Chapter 14: Numerical Differentiation

Uri M. Ascher and Chen Greif
Department of Computer Science
The University of British Columbia
{ascher,greif}@cs.ubc.ca

Slides for the book
A First Course in Numerical Methods (published by SIAM, 2011)
http://www.ec-securehost.com/SIAM/CS07.html

Why Numerical Differentiation?

- Simply for mathematical modeling.
- Obtaining derivatives of complicated functions by hand may be prone to errors.
- The function may not be known explicitly.

Why numerical differentiation?

- Numerical differentiation is a major tool in deriving methods for differential equations (see Chapter 16).
- Approximating derivatives is ubiquitous in continuous optimization and nonlinear equations (see Chapters 3 and 9).
- The need to estimate derivatives from discrete data often arises in applications.

What is numerical differentiation

- Given a function $f(x)$ that is differentiable in the vicinity of a point x_0, it is often necessary to estimate the derivative $f'(x)$ and higher derivatives using nearby values of f.
- Example 1.2 in Chapter 1 provides a simple instance of numerical differentiation. Here we consider the more complete picture. For instance, we ask
  - how to achieve more, higher order difference formulas in an easy and orderly fashion?
  - how to control or altogether avoid the strong cancellation error effect demonstrated in Example 1.3?

These and several other questions are considered here.

Outline

- Deriving formulas using Taylor series
- Richardson extrapolation
- Deriving formulas using polynomial interpolation
- Roundoff and data errors
- ~~*Differentiation matrices~~

*advanced

Deriving Formulas Using Taylor Series

From calculus, the derivative of a function f at x_0 is defined by

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0+h) - f(x_0)}{h}.$$

This uses values of f near the point of differentiation, x_0. To obtain a formula approximating a first derivative, we therefore typically choose equally spaced points nearby (e.g., $x_0 - h, x_0, x_0 + h, \ldots$, for a small, positive h) and construct an approximation from values of the function $f(x)$ at these points.

