Mathematics for Informatics

Numerical Mathematics 3
(lecture 10 of 12)

Francesco Dolce

francesco.dolce@fjfi.cvut.cz

Czech Technical University in Prague

Fall 2019/2020

created: December 9, 2019, 18:09

Outline

1. Smooth optimization
    - Optimization methods overview
    - Smooth optimization methods

2. Numerical Differentiation
    - Introduction and motivation
    - Newton's difference quotient

Outline

1. Smooth optimization
   - Optimization methods overview
   - Smooth optimization methods

2. Numerical Differentiation
   - Introduction and motivation
   - Newton's difference quotient

Examples of optimization in IT

- Clustering
- Classification
- Model fitting
- Recommender systems
- . . .

Optimization methods

Optimization methods can be:

1. **discrete**, when the support is made of several disconnected pieces (usually finite);
2. **smooth**, when the support is connected (we have a derivative).

Optimization methods

Optimization methods can be:

1. **discrete**, when the support is made of several disconnected pieces (usually finite);
2. **smooth**, when the support is connected (we have a derivative).

They are further distinguished based on how the method calculates a solution:

1. **direct**, a finite numeber of steps;
2. **iterative**, the solution is the limit of some approximate results;
3. **heuristic**, methods quickly producing a solution that may not be optimal.

Optimization methods

Optimization methods can be:

1. **discrete**, when the support is made of several disconnected pieces (usually finite);
2. **smooth**, when the support is connected (we have a derivative).

They are further distinguished based on how the method calculates a solution:

1. **direct**, a finite numeber of steps;
2. **iterative**, the solution is the limit of some approximate results;
3. **heuristic**, methods quickly producing a solution that may not be optimal.

Methods are also classified based on randomness:

1. **deterministic**;
2. **stochastic**, e.g., evolution, genetic algorithsm, ...

Gradient descent methods

Goal: find local minima of $f : D_f \to \mathbb{R}$, with $D_f \subset \mathbb{R}^n$.

We assume that f, its first and second derivatives exist and are continuous on D_f.

We shall describe an iterative deterministic method from the family of descent methods.

Descent method - general idea

Let $x^{(1)} \in D_f$.

We shall construct a sequence $x^{(k)}$, with $k = 1, 2, \ldots$, such that

$$x^{(k+1)} \ = \ x^{(k)} + t^{(k)} \Delta x^{(k)},$$

where $\Delta x^{(k)}$ is a suitable vector (in the direction of the descent) and $t^{(k)}$ is the length of the so-called step.

Our goal is to have $f\left(x^{(k+1)}\right) < f\left(x^{(k)}\right)$, except when $x^{(k)}$ is already a point of local minimum.

Descent method - algorithm overview

Let $x \in D_f$.

1. Select Δx;
2. Select $t > 0$;
3. Calculate $x + t\Delta x$ and store it in x;
4. Repeat this loop until the stopping criterion is satisfied.

Descent method - algorithm overview

Let $x \in D_f$.

1. Select Δx;
2. Select $t > 0$;
3. Calculate $x + t\Delta x$ and store it in x;
4. Repeat this loop until the stopping criterion is satisfied.

For instance, a suitable stopping criterion may be the closeness to 0 of the norm of the gradient.

Descent method - choice of t

For small t we approximately have:

$$f(x + t\Delta x) \approx f(x) + t\nabla f(x) \cdot \Delta x.$$

Thus, we need:

$$\nabla f(x) \cdot \Delta x < 0.$$

Descent method - choice of t

For small t we approximately have:

$$f(x + t\Delta x) \approx f(x) + t\nabla f(x) \cdot \Delta x.$$

Thus, we need:

$$\nabla f(x) \cdot \Delta x < 0.$$

The ideal choice of t is a point of local minimum of the mapping

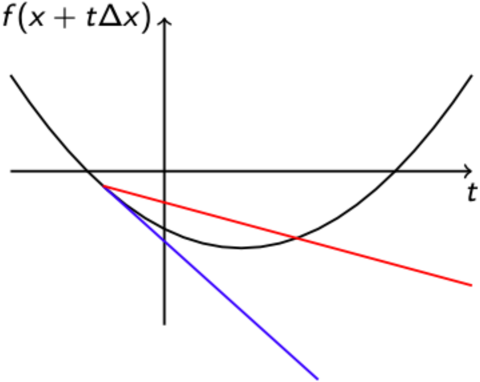
$$s \mapsto f(x + s\Delta x)$$

for some $s > 0$.

