HARRI HAKULA & VOLUNTEERS

MS-C1650 NUMERICAL ANALYSIS

APPLIED MATH

Contents

List of Figures

List of Tables

Introduction

This document covers the material of MS-C1650 Numerical Analysis.

Basic Concepts and Definitions

Floating-point arithmetic

Definition

A floating-point number with a base b and length n is defined as

$$x = \pm(.d_1 d_2 ... d_n)_b \cdot b^e,$$

where $m \leq e \leq M$ is the exponent and $(.d_1 d_2 ... d_n)$ is the mantissa. A floating-point number is normalized if $d_1 \neq 0$.

The book uses notation $d_1.d_2...d_n$.

IEEE:

$k = 2$, 64 bits; 1 for the sign, 11 for the exponent and 52 for the mantissa.

"double precision"

All floating-point systems have a machine epsilon that is the smallest defined number after zero. IEEE-standard uses subnormal numbers that fill (one way or another) the underflow gap $[0, \epsilon]$.

IEEE: Double precision

Exponent	Number	Type
0....0	$\pm(0.b_1 b_2 ... b_{52})_2 \cdot 2^{-1022}$	0 or subnormal
$0....01 = 1_{10}$	$\pm(1.b_1 b_2 ... b_{52})_2 \cdot 2^{-1022}$	Normalized
$\vdots$	$\vdots$	Note! Exponent
$01....1 = 1023_{10}$	$\cdots \quad \cdot 2^0$	= "real" + 1023
$\vdots$		
$11....10 = 2046_{10}$	$\cdots \quad \cdot 2^{1023}$	
11....1	$\pm\infty$, if $b_i = 0$, otherwise NaN	Exception

Exceptions: $\pm\infty$, NaN

$$\left.\begin{array}{l}\text{Overflow}\\ \text{Underflow}\end{array}\right\} \Rightarrow \text{value depends on the chosen rounding method}$$

Rounding

$$
\begin{aligned}
\text{down}: &\quad \hat{x} = \text{round}(x) \quad ; \hat{x} \le x \\
\text{up}: &\quad \hat{x} = \text{round}(x) \quad ; \hat{x} \ge x \\
\text{up}: &\quad \hat{x} = \text{round}(x) \quad ; \hat{x} \ge x \\
\text{towards } 0: &\quad \text{up or down, depending} \quad ; \hat{x} \in [0, x] \\
\text{nearest}: &\quad \hat{x} = \text{round}(x) \quad ; \text{the nearest, in case of a tie, the one with a rightmost zero}
\end{aligned}
$$

Assumption: rounding to the nearest value
Holds: $\text{round}(x) = x(1+\delta)$, where $|\delta| < \varepsilon$, (or $|\delta| < \frac{\varepsilon}{2}$, when default rounding mode is used.)
The standard gives:

$$
\begin{aligned}
a \oplus b = &\quad \text{round}(a+b) = &\quad (a+b)(1+\delta_1) \\
a \ominus b = &\quad \text{round}(a-b) = &\quad (a-b)(1+\delta_2) \\
a \otimes b = &\quad \text{round}(a \cdot b) \;\; = &\quad (a \cdot b)(1+\delta_3) \\
a \oslash b = &\quad \text{round}(a/b) \;\; = &\quad (a/b)(1+\delta_4)
\end{aligned}
$$

Documenting the rounding is non-trivial.

Condition numbers

Definition

A condition number describes how sensitive the output value is to a small change in the input argument. (A property of the function, not the algorithm)
Assumption: f: $\mathbb{R} \to \mathbb{R}$, $\hat{x}$ and x close to each other, e.g. $\hat{x} = \text{round}(x)$.
Question: How close is $y = f(x)$ to $\hat{y} = f(\hat{x})$?

Definition

Absolute condition number $C(x)$

$$|\hat{y} - y| \simeq C(x)|\hat{x} - x|$$

Definition

Relative condition number $\kappa(x)$

$$\left|\frac{\hat{y} - y}{y}\right| \simeq \kappa(x) \left|\frac{\hat{x} - x}{x}\right|$$

Model 1

$$\hat{y} - y = f(\hat{x}) - f(x) = \frac{f(\hat{x}) - f(x)}{\hat{x} - x}(\hat{x} - x)$$
$$\frac{f(\hat{x}) - f(x)}{\hat{x} - x} \simeq f'(x)$$
$$\Rightarrow C(x) = |f'(x)|$$

Model 2

Similarly,

$$\frac{\hat{y} - y}{y} = \frac{f(\hat{x}) - f(x)}{\hat{x} - x} \cdot \frac{\hat{x} - x}{x} \cdot \frac{x}{f(x)}$$
$$\frac{f(\hat{x}) - f(x)}{\hat{x} - x} \simeq f'(x)$$
$$\Rightarrow \kappa(x) = \left| \frac{x f'(x)}{f(x)} \right|$$

Lecture problem

Examine the two functions $f(x) = 2x$, $f(x) = \sqrt{2}$.

$$f(x) = 2x, \quad f'(x) = 2 \quad \Rightarrow \quad C(x) = 2, \quad \kappa(x) = 1$$
$$f(x) = x^{\frac{1}{2}}, \quad f'(x) = \frac{1}{2}x^{-\frac{1}{2}} \quad \Rightarrow \quad C(x) = \frac{1}{2}x^{-\frac{1}{2}}, \quad \kappa(x) = \frac{1}{2}$$

Stability in algorithms

$fl(x + y) \equiv \text{round}(x) \oplus \text{round}(y) = (x(1 + \delta_1) + y(1 + \delta_2))(1 + \delta_3)$

Forward error analysis FEA:

How much does the answer $fl(x + y)$ differ from the precise value $x + y$?

Backward error analysis BEA:

What problem yields the obtained precise value?

FEA:

$fl(x + y) = x + y + x(\delta_1 + \delta_2 + \delta_1\delta_3) + y(\delta_2 + \delta_3 + \delta_2\delta_3)$

Absolute error:

$$|fl(x + y) - (x + y)| \leq (|x| + |y|)(2\varepsilon + \varepsilon^2)$$

Relative error:

$$\frac{|fl(x+y)-(x+y)|}{x+y} \leq \frac{(|x|+|y|)(2\varepsilon+\varepsilon^2)}{|x+y|}$$

An interesting situation: $y \approx -x$

BEA:

$$fl(x+y) = x(1+\delta_1)(1+\delta_2) + y(1+\delta_2)(1+\delta_3)$$

Sum of two numbers is therefore backwards stable.

Relative error
$x(1+\delta_1)(1+\delta_2) \leq 2\varepsilon+\varepsilon^2$
Ditto: $y(1+\delta_2)(1+\delta_3)$

Also

A problem can be well-posed even when an algorithm is unstable. A well-posed problem can sometimes be approximated with an ill-conditioned function.

Numerical Differentiation

Difference quotient

Taylor: $f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(\xi), \quad \xi \in [x, x+h]$

Approximation for the derivative:

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

*discretization error: $O(h)$

Because $f'(x) = \frac{f(x+h)-f(x)}{h}$ is a first-order approximation, discretization error is $O(h^1)$.