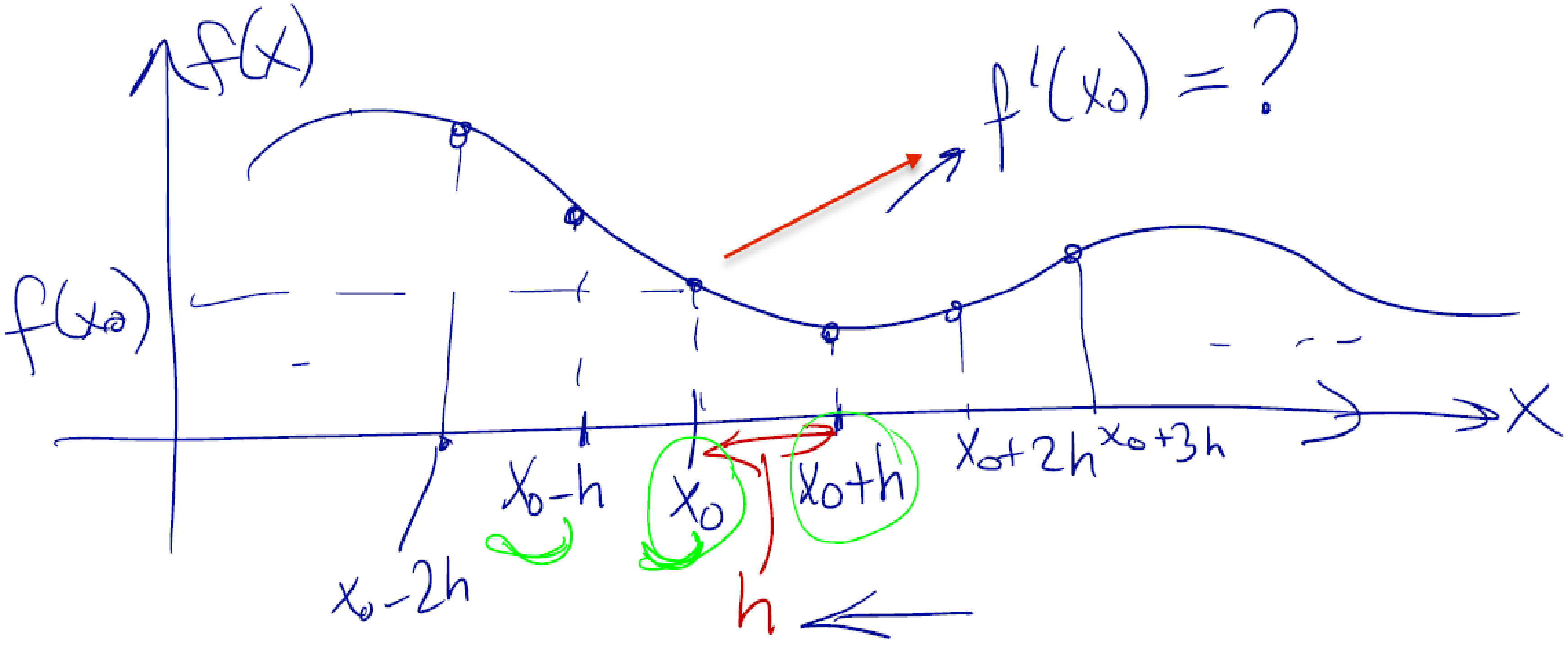
Deriving Formulas Using Taylor Series

The resulting discretization error, also called **truncation error**, shrinks as h shrinks, provided that the function $f(x)$ is *sufficiently smooth*. Throughout this section, indeed throughout this chapter and others, we obtain estimates for truncation errors which depend on some higher derivatives of $f(x)$. We will assume, unless specifically noted otherwise, that such high derivatives of f exist and are bounded: this is what we mean by "sufficiently smooth."

What determines an actual value for h depends on the specific application.

In the context of deriving formulas for use in the numerical solution of differential equations, an interval $[a,b]$ of a fixed size has to be traversed. This requires $(b-a)/h$ such h-steps, so the smaller the step size h the more computational effort is necessary. Thus, *efficiency* considerations would limit us from taking $h > 0$ to be incredibly small.

Deriving Formulas Using Taylor Series



Formula derivation

1. Two-point formulas
2. Three-point formulas
3. Five-point formulas
4. Three-point formulas for the second derivative

Deriving Formulas Using Taylor Series

- This is the most convenient, ad hoc approach.
- Start from Taylor's expansion, generally written for a small $h > 0$ as

$$
\begin{aligned}
f(x_0 \pm h) &= f(x_0) \pm h f'(x_0) + \frac{h^2}{2} f''(x_0) \pm \frac{h^3}{6} f'''(x_0) + \\
&+ \frac{h^4}{24} f^{(iv)}(x_0) \pm \frac{h^5}{120} f^{(v)}(x_0) + \frac{h^6}{720} f^{(vi)}(x_0) + \mathcal{O}(h^7).
\end{aligned}
$$

- Truncate this as needed and derive an expression for $f'(x_0)$.
- Simplest example is the forward difference of Example 1.2. Likewise, backward difference is obtained by writing $f(x_0 - h) = f(x_0) - h f'(x_0) + \frac{h^2}{2} f''(\xi)$, hence $f'(x_0)$ is approximated by $\frac{f(x_0) - f(x_0 - h)}{h}$ with truncation error $\frac{h}{2} f''(\xi)$ for some $x_0 - h \le \xi \le x_0$.
- The forward and backward formulas are one-sided, two-point formulas with truncation error $\mathcal{O}(h)$, i.e., they are 1st order methods.