We may use any method to solve this subproblem (e.g., analytic solution, Newton method, etc). We shall describe the **backtracking** method.

Backtracking

We have f, $\Delta x \in \mathbb{R}^n$, $x \in D_f$, and parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$.
Set $t := 1$. Replace t by βt until $f(x + t\Delta x) \leq f(x) + \alpha t \nabla f(x) \cdot \Delta x$.



The cut of the graph of f over the line $x + t\Delta x$ is in black, the tangent line $f(x) + t\nabla f(x) \cdot \Delta x$ is in blue, and the acceptable limit for descent $f(x) + \alpha t \nabla f(x) \cdot \Delta x$ is in red.

Choice of Δx (1/2)

Assume we are at the point x and start to move in the direction v. We have again

$$f(x+v) \approx f(x) + \nabla f(x) \cdot v.$$

In order to descend, we need $\nabla f(x) \cdot v < 0$. To find the best direction (where the descent is the greatest), we need to select a norm $\| \cdot \|$ on $\mathbb{R}^n$, and then solve

$$\Delta x = \text{argmin}\{\nabla f(x) \cdot v \ : \ \|v\| = 1\} \cdot .$$

For a symmetric positively definite matrix $P \in \mathbb{R}^{n,n}$ we define the following norm:

$$\|v\|_P = \sqrt{v^T P v}, \quad v \in \mathbb{R}^n.$$

The direction of the greatest descent is given by

$$\Delta x = -P^{-1} \nabla f(x)^T.$$

Choice of Δx (2/2)

The choice $\|\cdot\|_2$ (Euclidean norm) leads to the gradient method. The direction of the greatest descent of f at a point x is $-\nabla f(x)$.
That is, we set

$$\Delta x^{(k)} = -\nabla f(x^{(k)})\cdot$$

Is this the best possible choice? If we have more information on f, we may obtain better results using Newton's method:

$$\Delta x^{(k)} = -\left(\nabla^2 f(x^{(k)})\right)\left(\nabla f(x^{(k)})\right)^T.$$

The direction of the greatest descent is found with respect to the norm $\|\cdot\|_P$ with $P = \nabla^2 f(x^{(k)})$, i.e., the Hessian matrix of f at the point $x^{(k)}$.

Momentum

We shall add a **momentum** (sometimes also called acceleration) to the direction of the descent:

$$\Delta x^{(k)} = \gamma x^{(k-1)} \ + \ \text{argmin}\{\nabla f(x) \cdot v \ : \ \|v\| = 1\},$$

where $\gamma \in [0, 1)$ is the dampening parameter. For $\gamma = 0$ we retrieve the previous methods.

Curve fitting

Assume that we have some data $x_1, x_2, \ldots, x_N \in \mathbb{R}^n$ and an unknown (possibly non-deterministic) mapping $f : \mathbb{R}^n \to \mathbb{R}$ for which we have some observations $y_i = f(x_i)$.

We want to find an approximation of f.

We shall find a curve that best fits the data, i.e., minimize the following

$$\sum_{i=1}^{N} \left(y_i - \beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i,2} - \cdots - \beta_n x_{i_n}\right)^2 ,$$

with $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})^T$.

Polynomial regression

The task

$$\text{minimize} \quad \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i,2} - \cdots - \beta_n x_{i,n})^2$$

is interpreted as finding the best curve that models our data using

$$y_i = \beta_0 + x_i^T \beta + \varepsilon_i,$$

which is the approximation of the mapping f if the error term ε_i is small.

Polynomial regression

The task

$$\text{minimize} \quad \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i,2} - \cdots - \beta_n x_{i,n})^2$$

is interpreted as finding the best curve that models our data using

$$y_i = \beta_0 + x_i^T \beta + \varepsilon_i,$$

which is the approximation of the mapping f if the error term ε_i is small.

Why polynomial regression?

