CHAPTER 11

Numerical Differentiation

Differentiation is a basic mathematical operation with a wide range of applications in many areas of science. It is therefore important to have good methods to compute and manipulate derivatives. You probably learnt the basic rules of differentiation in school — symbolic methods suitable for pencil-and-paper calculations. Such methods are of limited value on computers since the most common programming environments do not have support for symbolic computations.

Another complication is the fact that in many practical applications a function is only known at a few isolated points. For example, we may measure the position of a car every minute via a GPS (Global Positioning System) unit, and we want to compute its speed. When the position is known at all times (as a mathematical function), we can find the speed by differentiation. But when the position is only known at isolated times, this is not possible.

The solution is to use approximate methods of differentiation. In our context, these are going to be numerical methods. We are going to present a number of such methods, but more importantly, we are going to present a general strategy for deriving numerical differentiation methods. In this way you will not only have a number of methods available to you, but you will also be able to develop new methods, tailored to special situations that you may encounter.

The basic strategy for deriving numerical differentiation methods is to evaluate a function at a few points, find the polynomial that interpolates the function at these points, and use the derivative of this polynomial as an approximation to the derivative of the function. This technique also allows us to keep track of the so-called *truncation error*, the mathematical error committed by differentiating the polynomial instead of the function itself. In addition to the truncation error,

there are also *round-off* errors, which are unavoidable when we use floating-point numbers to perform calculations with real numbers. It turns out that numerical differentiation is very *sensitive* to round-off errors, but these errors are quite easy to analyse.

If you just read through this chapter you may be overwhelmed by all the details and inequalities. But the key is to study the first and simplest method in section 11.1 in detail. If you understand this method, you should have no problems understanding the others as well, since both the derivation and the analysis is essentially the same for all the methods. The general strategy is summarised in section 11.2. Note also that the methods for numerical integration in Chapter 12 are derived and analysed in much the same way as the differentiation methods in this chapter.

11.1 A simple method for numerical differentiation

We start by introducing the simplest method for numerical differentiation, derive its error, and its sensitivity to round-off errors. The procedure used here for deriving the method and analysing the error is used over again in later sections to derive and analyse additional methods.

Let us first clarify what we mean by numerical differentiation.

Problem 11.1 (Numerical differentiation)**.** *Let f be a given function that is only known at a number of isolated points. The problem of numerical differentiation is to compute an approximation to the derivative f' of f by suitable combinations of the known values of f.*

A typical example is that f is given by a computer program (more specifically a function, procedure or method, depending on you choice of programming language), and you can call the program with a floating-point argument x and receive back a floating-point approximation of $f(x)$. The challenge is to compute an approximation to $f'(a)$ for some real number a when the only aid we have at our disposal is the program to compute values of f.

11.1.1 The basic idea

Since we are going to compute derivatives, we must be clear about how they are defined. Recall that $f'(a)$ is defined by

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}. \tag{11.1}$$

In the following we will assume that this limit exists; i.e., that f is differentiable at $x = a$. From (11.1) we immediately have a natural approximation to $f'(a)$; we simply pick a positive h and use the approximation

$$f'(a) \approx \frac{f(a+h) - f(a)}{h}. \tag{11.2}$$

Note that this corresponds to approximating f by the straight line p_1 that interpolates f at a and $a+h$, and then using $p_1'(a)$ as an approximation to $f'(a)$.

Observation 11.2. *The derivative of f at a can be approximated by*

$$f'(a) \approx \frac{f(a+h) - f(a)}{h}.$$

In a practical situation, the number a would be given, and we would have to locate the two nearest values a_1 and a_2 to the left and right of a such that $f(a_1)$ and $f(a_2)$ are known. Then we would use the approximation

$$f'(a) \approx \frac{f(a_2) - f(a_1)}{a_2 - a_1}.$$

In later sections, we will derive several formulas like (11.2). Which formula to use in a particular situation, and exactly how to apply it, will have to be decided in each case.

Example 11.3. Let us test the approximation (11.2) for the function $f(x) = \sin x$ at $a = 0.5$ (using 64-bit floating-point numbers). In this case we know that the exact derivative is $f'(x) = \cos x$ so $f'(a) = 0.87758256$. This makes it is easy to check the accuracy of the numerical method. We try with a few values of h and find

h	$\big(f(a+h) - f(a)\big)/h$	$E_1(f;a,h)$
10^{-1}	0.8521693479	2.5×10^{-2}
10^{-2}	0.8751708279	2.4×10^{-3}
10^{-3}	0.8773427029	2.4×10^{-4}
10^{-4}	0.8775585892	2.4×10^{-5}
10^{-5}	0.8775801647	2.4×10^{-6}
10^{-6}	0.8775823222	2.4×10^{-7}

where $E_1(f;a,h) = f'(a) - \big(f(a+h) - f(a)\big)/h$. In other words, the approximation seems to improve with decreasing h, as expected. More precisely, when h is reduced by a factor of 10, the error is reduced by the same factor. ∎

11.1.2 The truncation error

Whenever we use approximations, it is important to try and keep track of the error, if at all possible. To analyse the error in numerical differentiation, Taylor polynomials with remainders are useful. To analyse the error in the approximation above, we do a Taylor expansion of $f(a+h)$. We have

$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2}f''(\xi_h),$$

where ξ_h lies in the interval $(a, a+h)$. If we rearrange this formula, we obtain

$$f'(a) - \frac{f(a+h)-f(a)}{h} = -\frac{h}{2}f''(\xi_h). \tag{11.3}$$

This is often referred to as the *truncation error* of the approximation, and is a reasonable error formula, but it would be nice to get rid of ξ_h. We first take absolute values in (11.3),

$$\left| f'(a) - \frac{f(a+h)-f(a)}{h} \right| = \frac{h}{2}\left|f''(\xi_h)\right|.$$

Recall from the Extreme value theorem that if a function is continuous, then its maximum always exists on any closed and bounded interval. In our setting here, it is natural to let the closed and bounded interval be $[a, a+h]$. This leads to the following lemma.