Assumption: $f(x)$ and $f(x+h)$ are precise: $\delta_i < \varepsilon, i = 1,2$

$$\begin{aligned}\frac{f(x+h)(1+\delta_1)-(f(x)(1+\delta_2)}{h} =& \frac{f(x+h)-f(x)}{h} \\ &+ \frac{\delta_1 f(x+h) - \delta_2 f(x)}{h}\end{aligned}$$

$|$ rounding error $| \leq \frac{2\varepsilon|f(x)|}{h}$ (for small values of h)

Observed:

$$\left.\begin{array}{ll}\text{discretization error} & \sim h \\ \text{rounding error} & \sim \frac{1}{h}\end{array}\right\} \Rightarrow \text{balanced}$$

Example

$f(x) = \sin(x), \quad x = \frac{\pi}{4}; \quad f'(x) = \cos(x), \quad f''(x) = -\sin(x)$

$$\left.\begin{array}{ll} \text{discretization error} & \sim \frac{\sqrt{2}h}{4} \\ \text{rounding error} & \sim \frac{\sqrt{2}\varepsilon}{h} \end{array}\right\} \Rightarrow h = 2\sqrt{\varepsilon}$$

Note:
absolute condition number $C(x) = |-\sin(x)|$
relative condition number $\kappa(x) = |-\frac{x\sin(x)}{\cos(x)}|$,
when $x = \frac{\pi}{4}$, we obtain
$C(\frac{\pi}{4}) = \frac{1}{\sqrt{2}}$;
$\kappa(\frac{\pi}{4}) = \frac{\pi}{4}$.

It is therefore the difference quotient that makes the problem ill-conditioned.

Solving Equations

Bisection

The mean value theorem for continuous functions states that $f(x) = 0$ exists if $x_1 < x < x_2$ so that $f(x_1)$ and $f(x_2)$ have different signs.

The bisection algorithm is based on halving the interval so that the sign requirement applies.

Note that in practise the problem is to find an interval $[x_1, x_2]$.

Rate of convergence: How fast can we obtain the solution, that is, how fast does the error approach zero?

Analysis: Let us have an interval $[a, b]$. After k steps the interval examined is $\frac{|b-a|}{2^k}$ ($\to 0$, when $k \to \infty$). Let us centralize the solution by examining the interval 2δ:

$$\frac{|b-a|}{2^k} \leq 2\delta \Leftrightarrow 2^{k+1} \geq \frac{|b-a|}{\delta} \Leftrightarrow k \geq \log_2\left(\frac{|b-a|}{\delta}\right) - 1$$

The error decreases by a constant factor of $\frac{1}{2}$ on every step. Thus, the algorithm is linearly converging.

Newton's Method

Let the initial guess be x_0. The iteration $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ is Newton's method.

Connection to Taylor polynomial:

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x-x_0)^2}{2} f''(\xi), \quad \xi \in [x_0, x]$$

Let x_* be a zero of $f(x)$: $f(x_*) = 0$

Let us ignore the truncation error and write $x_1 = x_*$:

$0 = f(x_0) + (x_1 - x_0)f'(x_0)$

Theorem

If $f \in C^2$, the initial value x_0 is good enough and $f'(x_*) \neq 0$, Newton's iteration converges asymptotically to the zero x_* with quadratic speed.

Proof (quadraticity)

Taylor polynomial at x_k:

$$x_* = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{(x_* - x_k)^2}{2}\frac{f''(\xi_k)}{f'(x_k)}$$

Let us calculate the difference $x_{k+1} - x_*$:

$$x_{k+1} - x_* = \frac{f''(\xi_k)}{2f'(x_k)}(x_k - x_*)^2$$

With the assumption $\left|\frac{f''(\xi_k)}{2f'(x_k)}\right| \leq C$ the theorem is proved.
(In the book: $C_* = \left|\frac{f''(x_*)}{2f'(x_*)}\right|$ so that $\lim_{k\to\infty} \frac{|x_{k+1}-x_*|}{|x_k-x_*|^2} = C_*$)

Quasi Newton's Methods

In practise, finding the derivative $f'(x_k)$ can be difficult or unreasonably expensive.

Newton's iteration is modified by approximating the derivative with difference quotient:

Secant method

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}, \quad k = 1, 2, \dots$$

Thus, two initial guesses are needed to start the iteration.
The rate of convergence is $\frac{1+\sqrt{5}}{2} \simeq 1.62$.

Interpolation

Lagrange polynomials

Idea: Approximating a function $f(x)$ over the interval $x \in [a, b]$ with a polynomial $p(x)$ so that at the data points $(x_i, y_i), \quad i = 0, 1, ..., n$ the approximation is precise: $y_i = p(x_i)$.

Example

Data points: $(1, 2)$, $(2, 3)$, $(3, 6)$ $\quad ((x_i, y_i),\ i = 0, 1, 2)$

A possible interval: [1,3]; $\quad p_2(x) = \sum_{j=0}^{2} = c_j x^j$

A second order polynomial $\Leftrightarrow$ three unknown coefficients.

$\Rightarrow$ three data points define a unique second order polynomial

In matrix form (Vandermonde):

$$\begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix} \text{that is} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 6 \end{pmatrix}$$

$\Rightarrow c_0 = 3, \quad c_1 = -2, \quad c_2 = 1; \quad p_2(x) = x^2 - 2x + 3$

Unfortunately, this method is highly sensitive to error in the input values.

The complexity of solving a linear system of equations: $O(n^3)$

Idea: Let us replace the basis x^j with a "better" one. The best possible scenario:

$$p(x) = \sum_i y_i \varphi_i(x), \quad \text{when} \begin{cases} \varphi_i(x_i) = 1 \\ \varphi_i(x_j) = 0, \quad i \neq j. \end{cases}$$

We find that the construction of $\varphi_i(x)$ is simple.

Definition Lagrange polynomials

$\varphi_i(x) = \prod_{j \neq i} \dfrac{x - x_j}{x_i - x_j}; \quad p(x) = \sum y_i \varphi_i(x)$ is the so-called Lagrange's form

Example

$$\begin{cases} \varphi_0(x) = \frac{(x-2)(x-3)}{(1-2)(1-3)} \\ \varphi_1(x) = \frac{(x-1)(x-3)}{(2-1)(2-3)} \\ \varphi_2(x) = \frac{(x-1)(x-2)}{(3-1)(3-2)} \end{cases} \Rightarrow p(x) = 2\varphi_0(x) + 3\varphi_1(x) + 6\varphi_2(x) = x^2 - 2x + 3$$

Now the complexity is: $O(n^2)$

Side step 1:

The evaluation of a polynomial in basis x^j is linear: $O(n^2)$

Horner: $y = c_n; \quad y = yx + c_{n-1}; ...$ n steps $\Rightarrow y = \sum_{j=0}^{n} c_j x^j$

Side step 2:

Theorem Interpolation polynomial $p_n(x)$ is unique

Central idea for the proof:

$p_n(x)$ has n zeros. Let $p_n(x)$ and $q_n(x)$ be interpolation polynomials. $(p_n(x_i) - q_n(x_i)) = 0, \quad i = 0, 1, .., n$ so there is $n+1$ zeros. The difference: $p_n(x) - q_n(x) = 0$

Back to business

The Lagrange form can be written more efficiently in the so-called barycentric form, where the evaluation is faster.

