Survey of Applied Mathematics Techniques

Brian Wetton [1]

July 3, 2018

[1] www.math.ubc.ca/~wetton, wetton@math.ubc.ca

Lecture 5

Interpolation, Quadrature, and Collocation Methods.

5.1 Interpolation

5.1.1 Linear interpolation

In Lecture #2 we saw the error in linear (tangent line) approximation was given by

$$f(x) - L(x) = \frac{1}{2} f''(\theta)(x-a)^2$$

where $L(x)$ is the linear approximation based at a, $L(x) = f(a) + f'(a)(x-a)$ and θ is some point between x and a.

We can show a similar result for linear interpolation between $f(a)$ and $f(b)$, that is

$$|f(x) - I(x)| \leq \frac{1}{8} \|f''\|_\infty (b-a)^2 \tag{5.1}$$

where $x \in (a, b)$, $I(x)$ is the linear interpolation

$$I(x) = f(a) + (f(b) - f(a)) \frac{(x-a)}{(b-a)}$$

and the maximum of the second derivative is taken over that same interval. Before we proceed with the proof, let us move the problem to a reference interval $y \in [-1, 1]$. We map x to y linearly

$$x = a + (b-a)(y+1)/2 \quad \text{and} \quad y = 2(x-a)/(b-a) - 1$$

We can now formulate the problem in y, that is to bound the difference between

$$f(y) - I(y)$$

with $y \in (-1, 1)$ and

$$I(y) = f(a)\frac{1-y}{2} + f(b)\frac{1+y}{2}.$$

From the form above we can identify the functions $(1-y)/2$ and $(1+y)/2$ as *shape functions* of the given data. Each function has value 1 at the given data and 0 at the other.

Note 1. *Sometimes the reference interval [0,1] is more convenient.*

We can proceed with the estimate of $f(y) - I(y)$ at a given point $c \in (-1, 1)$ by considering the function

$$q(y) = f(y) - I(y) - \frac{f(c) - I(c)}{1 - c^2}(1 - y^2)$$

which has zeros at the three distinct points -1, c, and 1, and so (by Rollé's Theorem applied three times) a point $\theta \in (-1, 1)$ exists at which $q''(\theta) = 0$. Taking two derivatives of the function q above ($I'' \equiv 0$ since it is linear) we have

$$f(c) - I(c) = -\frac{1}{2}\frac{d^2 f}{dy^2}(\theta)(1 - c^2) \tag{5.2}$$

and since $|1 - c^2| \leq 1$ on the interval,

$$|f(c) - I(c)| \leq \frac{1}{2}\|\frac{d^2 f}{dy^2}\|_\infty. \tag{5.3}$$

Transforming the derivatives back to the original interval in x gives

$$\frac{d^2 f}{dx^2} = \frac{d^2 f}{dy^2}\frac{(b-a)^2}{4}$$

and we have the desired result (5.1). Considering $h = (b-a)$ the interval width, we call linear interpolation a *second order* method.

5.1.2 Cubic Hermite interpolation

Consider now interpolation where we know the values and derivatives of the function at a and b. We can construct a third order polynomial (cubic) approximation I to the function at that point. This is easiest to do in the reference interval

$$\begin{aligned} I(y) \quad &= \quad f(-1)(\frac{1}{2} - \frac{3}{4}y + \frac{1}{4}y^3) + f(1)(\frac{1}{2} + \frac{3}{4}y - \frac{1}{4}y^3) + \\ & \quad\quad f'(-1)(\frac{1}{4} - \frac{1}{4}y - \frac{1}{4}y^2 + \frac{1}{4}y^3) + f'(1)(-\frac{1}{4} - \frac{1}{4}y + \frac{1}{4}y^2 + \frac{1}{4}y^3) \end{aligned}$$

Notice that the shape function $\phi_1(y) = 1/2 - 3y/4 + y^3/4$ multiplying $f(-1)$ has the expected properties, $\phi_1(-1) = 1$, $\phi_1(-1) = 0$, $\phi_1'(-1) = 0$, and $\phi_1'(1) = 0$.

To examine the approximation properties of this interpolation at $y = c$ we play the same game as before and consider

$$q(y) = f(y) - I(y) - \frac{f(c) - I(c)}{(1 - c^2)^2}(1 - y^2)^2.$$

We have $q(-1) = q(c) = q(1) = q'(-1) = q'(1) = 0$ and after repeated uses of Rollé's Theorem we find that $q''''(\theta) = 0$ from which it follows that

$$f(c) - I(c) = \frac{1}{24}\frac{d^4 f}{dy^4}(1 - c^2)^2 \tag{5.4}$$

and so

$$|f(c) - l(c)| \leq \frac{1}{384}\|\frac{d^4 f}{dx^4}\|_\infty (b - a)^4.$$

We see that cubic Hermite interpolation is a fourth order approximation.