The data may be in the following form:

$$x_i = (z_i, z_i^2, z_i^3, \ldots, z_i^n)^T \quad \text{for some } z_i \in \mathbb{R}.$$

Ordinary least squares (1/2)

If $N > n$, the minimization problem can always be solved analytically:

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

where

$$\mathbb{X} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \cdots & x_{N,n} \end{pmatrix}.$$

The matrix $\mathbb{X}$ is given by

$$\sqrt{\sum_{i=1}^{N}(y_i - \beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i-2} - \cdots - \beta_n x_{i,n})^2} = \left\| \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} - \mathbb{X} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} \right\|_2$$

.

Ordinary least squares (2/2)

The solution

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

is called **ordinary least square estimate** (**OLS**).

Outline

1. Smooth optimization
   - Optimization methods overview
   - Smooth optimization methods

2. Numerical Differentiation
   - Introduction and motivation
   - Newton's difference quotient

Motivation

Differentiation is a basic mathematical operation. It is important to have good methods to compute and manipulate derivatives. Classical methods (real calculus or complex calculus) are of limited value on computers, since most common programming environments do not have support for symbolic computations.

Motivation

Differentiation is a basic mathematical operation. It is important to have good methods to compute and manipulate derivatives. Classical methods (real calculus or complex calculus) are of limited value on computers, since 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 (i.e., numerical methods) of differentiation.

Main idea

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.

Main idea

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 **truncation error**, the mathematical error committed by differentiating the polynomial instead of the function itself.

Main idea

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 **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, unavoidable when using floating-point numbers to perform calculations with real numbers.

Numerical differentiation is very sensitive to round-off errors, but these errors are quite easy to analyse.

The problem

Let f be a given function that is known at a number of isolated points.

The problem of **numerical differentiation** consists in computing an approximation to the derivative f' of f by suitable combinations of the known function values of f.

The basic idea (1/2)

The standard definition of $f'(a)$ is by a limit process,

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

In the following we will assume that this limit exists, in other words that f is differentiable at $x = a$.

The basic idea (1/2)

The standard definition of $f'(a)$ is by a limit process,

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

In the following we will assume that this limit exists, in other words that f is differentiable at $x = a$.

An immediate and natural approximation of $f'(a)$ is then

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

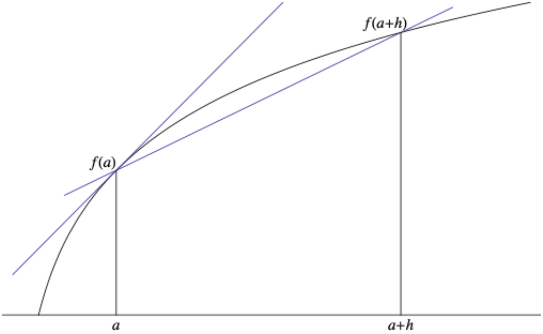
where h is a (small) positive number.
This approximation is called the **Newton's (difference) quotient**.

The basic idea (2/2)

The straight line p_1 that interpolates f at a and $a+h$ is given by

$$p_1(x) = f(a) + \frac{f(a+h) - f(a)}{h}(x - a).$$

The derivative $p_1(x)$ is exactly the Newton's quotient of f at a.
Equivalently, the slope of p_1 at a is an approximation to the slope of f at a.



Left-sided version

An alternative approximation is given by the left-sided version:

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

This approximation behaves similarly, and the analysis is also completely analogous to that of the more common right-sided version.

Example

Let us consider $f(x) = \sin x$ at $a = 0.5$ (using `binary64` floating-point numbers).

We know that the exact derivative is $f'(x) = \cos x$, so $f'(a) \approx 0.8775825619$ with 10 correct digits.

h	$\frac{f(a+h)-f(a)}{h}$	$f'(a) - \frac{f(a+h)-f(a)}{h}$
10^{-1}	0.8521693479	$2.5 \cdot 10^{-2}$
10^{-2}	0.8751708279	$2.4 \cdot 10^{-3}$
10^{-3}	0.8773427029	$2.4 \cdot 10^{-4}$
10^{-4}	0.8775585892	$2.4 \cdot 10^{-5}$
10^{-5}	0.8775801647	$2.4 \cdot 10^{-6}$
10^{-6}	0.8775823222	$2.4 \cdot 10^{-7}$

The approximation improves with decreasing h, as expected.

Truncation error

To analyse errors in numerical differentiation, we use Taylor polynomials with remainders.

Truncation error

To analyse errors in numerical differentiation, we use Taylor polynomials with remainders.

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

where $\xi_h \in (a, a+h)$.