Definition of a new basis polynomials: $\hat{\varphi}_i(x) = \prod_{j=0}^{n} \frac{x-x_j}{x-x_I}$ Then let $\varphi(x) = \prod_{j=0}^{n} (x - x_j)$ so $p(x) = \varphi(x) \sum_{i=0}^{n} \frac{w_i}{x-x_i}, \quad w_i = \frac{1}{\prod_{j \neq i} (x_i - x_j)}$

We have formed the first barycentric form. The calculation of the weights w_i is (n^2), but the evaluation is only $O(n)$.

We observe that if $y_i = 1$, then $p_n(x) = 1$. Therefore must be: $1 = \varphi(x) \sum_{i=0}^{n} \frac{w_i}{x-x_i}, \quad$ for all x.

Definition Barycentric interpolation formula

$$p(x) = \left(\sum_{i=0}^{n} \frac{w_i}{x - x_i} y_i \right) \Big/ \left(\sum_{i=0}^{n} \frac{w_i}{x - x_i} \right)$$

Example

$$
\begin{aligned}
&y_0 = 2, \quad y_1 = 3, \quad y_2 = 6 \\
&w_o = \frac{1}{(1-2)(1-3)} = \frac{1}{2}, \quad w_1 = \frac{1}{(2-1)(2-3)} = -1, \quad w_2 = \frac{1}{(3-1)(3-2)} = \frac{1}{2} \\
&p(x) = \left(\frac{2}{2(x-1)} - \frac{3}{x-2} + \frac{6}{2(x-3)}\right) \Big/ \left(\frac{1}{2(x-1)} - \frac{1}{x-2} + \frac{1}{2(x-3)}\right)
\end{aligned}
$$

Does this yield us the same result?

$$
\begin{aligned}
p(x) &= \left(\frac{x^2-2x+3}{(x-1)(x-2)(x-3)}\right) \Big/ \left(\frac{1}{(x-1)(x-2)(x-3)}\right) \\
&= x^2 - 2x + 3
\end{aligned}
$$

Hurray!

Newton polynomials

An extension to the natural basis is the set
$1, x - x_0, (x - x_0)(x -_1), \cdots, \prod_{j=0}^{n-1} (x - x_j).$

Definition Newton's interpolation polynomials

$p_n(x) = a_0 + a_1(x_1 - x_0) + ... + a_n \prod_{j=0}^{n-1},$

where a_i is chosen such that the interpolation condition is true for every x_i.

The construction is equivalent to solving a lower triangular matrix: $O(n^2)$

$$
\begin{aligned}
p(x_0) &= a_o = y_0 \\
p(x_1) &= a_= + a_1(x_1 - x_0) = y_1 \;\Rightarrow\; a_1 = \frac{y_1 - a_0}{x_1 - x_0} \\
&\vdots
\end{aligned}
$$

That is:

$$
\begin{pmatrix}
1 & & & & \\
1 & x_1 - x_0 & & & \\
1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & & \\
\vdots & & & \ddots & \\
1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \cdots & \prod_{j=0}^{n-1}(x_n - x_j)
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}
=
\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}
$$

Two remarks:

a) The order of the points makes no difference.*

*stabililty varies between permutations

b) Adding a data point doesn't affect the previously calculated coefficients.

In the barycentric interpolation above the weights w_i can also be updated incrementally.

Example

$$p(x) = a_0 + a_1(x-1) + a_2(x-1)(x-2)$$

$$\text{System:} \quad \begin{pmatrix} 1 & & \\ 1 & 1 & \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 6 \end{pmatrix} \quad \Rightarrow \quad \begin{cases} a_0 = 2 \\ a_1 = 1 \\ a_2 = 1 \end{cases}$$

$$p(x) = x^2 - 2x + 3$$

Potential problem: Overflow and underflow in large systems

Divided differences

Let us consider the interpolating polynomial in the natural basis:

$$p_n(x) = \sum_{k=0}^{n} a_k x^k$$

Notice that a_k are exaclty the coefficients of the Newton interpolation polynomial.

Definition Divided differences

k^{th}-order divided difference $f[x_0, x_1, x_2, \cdots, x_k] = a_k$, where a_k is the coefficient of the term x^k in the polynomial of degree k that interpolates the points x_i.

Why is this a sensible definition?

One data point: $f[x_j] = f_j = y_j$ (Correct!)

Two data points: $f[x_i, x_j] = \frac{f_j - f_i}{x_j - x_i} = \frac{f[x_j] - f[x_i]}{x_j - x_i}$

Three data points: $f[x_i, x_j, x_k] = \frac{f[x_j, x_k] - f[x_i, x_j]}{x_k - x_i}$

Theorem

$$f[x_o, x_1, \cdots, x_k] = \frac{f[x_1, \cdots, x_k] - f[x_0, \cdots, x_{k-1}]}{x_k - x_0}$$

Proof

Three interpolating polynomials:

p of degree k; $(x_o, f_0), \cdots, (x_k, f_k)$
q of degree $k-1$; $(x_o, f_0), \cdots, (x_{k-1}, f_{k-1})$
r of degree $k-1$; $(x_1, f_1), \cdots, (x_k, f_k)$
Claim: $p(x) = q(x) + \frac{x-x_0}{x_k-x_0}(r(x) - q(x)$
$x_0 : p(x_0) = q(x_0) = f_0$
$x_1, \cdots, x_{k-1} : p(x_i) = q(x_i) = r(x_i) = f_i, \quad i = 1, \cdots, k-1$
$x_k : p(x_k) = r(x_k) = f_k$; RHS : $q(x_k) + 1 \cdot (r(x_k) - q(x_k) = r(x_k)$ □

Example

$$f[x_0] = 2$$
$$f[x_1] = 3 \quad f[x_0, x_1] = \frac{3-2}{2-1} = 1$$
$$f[x_2] = 6 \quad f[x_1, x_2] = \frac{6-3}{3-2} = 3$$
$$f[x_0.x_1, x_2] = \frac{3-1}{3-1} = 1$$

We have gained the exact coefficients a_k!

Interpolation error

$R(x) = f(x) - p(x)$ Let us assume that f differentiable $(n+1)$ times. Let x' be some point other than x_i.

Formation of an aiding function: $h(x) = f(x) - p(x) - c \cdot w(x)$, where $W(x) = \prod_{j=0}^{n}(x - x_j)$ and $c = \frac{f(x')-p(x')}{w(x')}$. The zeros of the function $h(x)$ are $x_0, \cdots, x_n$ (n+1 zeros) and x'. Hence, there are at least $n+2$ zeros. By using Rolle's theorem, we can conclude that $h^{(n+1)}$ has at least one zero, denoted by ξ. $h^{(n+1)} = f^{(n+1)}(x) - p^{(n+1)}(x) - c^w(n+1)(x) = f^{(n+1)}(x) - c(n+1)! \Rightarrow h^{(n+1)}(\xi) = f^{(n+1)}(\xi) - c(n+1)! = 0 \Rightarrow c = \frac{f^{(n+1)}(\xi)}{(n+1)!}$

At the point x' : $\quad R(x') = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^{n}(x' - x_j)$

Theorem

$R(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^{n}(x - x_j)$, where $\xi = \xi(x)$

Mark that by the definition of $h(x)$ the constant c is the coefficient of the highest order term. Based on the previous result, c is some divided difference: $f[x_0, \cdots, x_n, x] = \frac{1}{(n+1)!} f^{(n+1)}(\xi(x))$.