Deriving Formulas Using Taylor Series

Higher order formulas

- We can easily derive 2nd order, three-point formulas.
  - Centered: subtract expressions for $f(x_0+h)$ and $f(x_0-h)$:

$$f(x_0+h)-f(x_0-h)=2hf'(x_0)+\frac{2h^3}{6}f'''(x_0)+\frac{2h^5}{120}f^{(v)}(x_0)+\mathcal{O}(h^7$$

    Thus, $f'(x_0)$ is approximated by $\frac{f(x_0+h)-f(x_0-h)}{2h}$ with truncation error $-\frac{h^2}{6}f'''(\xi)$.
  - One-sided: Please verify that the approximation $\frac{1}{2h}\left(-3f(x_0)+4f(x_0+h)-f(x_0+2h)\right)$ has truncation error $\frac{h^2}{3}f'''(\xi)$ where $\xi\in[x_0,x_0+2h]$.
- Using five points we can derive 4th order formulas, e.g., the formula

$$f'(x_0)\approx\frac{1}{12h}\left(f(x_0-2h)-8f(x_0-h)+8f(x_0+h)-f(x_0+2h)\right),$$

  has the truncation error $e(h)=\frac{h^4}{30}f^{(v)}(\xi)$.
- A similar approach may be used to eliminate for the 2nd derivative. For instance, the famous centred, three-point, 2nd order formula is

$$f''(x_0)=\frac{1}{h^2}\left(f(x_0-h)-2f(x_0)+f(x_0+h)\right)-\frac{h^2}{12}f^{(iv)}(\xi).$$

Deriving Formulas Using Taylor Series

Ex 14.1 :

Example

Approximate $f'(0)$ for $f(x) = e^x$ using methods of order 1, 2 and 4. Record and plot errors for $h = 10^{-k}, \ k = 1, \ldots, 5$.