Lemma 11.4. *Suppose f has continuous derivatives up to order two near a. If the derivative $f'(a)$ is approximated by*

$$\frac{f(a+h)-f(a)}{h},$$

then the truncation error *is bounded by*

$$E(f;a,h) = \left| f'(a) - \frac{f(a+h)-f(a)}{h} \right| \leq \frac{h}{2} \max_{x\in[a,a+h]} \left|f''(x)\right|. \tag{11.4}$$

Let us check that the error formula (11.3) agrees with the numerical values in example 11.3. We have $f''(x) = -\sin x$, so the absolute value of the right-hand side in (11.3) becomes

$$E(\sin; 0.5, h) = \frac{h}{2}\sin\xi_h,$$

where $\xi_h \in (0.5, 0.5+h)$. For $h = 0.1$ we therefore have that the error must lie in the interval

$$[0.05\sin 0.5,\, 0.05\sin 0.6] = [2.397\times 10^{-2},\, 2.823\times 10^{-2}],$$

and the right end of the interval is the maximum value of the right-hand side in (11.4). When h is reduced by a factor of 10, the number $h/2$ is reduced by the same factor, while ξ_h is restricted to an interval whose width is also reduced by a factor of 10. This means that ξ_h will approach 0.5 so $\sin\xi_h$ will approach the lower value $\sin 0.5 \approx 0.479426$. For $h = 10^{-n}$, the error will therefore tend to

$$\frac{10^{-n}}{2}\sin 0.5 \approx \frac{0.2397}{10^n},$$

which is in complete agreement with example 11.3.

This is true in general. If f'' is continuous, then ξ_h will approach a when h goes to zero. But even when $h > 0$, the error in using the approximation $f''(\xi_h) \approx f''(a)$ is usually acceptable. This is the case since it is usually only necessary to know the magnitude of the error, i.e., it is sufficient to know the error with one or two correct digits.

Observation 11.5. *The truncation error is given approximately by*

$$\left| f'(a) - \frac{f(a+h)-f(a)}{h} \right| \approx \frac{h}{2}\left|f''(a)\right|.$$

11.1.3 The round-off error

So far, we have just considered the mathematical error committed when $f'(a)$ is approximated by $\big(f(a+h)-f(a)\big)/h$. But what about the round-off error? In fact, when we compute this approximation we have to perform the one critical operation $f(a+h)-f(a)$ — subtraction of two almost equal numbers — which we know from chapter 5 may lead to large round-off errors. Let us continue example 11.3 and see what happens if we use smaller values of h.

Example 11.6. Recall that we estimated the derivative of $f(x) = \sin x$ at $a = 0.5$ and that the correct value with ten digits is $f'(0.5) \approx 0.8775825619$. If we check

values of h from 10^{-7} and smaller we find

h	$\big(f(a+h)-f(a)\big)/h$	$E(f;a,h)$
10^{-7}	0. 8775825372	2.5×10^{-8}
10^{-8}	0.8775825622	-2.9×10^{-10}
10^{-9}	0.8775825622	-2.9×10^{-10}
10^{-11}	0.8775813409	1.2×10^{-6}
10^{-14}	0.8770761895	5.1×10^{-4}
10^{-15}	0.8881784197	-1.1×10^{-2}
10^{-16}	1.110223025	-2.3×10^{-1}
10^{-17}	0.000000000	8.8×10^{-1}

This shows very clearly that something quite dramatic happens. Ultimately, when we come to $h = 10^{-17}$, the derivative is computed as zero. ∎

If $\overline{f(a)}$ is the floating-point number closest to $f(a)$, we know from lemma 5.6 that the relative error ϵ in this approximation will be bounded by 5×2^{-53} since floating-point numbers are represented in binary ($\beta = 2$) with 53 bits for the significand ($m = 53$). We therefore have $|\epsilon| \le 5\times2^{-53} \approx 6\times10^{-16}$. In practice, the real upper bound on ϵ is usually smaller, and in the following we will denote this upper bound by ϵ^*. This means that a definite upper bound on ϵ^* is 6×10^{-16}.

Notation 11.7. *The maximum relative error when a real number is represented by a floating-point number is denoted by ϵ^*.*

There is a handy way to express the relative error in $f(a)$. If we denote the computed value of $f(a)$ by $\overline{f(a)}$, we will have

$$\overline{f(a)} = f(a)(1+\epsilon)$$

which corresponds to the relative error being $|\epsilon|$.

Observation 11.8. *Suppose that $f(a)$ is computed with 64-bit floating-point numbers and that no underflow or overflow occurs. Then the computed value $\overline{f(a)}$ satisfies*

$$\overline{f(a)} = f(a)(1+\epsilon) \tag{11.5}$$

where $|\epsilon| \le \epsilon^*$, and ϵ depends on both a and f.

The computation of $f(a+h)$ is of course also affected by round-off error, so we have

$$\overline{f(a)} = f(a)(1+\epsilon_1), \quad \overline{f(a+h)} = f(a+h)(1+\epsilon_2) \tag{11.6}$$

where $|\epsilon_i| \leq \epsilon^*$ for $i = 1, 2$. Here we should really write $\epsilon_2 = \epsilon_2(h)$, because the exact round-off error in $\overline{f(a+h)}$ will inevitably depend on h in a rather random way.

The next step is to see how these errors affect the computed approximation of $f'(a)$. Recall from example 5.11 that the main source of round-off in subtraction is the replacement of the numbers to be subtracted by the nearest floating-point numbers. We therefore consider the computed approximation to be

$$\frac{\overline{f(a+h)} - \overline{f(a)}}{h}.$$

If we insert the expressions (11.6), and also make use of lemma 11.4, we obtain

$$\begin{aligned} \frac{\overline{f(a+h)} - \overline{f(a)}}{h} &= \frac{f(a+h) - f(a)}{h} + \frac{f(a+h)\epsilon_2 - f(a)\epsilon_1}{h} \\ &= f'(a) + \frac{h}{2} f''(\xi_h) + \frac{f(a+h)\epsilon_2 - f(a)\epsilon_1}{h} \end{aligned} \tag{11.7}$$

where $\xi_h \in (a, a+h)$. This shows that the total error in the computed approximation to the derivative consists of two parts: The truncation error that we derived in the previous section, plus the last term on the right in (11.7), which is due to the round-off in floating-point numbers. The truncation error is proportional to h and therefore tends to 0 when h tends to 0. The error due to round-off however, is proportional to $1/h$ and therefore becomes large when h tends to 0.

