to zero. Formulae for the missing data look like

$$p(c_1) = ay_0 + bd_0 + cf_1 + df_2 + ey_1 + fd_1 \tag{35}$$

where (for instance)

$$b = -\frac{\left(3c_1c_2 - 5c_2^2 - c_1 + 3c_2\right) c_1 \left(c_1 - 1\right)^3}{2c_2 \left(10c_1c_2 - 5c_1 - 5c_2 + 3\right)} \,. \tag{36}$$

Notice the nontrivial combination of c_1 and c_2 in the denominator. For this technique to succeed, that polynomial cannot be zero. But in fact the problem is not poised if

$$2c_1c_2 \left(c_2 - 1\right) \left(c_1 - 1\right) \left(10c_1c_2 - 5c_1 - 5c_2 + 3\right) \left(c_2 - c_1\right) = 0 \,. \tag{37}$$

The forbidden c_1 and c_2

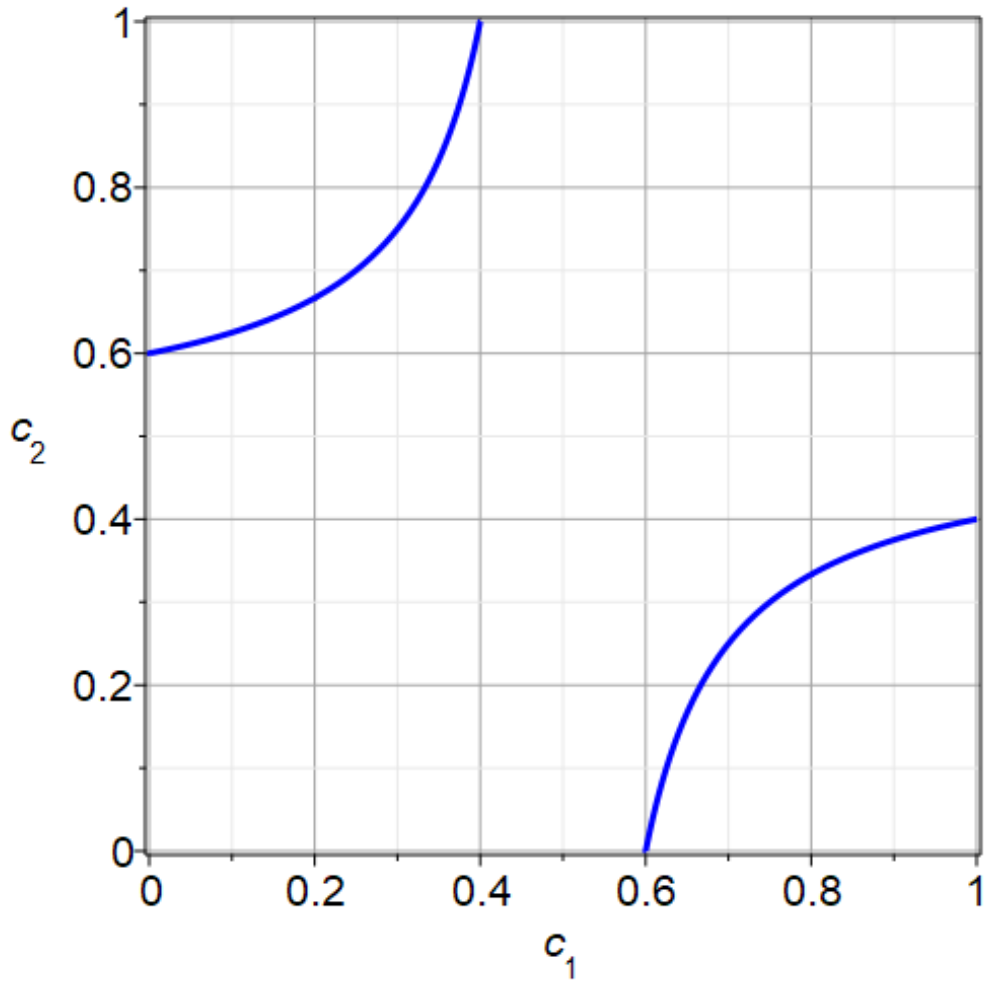

Figure 3: If the point (c_1, c_2) lies on the blue curves, or else if $c_1 c_2 (c_1 - c_2) = 0$, the Birkhoff interpolation problem is not poised.

Human vs Computer

Any given Birkhoff interpolation problem of explicit dimension can instead be solved directly by setting up a linear system of equations. The matrix will be a submatrix of a confluent Vandermonde matrix. For computers, it's not clear at first that the Butcher factor technique offers much advantage. The matrices needed are smaller, yes, but the setup is more involved.

A *big* disadvantage of the Vandermonde approach if floating-point arithmetic is involved is that this changes the basis, which if the grade is at all large may introduce serious numerical instability. It is (generally speaking) *much* better to stay in the same polynomial basis, if you can. Indeed, my preferred way to write a Butcher factor is in a Lagrange interpolational basis!