Piecewise polynomial approximation

Idea: Let us subdivide an interval $[a, b]$ into subintervals of length $h = \frac{b-a}{n}$, where n is the number of subintervals. Each subinterval will be approximated separately with a low-degree polynomial.

Linear piecewise interpolation polynomial (Interpolant)

$l(x) = f(x_{i-1})\frac{x-x_i}{x_{i-1}-x_i} + f(x_i)\frac{x-x_i}{x_i-x_{i-1}}, \quad x \in [x_i, x_{i-1}]$ Interpolation error: $f(x) - l(x) = \frac{f''(\xi)}{2!}(x - x_{i-1})(x - x_i)$ Assume: $|f''(x)| \leq M$: $|f(x) - l(x)| \leq M\frac{h^2}{8}, \quad x \in [x_{i-1}, x_i]$

If the derivative is bounded over the whole interval $[a, b]$ the error is the same.

Hermite interpolation

We require the derivative to be continuous. Let $p(x)$ be a third order polynomial. The derivative of $p(x)$ is quadratic: $p'(x) = f'(x_{x-1})\frac{x-x_1}{x_{x-1}-x_i} + f'(x_i)\frac{x-x_{x-i}}{x_i-x_{i-1}} + \alpha(x - x_{i-1})(x - x_i)$ Must be: $p'(x_i) = f'(x_i)$. We must fit parameter α to the data: Integrating: $p(x) = -\frac{f'(x_{i-1})}{h}\int_{x_{i-1}}^{x}(t - x_i)dt + \frac{f'(x_i)}{h}\int_{x_{i-1}}^{x}(t - x_{i-1})dt + \alpha\int_{x_{i-1}}^{x}(t - x_i)(t - x_i)dt + C$ Instantly: $p(x_{i-1}) = f(x_{i-1}) \Rightarrow C = f(x_{i-1})$ Accordingly: $p(x_i) = f(x_i) \Rightarrow \alpha = \frac{3}{h^2}(f'(x_{i-1}) + f'(x_i)) + \frac{6}{h^2}(f(x_{i-1}) - f(x_i))$

Splines

If we abandon the need to fit the derivative, we can construct a third order polynomial which has two continuous derivatives at points $x_i : s(x)$

Problem: To choose the coefficients we require the solution to a global problem. For each interval we derive a single spline: $s(x)$

Construction: Let us assume that $z_i = s''(x_i), \quad i = 1, \cdots, n-1$ is known. In addition, $h = x_i - x_{i-1}$ (=constant).

Interval: $[x_{i-1}, x_i]$: $\quad s_i''(x) = \frac{1}{h}z_{i-1}(x_i - x) + \frac{1}{h}z_i(x - xi - 1)$ By integrating twice:

$s_i(x) = \frac{1}{h}z_{i-1}\frac{(x_i-x)^3}{6} + \frac{1}{h}z_i\frac{(x-x_{i-1})^3}{6} + C_i(x - x_{i-1}) + D_i$

Interpolation condition attaches the constants C_i and D_i:

$D_i = f_{i-1} - \frac{h^2}{6}z_{i-1}$

$C_i = \frac{1}{h}[f_i - f_{i-1} + \frac{h^2}{6}(z_{i-1} - z_i)]$

We have derived a formula that evaluates the spline over every subinterval. However, we still must solve z_i and set a value for the boundaries z_0 and z_n.

By calculating the derivative of $s(x)$ and then exploiting continuity:

$s_i'(x_i) = s_{i+1}'(x_i)$:
$\frac{h}{2}z_i + \frac{1}{h}(f_i - f_{i-1}) + \frac{h^2}{6}(z_{i-1} - z_i) =$
$-\frac{h}{2}z_i + \frac{1}{h}(f_{i+1} - f_i) + \frac{h^2}{6}(z_i - z_{i-1}), \quad i = 1, \cdots n-1,$
which is a tridiagonal matrix:

$$\begin{aligned}\frac{2h}{3}z_i + \frac{h}{6}z_{i-1} + \frac{h}{6}z_{i+1} &= -\frac{2}{h}f_i + \frac{1}{h}f_{i-1} + \frac{1}{h}f_{i+1}\\ &= \frac{1}{h}(f_{i+1} - 2f_i + f_{i-1})\\ &= b_i\end{aligned}$$

Taking z_0 and z_n to the right side
$b_1 = \frac{1}{h}(f_2 - 2f_1 + f_0) - \frac{h}{6}z_0,$
$b_{n-1} = \frac{1}{h}(f_n - 2f_{n-1} + f_{n-2}) - \frac{h}{6}z_n$
We form a so-called natural spline by choosing $z_0 = z_n = 0$
Other options for choosing the value for z_0 and z_n:
a) The first derivative at the end points is precise.
b) The third derivative is continuous at x_1 and x_{n-1}, this is known as the not-a-knot condition.

Bezier

Bernstein Polynomials; $B_k^n(t)$, $t \in [0,1]$

Definition $B_k^n(t) = \binom{n}{k} t^k (1-t)^{n-k}$

Bernstein polynomials have useful properties:

1) $\sum_{k=0}^{n} B_k^n(t) = 1 \ (= (t+1-t)^n)$

2) $0 \leq B_k^n(t) \leq 1,$ for each $k, n \geq 0$

3) $B_0^n(0) = B_n^n(1) = 1$, otherwise $B_k^n(0) = B_k^n(1) = 0$

From combinatorics we obtain the fundamental property of recursion:

$$B_k^n(t) = (1-t)B_k^{n-1}(t) + tB_{k-1}^{n-1}(t)$$

Bézier Curves

Let us use the notation $x^k \in \mathbb{R}^n$ (point).

Definition

Given is the set of points $x = x^1, ..., x^k \in \mathbb{R}^n$, the convex hull of which is:

$$CHull(x) = \{y \in \mathbb{R}^n \mid y = \sum_{i=1}^{k} a_i x^i,\ a_i \geq 0, \sum_{i=1}^{k} a_i = 1\}$$

Definition: Bezier curve

The following curve, determined by the set of points x, is a Bezier curve.

$$\beta^n(t) = \sum_{k=0}^{n} x^k B_k^n(t)$$

The Bezier curve $\beta^n(t)$ is within the convex hull formed by points x (control points, Bezier-points). It follows from the properties of Bernstein polynomials that $\beta^n(t)$ passes through the first and last control point.

Closed curves: control points: $x^0 = x^n$

If the closed curve is to be smooth at the starting point, the tangent vectors at the endpoints must be codirectional.

