Numerical Differentiation

KEY WORDS. Numerical differentiation, error and order of approximation, Richardson extrapolation.

GOAL.

- To understand the derivation of numerical differentiation formulas and their errors.
- To understand the application of numerical differentiation formulas in the solution of differential equations.

1 Introduction

So far, we have used the interpolating polynomial to approximate values of a function $f(x)$ at points where $f(x)$ is unknown. Another use of the interpolating polynomial of equal or even greater importance in practice is the imitation of the fundamental operations of calculus. In all of these applications, the basic idea is extremely simple: instead of performing the operation on the function $f(x)$ which may be difficult or, in cases where $f(x)$ is known at discrete points only, impossible, the operation is performed on a suitable interpolating polynomial. In this chapter, we consider the operation of differentiation.

2 Numerical Differentiation Formulas

Consider the case when $n = 1$ so that

$$p_1(x) = f_0 + f[x_0, x_1](x - x_0).$$

Then

$$p_1'(x) = f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0}. \tag{1}$$

If we set $x = x_0$ and $x_1 = x + h$, then $x_1 - x_0 = h$ and

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \equiv D_h^F(f), \tag{2}$$

which is called the *forward difference formula.* From elementary calculus,

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}. \tag{3}$$

Of interest in computations is how good this approximation is. This is measured by estimating the error

$$E = f'(x) - \frac{f(x+h) - f(x)}{h}.$$

This could be done using (11). An alternative approach is to use Taylor's Theorem in the following way. Since

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(\xi),$$

for some ξ between x and $x+h$, then

$$\frac{f(x+h) - f(x)}{h} = f'(x) + \frac{1}{2}hf''(\xi),$$

so that

$$E = -\frac{1}{2}hf''(\xi).$$

If, for every $t \in [a,b]$, $|f''(t)| \leq M_2$, where M_2 is a positive constant, then

$$|E| \leq \frac{1}{2}hM_2,$$

and we write

$$|E| = O(h). \tag{4}$$

Roughly speaking, this means that if h is halved, the error will be reduced by a factor of 2. Since h appears in (4) raised to the power 1, the forward difference formula (2) is said to be *first order*.

EXAMPLE 1. Using $D_h^F(f)$ of (2), approximate the derivative of the function $f(x) = e^x$ at $x = 0$. Results for various values of h are given in Table 1. Ratio is the quantity $D_h^F(f)/D_{h/2}^F(f)$ and shows that, for h sufficiently small, the error is reduced by a factor of two when h is halved.

h	D_h^F	Error	Ratio
0.2000	1.1070	0.1070	0
0.1000	1.0517	0.0517	2.0695
0.0500	1.0254	0.0254	2.0340
0.0250	1.0126	0.0126	2.0168
0.0125	1.0063	0.0063	2.0084

Table 1. Example of numerical differentiation using (2).

Returning to (1) and setting $x_0 = x - h$ and $x_1 = x + h$, we have $x_1 - x_0 = 2h$ and

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \equiv D_h^C(f), \tag{5}$$

the *central difference formula*. Note that, in this case, x is not an interpolation point but $\omega_2'(x) = 2x - (x_0 + x_1) = 0$; cf., (12). To derive an expression for the error in the approximation $D_h^C(f)$, note that

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \frac{1}{6}h^3 f'''(\xi_1),$$

and

$$f(x-h) = f(x) - hf'(x) + \frac{1}{2}h^2 f''(x) - \frac{1}{6}h^3 f'''(\xi_2),$$

where ξ_1 and ξ_2 lie between x and $x+h$, and $x-h$ and x, respectively. Then

$$E = f'(x) - \frac{f(x+h) - f(x-h)}{2h} = -\frac{h^2}{6}\left(\frac{f'''(\xi_1) + f'''(\xi_2)}{2}\right) = -\frac{h^2}{6} f'''(\xi), \tag{6}$$

for some ξ between ξ_1 and ξ_2, on using the Intermediate Value Theorem. Thus, in this case,

$$E = O(h^2),$$

and the central difference formula (5) is *second order*.