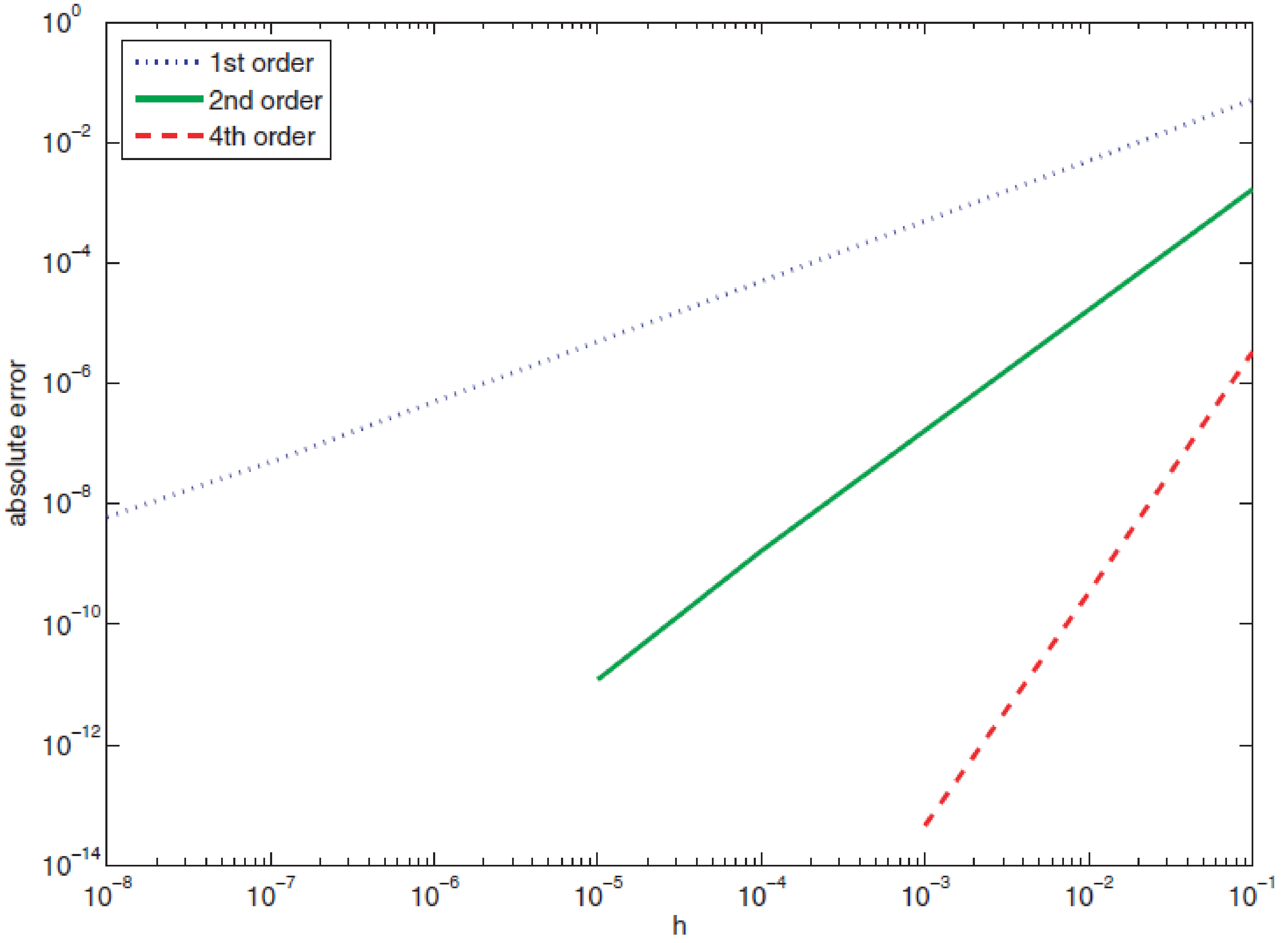
$f(x_0+h) - f(x_0)$

h	$\frac{e^h - 1}{h} - 1$	$\frac{e^h - e^{-h}}{2h} - 1$	$\frac{-e^{2h} + 8e^h - 8e^{-h} + e^{-2h}}{12h} - 1$
0.1	5.17e-2	1.67e-3	3.33e-6
0.01	5.02e-3	1.67e-5	3.33e-10
0.001	5.0e-4	1.67e-7	−4.54e-14
0.0001	5.0e-5	1.67e-9	−2.60e-13
0.00001	5.0e-6	1.21e-11	−3.63e-12

Deriving Formulas Using Taylor Series

Figure displays the errors for values $h = 10^{-k}$, $k = 1, \ldots, 8$. On the log-log scale of the figure, different orders of accuracy appear as different slopes of straight lines. Where the straight lines terminate (going from right to left in h) is where roundoff error takes over.

Before roundoff error dominates, the higher order methods perform very well.



Deriving Formulas Using Taylor Series

Difference formulas

Notation: $f_j = f(x_0 + jh),\ j = 0, \pm 1, \pm 2, \ldots$

Derivative	points	type	order	formula
$hf'(x_0)$	2	forward	1	$f_1 - f_0$
	2	backward	1	$f_0 - f_{-1}$
$2hf'(x_0)$	3	centred	2	$f_1 - f_{-1}$
	3	forward	2	$-3f_0 + 4f_1 - f_2$
$12hf'(x_0)$	5	centred	4	$f_{-2} - 8f_{-1} + 8f_1 - f_2$
$h^2 f''(x_0)$	3	centred	2	$f_{-1} - 2f_0 + f_1$
$12h^2 f''(x_0)$	5	centred	4	$-f_{-2} + 16f_{-1} - 30f_0 + 16f_1 - f_2$

Deriving Formulas Using Taylor Series

Taylor series approach assessment

- Simple and natural to use.
- Directly obtain both formula and its truncation error estimate.
- However, this approach is ad hoc and does not automatically generalize:
  - Sometimes we need to approximate with a non-uniform step size, e.g., approximate $f'(x_0)$ using $f(x_0)$, $f(x_0+h)$, $f(x_0-h/3)$. This would require an individual treatment.
  - Tedious work (and increased chances for human error) may be required for deriving high order formulas.

Richardson Extrapolation

- This is a straightforward, favourite technique based on a simple yet general principle for deriving higher order formulas from lower order ones.
- More limited than the general Taylor series approach, but methodical and easily applied.
- Can be applied repeatedly for generating methods of higher and higher order.
- Used in classical methods for numerical integration and differential equations (Chapters 15, 16).
- **Basic idea**: use lower order formula with more than one step size, and combine to eliminate the leading term of the truncation error.