Let us differentiate:

$$\frac{d}{dt}\beta^n(t) = \frac{d}{dt}\sum_{k=0}^{n} x^k B_k^n(t)$$

Bernstein: $\frac{d}{dt}B_k^n(t) = n(B_{k-1}^{n-1}(t) - B_k^{n-1}(t))$

$$\begin{aligned}\frac{d}{dt}\beta^n(t) &= n\sum_{k=0}^{n}(B_{k-1}^{n-1}(t) - B_k^{n-1}(t))x^k \\ &= n\sum_{k=0}^{n-1}(x^{k+1} - x^k)B_k^{n-1}(t)\end{aligned}$$

Note that the derivative of the Bezier curve is also a Bezier curve!

Thus, we obtain:

$$\begin{cases}\frac{d}{dt}\beta^n(0) = n(x^1 - x^0) \\ \frac{d}{dt}\beta^n(1) = n(x^n - x^{n-1})\end{cases}$$

Geometrically: x^0, x^1, x^{n-1} are on the same line and x^0 is between x^1 and x^{n-1}.

Lifting algorithm

The control points uniquely define a curve, but the opposite does not hold true.

Now, the following applies:

$$\beta^n(t) = \sum_{k=0}^{n} x^k B_k^n(t) = \sum_{k=0}^{n+1} y^k B_k^{n+1}(t) = \alpha^{n+1}(t)$$

By setting $x^{-1} = x^{n+1} = 0$, we obtain the condition

$$y^k = \left(1 - \frac{k}{n+1}\right)x^k + \left(\frac{k}{n+1}\right)x^{k-1} .$$

De Casteljau Algorithm

The previously described ideas can be combined into a practical algorithm. Let the control points be x^0, x^1,...,x^n:

(1) The constant curves are defined: $\beta_i^0(t) = x^i$

(2) $\beta_i^r(t) = (1-t)\beta_i^{r-1}(t) + t\beta_{i+1}^{r-1}(t)\,;\ r = 1, ..., n\,;\ i = 0, ..., n-r.$

The algorithm ends with the curve $\beta_0^n(t)$.

Numerical Integration

Monte Carlo

Central limit theorem

Let X_i be independent and identically distributed random variables with an expected value μ and a variance σ^2. In this case, for the sample average $A_N = \frac{1}{N}\sum_{i=1}^{N} X_i$ we have the variance

$$\mathrm{Var}(A_N) = \frac{1}{N^2}\sum_{i=1}^{N}\mathrm{Var}(X_i) = \frac{\sigma^2}{N}.$$

The standard deviation σ has the same units as X_i: $\sigma(A_N) = \frac{\sigma}{\sqrt{N}}$.

Thus, the speed of convergence for Monte Carlo methods is of the order $O(\frac{1}{\sqrt{N}})$, where N is the amount of integration points. Remarkably, this holds regardless of the dimension!

Buffon's needle

The distance between two lines is denoted by D. What is the probability that a dropped needle with the length L intersects a line?

Let y be the distance from the center of the needle to the closest line and θ the angle shown in Figure 1.

Figure 1: Buffon's needle.

Let us choose $L = D = 1$; y and θ random variables with distributions $y \sim \mathrm{Unif}(0, \frac{1}{2})$, $\theta \sim \mathrm{Unif}(0, \pi)$. The condition for intersection: $y \leq \frac{1}{2}\sin\theta$.

Determining the probability requires calculating the ratio of areas: Possible configurations are the points $[0, \pi] \times [0, \frac{1}{2}]$ i.e. the area $\frac{\pi}{2}$, the condition is fulfilled by $\int_0^{\pi} \frac{1}{2}\sin\theta \, d\theta = 1$;

$$P = \frac{1}{(\frac{\pi}{2})} = \frac{2}{\pi}$$

Hence the approximation: $\pi \approx 2($ #drops / #intersections $)$.

Example Difficult geometry

$$I = \iiint_V \gamma(x,y,z)\, dx\, dy\, dz, \text{ for the density } \gamma(x,y,z) = e^{z/2}.$$

V is defined by the inequations $\begin{cases} xyz \leq 1, \\ -5 \leq x,y,z \leq 5. \end{cases}$

Due to the exponential distribution of the density, the volume and mass integrals over the same region V converge in a different manner: the standard deviation of the volume is lower.

In many cases, a suitable change of variables turns the situation around: $u = e^{z/2} \quad : -5 \leq z \leq 5 \quad \rightarrow e^{-2.5} \approx 0.08 \leq u \leq e^{2.5} \approx 12.2$

$$I = 2\int_{e^{-2.5}}^{e^{2.5}} \int_{-5}^{5} \int_{-5}^{5} \begin{cases} 0, 2xy \ln u > 1 \\ 1, 2xy \ln u \leq 1 \end{cases} \quad dx\, dy\, du$$

To halve the standard deviation one must typically quadruple the integration points. A custom fitted distribution is usually more efficient.

Example Higher dimension

Let us examine the general case:

$$I = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_n}^{b_n} f(x_1, x_2, \ldots, x_n)\, dx_n \ldots\, dx_2\, dx_1, \left(\int \int \cdots \int_V f\, dV \right)$$

where the limits of inner integrals can be functions of outer variables: $a_2 \equiv a_2(x_1)$, $b_n \equiv b_n(x_1, \ldots, x_{n-1})$.

Proceeding as above, we obtain limits of the surrounding volume (min/max) for each dimension: $[A_1, B_1], [A_2, B_2], \ldots, [A_n, B_n]$ and $\hat{V} = [A_1, B_1] \times [A_2, B_2] \times \cdots \times [A_n, B_n]$.

Let us define a function g so that

$$g(x_1, x_2, \ldots, x_n) = \begin{cases} 0, \text{ if } (x_1, \ldots, x_n) \in \hat{V} \setminus V \\ 1, \text{ if } (x_1, \ldots, x_n) \in \hat{V} \end{cases}$$

$$I \approx \sum_{i=1}^{N} g_i \left(\frac{|\hat{V}|}{N} \right), \text{ where } |\hat{V}| = (B_1 - A_1) \ldots (B_n - A_n).$$

High-dimensional problems are of great interest currently. Monte Carlo methods are natural, however, the slow rate of convergence is problematic.

Example MATLAB Another estimation of π

The area of a circle: $A = \pi r^2$

Let us set $r = 1$, in which case $\hat{V} = [-1, 1] \times [-1, 1]$ and $|\hat{V}| = 4$.