Below we will see that the error formula (11.7) provides a reasonable explanation of example 11.6. First we tidy up the expression a little bit and sum it up in a theorem.

Theorem 11.9. *Suppose that f and its first two derivatives are continuous near a. When the derivative of f at a is approximated by*

$$\frac{f(a+h) - f(a)}{h},$$

the error in the computed approximation is given by

$$\left| f'(a) - \frac{\overline{f(a+h)} - \overline{f(a)}}{h} \right| \leq \frac{h}{2} M_1 + \frac{2\epsilon^*}{h} M_2, \tag{11.8}$$

where

$$M_1 = \max_{x \in [a, a+h]} \left| f''(a) \right|, \quad M_2 = \max_{x \in [a, a+h]} \left| f(a) \right|.$$

Proof. To get to (11.8) we have rearranged (11.7) and used the triangle inequality. We have also replaced $\left|f''(\xi_h)\right|$ by its maximum on the interval $[a, a+h]$, as in (11.4). Similarly, we have replaced $f(a)$ and $f(a+h)$ by their common maximum on $[a, a+h]$. The last term then follows by applying the triangle inequality to the last term in (11.7) and replacing $|\epsilon_1|$ and $|\epsilon_2(h)|$ by the upper bound ϵ^*. ∎

The inequality (11.8) can be replaced by an approximate equality by making the approximations $M_1 \approx \left|f''(a)\right|$ and $M_2 \approx \left|f(a)\right|$, just like in observation 11.8 and using the maximum of ϵ_1 and ϵ_2 in (11.7), which we denote $\epsilon(h)$.

Observation 11.10. *The inequality* (11.8) *is approximately equivalent to*

$$\left| f'(a) - \frac{\overline{f(a+h)} - \overline{f(a)}}{h} \right| \approx \frac{h}{2}\left|f''(a)\right| + \frac{2\,|\epsilon(h)|}{h}\left|f(a)\right|. \tag{11.9}$$

Let us check how well observation 11.10 agrees with the computations in examples 11.3 and 11.6.

Example 11.11. For large values of h the first term on the right in (11.9) will dominate the error and we have already seen that this agrees very well with the computed values in example 11.3. The question is how well the numbers in example 11.6 can be modelled when h becomes smaller.

To estimate the size of $\epsilon(h)$, we consider the case when $h = 10^{-16}$. Then the observed error is -2.3×10^{-1} so we should have

$$\frac{10^{-16}}{2}\sin 0.5 - \frac{2\epsilon\left(10^{-16}\right)}{10^{-16}} = -2.3 \times 10^{-1}.$$

We solve this equation and find

$$\epsilon\left(10^{-16}\right) = \frac{10^{-16}}{2}\left(2.3 \times 10^{-1} + \frac{10^{-16}}{2}\sin 0.5\right) = 1.2 \times 10^{-17}.$$

If we try some other values of h we find

$$\epsilon\left(10^{-11}\right) = -6.1 \times 10^{-18}, \quad \epsilon\left(10^{-13}\right) = 2.4 \times 10^{-18}, \quad \epsilon\left(10^{-15}\right) = 5.3 \times 10^{-18}.$$

We observe that all these values are considerably smaller than the upper limit 6×10^{-16} which we mentioned above.

Figure 11.1 shows plots of the error. The numerical approximation has been computed for the values $n = 0.01i$, $i = 0, \ldots, 200$ and plotted in a log-log plot. The errors are shown as isolated dots, and the function

$$g(h) = \frac{h}{2}\sin 0.5 + \epsilon^* \frac{2}{h}\sin 0.5 \tag{11.10}$$

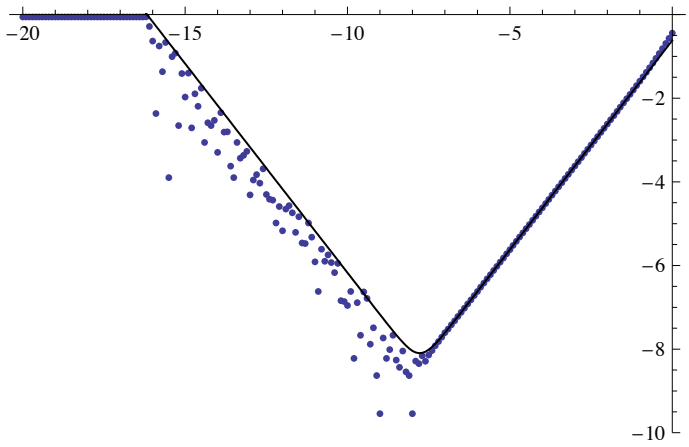

Figure 11.1. Numerical approximation of the derivative of $f(x) = \sin x$ at $x = 0.5$ using the approximation in lemma 11.4. The plot is a $\log_{10}$-$\log_{10}$ plot which shows the logarithm to base 10 of the absolute value of the total error as a function of the logarithm to base 10 of h, based on 200 values of h. The point -10 on the horizontal axis therefore corresponds $h = 10^{-10}$, and the point -6 on the vertical axis corresponds to an error of 10^{-6}. The plot also includes the function given by (11.10).

with $\epsilon^* = 7 \times 10^{-17}$ is shown as a solid graph. It seems like this choice of ϵ^* makes $g(h)$ a reasonable upper bound on the error. ∎

11.1.4 Optimal choice of h

Figure 11.1 indicates that there is an optimal value of h which minimises the total error. We can find this mathematically by minimising the upper bound in (11.9), with $|\epsilon(h)|$ replaced by the upper bound ϵ^*. This gives

$$g(h) = \frac{h}{2}\left|f''(a)\right| + \frac{2\epsilon^*}{h}\left|f(a)\right|. \tag{11.11}$$

To find the value of h which minimises this expression, we differentiate with respect to h and set the derivative to zero. We find

$$g'(h) = \frac{\left|f''(a)\right|}{2} - \frac{2\epsilon^*}{h^2}\left|f(a)\right|.$$

If we solve the equation $g(h) = 0$, we obtain the approximate optimal value.