In general, under intuitive conditions, if you fit n data for a function f with a $n-1$ degree polynomial it will give an order n interpolant. In higher dimensions the allowable placement of interpolation points becomes an interesting question.

Note 2. *We have done* interpolation*, that is approximating function values inside the scaled interval [-1,1]. We could use the same formulas for* extrapolation *outside the interval. However the final expressions cannot be used since they involve bounds of functions inside the interval, but the original expressions (5.2) and (5.4) can be used. Note that the error bound grows rapidly as the extraoplation point moves outside the interval. In this setting, we can say that the tangent line approximation is always an extrapolation procedure.*

5.2 Derivative Approximation

We can use the derivatives of the interpolant to find approximations to the derivative of the original function. If we consider linear interpolation, the derivative of the interpolant is just a constant, $(f(1) - f(-1))/2$. This is a first order approximation everywhere except at the midpoint $y = 0$, where it is second order accurate (centred differencing). This can be seen by expanding the expression above in Taylor series about $y = 0$. It is also possible to see using the same machinery that we used above. Consider linear interpolation, and the function we used to derive the error expression at $y = c$:

$$q(y) = f(y) - I(y) - \frac{f(c) - I(c)}{1 - c^2}(1 - y^2).$$

This q was constructed so that $q(-1) = q(c) = q(1) = 0$ so that we had $q''(\theta_1) = 0$ and the second derivative of the last term gave us information about the quantity that we wanted, $f(c) - I(c)$. Let us modify this idea to give us information about $f'(c) - I(c)$:

$$Q(y) = f(y) - I(y) - \frac{f(c) - I(c)}{1 - c^2}(1 - y^2) + \frac{f'(c) - I'(c) - 2c\frac{f(c) - I(c)}{1 - c^2}}{1 - c^2}(1 - y^2)(y - c)$$

where $(1-y^2)(y-c)/(1-c^2)$ is the functions with zero value at $y = \pm 1, c$ and derivative value 1 at $y = c$ so that $Q(-1) = Q(c) = Q'(c) = Q(1) = 0$. Thus we have $Q''''(\theta_2) = 0$ which results in

$$f'(c) - I'(c) = 2c\frac{f(c) - I(c)}{1 - c^2} + \frac{1}{6}f'''(\theta_2)(1 - c^2).$$

We can use the value interpolation result to simplify the first term on the RHS

$$f'(c) - I'(c) = cf''(\theta_1) + \frac{1}{6}f'''(\theta_2)(1 - c^2).$$

If we then scale back to an original interval of length h, remembering that $d/dy = \frac{h}{2}d/dx$ we obtain

$$f'(x) - I'(x) = (x - x_m)\frac{h}{2}f''(\theta_1) + \frac{h^2}{24}f'''(\theta_2)c(x).$$

where the primes above are now x derivatives, x_m is the midpoint of the interval, $c(x)$ is the quadratic with values 1 at x_m and zero at the interval end points, and the θ values have been mapped. This is the desired result: notice that the dominant first order error vanishes at the interval midpoint.

You can use the same kind of argument to show that with the cubic Hermite interpolant I, we have I' approximates f' to third order accuracy in the interval, with increased (fourth order) accuracy in the interval centre. This fact will be used in the derivation of the method used for the collocation method described in Section 5.4.2.

5.3 Numerical Quadrature

Quadrature is an alternate term for numerical integration. We consider that an integral from a to b has been broken into N subintervals of length $h = (b-a)/N$ and consider the integrals on these subintervals mapped to the reference interval [-1,1].

Just as we approximated derivatives by derivatives of interpolating functions, we can approximate integrals in the same way. If we have

$$f(y) \approx I(y) = \sum_{j=1}^{q} f(y_j)\phi_j(y)$$

then we take

$$\int_{-1}^{1} f(y)dy \approx \int_{-1}^{1} I(y)dy = \sum_{j=1}^{q} f(y_j) \int_{-1}^{1} \phi_j(y)dy$$

where we have an interpolant based on q function values at points y_j with shape functions ϕ_j. This gives us the general form of a quadrature rule on the reference interval

$$\int_{-1}^{1} f(y)dy \approx \sum_{j=1}^{q} w_j f(y_j)$$

where w_j are the *quadrature weights*, given by $\int_{-1}^{1} \phi_j(y)dy$ in the case of a quadrature coming from an interpolant. The definition can be extended naturally to interpolants that include derivative values.