Counter: $g_i = \begin{cases} 1, \text{if the point is inside the circle} \\ 0, \text{otherwise.} \end{cases}$

The routine: (N denotes the number of points)

```
numberin = 0
for i = 1:N
    x = 2 * rand - 1
    y = 2 * rand - 2
    if x^2 + y^2 < 1
        numberin = numberin + 1
    end
end
pio4 = numberin / N         // ratio of areas = pi/4
piapprox = 4 * pio4
```

Spread? $\mathrm{Var}(aX) = a^2 \mathrm{Var}(X)$

$\mathrm{Var}(X_i) = \mathrm{E}(X_i^2) - (\mathrm{E}(X_i))^2$, and here: $X_i^2 = X_i \ (= g_i)$

```
varpio4 = (pio4 - pio4^2) / N
varpi = 16 * varpio4
stdpi = sqrt(varpi)
```

Newton-Cotes

Idea: Let us approximate the integral $\int_a^b f(x)\,dx$ by integrating an interpolant of the function f.

$$\text{Lagrange:} \int_a^b f(x)\,dx \approx \sum_{i=0}^{n} f(x_i) \int_a^b \left(\prod_{\substack{j=0 \\ i \neq j}}^{n} \frac{x - x_j}{x_i - x_j} \right) dx$$

The familiar trapezoidal rule is obtained by choosing $n = 1$:

$$p_1(x) = f(a)\frac{x-b}{a-b} + f(b)\frac{x-a}{b-a},$$
$$\text{i.e.} \int_a^b f(x)\,dx \simeq \int_a^b p_1(x)\,dx = \frac{b-a}{2}\left[f(a) + f(b)\right]$$

The error is the integral of the interpolation error; for the trapezoid:

$$\begin{aligned} \int_a^b f(x)\,dx - \int_a^b p_1(x)\,dx &= \frac{1}{2}\int_a^b f''(\xi(x))(x-a)(x-b)\,dx \\ &= \frac{1}{2} f''(\eta) \int_a^b (x-a)(x-b)\,dx \\ &= -\frac{1}{12}(b-a)^3 f''(\eta) \end{aligned}$$

Over n subintervals:

$$\int_a^b f(x)\,dx \simeq \frac{h}{2}\left[f(x_0)+2f(x_1)+2f(x_2)+\cdots+f(x_n)\right]$$

and an error of $O(h^2)$.

What about $n=2$?

$$\text{Required: } \int_a^b f(x)\,dx \approx A_1 f(a) + A_2 f\left(\frac{a+b}{2}\right) + A_3 f(b),$$

accurate for all second-degree (or lower) polynomials. Evidently, the coefficients A_i are obtained from the integrals of the polynomial bases. Let us proceed with the undefined coefficients:

$$\begin{aligned}
\int_a^b 1\,dx &= b-a & &\Rightarrow A_1 + A_2 + A_3 = b-a\\
\int_a^b x\,dx &= \frac{1}{2}(b^2-a^2) & &\Rightarrow A_1 a + A_2\frac{a+b}{2} + A_3 b = \frac{1}{2}(b^2-a^2)\\
\int_a^b x^2\,dx &= \frac{1}{3}(b^3-a^3) & &\Rightarrow A_1 a^2 + A_2\left(\frac{a+b}{2}\right)^2 + A_3 b^2 = \frac{1}{3}(b^3-a3)
\end{aligned}$$

We obtain: $A_1 = A_3 = \frac{b-a}{6}$, $A_2 = \frac{4(b-a)}{6}$

This is known as Simpson's rule:

$$\int_a^b f(x)\,dx \simeq \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

Over n subintervals:

$$\begin{aligned}
\int_a^b f(x)\,dx \simeq \frac{h}{6}[&f(x_0)+4f(x_{1/2})+2f(x_1)+\dots\\
&+2f(x_{n-1})+4f(x_{n-1/2})+f(x_n)]
\end{aligned}$$

And the error: (Accurate for polynomials of degree three)

For one interval: $\frac{1}{2880}(b-a)^5 f^{(4)}(\xi)$ and for n subintervals $O(h^4)$.

Gaussian quadrature

Idea: Let us choose the points and weights simultaneously.

$$\text{Problem: } n=1: \int_a^b f(x)\,dx \simeq A_0 f(x_0) + A_1 f(x_1)$$

As above:

$$\begin{aligned}
\int_a^b 1\,dx &= b-a & &\Rightarrow A_0 + A_1 = b-a\\
\int_a^b x\,dx &= \frac{1}{2}(b^2-a^2) & &\Rightarrow A_0 x_0 + A_1 x_1 = \frac{1}{2}(b^2-a^2)\\
\int_a^b x^2\,dx &= \frac{1}{3}(b^3-a^3) & &\Rightarrow A_0 x_0^2 + A_1 x_1^2 = \frac{1}{3}(b^3-a3)\\
&\vdots
\end{aligned}$$

We have a nonlinear system of equations!

The answer: Orthogonal polynomials

Definition Orthogonal polynomials

Two polynomials are are orthogonal over the interval $[a, b]$ if their inner product is zero.

$$\langle p, q \rangle = \int_a^b p(x)q(x)\,dx = 0$$

For orthonormal polynomials $\langle p, p \rangle = \langle q, q \rangle = 1$.

Gram-Schmidt: $\{1, x, x^2, \dots\} \to \{q_0, q_1, q_2, \dots\} \Leftarrow$ orthonormal

$$q_0 = 1/\left[\int_a^b 1^2\,dx\right]^{1/2} = \frac{1}{\sqrt{b-a}} \quad \text{n.b. } ||q(x)|| \equiv \left[\int_a^b (q(x))^2\,dx\right]^{1/2}$$

For $j = 1, 2, \dots$

$$\begin{aligned}\tilde{q}_j(x) &= xq_{j-1}(x) - \sum_{i=0}^{j-1} \langle\, xq_{j-1}(x), q_i(x)\,\rangle\, q_i(x) \\ q_j(x) &= \tilde{q}_j(x) \,/\, ||\tilde{q}_j(x)||\end{aligned}$$

Observation: $q_{j-1}(x)$ is orthogonal to all polynomials of degree $j-2$ or less.

$$\langle\, xq_{j-1}(x), q_i(x)\,\rangle = \langle\, q_{j-1}(x), xq_i(x)\,\rangle = 0,\ i \le j-3$$

$$\begin{aligned}\Rightarrow \tilde{q}_j(x) = xq_{j-1}(x) &- \langle\, xq_{j-1}(x), q_{j-1}(x)\,\rangle\, q_{j-1}(x) \\ &- \langle\, xq_{j-1}(x), q_{j-2}(x)\,\rangle\, q_{j-2}(x)\end{aligned}$$

Alas, we obtain a recursion of three terms!

Quadrature points are zeros of orthogonal polynomials:

Theorem

Let $x_0, x_1, \dots, x_n$ be the zeros of the orthogonal polynomial $q_{n+1}(x)$ over the interval $[a, b]$, in which case

$$\int_a^b f(x)\,dx \simeq \sum_{i=0}^{n} A_i f(x_i)\,,$$

$$\text{where } A_i = \int_a^b \varphi_i(x)\,dx\,,\ \varphi_i(x) = \prod_{\substack{j=0 \\ i\ne j}}^{n} \frac{x-x_j}{x_i-x_i}\,,$$

is accurate for all polynomials of degree $2n+1$ or less.

Proof

Let f be a polynomial of degree $2n+1$ or less. When f is divided by q_{n+1}, the remainder is of degree n or less. The division algorithm:

$$f = q_{n+1} p_n + r_n \text{ and } f(x_i) = r_n(x_i)\,,\; q_{n+1}(x_i) = 0\,.$$

Let us integrate:

$$\begin{aligned}\int_a^b f(x)\,dx &= \int_a^b q_{n+1}(x) p_n(x)\,dx + \int_a^b r_n(x)\,dx \\ &= \int_a^b r_n(x)\,dx\,,\text{ because } \langle\, q_{n+1}(x), p_n(x)\,\rangle = 0 \\ &= \sum_{i=0}^{n} A_i r_n(x_i) = \sum_{i=0}^{n} A_i f(x_i) \qquad \square\end{aligned}$$

Definition Weighted orthogonal polynomials

Let us define the inner product

$$\langle p, q\rangle_w = \int_a^b p(x) q(x) w(x)\,dx\,,$$

where $w(x)$ is a positive weight function.