Lemma 11.12. *Let f be a function with continuous derivatives up to order 2. If the derivative of f at a is approximated as in lemma 11.4, then the value of h which minimises the total error (truncation error + round-off error) is*

approximately

$$h^* \approx 2\frac{\sqrt{\epsilon^* \left|f(a)\right|}}{\sqrt{\left|f''(a)\right|}}.$$

It is easy to see that the optimal value of h is the value that balances the two terms in (11.11)l, i.e., the truncation error and the round-off error are equal. In the example with $f(x) = \sin x$ and $a = 0.5$ we can use $\epsilon^* = 7 \times 10^{-17}$ which gives

$$h^* = 2\sqrt{\epsilon} = 2\sqrt{7 \times 10^{-17}} \approx 1.7 \times 10^{-8}.$$

11.2 Summary of the general strategy

Before we continue, let us sum up the derivaton and analysis of the numerical differentiation method in section 11.1, since we will use this over and over again.

The first step was to derive the numerical method. In section 11.1 this was very simple since the method came straight out of the definition of the derivative. Just before observation 11.2 we indicated that the method can also be derived by approximating f by a polynomial p and using $p'(a)$ as an approximation to $f'(a)$. This is the general approach that we will use below.

Once the numerical method is known, we estimate the mathematical error in the approximation, *the truncation error*. This we do by performing Taylor expansions with remainders. For numerical differentiation methods which provide estimates of a derivative at a point a, we replace all function values at points other than a by Taylor polynomials with remainders. There may be a challenge to choose the degree of the Taylor polynomial.

The next task is to estimate the total error, including round-off error. We consider the difference between the derivative to be computed and the computed approximation, and replace the computed function evaluations by expressions like the ones in observation 11.8. This will result in an expression involving the mathematical approximation to the derivative. This can be simplified in the same way as when the truncation error was estimated, with the addition of an expression involving the relative round-off errors in the function evaluations. These expressions can then be simplified to something like (11.8) or (11.9).

As a final step, the optimal value of h can be found by minimising the total error.

Procedure 11.13. *The following is a general procedure for deriving numerical methods for differentiation:*

1. *Interpolate the function f by a polynomial p at suitable points.*
2. *Approximate the derivative of f by the derivative of p. This makes it possible to express the approximation in terms of function values of f.*
3. *Derive an estimate for the error by expanding the function values (other than the one at a) in Taylor series with remainders.*
4. *Derive an estimate of the round-off error by assuming that the relative errors in the function values are bounded by ϵ^*. By minimising the total error, an optimal step length h can be determined.*

11.3 A simple, symmetric method

The numerical differentiation method in section 11.1 is not symmetric about a, so let us try and derive a symmetric method.

11.3.1 Construction of the method

We want to find an approximation to $f'(a)$ using values of f near a. To obtain a symmetric method, we assume that $f(a-h)$, $f(a)$, and $f(a+h)$ are known values, and we want to find an approximation to $f'(a)$ using these values. The strategy is to determine the quadratic polynomial p_2 that interpolates f at $a-h$, a and $a+h$, and then we use $p_2'(a)$ as an approximation to $f'(a)$.

We write p_2 in Newton form,

$$p_2(x) = f[a-h] + f[a-h,a](x-(a-h)) + f[a-h,a,a+h](x-(a-h))(x-a). \quad (11.12)$$

We differentiate and find

$$p_2'(x) = f[a-h,a] + f[a-h,a,a+h](2x-2a+h).$$

Setting $x = a$ yields

$$p_2'(a) = f[a-h,a] + f[a-h,a,a+h]h.$$

To get a practically useful formula we must express the divided differences in terms of function values. If we expand the second expression we obtain

$$p_2'(a) = f[a-h,a] + \frac{f[a,a+h]-f[a-h,a]}{2h}h = \frac{f[a,a+h]+f[a-h,a]}{2} \quad (11.13)$$

The two first order differences are

$$f[a-h,a] = \frac{f(a)-f(a-h)}{h}, \quad f[a,a+h] = \frac{f(a+h)-f(a)}{h},$$

and if we insert this in (11.13) we end up with

$$p_2'(a) = \frac{f(a+h)-f(a-h)}{2h}.$$

Lemma 11.14. *Let f be a given function, and let a and h be given numbers. If $f(a-h)$, $f(a)$, $f(a+h)$ are known values, then $f'(a)$ can be approximated by $p_2'(a)$ where p_2 is the quadratic polynomial that interpolates f at $a-h$, a, and $a+h$. The approximation is given by*

$$f'(a) \approx p_2'(a) = \frac{f(a+h)-f(a-h)}{2h}. \tag{11.14}$$

Let us test this approximation on the function $f(x) = \sin x$ at $a = 0.5$ so we can compare with the method in section 11.1.

Example 11.15. We test the approximation (11.14) with the same values of h as in examples 11.3 and 11.6. Recall that $f'(0.5) \approx 0.8775825619$ with ten correct decimals. The results are

h	$\big(f(a+h)-f(a-h)\big)/(2h)$	$E(f;a,h)$
10^{-1}	0.8761206554	1.5×10^{-3}
10^{-2}	0.8775679356	1.5×10^{-5}
10^{-3}	0.8775824156	1.5×10^{-7}
10^{-4}	0.8775825604	1.5×10^{-9}
10^{-5}	0.8775825619	1.8×10^{-11}
10^{-6}	0.8775825619	-7.5×10^{-12}
10^{-7}	0.8775825616	2.7×10^{-10}
10^{-8}	0.8775825622	-2.9×10^{-10}
10^{-11}	0.8775813409	1.2×10^{-6}
10^{-13}	0.8776313010	-4.9×10^{-5}
10^{-15}	0.8881784197	-1.1×10^{-2}
10^{-17}	0.0000000000	8.8×10^{-1}

If we compare with examples 11.3 and 11.6, the errors are generally smaller. In particular we note that when h is reduced by a factor of 10, the error is reduced

by a factor of 100, at least as long as h is not too small. However, when h becomes smaller than about 10^{-6}, the error becomes larger. It therefore seems like the truncation error is smaller than in the first method, but the round-off error makes it impossible to get accurate results for small values of h. The optimal value of h seems to be $h^* \approx 10^{-6}$, which is larger than for the first method, but the error is then about 10^{-12}, which is smaller than the best we could do with the first method. ∎

11.3.2 Truncation error

Let us attempt to estimate the truncation error for the method in lemma 11.14. The idea is to do replace $f(a-h)$ and $f(a+h)$ with Taylor expansions about a. We use the Taylor expansions

$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2}f''(a) + \frac{h^3}{6}f'''(\xi_1),$$
$$f(a-h) = f(a) - hf'(a) + \frac{h^2}{2}f''(a) - \frac{h^3}{6}f'''(\xi_2),$$

where $\xi_1 \in (a, a+h)$ and $\xi_2 \in (a-h, a)$. If we subtract the second formula from the first and divide by $2h$, we obtain

$$\frac{f(a+h) - f(a-h)}{2h} = f'(a) + \frac{h^2}{12}\big(f'''(\xi_1) + f'''(\xi_2)\big). \tag{11.15}$$

This leads to the following lemma.