Example 2. Repeat Example 1 using $D_h^C(f)$ of (5). From Table 2, we see that the error is now reduced by a factor of 4 when the value of h is halved.

h	D_h^C	Error	Ratio
0.2000	1.0067	0.0067	0
0.1000	1.0017	0.0017	4.0060
0.0500	1.0004	0.0004	4.0015
0.0250	1.0001	0.0001	4.0004
0.0125	1.0000	0.0000	4.0001

Table 2. Example of numerical differentiation using (5).

To obtain an approximation to $f''(x)$, consider

$$p_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1),$$

the polynomial of degree ≤ 2 interpolating $f(x)$ at x_0, x_1, and x_2. Then

$$p_2''(x) = 2f[x_0, x_1, x_2],$$

and if $x = x_1$, $x_0 = x - h$ and $x_2 = x + h$, it follows that

$$p_2''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} \equiv D_h'' f(x). \tag{7}$$

Using Taylor series expansions, it can be shown that

$$f''(x) - D_h'' f(x) = -\frac{h^2}{12} f^{(4)}(\xi_x), \tag{8}$$

where ξ_x lies between $x - h$ and $x + h$.

3 The Error in Numerical Differentiation

Let $f(x)$ be a function defined on an interval I containing the set of points $S = \{x_i\}_{i=0}^n$, and let $p_n(x)$ be the polynomial of degree $\leq n$ interpolating $f(x)$ at these points. We wish to approximate $f'(x)$ by $p_n'(x)$ for $x \in I$, and derive a formula for the error that is expected in this approximation. If $f(x)$ has continuous derivatives of order $n+1$ in I and if $x \in I$, then we have seen in Chapter 3 that

$$f(x) - p_n(x) = \omega_{n+1}(x)g(x), \tag{9}$$

where

$$\omega_{n+1}(x) \equiv (x - x_0)(x - x_1)\cdots(x - x_n) = \prod_{j=0}^{n}(x - x_j),$$

and

$$g(x) = \frac{1}{(n+1)!}f^{(n+1)}(\xi_x), \tag{10}$$

where ξ_x is some unknown point in the smallest interval containing the set S and x. On differentiating (9), assuming that all derivatives exist, we obtain

$$f'(x) - p_n'(x) = \omega_{n+1}'(x)g(x) + \omega_{n+1}(x)g'(x).$$

If x is arbitrary, this formula is almost useless in practice as we do not know $g(x)$ as a function of x, because the unknown point ξ_x in (10) is a function of x, and, as a consequence, we cannot perform the differentiation. However, if $x = x_j$, one of the interpolating points, then

$$\omega_{n+1}(x_j) = 0,$$

and

$$f'(x_j) - p_n'(x_j) = \omega_{n+1}'(x_j)g(x_j), \tag{11}$$

where

$$\omega_{n+1}'(x_j) = \prod_{i=0, i\neq j}^{n}(x_j - x_i).$$

Note also that if $x \in I$ is such that

$$\omega_{n+1}'(x) = 0,$$

then

$$f'(x) - p_n'(x) = \omega_{n+1}(x)g'(x). \tag{12}$$

Remark on the accuracy on numerical differentiation: The basic idea of numerical differentiation is very simple. Given the data $\{(x_i, f_i)\}_{i=0}^n$, determine the interpolating polynomial $p_n(x)$ passing through these points. Then differentiate this polynomial to obtain $p_n'(x)$, whose value for any given x is taken as an approximation to $f'(x)$. This is basically an unstable process and we cannot normally expect great accuracy, even when the original data are correct. For one thing, very little can be said about the accuracy at a nontabular point. The graph of the interpolating polynomial will generally oscillate about the true curve and, at a nontabular point, the slope of the interpolating polynomial, $p'_n(x)$, may be quite different from the slope of the true function. In addition, the values of $f(x)$ at the interpolating points will normally be inaccurate because of rounding or other errors. It is entirely possible for the displacements at the tabular point to be very small while the slopes may be considerably different.