Truncation error

To analyse errors in numerical differentiation, we use Taylor polynomials with remainders.

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

where $\xi_h \in (a, a+h)$.

Thus

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

This is called the **truncation error** of the approximation.

Example

Let us consider again $f(x) = \sin x$ at $a = 0.5$. We have $f''(x) = -\sin x$, so that the truncation error is

$$f'(a) - \frac{f(a+h) - f(a)}{h} = \frac{h}{2} \sin \xi_h \qquad \text{with } \xi_h \in (0.5, 0.5 + h).$$

Example

Let us consider again $f(x) = \sin x$ at $a = 0.5$. We have $f''(x) = -\sin x$, so that the truncation error is

$$f'(a) - \frac{f(a+h) - f(a)}{h} = \frac{h}{2} \sin \xi_h \qquad \text{with } \xi_h \in (0.5, 0.5+h).$$

For $h = 0.1$ the error lies in the interval

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

Example

Let us consider again $f(x) = \sin x$ at $a = 0.5$. We have $f''(x) = -\sin x$, so that the truncation error is

$$f'(a) - \frac{f(a+h) - f(a)}{h} = \frac{h}{2}\sin \xi_h \qquad \text{with } \xi_h \in (0.5, 0.5 + h).$$

For $h = 0.1$ the error lies in the interval

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

As h became even smaller, the number ξ_h will approach 0.5 and $\sin \xi_h$ will approach $\sin 0.5 \approx 0.479426$. So, for $h = 10^{-n}$, the error tends to

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

Approximation of the truncation error

In general, if $f''(x)$ is continuous, then ξ_h will approach a when h goes to zero. Thus we can approximate $f''(\xi_h) \approx f''(a)$.

Theorem

The truncation error when using Newton's quotient to approximate $f'(a)$ is given approximately by

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

Upper bound on the truncation error

The exact value of the truncation error is given by

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

Upper bound on the truncation error

The exact value of the truncation error is given by

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

Theorem

Suppose f has continuous derivatives up to order two near a.
The truncation error is bounded by

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

Note that we included the extremal values of the interval $[a, a+h]$.

Round-off error - an example

When computing the approximation with small values of h we have to perform the critical operation $f(a+h) - f(a)$, i.e., the subtraction of two almost equal numbers. This may lead to large round-off errors.

Round-off error - an example

When computing the approximation with small values of h we have to perform the critical operation $f(a+h) - f(a)$, i.e., the subtraction of two almost equal numbers. This may lead to large round-off errors.

Let us consider again $f(x) = \sin$ at $a = 0.5$ and the correct value with ten digits is $f'(0.5) \approx 0.8775825619$. If we check values of h smaller than 10^{-6} we find

h	$\dfrac{f(a+h)-f(a)}{h}$	$f'(a) - \dfrac{f(a+h)-f(a)}{h}$
10^{-7}	0.8775825372	$2.5 \cdot 10^{-8}$
10^{-8}	0.8775825622	$-2.9 \cdot 10^{-10}$
10^{-9}	0.8775825622	$-2.9 \cdot 10^{-10}$
10^{-11}	0.8775813409	$1.2 \cdot 10^{-6}$
10^{-14}	0.8770761895	$5.1 \cdot 10^{-4}$
10^{-15}	0.8881784197	$-1.1 \cdot 10^{-2}$
10^{-16}	1.110223025	$-2.3 \cdot 10^{-1}$
10^{-17}	0.0000000000	$8.8 \cdot 10^{-1}$

Round-off error in the function values

In a previous lecture we saw that the relative error ϵ_1 in double precision is bounded by

$$|\epsilon_1| = \left| \frac{\mathrm{fl}(f(a)) - f(a)}{f(a)} \right| \leq 5 \cdot 2^{-53} \approx 6 \cdot 10^{-16}.$$

Note that ϵ_1 depends both on a and f.

Round-off error in the function values

In a previous lecture we saw that the relative error ϵ_1 in double precision is bounded by

$$|\epsilon_1| = \left|\frac{\mathrm{fl}(f(a)) - f(a)}{f(a)}\right| \leq 5 \cdot 2^{-53} \approx 6 \cdot 10^{-16}.$$

Note that ϵ_1 depends both on a and f.