Lemma 11.16. *Suppose that f and its first three derivatives are continuous near a, and suppose we approximate $f'(a)$ by*

$$\frac{f(a+h) - f(a-h)}{2h}. \tag{11.16}$$

The truncation error in this approximation is bounded by

$$\big|E_2(f;a,h)\big| = \left|f'(a) - \frac{f(a+h)-f(a-h)}{2h}\right| \leq \frac{h^2}{6}\max_{x\in[a-h,a+h]}\big|f'''(x)\big|. \tag{11.17}$$

Proof. What remains is to simplify the last term in (11.15) to the term on the

right in (11.17). This follows from

$$\begin{aligned}\left|f'''(\xi_1)+f'''(\xi_2)\right| &\le \max_{x\in[a,a+h]}\left|f'''(x)\right| + \max_{x\in[a-h,a]}\left|f'''(x)\right| \\ &\le \max_{x\in[a-h,a+h]}\left|f'''(x)\right| + \max_{x\in[a-h,a+h]}\left|f'''(x)\right| \\ &= 2\max_{x\in[a-h,a+h]}\left|f'''(x)\right|.\end{aligned}$$

The last inequality is true because the width of the intervals over which we take the maximums are increased, so the maximum values may also increase. ∎

The error formula (11.17) confirms the numerical behaviour we saw in example 11.15 for small values of h since the error is proportional to h^2: When h is reduced by a factor of 10, the error is reduced by a factor 10^2.

11.3.3 Round-off error

The round-off error may be estimated just like for the first method. When the approximation (11.16) is computed, the values $f(a-h)$ and $f(a+h)$ are replaced by the nearest floating point numbers $\overline{f(a-h)}$ and $\overline{f(a+h)}$ which can be expressed as

$$\overline{f(a+h)} = f(a+h)(1+\epsilon_1), \quad \overline{f(a-h)} = f(a-h)(1+\epsilon_2),$$

where both ϵ_1 and ϵ_2 depend on h and satisfy $|\epsilon_i| \le \epsilon^*$ for $i = 1, 2$. Using these expressions we obtain

$$\frac{\overline{f(a+h)}-\overline{f(a-h)}}{2h} = \frac{f(a+h)-f(a-h)}{2h} + \frac{f(a+h)\epsilon_1 - f(a-h)\epsilon_2}{2h}.$$

We insert (11.15) and get the relation

$$\frac{\overline{f(a+h)}-\overline{f(a-h)}}{2h} = f'(a) + \frac{h^2}{12}\left(f'''(\xi_1)+f'''(\xi_2)\right) + \frac{f(a+h)\epsilon_1 - f(a-h)\epsilon_2}{2h}.$$

This leads to an estimate of the total error if we use the same technique as in the proof of lemma 11.8.

Theorem 11.17. *Let f be a given function with continuous derivatives up to order three, and let a and h be given numbers. Then the error in the approximation*

$$f'(a) \approx \frac{f(a+h)-f(a-h)}{2h},$$

including round-off error and truncation error, is bounded by

$$\left| f'(a) - \frac{\overline{f(a+h)} - \overline{f(a-h)}}{2h} \right| \leq \frac{h^2}{6} M_1 + \frac{\epsilon^*}{h} M_2 \tag{11.18}$$

where

$$M_1 = \max_{x \in [a-h, a+h]} \left| f'''(x) \right|, \quad M_2 = \max_{x \in [a-h, a+h]} \left| f(x) \right|. \tag{11.19}$$

In practice, the interesting values of h will usually be so small that there is very little error in making the approximations

$$M_1 = \max_{x \in [a-h, a+h]} \left| f'''(x) \right| \approx \left| f'''(a) \right|, \quad M_2 = \max_{x \in [a-h, a+h]} \left| f(x) \right| \approx \left| f(a) \right|.$$

If we make this simplification in (11.18) we obtain a slightly simpler error estimate.

Observation 11.18. *The error* (11.18) *is approximately bounded by*

$$\left| f'(a) - \frac{\overline{f(a+h)} - \overline{f(a-h)}}{2h} \right| \lesssim \frac{h^2}{6} \left| f'''(a) \right| + \frac{\epsilon^* \left| f(a) \right|}{h}. \tag{11.20}$$

A plot of how the error behaves in this approximation, together with the estimate of the error on the right in (11.20), is shown in figure 11.2.

11.3.4 Optimal choice of h

As for the first numerical differentiation method, we can find an optimal value of h which minimises the error. The error is minimised when the truncation error and the round-off error have the same magnitude. We can find this value of h if we differentiate the right-hand side of (11.18) with respect to h and set the derivative to 0. This leads to the equation

$$\frac{h}{3} M_1 - \frac{\epsilon^*}{h^2} M_2 = 0$$

which has the solution

$$h^* = \frac{\sqrt[3]{3\epsilon^* M_2}}{\sqrt[3]{M_1}} \approx \frac{\sqrt[3]{3\epsilon^* \left| f(a) \right|}}{\sqrt[3]{\left| f'''(a) \right|}}.$$

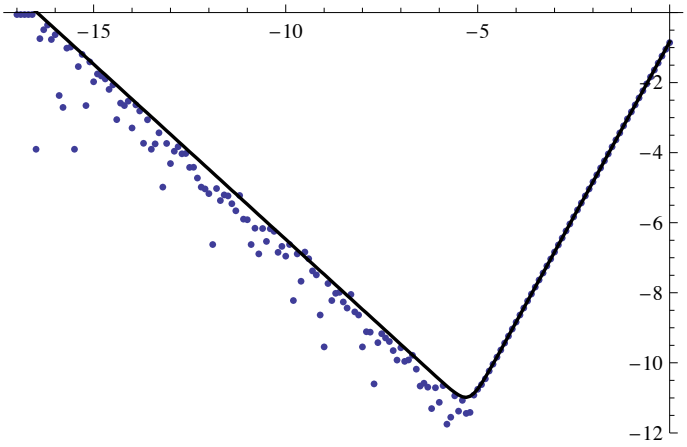

Figure 11.2. Log-log plot of the error in the approximation to the derivative of $f(x) = \sin x$ at $x = 1/2$ for values of h in the interval $[0, 10^{-17}]$, using the method in theorem 11.17. The function plotted is the right-hand side of (11.20) with $\epsilon^* = 7 \times 10^{-17}$, as a function of h.