4 The Method of Undetermined Coefficients

The method of undetermined coefficients is a procedure used in deriving formulas for numerical differentiation and numerical integration. We describe this method by means of an example. Suppose

$$f'(x) \approx D_h f(x) \equiv Af(x-h) + Bf(x) + Cf(x+h), \tag{13}$$

where A, B, C are constants to be determined so that $D_h f(x)$ is as accurate an approximation as possible. Now, from Taylor series,

$$f(x-h) = f(x) - hf'(x) + \frac{1}{2!}h^2 f''(x) - \frac{1}{3!}h^3 f'''(x) + \frac{1}{4!}h^4 f^{(4)}(x) + \cdots \tag{14}$$

$$f(x) = f(x) \tag{15}$$

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2!}h^2 f''(x) + \frac{1}{3!}h^3 f'''(x) + \frac{1}{4!}h^4 f^{(4)}(x) + \cdots \tag{16}$$

If we multiply (14), (15), (16) by A, B, C, respectively, and add, then, from (13), we have

$$\begin{aligned} D_h f(x) &= (A+B+C)f(x) + (C-A)hf'(x) + (A+C)\frac{1}{2}h^2 f''(x) + (C-A)\frac{1}{3!}h^3 f'''(x) \\ &\quad + (A+C)\frac{1}{4!}h^4 f^{(4)}(x) + \cdots \end{aligned}$$

If

$$D_h f(x) \approx f'(x)$$

for an arbitrary function $f(x)$, we require

$$\begin{aligned} A+B+C &= 0, \\ h(C-A) &= 1, \\ A+C &= 0. \end{aligned} \tag{17}$$

On solving this system, we obtain

$$A = -C = -1/2h, \quad B = 0.$$

Then

$$D_h f(x) \equiv \frac{f(x+h) - f(x-h)}{2h}$$

and

$$D_h f(x) = f'(x) + \frac{1}{3!} h^2 f'''(x) + \cdots$$

An equivalent approach is to choose the coefficients A, B and C so that $D_h f(x)$ is exact for polynomials of as high degree as possible. This is equivalent to making

$$D_h f(x) = f'(x)$$

for

$$f(x) = x^j, \quad j = 0, 1, 2, \ldots.$$

If $f(x) = 1$ then

$$D_h' f(x) = f'(x)$$

gives

$$A + B + C = 0.$$

Then, if $f(x) = x$ and $f(x) = x^2$, we obtain $h(A-C) = 1$ and $h^2(A+C)/2 = 0$, respectively. These are exactly the equations of (17) but are much easier to obtain.

Consider now the problem of determining an approximation to the second derivative thus:

$$f''(x) \approx D_h'' f(x) \equiv Af(x-h) + Bf(x) + Cf(x+h).$$

We choose the coefficients A, B and C so that $D_h^{(2)} f(x) = f''(x)$ is exact for $f(x) = x^j$ for $j = 0, 1, 2, \ldots$. If $f(x) = 1$ then

$$D_h^{(2)} f(x) = f''(x)$$

gives

$$A + B + C = 0.$$

Then, if $f(x) = x$ and $f(x) = x^2$, we obtain $h(A-C) = 0$ and $h^2(A+C)/2 = 1$, respectively. Thus, $A = C = 1/h^2$ and $B = -2/h^2$, and

$$D_h^{(2)} f(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}. \tag{18}$$

Note that, when $f(x) = x^3$,

$$D_h^{(2)} f(x) = f''(x),$$

but, when $f(x) = x^4$,

$$D_h^{(2)}(x) \neq f''(x).$$

Using the Taylor series approach,

$$f''(x) - D_h^{(2)} f(x) \approx -\frac{h^2}{12} f^{(4)}(x). \tag{19}$$

5 An Application of Numerical Differentiation Formulas

Consider the second order linear two–point boundary value problem,

$$\begin{cases} -u''(x) + \kappa u(x) = f(x), \quad x \in I, \\ \\ u(0) = g_0, \quad u(1) = g_1, \end{cases} \tag{20}$$