Let us denote by ϵ^* the maximum relative error that occurs when real numbers are represented by floating-point numbers, and there is no underflow or overflow. Thus

$$\mathrm{fl}(f(a)) = f(a)(1+\epsilon_1) \qquad \text{and} \qquad \mathrm{fl}(f(a+h)) = f(a+h)(1+\epsilon_2),$$

where $|\epsilon_i| \leq \epsilon^*$ for $i = 1, 2$.

Round-off error in the derivative (1/2)

The main source of round-off in subtraction is the replacement of the numbers to be subtracted by nearest floating-point numbers. We therefore consider the computed approximation to be

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

and ignore the error in the division by h.

Round-off error in the derivative (1/2)

The main source of round-off in subtraction is the replacement of the numbers to be subtracted by nearest floating-point numbers. We therefore consider the computed approximation to be

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

and ignore the error in the division by h. Hence

$$\begin{aligned} f'(a) - \frac{\mathrm{fl}(f(a+h)) - \mathrm{fl}(f(a))}{h} &= f'(a) - \frac{f(a+h) - f(a)}{h} - \frac{f(a+h)\epsilon_2 - f(a)\epsilon_1}{h} \\ &= -\frac{h}{2} f''(\xi_h) - \frac{f(a+h)\epsilon_2 - f(a)\epsilon_1}{h}, \end{aligned}$$

where $\xi_h \in (a, a+h)$.

Round-off error in the derivative (1/2)

The main source of round-off in subtraction is the replacement of the numbers to be subtracted by nearest floating-point numbers. We therefore consider the computed approximation to be

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

and ignore the error in the division by h. Hence

$$\begin{aligned} f'(a) - \frac{\mathrm{fl}(f(a+h)) - \mathrm{fl}(f(a))}{h} &= f'(a) - \frac{f(a+h) - f(a)}{h} - \frac{f(a+h)\epsilon_2 - f(a)\epsilon_1}{h} \\ &= -\frac{h}{2} f''(\xi_h) - \frac{f(a+h)\epsilon_2 - f(a)\epsilon_1}{h}, \end{aligned}$$

where $\xi_h \in (a, a+h)$.

The truncation error is proportional to h, while the round-off error is proportional to $1/h$.

Round-off error in the derivative (2/2)

When h is small, we may assume that $f(a+h) \approx f(a)$. Thus

$$f'(a) - \frac{\mathrm{fl}(f(a+h)) - \mathrm{fl}(f(a))}{h} \approx -\frac{h}{2}f''(a) - \frac{\epsilon_2 - \epsilon_1}{h}f(a).$$

Round-off error in the derivative (2/2)

When h is small, we may assume that $f(a+h) \approx f(a)$. Thus

$$f'(a) - \frac{\mathrm{fl}(f(a+h)) - \mathrm{fl}(f(a))}{h} \approx -\frac{h}{2}f''(a) - \frac{\epsilon_2 - \epsilon_1}{h}f(a).$$

The most uncertain term is $\epsilon_2 - \epsilon_1$. The magnitude of relative errors in `binary64` is about 10^{-17}. If they are of opposite signs, this magnitude may be doubled, so we replace $\epsilon_2 - \epsilon_1$ by $2\tilde{\epsilon}(h)$.

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

Round-off error in the derivative (2/2)

When h is small, we may assume that $f(a+h) \approx f(a)$. Thus

$$f'(a) - \frac{\mathrm{fl}(f(a+h)) - \mathrm{fl}(f(a))}{h} \approx -\frac{h}{2}f''(a) - \frac{\epsilon_2 - \epsilon_1}{h}f(a).$$

The most uncertain term is $\epsilon_2 - \epsilon_1$. The magnitude of relative errors in `binary64` is about 10^{-17}. If they are of opposite signs, this magnitude may be doubled, so we replace $\epsilon_2 - \epsilon_1$ by $2\tilde{\epsilon}(h)$.