At the end of section 11.1.4 we saw that a reasonable value for ϵ^* was $\epsilon^* = 7 \times 10^{-17}$. The optimal value of h in example 11.15, where $f(x) = \sin x$ and $a = 1/2$, then becomes $h = 4.6 \times 10^{-6}$. For this value of h the approximation is $f'(0.5) \approx 0.877582561887$ with error 3.1×10^{-12}.

11.4 A four-point method for differentiation

In a way, the two methods for numerical differentiation that we have considered so far are the same. If we use a step length of $2h$ in the first method, the approximation becomes

$$f'(a) \approx \frac{f(a+2h) - f(a)}{2h}.$$

The analysis of the symmetric method shows that the approximation is considerably better if we associate the approximation with the midpoint between a and $a+h$,

$$f'(a+h) \approx \frac{f(a+2h) - f(a)}{2h}.$$

At the point $a+h$ the approximation is proportional to h^2 rather than h, and this makes a big difference as to how quickly the error goes to zero, as is evident if we compare examples 11.3 and 11.15. In this section we derive another method for which the truncation error is proportional to h^4.

The computations below may seem overwhelming, and have in fact been done with the help of a computer to save time and reduce the risk of miscal-

culations. The method is included here just to illustrate that the principle for deriving both the method and the error terms is just the same as for the simple symmetric method in the previous section. To save space we have only included one highlighted box, where both the approximation method and the total error are given.

11.4.1 Derivation of the method

We want better accuracy than the symmetric method which was based on interpolation with a quadratic polynomial. It is therefore natural to base the approximation on a cubic polynomial, which can interpolate four points. We have seen the advantage of symmetry, so we choose the interpolation points $x_0 = a-2h$, $x_1 = a-h$, $x_2 = a+h$, and $x_3 = a+2h$. The cubic polynomial that interpolates f at these points is

$$p_3(x) = f(x_0) + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) \\ + f[x_0,x_1,x_2,x_3](x-x_0)(x-x_1)(x-x_2).$$

and its derivative is

$$p_3'(x) = f[x_0,x_1] + f[x_0,x_1,x_2](2x-x_0-x_1) \\ + f[x_0,x_1,x_2,x_3]\big((x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\big).$$

If we evaluate this expression at $x = a$ and simplify (this is quite a bit of work), we find that the resulting approximation of $f'(a)$ is

$$f'(a) \approx p_3'(a) = \frac{f(a-2h) - 8f(a-h) + 8f(a+h) - f(a+2h)}{12h}. \tag{11.21}$$

11.4.2 Truncation error

To estimate the error, we expand the four terms in the numerator in (11.21) in Taylor series,

$$f(a-2h) = f(a) - 2hf'(a) + 2h^2f''(a) - \frac{4h^3}{3}f'''(a) + \frac{2h^4}{3}f^{(iv)}(a) - \frac{4h^5}{15}f^{(v)}(\xi_1),$$
$$f(a-h) = f(a) - hf'(a) + \frac{h^2}{2}f''(a) - \frac{h^3}{6}f'''(a) + \frac{h^4}{24}f^{(iv)}(a) - \frac{h^5}{120}f^{(v)}(\xi_2),$$
$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2}f''(a) + \frac{h^3}{6}f'''(a) + \frac{h^4}{24}f^{(iv)}(a) + \frac{h^5}{120}f^{(v)}(\xi_3),$$
$$f(a+2h) = f(a) + 2hf'(a) + 2h^2f''(a) + \frac{4h^3}{3}f'''(a) + \frac{2h^4}{3}f^{(iv)}(a) + \frac{4h^5}{15}f^{(v)}(\xi_4),$$

where $\xi_1 \in (a-2h, a)$, $\xi_2 \in (a-h, a)$, $\xi_3 \in (a, a+h)$, and $\xi_4 \in (a, a+2h)$. If we insert this into the formula for $p_3'(a)$ we obtain

$$\frac{f(a-2h)-8f(a-h)+8f(a+h)-f(a+2h)}{12h} = f'(a) - \frac{h^4}{45}f^{(v)}(\xi_1) + \frac{h^4}{180}f^{(v)}(\xi_2) + \frac{h^4}{180}f^{(v)}(\xi_3) - \frac{h^4}{45}f^{(v)}(\xi_4).$$

If we use the same trick as for the symmetric method, we can combine all last four terms in and obtain an upper bound on the truncation error. The result is

$$\left| f'(a) - \frac{f(a-2h)-8f(a-h)+8f(a+h)-f(a+2h)}{12h} \right| \leq \frac{h^4}{18}M \tag{11.22}$$

where

$$M = \max_{x\in[a-2h,a+2h]} \left| f^{(v)}(x) \right|.$$

11.4.3 Round-off error

The truncation error is derived in the same way as before. The quantities we actually compute are

$$\overline{f(a-2h)} = f(a-2h)(1+\epsilon_1), \quad \overline{f(a+2h)} = f(a+2h)(1+\epsilon_3),$$
$$\overline{f(a-h)} = f(a-h)(1+\epsilon_2), \quad \overline{f(a+h)} = f(a+h)(1+\epsilon_4).$$

We estimate the difference between $f'(a)$ and the computed approximation, make use of the estimate (11.22), combine the function values that are multiplied by ϵs, and approximate the maximum values by function values at a. We sum up the result.