where $I = [0, 1]$. Let $\{x_i\}_{i=0}^{N+1}$ be a uniform partition of the interval I such that $x_i = ih, i = 0, 1, \ldots, N+1$, and $(N+1)h = 1$. Then

$$-u''(x_i) + \kappa u(x_i) = f(x_i), \quad i = 1, \ldots, N,$$

or, using (18),

$$-\frac{u(x_{i-1}) - 2u(x_i) + u(x_{i+1})}{h^2} + E_i + \kappa u(x_i) = f(x_i), \quad i = 1, \ldots, N,$$

where, from (8),

$$|E_i| = O(h^2).$$

Thus, the basic finite difference method for solving (20) involves determining the numbers $\{u_i^h\}_{i=0}^{N+1}$, where $u_i^h \approx u(x_i)$, such that

$$\begin{cases} u_0^h = g_0, \\ \\ -\frac{u_{i-1}^h - 2u_i^h + u_{i+1}^h}{h^2} + \kappa u_i^h = f(x_i), \quad i = 1, \ldots, N, \\ \\ u_{N+1}^h = g_1; \end{cases} \tag{21}$$

or

$$\begin{cases} 2u_1^h - u_2^h + h^2 \kappa u_1^h = h^2 f(x_1) + g_0, \\ \\ -u_{i-1}^h + 2u_i^h - u_{i+1}^h + h^2 \kappa u_i^h = h^2 f(x_i), \quad i = 2, \ldots, N-1, \\ \\ -u_{N-1}^h + 2u_N^h + h^2 \kappa u_N^h = h^2 f(x_N) + g_1. \end{cases} \tag{22}$$

Equations (22) may be written in the form

$$A\mathbf{u} = \mathbf{f}, \tag{23}$$

where

$$A = \begin{bmatrix} 2 + h^2\kappa & -1 & & & \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 + h^2\kappa \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1^h \\ u_2^h \\ \vdots \\ \vdots \\ u_{N-1}^h \\ u_N^h \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} h^2 f(x_1) + g_0 \\ h^2 f(x_2) \\ \vdots \\ \vdots \\ h^2 f(x_{N-1}) \\ h^2 f(x_N) + g_1 \end{bmatrix}.$$

This tridiagonal system of equations can be solved using the MATLAB function *tridisolve* available from

`www.mathworks.com/Moler`

It can be shown that the difference problem (21) has a unique solution $\{u_i^h\}_{i=0}^{N+1}$, which is second order accurate; that is,

$$|u(x_i) - u_i^h| = O(h^2), \quad j = 1, \ldots, N.$$

Example 5.1. Consider $f(x) = \sin(x) - 1$, $\kappa = 0$, $g_0 = g_1 = 0$, the boundary value problem becomes

$$\begin{cases} -u''(x) = -\sin(x) + 1, \quad x \in [0, \pi], \\ \\ u(0) = 0, \quad u(1) = 1, \end{cases} \tag{24}$$

with exact solution $u(x) = -\sin(x) - \frac{x^2}{2} + \pi\frac{x}{2}$. The A matrix and $\mathbf{f}$ vector can be constructed as shown above. The Matlab implementation is the following.

```
function boundary_value_problem (N)

x = linspace(0, pi, N+1);
f = -(sin(x) - 1);
h = pi/N;

A = zeros(N+1, N+1);
A(1, 1) = 1;
A(N+1, N+1) = 1;
for i=2:N
 A(i, i-1) = -1/h/h;
 A(i, i) =  2/h/h;
 A(i, i+1) = -1/h/h;
end
b = zeros(N+1, 0);
b(1, 1) = 0;
b(N+1, 1) = 0;
for i=2:N
b(i, 1) = f(i);
end

u = inv(A)*b;

plot(x, u, 'go');
hold on
plot(x, -sin(x) - x.*x/2 + pi*x/2, '-k');
legend ('numerical approximation', 'true solution')
```

```
err = 0
for i=1:N+1
err = err + abs(u(i)-(-sin(x(i))-x(i)*x(i)/2 + pi*x(i)/2));
end
err/N

end
```