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

Hence

$$\tilde{\epsilon}(h) \approx -\frac{h}{2f(a)}\left(f'(a) - \frac{\mathrm{fl}(f(a+h)) - \mathrm{fl}(f(a))}{h} + \frac{h}{2}f''(a).\right)$$

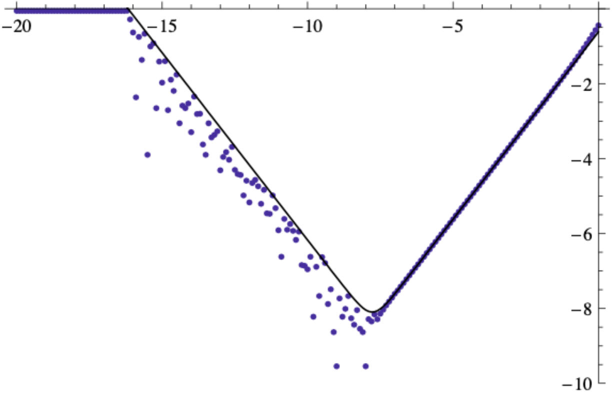
Round-off error in the derivative - an example (1/2)

Let us consider again $f(x) = \sin$ at $a = 0.5$ and the correct value with ten digits is $f'(0.5) \approx 0.8775825619$.

h	$\frac{f(a+h)-f(a)}{h}$	$f'(a) - \frac{f(a+h)-f(a)}{h}$	$\tilde{\epsilon}(h)$
10^{-7}	0.85775825372	$2.5 \cdot 10^{-8}$	-7.6×10^{-17}
10^{-8}	0.8775825622	$-2.9 \cdot 10^{-10}$	2.8×10^{-17}
10^{-9}	0.8775825622	$-2.9 \cdot 10^{-10} \downarrow$	$5.5 \times 10^{-19} \downarrow$
10^{-11}	0.8775813409	$1.2 \cdot 10^{-6} \uparrow$	$-1.3 \times 10^{-17} \uparrow$
10^{-14}	0.8770761895	$5.1 \cdot 10^{-4}$	5.3×10^{-18}
10^{-15}	0.8881784197	$-1.1 \cdot 10^{-2}$	1.1×10^{-17}
10^{-16}	1.110223025	$-2.3 \cdot 10^{-1}$	2.4×10^{-17}
10^{-17}	0.0000000000	$-9.2 \cdot 10^{-1}$	-9.2×10^{-18}

Round-off error in the derivative - an example (2/2)

Numerical approximation of the derivative of $f(x) = \sin x$ at $x = 0.5$ using Newton's quotient. The plot is a $\log_{10} - \log_{10}$ plot.



The point -10 on the horizontal axis corresponds to $h = 10^{-10}$, and the point -6 on the vertical axis corresponds to an error of 10^{-6}.

Approximation of the round-off error

The round-off error is given by

$$\begin{aligned}\left|f'(a) - \frac{\mathrm{fl}(f(a+h)) - \mathrm{fl}(f(a))}{h}\right| &\approx \left|-\frac{h}{2}f''(a) - \frac{\epsilon_2 - \epsilon_1}{h}f(a)\right| \\ &\leq \frac{h}{2}\left|f''(a)\right| + \frac{|\epsilon_2 - \epsilon_1|}{h}\left|f(a)\right| \\ &\leq \frac{h}{2}\left|f''(a)\right| + \frac{\epsilon_2| + |\epsilon_1|}{h}|f(a)| \\ &\leq \frac{h}{2}|f''(a)| + \frac{2\max\{\epsilon_1, \epsilon_2\}}{h}|f(a)|\end{aligned}$$

Approximation of the round-off error

The round-off error is given by

$$\begin{aligned}\left|f'(a) - \frac{\mathrm{fl}(f(a+h)) - \mathrm{fl}(f(a))}{h}\right| &\approx \left|-\frac{h}{2}f''(a) - \frac{\epsilon_2 - \epsilon_1}{h}f(a)\right| \\ &\leq \frac{h}{2}\left|f''(a)\right| + \frac{\left|\epsilon_2 - \epsilon_1\right|}{h}\left|f(a)\right| \\ &\leq \frac{h}{2}\left|f''(a)\right| + \frac{\epsilon_2| + |\epsilon_1|}{h}|f(a)| \\ &\leq \frac{h}{2}|f''(a)| + \frac{2\max\{\epsilon_1, \epsilon_2\}}{h}|f(a)|\end{aligned}$$

Theorem

The round-off error when using Newton's quotient to approximate $f'(a)$ *is roughly bounded by*

$$\left|f'(a) - \frac{fl(f(a+h) - fl(f(a))}{h}\right| \lesssim \frac{h}{2}|f''(a)| + \frac{2\epsilon(h)}{h}|f(a)|$$

where $\epsilon(h) = \max\{\epsilon_1, \epsilon_2\}$.