Richardson Extrapolation

Example of method derivation

Goal: Derive a centred, 4th order formula for the 2nd derivative.

- Write the centred 3-point formula for f_0 once for h and once for $2h$,

$$\begin{aligned} f''(x_0) &= \frac{1}{h^2}(f_{-1} - 2f_0 + f_1) - \frac{h^2}{12} f^{(iv)}(x_0) + \mathcal{O}(h^4) \\ &= \frac{1}{(2h)^2}(f_{-2} - 2f_0 + f_2) - \frac{(2h)^2}{12} f^{(iv)}(x_0) + \mathcal{O}(h^4). \end{aligned}$$

- The leading term of the truncation error in the second line is 4 times larger than in the first. So multiply the first line by 4 and subtract the 2nd line:

$$3f''(x_0) = \frac{4}{h^2}(f_{-1} - 2f_0 + f_1) - \frac{1}{(2h)^2}(f_{-2} - 2f_0 + f_2) + \mathcal{O}(h^4).$$

- Rearrange:

$$f''(x_0) = \frac{1}{12h^2}(-f_{-2} + 16f_{-1} - 30f_0 + 16f_1 - f_2) + \mathcal{O}(h^4).$$

Richardson Extrapolation

Error Estimation

It is often desirable to obtain a practical estimate for the error committed in a numerical differentiation formula without resorting to precise knowledge of the higher derivatives of $f(x)$.

Example 14.3. Instead of combining the two second order expressions in Example 14.2 to obtain a higher order formula, we could use them to obtain an *error estimate*. Thus, subtracting one from the other and ignoring $\mathcal{O}(h^4)$ terms, we clearly obtain an expression for $-\frac{3h^2}{12} f^{(iv)}(x_0)$ in terms of computable values of f at x_0, $x_0 \pm h$, and $x_0 \pm 2h$. Dividing by 3, this yields the computable error estimate

$$e(h) \approx \frac{1}{3}\left[\frac{1}{4h^2}\left(f(x_0-2h)-2f(x_0)+f(x_0+2h)\right) - \frac{1}{h^2}\left(f(x_0-h)-2f(x_0)+f(x_0+h)\right)\right],$$

applicable for the classical three-point formula.

Richardson Extrapolation

Advantages

- Simplicity
- Generality

Disadvantages

- Resulting formulas are typically noncompact: they use more neighboring points involving a wider spacing than perhaps necessary.
- The explicit reliance on the existence of higher and higher *nicely bounded* derivatives of f, so that neglecting higher order error terms is enabled, exposes vulnerability when this assumption is violated.

Deriving Formulas Using Lagrange Polynomial Interpolation

In some cases, Taylor series can become difficult to apply. Reasons are

- when desired order is high
- when the points used are not equally spaced

Thus, the complication of using Taylor expansions quickly increases, especially if a high order method is desired in such a case.

A *general approach* for deriving approximate differentiation formulas at some point x_0 is to choose a few nearby points, interpolate by a polynomial, and then differentiate the interpolant.

Deriving Formulas Using Lagrange Polynomial Interpolation

Consider the derivation of a formula involving the points $x_0, x_1, \ldots, x_n$, not ordered in any particular fashion. The interpolating polynomial of degree at most n is

$$p(x) = \sum_{j=0}^{n} f(x_j) L_j(x),$$

$$L_j(x) = \frac{(x - x_0)\cdots(x - x_{j-1})(x - x_{j+1})\cdots(x - x_n)}{(x_j - x_0)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_n)}.$$

Lagrange polynomials

Taking the derivative of $p(x)$ and substituting $x = x_0$ yields the *general formula for numerical differentiation*

$$p'(x_0) = \sum_{j=0}^{n} f(x_j) L_j'(x_0).$$

This formula does not require the points $x_0, x_1, \ldots, x_n$ to be equidistant!