Observation 11.19. *Suppose that f and its first five derivatives are continuous. If $f'(a)$ is approximated by*

$$f'(a) \approx \frac{f(a-2h)-8f(a-h)+8f(a+h)-f(a+2h)}{12h},$$

the total error is approximately bounded by

$$\left| f'(a) - \frac{\overline{f(a-2h)}-8\overline{f(a-h)}+8\overline{f(a+h)}-\overline{f(a+2h)}}{12h} \right| \lesssim \frac{h^4}{18}\left|f^{(v)}(a)\right| + \frac{3\epsilon^*}{h}\left|f(a)\right|. \tag{11.23}$$

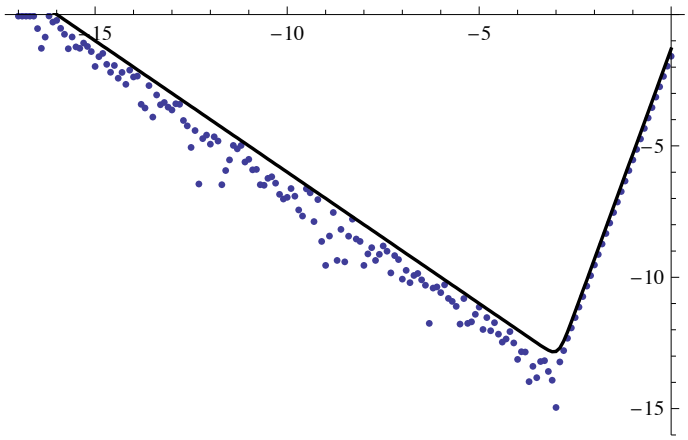

Figure 11.3. Log-log plot of the error in the approximation to the derivative of $f(x) = \sin x$ at $x = 1/2$, using the method in observation 11.19, with h in the interval $[0, 10^{-17}]$. The function plotted is the right-hand side of (11.23) with $\epsilon^* = 7 \times 10^{-17}$.

A plot of the error in the approximation for the $\sin x$ example is shown in figure 11.3.

11.4.4 Optimal value of h

From observation 11.19 we can compute the optimal value of h by differentiating the right-hand side with respect to h and setting it to zero,

$$\frac{2h^3}{9}\left|f^{(v)}(a)\right| - \frac{3\epsilon^*}{h^2}\left|f(a)\right| = 0$$

which has the solution

$$h^* = \frac{\sqrt[5]{27\epsilon^*\left|f(a)\right|}}{\sqrt[5]{2\left|f^{(v)}(a)\right|}}.$$

For the case above with $f(x) = \sin x$ and $a = 0.5$ the solution is $h^* \approx 8.8 \times 10^{-4}$. For this value of h the actual error is 10^{-14}.

11.5 Numerical approximation of the second derivative

We consider one more method for numerical approximation of derivatives, this time of the second derivative. The approach is the same: We approximate f by a polynomial and approximate the second derivative of f by the second derivative of the polynomial. As in the other cases, the error analysis is based on expansion in Taylor series.

11.5.1 Derivation of the method

Since we are going to find an approximation to the second derivative, we have to approximate f by a polynomial of degree at least two, otherwise the second derivative is identically 0. The simplest is therefore to use a quadratic polynomial, and for symmetry we want it to interpolate f at $a-h$, a, and $a+h$. The resulting polynomial p_2 is the one we used in section 11.3 and it is given in equation (11.12). The second derivative of p_2 is constant, and the approximation of $f'(a)$ is

$$f''(a) \approx p_2''(a) = f[a-h, a, a+h].$$

The divided difference is easy to expand.

Lemma 11.20. *The second derivative of a function f at a can be approximated by*

$$f''(a) \approx \frac{f(a+h) - 2f(a) + f(a-h)}{h^2}. \tag{11.24}$$

11.5.2 The truncation error

Estimation of the error goes as in the other cases. The Taylor series of $f(a-h)$ and $f(a+h)$ are

$$f(a-h) = f(a) - hf'(a) + \frac{h^2}{2}f''(a) - \frac{h^3}{6}f'''(a) + \frac{h^4}{24}f^{(iv)}(\xi_1),$$
$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2}f''(a) + \frac{h^3}{6}f'''(a) + \frac{h^4}{24}f^{(iv)}(\xi_2),$$

where $\xi_1 \in (a-h, a)$ and $\xi_2 \in (a, a+h)$. If we insert these Taylor series in (11.24) we obtain

$$\frac{f(a+h) - 2f(a) + f(a-h)}{h^2} = f''(a) + \frac{h^2}{24}\big(f^{(iv)}(\xi_1) + f^{(iv)}(\xi_2)\big).$$

From this we obtain an expression for the truncation error.

Lemma 11.21. *Suppose f and its first three derivatives are continuous near a. If the second derivative $f''(a)$ is approximated by*

$$f''(a) \approx \frac{f(a+h) - 2f(a) + f(a-h)}{h^2},$$

the error is bounded by

$$\left| f''(a) - \frac{f(a+h) - 2f(a) + f(a-h)}{h^2} \right| \leq \frac{h^2}{12} \max_{x \in [a-h, a+h]} \left| f'''(x) \right|. \tag{11.25}$$

11.5.3 Round-off error

The round-off error can also be estimated as before. Instead of computing the exact values, we compute $\overline{f(a-h)}$, $\overline{f(a)}$, and $\overline{f(a+h)}$, which are linked to the exact values by

$$\overline{f(a-h)} = f(a-h)(1+\epsilon_1), \quad \overline{f(a)} = f(a)(1+\epsilon_2), \quad \overline{f(a+h)} = f(a+h)(1+\epsilon_3),$$

where $|\epsilon_i| \leq \epsilon^*$ for $i = 1, 2, 3$. The difference between $f''(a)$ and the computed approximation is therefore

$$\begin{aligned} f''(a) - \frac{\overline{f(a+h)} - 2\overline{f(a)} + \overline{f(a-h)}}{h^2} \\ = -\frac{h^2}{24}\left(f''''(\xi_1) + f''''(\xi_2)\right) - \frac{\epsilon_1 f(a-h) - \epsilon_2 f(a) + \epsilon_3 f(a+h)}{h^2}. \end{aligned}$$

If we combine terms on the right as before, we end up with the following theorem.