Theorem

Building on our previous Theorem (q_{n+1} w-orthogonal):

$$\int_a^b f(x) w(x)\,dx \simeq \sum_{i=0}^{n} A_i f(x_i)\,,\text{where } A_i = \int_a^b \varphi_i(x) w(x)\,dx\,.$$

Once again, accurate for polynomials of degree $2n+1$ or less.

Example Gaussian quadrature: $x \in [-1, 1]$, $n = 1$

The zeros do not require normalization.
Basis: $\{1, x, x^2\}$

$$\begin{aligned}\text{Gram-Schmidt:}\quad \tilde{q}_0 &= 1 \\ \tilde{q}_1 &= x - \frac{\langle\, x, 1\,\rangle}{\langle\, 1, 1\,\rangle} \cdot 1 = x - \frac{\int_{-1}^{1} x\,dx}{\int_{-1}^{1} 1\,dx} \cdot 1 = x \\ \tilde{q}_2 &= x^2 - \frac{\langle\, x^2, 1\,\rangle}{\langle\, 1, 1\,\rangle} \cdot 1 - \frac{\langle\, x^2, x\,\rangle}{\langle\, x, x\,\rangle} x = x^2 - \frac{1}{3}\end{aligned}$$

The roots of $\tilde{q}_2$: $\pm\frac{1}{\sqrt{3}}$

Thus, the formula is: $$\int_{-1}^{1} f(x)\,dx \simeq A_0 f\left(-\frac{1}{\sqrt{3}}\right) + A_1 f\left(\frac{1}{\sqrt{3}}\right)$$

This is accurate all the way up to x^3.

Initial Value Problems

General problem:

$$\begin{cases} y'(t) = f(t, y(t)) & t \geq t_0 \\ y(t_0) = y_0 \end{cases} \tag{1}$$

Let us assume that we have considered the question of whether or not a solution exists and whether it is the only solution. Let us especially assume that the function f is continuous and Lipschitz continuous in y: for each y_1, y_2, $t \in [a, b]$,

$$|f(t, y_2) - f(t, y_1)| \leq L|y_2 - y_1| \tag{2}$$

,

where L is a constant, $t_0 \in [a, b]$.

The numerical solution approximates the solution curve determined by the initial value. Ordinary methods approximate the solution at time t_{k+1} using the solution at time t_k. Multistep methods use deeper dependence.

Euler's method

Constant step size h; $y_0 = y(t_0)$:

$$y_{k+1} = y_k + hf(t_k, y_k), \quad k = 0, 1, ... \tag{3}$$

We get from one point to another on the solution curve by moving along the tangent line.

Method follows directly from Taylor's theorem:

$$\begin{aligned} y(t_{k+1}) &= y(t_k) + hy'(t_k) + \frac{h^2}{2}y''(\xi_k) \\ &= y(t_k) + hf(t_k, y(t_k)) + \frac{h^2}{2}y''(\xi_k), \quad \xi_k \in [t_k, t_{k+1}] \end{aligned} \tag{4}$$

Types of errors: The truncation error (local) and the global error

Now:

$$\frac{y_{k+1} - y_k}{h} = f(t_k, y_k) \tag{5}$$

.

Inserting the solution $y(t_k)$:

$$\frac{y(t_{k+1}) - y(t_k)}{h} = f(t_k, y(t_k)) + \frac{h}{2} y''(\xi_k) \tag{6}$$

,

where $\frac{h}{2} y''(\xi_k)$ is the local error $O(h)$.

Euler's method is first order.

NB: Often the truncation error is described as $O(h^2)$. Here we are considering an approximation, which has on the left side the approximation of the derivative.

The method is consistent:

$$\lim_{h \to 0} \frac{y(t_{k+1}) - y(t_k)}{h} = y'(t_k) = f(t_k, y(t_k)) \tag{7}$$

The truncation error → o , when $h \to 0$.

What about the global error? At time t_k: $|y(t_k) - y_k| \leq ?$

Convergent method: $\max|y(t_k) - y_k| \to$ o , when $h \to 0$.

Theorem

Let us assume that the general problem is well-posed. Let $T \in [a, b]$, $T > t_0$ and $h = (T - t_0)/N$. Let

$$y_{k+1} = y_k + hf(t_k, y_k), \quad k = 0, 1, ..., N - 1$$

Let us assume that $y_0 \to y(t_0)$, when $h \to 0$. Thus, for every k with $t_k \in [t_0, T]$, $y_k \to y(t_k)$, when $h \to 0$ and $\max_k |y(t_k) - y_k| \to 0$.

Proof: Let us denote $d_j = y(t_j) - y_j$.

Subtracting Taylor and Euler:

$$d_{k+1} = d_k + h[f(t_k, y(t_k)) - f(t_k, y_k)] + \frac{h^2}{2} y''(\xi_k) \tag{8}$$

Lipschitz and $|y''(t)| \leq M$:

$$\begin{aligned} |d_{k+1}| &\leq |d_k| + hL|d_k| + \frac{h^2}{2} M \\ &= (1 + hL)|d_k| + \frac{h^2}{2} M \end{aligned} \tag{9}$$

Generally holds:

$$\begin{aligned} & \gamma_{k+1} \leq (1 + \alpha)\gamma_k + \beta, \quad \alpha > 0, \beta \geq 0, k = 0, 1, ... \\ \Rightarrow \quad & \gamma_n \leq e^{n\alpha} y_0 + \frac{e^{n\alpha} - 1}{\alpha} \beta \end{aligned} \tag{10}$$

Thus,

$$|d_{k+1}| \leq e^{(k+1)hL}|d_0| + \frac{e^{(k+1)hL} - 1}{L}\frac{h}{2}M \tag{11}$$

$kh \leq T - t_0$:

$$\max_k |d_k| \leq e^{L(T-t_0)}|d_0| + \frac{e^{L(T-t_0)-1}}{L}\frac{h}{2}M \tag{12}$$

,

where $|d_0| \to 0$ and $\frac{h}{2}M \to 0$, when $h \to 0$. □

Thus, the global error $O(h)$ is obtained with Euler's method. With the same technique, it is possible to examine the effect of the rounding error. Let us calculate the difference of the floating-point solution and the exact arithmetic (with corresponding denotations)

$$\begin{aligned} |d_{k+1}| &\leq (1 + hL)|d_k| + \delta \\ \Rightarrow |d_{k+1}| &\leq e^{L(T-t_0)}|d_0| + \frac{e^{L(T-t_0)} - 1}{hL}\delta \end{aligned} \tag{13}$$

where $|d_0|$ is the error at the beginning, and the latter term dominates when h is small.

Guideline: Minimize the global error without forgetting the rounding!