If we consider the quadrature that comes from linear interpolation, we have

$$\int_{-1}^{1} f(y)dy \approx f(-1) + f(1)$$

which is known as the Trapezoidal Rule. We can get an error estimate for this rule using (5.2). Note that this gives a better constant than the global bound on the interval (5.3). We have

$$\left|\int_{-1}^{1} (f(y) - I(y))dy\right| \leq \int_{-1}^{1} |f(y) - I(y)|dy \leq \frac{h^2}{8} K_2 \int_{-1}^{1} (1 - y^2)dy = \frac{h^2}{6} K_2$$

where $K_2 = \max |d^2 f/dx^2|$ over the original interval of length h. This is the right context to identify the order of accuracy of the quadrature, second order in this case. If we transform back to the original interval of length h:

$$\left|\int (f(x) - I(x))dx\right| \leq \frac{h^3}{12} K_2.$$

While locally third order, the total integral is over $N = (b - a)/h$ subintervals and this total integral will then be second order accurate.

Theorem 1. *If a quadrature rule is exact for polynomials of order q then it has order $q + 1$ accuracy.*

This is easy to prove using the Taylor polynomial approximation of order q of f, for which the quadrature is exact.

It would seem that the game now is to use interpolants with q points that fit polynomials of order $q - 1$ and so give order q quadrature rules, but we can do better. The idea is that interpolants have to be accurate at all points in the interval, but quadrature only has to get the integral correct. Another way to view this is that we can introduce interpolating points for which the shape functions integrate to zero to get higher order accuracy with fewer quadrature points.

Gaussian quadrature gives order $2q$ with q quadrature points. The first two examples are

- $q = 1$: $y_1 = 0$, $m_1 = 2$ (midpoint rule, second order accurate)
- $q = 2$: $y_{1,2} = \pm 1/\sqrt{3}$, $m_{1,2} = 1$ (fourth order)

Gaussian quadrature is discussed further in a hand-written appendix to these notes.

There is also a family of high order quadrature methods that include the end points that have accuracy $2q - 2$ with q points. The first is $q = 2$ Trapezoidal Rule discussed above, and the next is Simpson's rule, $q = 3$: $y_{1,2} = \pm 1$, $m_{1,2} =$

1/3, $y_3 = 0$, $m_3 = 4/3$. This can be derived from a quadratic interpolant and the extra order of accuracy comes from the cancellation of the next order error term due to symmetry.

Note 3. *We will be discussing quadrature again in the context of the Finite Element Method in Lecture #7 next week.*

5.4 Collocation Methods

Collocation methods for solving differential equations involve discretizing on sub-intervals, with interpolation points on the subintervals and the interpolating values as the unknowns. The equations are obtained by applying boundary and interface conditions at the subinterval ends and applying the differential equations to the interpolants at given points, called collocation points. A well known and highly adaptable collocation code for boundary value problems is available in MATLAB, the `bvp4c` routine. Some of the details of the algorithm on which it is based are described below.

5.4.1 Aside: Writing higher order equations as a first order system

Our original and favourite problem is the second order periodic boundary value problem for $u(x)$:

$$-u'' + u = f(x).$$

The `bvp4c` collocation method described below works only for first order systems. There is a trick to converting higher order problems into first order systems. Here, we introduce $v = u'$. Now the problem is equivalent to the system

$$\begin{aligned} u' &= v \\ v' &= u - f \end{aligned}$$

For periodicity we would add the conditions $u(0) = u(1)$ and $v(0) = v(1)$. Note that if we solve this system, v will be u' and u will solve the original problem. Third order scalar ODEs can be converted into a first order vector system with three components, etc.

This idea is also used when time stepping equations with higher order time derivatives. Almost all time stepping methods are designed for first order systems.