Human considerations

The gain in insight is potentially considerable, and moreover one can (as John did in 1967) solve problems of symbolic dimension, for arbitrary integers k. [One can do that with the code as well, but only by solving a few cases $k = 1$, $k = 2$, $k = 7$, whatever, and *guessing* the general form. If we're lucky we could prove it afterwards.]

A disadvantage of the Butcher factor is that sometimes, even if the problem is poised, the technique can fail. The Butcher factor is not allowed to have a zero at any of the nodes (and if we are unlucky, this can happen, say for symmetry reasons). The technique needs to be used carefully.

One such example

Suppose $\tau = [-1, 0, 1]$ are the nodes, and we know $p(\tau_0) = \rho_0$, $p''(\tau_1) = d_2$, and $p(\tau_2) = \rho_2$. This problem is poised: using the Vandermonde approach produces $p(z) = (\rho_0 + \rho_2 + d_2)/2 + (\rho_2 - \rho_0)z/2 + d_2 z^2/2$ easily enough. But asking for a Butcher factor that makes zero the two residues at $z = 0$ corresponding to $p(0)$ and $p'(0)$ forces $B(z) = \alpha(1 - z^2)$, which cancels the factors $(z + 1)(z - 1)$ of the node polynomial.

So the technique doesn't always work. [N.B. My code actually does work, for this case, because it's a bit more clever; but if instead we specify $p'''(0)$ and not $p''(0)$ then it fails, encountering what appears to be a bug. Sigh. Perhaps it's "too clever."]

Even so, the method can be useful

If one has code to evaluate Hermite interpolational polynomials already, and to differentiate them, then the "fill-in" technique allows easy conversion of Birkhoff data to usable Hermite interpolational polynomials, with *no* introduced numerical instability.

If time permits: Quadrature

Another trick is that one can directly integrate Hermite data: if $p(t)$ has known local Taylor series at various nodes, then its antiderivative $P(t)$ with $P(t_0) = 0$ and $P'(t) = p(t)$ has local Taylor series known at those nodes, as well, except for the as-yet undefined function values at t_1, t_2, ... That's a Birkhoff interpolational problem.

Example

Suppose $\tau = [-1, 0, 1]$ and $p(\tau) = [p_0, p_1, p_2]$. Then $P(\tau) = [[0,p_0], [NaN,p_1], [NaN,p_2]]$ (Maple uses `undefined` for `NaN`). Using the fill-in technique on this gives

$$P(\tau) = \left[[0, p_0], \left[\frac{5p_0}{12} + \frac{2p_1}{3} - \frac{p_2}{12}, p_1\right], \left[\frac{p_0}{3} + \frac{4p_1}{3} + \frac{p_2}{3}, p_2\right]\right] \quad (38)$$

and we see the familiar Simpson's rule formula pop out at the end. Note that p was grade 2, while P must be grade 3. [As is well-known, Simpson's rule has an extra degree of accuracy, one more than is shown here, because one residue is zero.]

If instead the nodes are $[-1, -1/2, 1/2, 1]$ then the integral at the end is $\frac{p_0}{9} + \frac{8p_1}{9} + \frac{8p_2}{9} + \frac{p_3}{9}$ which is accurate at least for p of grade 3.

This isn't completely straightforward if there are other missing data, and I have examples where the approach surprisingly fails.

But it's pretty good

Suppose for example that the nodes are $[-1, -1/2, 1/2, 1]$ and we know $f(-1) = p_0$, $f(1/2) = p_2$, and $f(1) = p_3$ but for some reason we only know $f'(-1/2) = d_1$. Finding a polynomial interpolant for that data is already a Birkhoff problem. Now suppose that what we really want is not f but its integral across the interval. Then the Maple code tells us that

$$\int_{-1}^{1} f(x)\,dx \approx \frac{25p_0}{9} + \frac{8d_1}{3} - \frac{16p_2}{9} + p_3 \tag{39}$$

and this formula will be exact for polynomials $f(x)$ of grade 3.

Open problems

Some years ago, Laureano Gonzalez-Vega and co-workers discussed open problems in Birkhoff interpolation. Many cases are still open; the only progress I am aware of is the 2001 paper by Rouillier, Safey El-Din, and Schost.

These open problems have some combinatorial flavour as well as analytic flavour: Given a real interpolation interval and a collection of sets of integers which describe which derivatives are known, the question is *for which sets of integers* are the Birkhoff interpolation problems poised for *every* choice of nodes in the interval?

In 2010 when we first published on the use of the Butcher factor for Birkhoff interpolation, we had hopes that this new approach would allow some progress on the open problem. But, life has interfered in the meantime and we have been unable to devote thought or attention to the opportunity. We still think there is a chance, though!

Thank you for listening.

This work was supported by NSERC, and by the Spanish MICINN. I am very grateful to CUNEF University for the opportunity to share this material here.

I would like to particularly thank Erik Postma and my other co-authors, and especially to thank John C. Butcher for teaching me the contour integral technique for interpolation, by which I (re)derived all these formulae.