Upper bound on the round-off error

Theorem

Suppose f has continuous derivatives up to order two near a.
The round-off error is (approximately) bounded by

$$\left| f'(a) - \frac{fl(f(a+h)) - fl(f(a))}{h} \right| \leq \frac{h}{2} M_1 \; + \; \frac{2\epsilon^*}{h} M_2.$$

where $M_1 = \max_{a \in [a, a+h]} |f''(x)|$ *and* $M_2 = \max_{x \in [a, a+h]} |f(x)|$.

Optimal choice of h

The previous example shows that there is an optimal value of h which minimises the total error. If, instead of ϵ_h we consider ϵ^* we have the error estimate

$$E(a,h) = \frac{h}{2}|f''(a)| \; + \; \frac{2\epsilon^*}{h}|f(a)|.$$

Optimal choice of h

The previous example shows that there is an optimal value of h which minimises the total error. If, instead of ϵ_h we consider ϵ^* we have the error estimate

$$E(a,h) = \frac{h}{2}|f''(a)| \ + \ \frac{2\epsilon^*}{h}|f(a)|.$$

To find the value of h which minimises this expression, let us consider

$$E(a,h)' = \frac{|f''(a)|}{2} - \frac{2\epsilon^*}{h^2}|f(a)|.$$

Optimal choice of h

The previous example shows that there is an optimal value of h which minimises the total error. If, instead of ϵ_h we consider ϵ^* we have the error estimate

$$E(a,h) = \frac{h}{2}|f''(a)| \; + \; \frac{2\epsilon^*}{h}|f(a)|.$$

To find the value of h which minimises this expression, let us consider

$$E(a,h)' = \frac{|f''(a)|}{2} - \frac{2\epsilon^*}{h^2}|f(a)|.$$

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

Theorem

Suppose f has continuous derivatives up to order two near a. The value of h which minimizes the total error (truncation error + round-off error) is approximately

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

Other methods (1/2)

- Symmetric version of Newton's quotient
  We find an approximation to $f'(a)$ using the values: $f(a-h)$, $f(a)$ and $f(a+h)$.

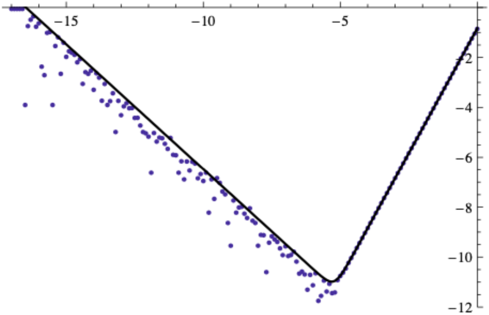
$$f'(a) \approx \frac{f(a+h)-f(a-h)}{2h}.$$

Other methods (1/2)

- Symmetric version of Newton's quotient
  We find an approximation to $f'(a)$ using the values: $f(a-h)$, $f(a)$ and $f(a+h)$.

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

Error: $\left| f'(a) - \frac{\mathrm{fl}(f(a+h) - \mathrm{fl}(f(a-h)))}{2h} \right| \lesssim \frac{h^2}{6}|f'''(a)| + \frac{\epsilon^*|f(a)|}{h}.$



Other methods (2/2)

- A four-point differentiation method
  We choose as interpolation points $a-2h$, $a-h$, $a+h$ and $a+2h$.

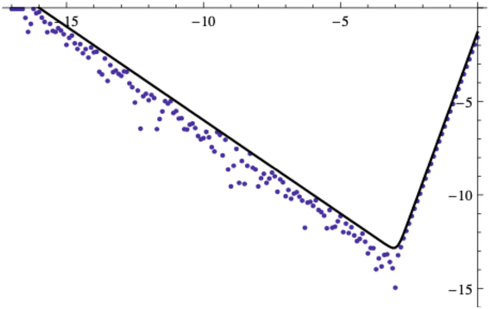
$$f'(a) \approx \frac{f(a-2h) - 8f(a-h) + 8f(a+h) - f(a+2h)}{12h}.$$

Other methods (2/2)

- A four-point differentiation method
  We choose as interpolation points $a-2h$, $a-h$, $a+h$ and $a+2h$.

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

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


−15
−10
−5
−5
−10
−15