Deriving Formulas Using Lagrange Polynomial Interpolation

Differentiation Using Equidistant Points

To obtain cleaner expressions, let us next assume equidistant spacing. We will get to the more general case later. Thus, assume that the points x_i are distributed around x_0 and given as $x_0 - lh,\ x_0 - (l-1)h, \ldots, x_0 - h, x_0, x_0 + h, \ldots, x_0 + uh$, where l and u are nonnegative integers and $n = l + u$. There is an obvious shift in index involved, from 0 to $-l$, because we want to emphasize that the points $x_i = x_0 + ih$ are generally on both sides of x_0, where we seek to approximate $f'(x_0)$.

Evaluating the weights of the formula, we get

$$L_0'(x_0) = \sum_{\substack{k=-l \\ k\neq 0}}^{u} \frac{1}{x_0 - x_k} = \frac{1}{h} \sum_{\substack{k=-l \\ k\neq 0}}^{u} \left(\frac{1}{-k} \right),$$

$$L_j'(x_0) = \frac{1}{x_j - x_0} \prod_{\substack{k=-l \\ k\neq 0 \\ k\neq j}}^{u} \frac{x_0 - x_k}{x_j - x_k} = \frac{1}{jh} \prod_{\substack{k=-l \\ k\neq 0 \\ k\neq j}}^{u} \left(\frac{-k}{j-k} \right) \quad \text{for } j \neq 0.$$

Deriving Formulas Using Lagrange Polynomial Interpolation

Differentiation Using Equidistant Points

$$p'(x_0) = h^{-1} \sum_{j=-l}^{u} a_j f(x_j).$$

where

$$a_j = h L_j'(x_0), \quad j = -l, \ldots, u,$$

Deriving Formulas Using Lagrange Polynomial Interpolation

Differentiation Using Equidistant Points

Interpolation Error

$$f(x) - p_n(x) = f[x_{-l}, x_{-l+1}, \ldots, x_u, x] \prod_{k=-l}^{u} (x - x_k).$$

$$f'(x_0) - p_n'(x_0) = \left[\frac{f^{(n+1)}(\xi)}{(n+1)!} l!u! \right] h^n \qquad \text{for some } \xi,\ x_{-l} \le \xi \le x_u.$$

Deriving Formulas Using Lagrange Polynomial Interpolation

Differentiation Using Equidistant Points

Numerical Differentiation.

Based on the points $x_i = x_0 + ih$, $i = -l, \ldots, u$, where $l + u = n$, an nth order formula approximating $f'(x_0)$ is given by

$$f'(x_0) \approx \frac{1}{h} \sum_{j=-l}^{u} a_j f(x_j),$$

where

$$a_j = \begin{cases} -\displaystyle\sum_{\substack{k=-l \\ k \neq 0}}^{u} \left(\frac{1}{k}\right), & j = 0, \\ \displaystyle\frac{1}{j} \prod_{\substack{k=-l \\ k \neq 0 \\ k \neq j}}^{u} \left(\frac{k}{k-j}\right), & j \neq 0. \end{cases}$$

Deriving Formulas Using Lagrange Polynomial Interpolation

Nonuniformly Spaced Points

Recall a major advantage of the current more general but complicated approach, in that the points need not be equally spaced.

Example 14.6. Suppose that the points x_{-1}, $x_0 = x_{-1} + h_0$, and $x_1 = x_0 + h_1$ are to be used to derive a second order formula for $f'(x_0)$ which holds even when $h_0 \neq h_1$. Such a setup arises often in practice.

$$f'(x_0) \approx \frac{f(x_1) - f(x_{-1})}{h_0 + h_1},$$

Instead, consider $p_2'(x_0) = \sum_{j=-1}^{1} f(x_j) L_j'(x_0)$, where p_2 is the interpolating polynomial in Lagrange form. We have

$$f'(x_0) \approx \frac{h_1 - h_0}{h_0 h_1} f(x_0) + \frac{1}{h_0 + h_1} \left(\frac{h_0}{h_1} f(x_1) - \frac{h_1}{h_0} f(x_{-1}) \right)$$