Theorem 11.22. *Suppose f and its first three derivatives are continuous near a, and that $f''(a)$ is approximated by*

$$f''(a) \approx \frac{f(a+h) - 2f(a) + f(a-h)}{h^2}.$$

Then the total error (truncation error + round-off error) in the computed approximation is bounded by

$$\left| f''(a) - \frac{\overline{f(a+h)} - 2\overline{f(a)} + \overline{f(a-h)}}{h^2} \right| \leq \frac{h^2}{12} M_1 + \frac{3\epsilon^*}{h^2} M_2. \tag{11.26}$$

where

$$M_1 = \max_{x \in [a-h, a+h]} \left| f^{(iv)}(x) \right|, \quad M_2 = \max_{x \in [a-h, a+h]} \left| f(x) \right|.$$

As before, we can simplify the right-hand side to

$$\frac{h^2}{12} \left| f^{(iv)}(a) \right| + \frac{3\epsilon^*}{h^2} \left| f(a) \right| \tag{11.27}$$

if we can tolerate a slightly approximate upper bound.

Figure 11.4 shows the errors in the approximation to the second derivative given in theorem 11.22 when $f(x) = \sin x$ and $a = 0.5$ and for h in the range

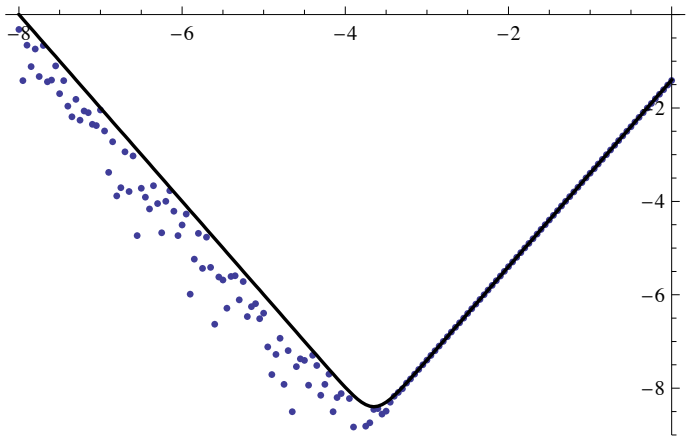

Figure 11.4. Log-log plot of the error in the approximation to the derivative of $f(x) = \sin x$ at $x = 1/2$ for h in the interval $[0, 10^{-8}]$, using the method in theorem 11.22. The function plotted is the right-hand side of (11.23) with $\epsilon^* = 7 \times 10^{-17}$.

$[0, 10^{-8}]$. The solid graph gives the function in (11.27) which describes the upper limit on the error as function of h, with $\epsilon^* = 7 \times 10^{-17}$. For h smaller than 10^{-8}, the approximation becomes 0, and the error constant. Recall that for the approximations to the first derivative, this did not happen until h was about 10^{-17}. This illustrates the fact that the higher the derivative, the more problematic is the round-off error, and the more difficult it is to approximate the derivative with numerical methods like the ones we study here.

11.5.4 Optimal value of h

Again, we find the optimal value of h by minimising the right-hand side of (11.26). To do this we find the derivative with respect to h and set it to 0,

$$\frac{h}{6} M_1 - \frac{6\epsilon^*}{h^3} M_2 = 0.$$

As usual it does not make much difference if we use the approximations $M_1 \approx \left| f'''(a) \right|$ and $M_2 = \left| f(a) \right|$.

Observation 11.23. *The upper bound on the total error* (11.26) *is minimised when h has the value*

$$h^* = \frac{\sqrt[4]{36\epsilon^* \left| f(a) \right|}}{\sqrt[4]{\left| f^{(iv)}(a) \right|}}.$$

When $f(x) = \sin x$ and $a = 0.5$ this gives $h^* = 2.2 \times 10^{-4}$ if we use the value $\epsilon^* = 7 \times 10^{-17}$. Then the approximation to $f''(a) = -\sin a$ is -0.4794255352 with an actual error of 3.4×10^{-9}.

11.6 Summary

In this chapter we have derived three methods for numerical differentiation. All these methods and their error analyses may seem rather overwhelming, but they all follow the general recipe in procedure 12.15. Perhaps the most delicate part of the procedure is to choose the degree of the Taylor polynomials. This is discussed in exercise 4.

It is procedure 12.15 that is the main content of this chapter. The individual methods are important in practice, but also serve as examples of how this procedure is implemented, and should show you how to derive other methods more suitable for your specific needs.

Exercises

11.1 **a)** Write a program that implements the numerical differentiation method

$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h},$$

and test the method on the function $f(x) = e^x$ at $a = 1$.

b) Determine the optimal value of h given in section 11.3.4 which minimises the total error. Use $\epsilon^* = 7 \times 10^{-17}$.

c) Use your program to determine the optimal value h of experimentally.

d) Use the optimal value of h that you found in (c) to determine a better value for ϵ^* in this specific example.

11.2 Repeat exercise 1, but compute the second derivative using the approximation

$$f''(a) \approx \frac{f(a+h) - 2f(a) + f(a-h)}{h^2}.$$

In (b) you should use the value of h given in observation 11.23.

11.3 **a)** Suppose that we want to derive a method for approximating the derivative of f at a which has the form

$$f'(a) \approx c_1 f(a-h) + c_2 f(a+h), \quad c_1, c_2 \in \mathbb{R}.$$

We want the method to be exact when $f(x) = 1$ and $f(x) = x$. Use these conditions to determine c_1 and c_2.

b) Show that the method in (a) is exact for all polynomials of degree 1, and compare it to the methods we have discussed in this chapter.

c) Use the procedure in (a) and (b) to derive a method for approximating the second derivative of f,

$$f''(a) \approx c_1 f(a-h) + c_2 f(a) + c_3 f(a+h), \quad c_1, c_2, c_3 \in \mathbb{R},$$

by requiring that the method should be exact when $f(x) = 1$, x and x^2.

d) Show that the method in (c) is exact for all quadratic polynomials.

11.4 It may sometimes be difficult to judge how many terms to include in the Taylor series used in the analysis of numerical methods. In this exercise we are going to see how this can be done. We use the numerical approximation

$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h}$$

in section 11.3 for our experiments.

a) Do the same derivation as section 11.3.2, but include only two terms in the Taylor series (plus remainder). What happens?

b) Do the same derivation as section 11.3.2, but include four terms in the Taylor series (plus remainder). What happens now?