5.4.2 Some details of the MATLAB `bvp4c` routine

We consider a vector, first order system for $\mathbf{u}(x)$ (with m components) on the unit interval. The interval is divided into N subintervals and the unknowns will be the $m(N+1)$ values at the subinterval ends. First order equations

$$\mathbf{u}' = \mathbf{g}(\mathbf{u}, x) \tag{5.5}$$

are given (could be nonlinear) along with m boundary conditions (also possibly nonlinear):

$$\mathbf{b}(\mathbf{u}(0), \mathbf{u}(1)) = 0. \tag{5.6}$$

To approximate the problem, we start by adding the m equations for the boundary conditions directly into the discrete system. To approximate the equations, we proceed as follows: the unknowns at subinterval ends lead to derivative values at the subinterval ends by applying the differential equation (5.5). These values provide the data for a Hermite cubic polynomial fit in each subinterval. There are now two ways to compute approximate derivatives at subinterval centres, both of which are fourth order accurate: first by evaluating the derivative of the cubic, and second by evaluating the approximation of $\mathbf{u}$ at subinterval centres and applying (5.5). The residual in the difference of these approximations provides mN equations, specifying the method.

Note that the method satisfies the boundary conditions exactly and is consistent with a fourth order truncation error. Proving convergence requires adding conditions to the types of equations (5.5) and boundary conditions (5.6) considered as not all problems in this form are well-posed.

For nonlinear problems `bvp4c` computes derivatives for Newton-type interations with numerical approximation. For this reason it is good to give it problems after scaling. The routine also has heuristic techniques to estimate the local error in the solution and will adapt the subintervals to be smaller where the truncation error is estimated to be larger. Notice that nothing in the method required uniform sizes to the subintervals. Finally, the routine allows for the differential equations and boundary conditions to have parameters. Additional "boundary" conditions are added for each parameter.

Note 4. *The resulting approximation will be C_1 in x since values and derivatives match at interval end points. For the second order problem, the solution will be C_2 since its derivative is one of the components of the vector solution and it is C_1.*

Note 5. *The form for the equations (5.5) and boundary conditions (5.6) is very general, but does require the derivatives to be given explicity in terms of the values. This is not true, for example, in the equations that came for the nonlinear stretched wire equations in Lecture #3.*

5.5 MATLAB `bvp4c` Examples

Some computational examples of the use of `bvp4c` are given below. The code will be posted on the web page of the lecture notes. More than just examples of this particular code, this is an example of the kind of robust, accurate numerical codes that exist for some problems and are available in commercial and freeware packages. Although this code takes a while to learn how to use, there is no reason to solve these types of boundary value problems with your own codes when this is available. There are plenty of computational problems left that do not fit into available software so (as applied mathematicians) we will not run out of work to do.

5.5.1 Problem A2

The approach for Problem A formulated as a first order system as shown in Section 5.4.1 are applied to the test problem of Lecture #2 in the `bvp_probA.m` code.

5.5.2 Problem A, nonlinear version

The code above is easily modified to handle the nonlinear version of the problem introduced in the Lecture #3 notes. We use the linear problem to provide an initial guess. This is in the `bvp_probAnonlin.m` code.

5.5.3 Eigenvalue problem

Consider now the "easy" eigenvalue problem from Lecture #4. That is, find the smallest eigenvalue λ of

$$-u'' = \lambda u$$

with $u(0) = 0$ and $u' + u = 0$ at $x = 1$. We want it to be normalized so that

$$\int_0^1 u^2(x)dx = 1.$$

This can be implemented by adding a third component $y(x)$ to the ODE system, with

$$\frac{dy}{dx} = u^2$$

and $y(0) = 0$ and $y(1) = 1$. Some care has to be made with the initial guess to the solution so that we get the right eigenvalue (remember there are an infinite number). This is in the `bvp_eigen.m` code.

5.6 Lecture #5 Problems

Problem 1. *Show that the derivative of the Hermite interpolant at the centre of the interval gives a fourth order accurate approximation.*

Problem 2. *Consider quadratic approximation based on the points $y = 0, \pm 1$ in the reference interval. Derive an error expression for its use to approximate derivatives. Note that is will be third order accurate at isolated points in the interval.*

Problem 3. *Show that the collocation method described in Section 5.4.2, when applied to our usual boundary value problem A2 on a uniform grid, is equivalent to a higher order Finite Difference method.*

Problem 4. *Code up the collocation method described in Section 5.4.2. Check that fourth order convergence is achieved in values and derivatives of* u *when applied to the test problem in Lecture #2.*

Problem 5. *Consider* $f \in C_2$ *and its linear interpolation at a point* x*. The error in linear interpolation involves the value of the second derivative at* θ *that depends on* x*. Fix the map* $\theta(x)$ *(which could be muti-valued) by taking the smallest value. What can you say about the regularity of* $\theta(x)$*? Can you show it is measurable? Assume more regularity on* f *if you need it in your argument.*