Deriving Formulas Using Lagrange Polynomial Interpolation

Nonuniformly Spaced Points

Table 14.2. *Errors in the numerical differentiation of* $f(x) = e^x$ *at* $x = 0$ *using three-point methods on a nonuniform mesh, Example 14.6. The error* e_g *in the more elaborate method is second order, whereas the simpler method yields first order accuracy* e_s*. The error value 0 at the bottom of the second column is a lucky break.*

h	e_g	e_s
0.1	8.44e-4	2.63e-2
0.01	8.34e-6	2.51e-3
0.001	8.33e-8	2.50e-4
0.0001	8.35e-10	2.50e-5
0.00001	0	2.50e-6

Roundoff and Data Errors

- Recall Example 1.3: for very small h and smooth f we may encounter severe cancellation error amplified by h^{-1} when approximating the derivative f'. Then, roundoff error takes over, and it grows as h decreases.
- Clearly, this problem happens with all numerical differentiation methods we have seen thus far (e.g. see our table of difference formulas).
- In fact, for higher order methods roundoff error dominates sooner (i.e. for larger h) because truncation error decreases faster.

Roundoff and Data Errors

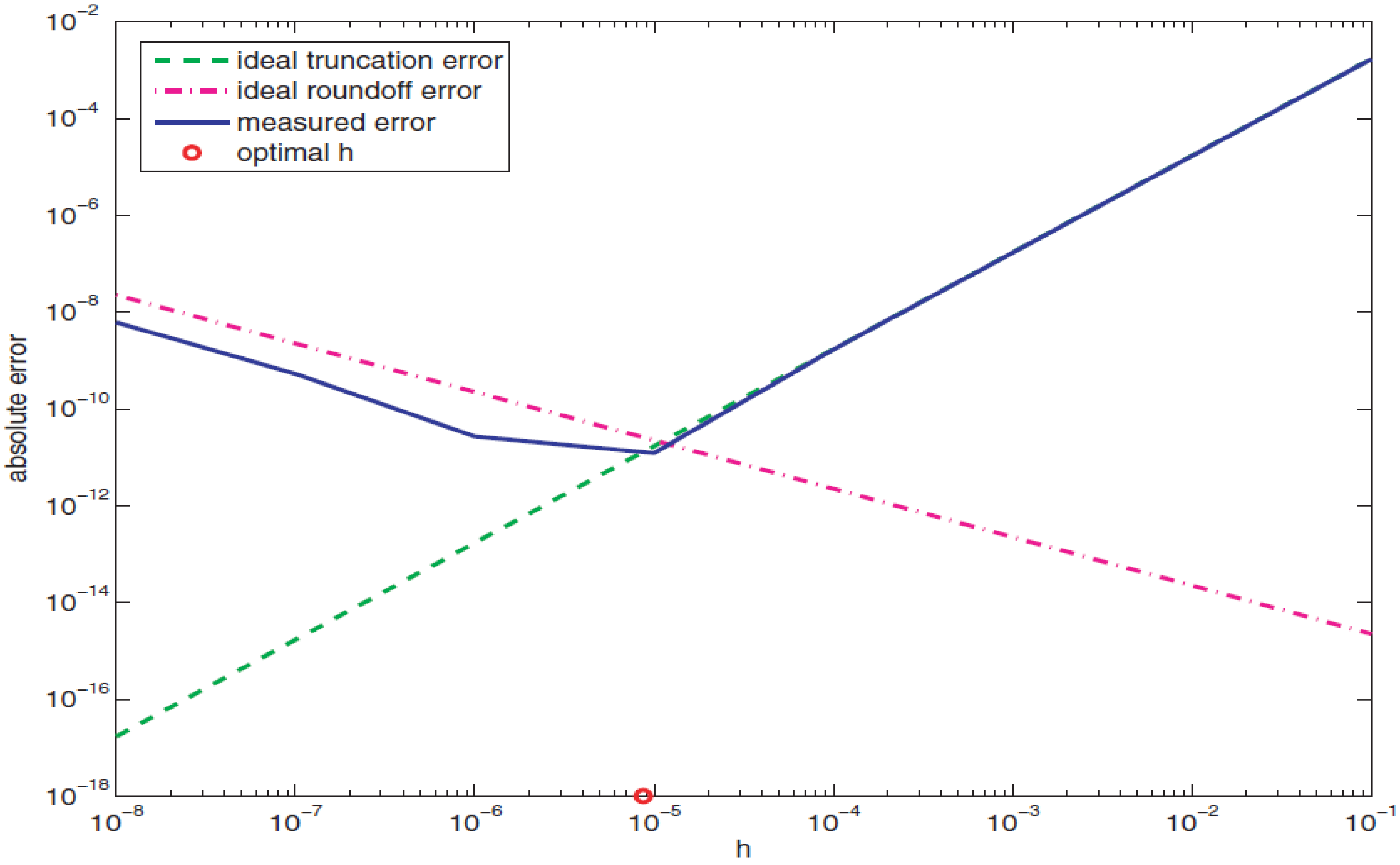


Figure 14.2. *The measured error roughly equals truncation error plus roundoff error. The former decreases but the latter grows as h decreases. The "ideal roundoff error" is just η/h. Note the log-log scale of the plot. A red circle marks the "optimal h" value for Example* 14.1.

Roundoff and Data Errors

Example

Consider the 2nd order method $f'(x_0) \simeq D_h = \frac{f(x_0+h)-f(x_0-h)}{2h}$.

- Denote $\mathrm{fl}(f(x)) \equiv \overline{f}(x) = f(x) + e_r(x), \quad |e_r(x)| \le \epsilon$, where ϵ depends on the **rounding unit**. Assuming exact arithmetic for simplicity, $\overline{D}_h = \frac{\overline{f}(x_0+h)-\overline{f}(x_0-h)}{2h}$.
- Obtain

$$
\begin{aligned}
|\overline{D}_h - D_h| &= \left| \frac{\overline{f}(x_0+h)-\overline{f}(x_0-h)}{2h} - \frac{f(x_0+h)-f(x_0-h)}{2h} \right| \\
&= \left| \frac{e_r(x_0+h)-e_r(x_0-h)}{2h} \right| \\
&\le \left| \frac{e_r(x_0+h)}{2h} \right| + \left| \frac{e_r(x_0-h)}{2h} \right| \le \frac{\epsilon}{h}.
\end{aligned}
$$

- So, if $|f'''(\xi)| \le M$ then

$$
\begin{aligned}
\left|f'(x_0) - \overline{D}_h\right| &= \left|(f'(x_0) - D_h) + (D_h - \overline{D}_h)\right| \\
&\le |f'(x_0) - D_h| + \left|D_h - \overline{D}_h\right| \le \frac{h^2 M}{6} + \frac{\epsilon}{h}.
\end{aligned}
$$

- "Theoretically optimal" h is where this bound is minimized: $h_* = (3\epsilon/M)^{1/3}$.

Roundoff and Data Errors

How bad can this problem be, and what can we do?

- The answer highly depends on the application!
- For discretization of differential equations, can usually keep $h \gg \epsilon$ using IEEE standard "double precision" word (but not "single precision"); see Chapter 2.
- For calculating gradient and Hessian in a more controlled optimization context (also, constrained multibody simulations), it is usually fine to use h "not too small". An alternative is automatic differentiation methods, not covered here, which do not suffer from roundoff error problems.
- For applications involving numerical differentiation of measured data which may contain noise, it is easy to get stuck! A possible remedy is to smooth the data (i.e., approximate the data by a smooth function), and only then differentiate.

The main lesson from this analysis is not to take h too close to the rounding unit η.

Roundoff and Data Errors

Example: differentiating noisy data

We create a function $f(x)$ to be differentiated by first sampling $\sin(x)$ on $[0, 2\pi]$ with $h = .01$, and then adding 1% Gaussian noise to these 200π values. The result is in the left panel below.
Next, numerical differentiation approximately gives $\cos(x)$ plus the noise magnified by $1/h = 100$. The (unacceptable) result is depicted in the right panel below.