Explicit and implicit method

Quadrature:

$$\begin{aligned} y(t+h) &= y(t) + \int_t^{t+h} f(s, y(s))ds \\ &= y(t) + \frac{h}{2}[f(t, y(t)) + f(t+h, y(t+h))] + O(h^3) \end{aligned} \tag{14}$$

leads to the trapezoidal:

$$y_{k+1} = y_k + \frac{h}{2}[f(t_k, y_k) + f(t_{k+1}, y_{k+1})] \tag{15}$$

The method is implicit: y_{k+1} must be solved at every step using some solution method. Euler's method is explicit: y_{k+1} is obtained by addition; y_{k+1} appears only on one side of the equation.

Idea: Predict and correct.

Heun's method:

$\tilde{y}_{k+\alpha} = y_k + \alpha h f(t_k, y_k)$; prediction

$y_{k+1} = y_k + \beta h f(t_k, y_k) + \gamma h f(t_k + \alpha h, \tilde{y}_{k+\alpha})$; correction

Three parameters: α, β, $\gamma \Rightarrow$ Let us fit them in Taylor's theorem.

Heun: $\alpha = 1$, $\beta = \gamma = \frac{1}{2}$

Generally: $\beta + \gamma = 1$, $\alpha\gamma = \frac{1}{2}$

For all methods like this, the truncation error is $O(h^2)$.

Synthesis

Synthesis: $y_{k+1} = y_k + h\Psi(t_k, y_k, h)$

a) consistency: $\lim_{h\to 0} \Psi(t, y, h) = f(t, y)$

b) stability: If there exists a constant K and a step size $h_0 > 0$ such that $|y_n - \tilde{y}_n| \le K|y_0 - \tilde{y}_0|$, where $y_n, \tilde{y}_n$ and $y_0, \tilde{y}_0$ are initial conditions, which holds when $h \le h_0$ and $nh \le T - t_0$, the method is stable.

c) a) & b) $\Rightarrow$ the method is convergent

If the truncation error is of the following form

$$\tau(t, h) = \frac{y(t+h) - y(t)}{h} - \Psi(t, y(t), h)$$

the global error of the stable method, which has the truncation error $O(h^p)$, is $O(h^p)$.

NB! The proof is similar to the one shown for Euler's method.

$$\gamma_{k+1} \le (1+\alpha)\gamma_k + \beta \quad \Rightarrow \quad \gamma_n \le e^{n\alpha}\gamma_0 + \frac{e^{n\alpha}-1}{\alpha}\beta$$

$$\begin{aligned}
\gamma_n &\le (1+\alpha)^2\gamma_{n-2} + [(1+\alpha)+1]\beta \\
&\le (1+\alpha)^n\gamma_0 + [\sum_{j=0}^{n-1}(1+\alpha)^j]\beta \\
&= (1+\alpha)^n\gamma_0 + \frac{(1+\alpha)^n - 1}{\alpha}\beta
\end{aligned}$$

$$(1+\alpha) \le e^\alpha = 1 + \alpha + \frac{\alpha^2}{2}e^\xi, \quad \xi \in (0, \alpha)$$

For Euler systems:

$$\boldsymbol{y}' = f(t, \boldsymbol{y}), \quad \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \quad \Rightarrow \quad \boldsymbol{y}_{k+1} = \boldsymbol{y}_k + h\boldsymbol{f}(t_k, \boldsymbol{y}_k)$$

Components: $y_{i,k+1} = y_{ik} + hf_i(t_k, y_{1k}, \ldots, y_{nk}), \quad i = 1, \ldots, n$

Multistep methods

Let us consider (once again) the integral

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(s, y(s)))\, ds$$

Idea: Let us replace the f(t,y) with a suitable interpolation polynomial, which takes the solution history into consideration.

If t_{k+1} is taken into account, the method is implicit.

Adams-Bashforth: Explicit

Let us interpolate at points t_k, t_{k+1},...,t_{k-m+1}; $p_{m-1}(s)$

$$y_{k+1} = y_k + \int_{t_k}^{t_{k+1}} p_{m-1}(s)\, ds = y_k + h\sum_{l=0}^{m-1} b_l\, f(t_{k-l}, y_{k-l}) \tag{16}$$

where

$$b_l = \frac{1}{h}\int_{t_k}^{t_{k+1}} \left(\prod_{\substack{j=0 \\ j\neq l}}^{m-1} \frac{s - t_{k-j}}{t_{k-l} - t_{k-j}} \right) ds$$

If m = 1, the Euler's method is obtained!

Lecture exercise: What kind of method is obtained, when m = 2?

$$y_{k+1} = y_k + h[\frac{3}{2}f(t_k, y_k) - \frac{1}{2}f(t_{k-1}, y_{k-1})]$$

The truncation error is $O(h^m)$. (The error of the integral is $O(h^{m+1})$.)

Adams-Moulton: Implicit

Let us consider the point t_{k+1} as well; $q_m(s)$.

$$y_{k+1} = y_k + h\sum_{l=0}^{m} c_l f(t_{k+1-l}, y_{k+1-l} \tag{17}$$

where

$$c_l = \frac{1}{h}\int_{t_k}^{t_{k+1}} \left(\prod_{\substack{j=0 \\ j\neq l}}^{m} \frac{s - t_{k+1-j}}{t_{k+1-l} - t_{k+1-j}} \right) ds$$

If m = 0, we obtain $y_{k+1} = y_k + hf(t_{k+1}, y_{k+1})$, which is so called implicit Euler's method.

Lecture exercise: What kind of method is obtained, when m = 1?

$$y_{k+1} = y_k + \frac{h}{2}[f(t_{k+1}, y_{k+1}) + f(t_k, y_k)]$$

which is a trapezoidal!

The truncation error is $O(h^{m+1})$.

General format: $\sum_{l=0}^{m} a_l y_{k+1} = h\sum_{l=0}^{m} b_l f(t_{k+l}, y_{k+l})$,

a_m = 1, b_m = 0 ⇒ explicit, otherwise implicit

The high order of the truncation error does not implicate stability!

$$y_{k+1} - 3y_{k+1} + 2y_k = h[\frac{13}{12}f(t_{k+2}, y_{k+2}) - \frac{5}{3}f(t_{k+1}, y_{k+1}) - \frac{5}{12}f(t_k, y_k)]$$

Exercise: y' = 0, y(0) = 1

$y_1 = 1 + \delta$

$y_2 = 3y_1 - 2y_0 = 1 + 3\delta$

...

$y_k = 3y_{k-1} - 2y_{k-2} = 1 + (2^k - 1)\delta$

$\delta \sim 2^{-53} \Rightarrow k = 100$ gives us an error $\sim 2^{47}$ (!)

Stiff Equations

Problem: The solution contains different time scales.

Example:

$$\begin{cases} y_1' = -100y_1 & +y_2 \\ y_2' = & -\frac{1}{10}y_2 \end{cases} \Leftrightarrow y' = Ay, \quad A = \begin{bmatrix} -100 & 1 \\ 0 & -\frac{1}{10} \end{bmatrix}$$

Solution:

$$\begin{cases} y_1(t) = e^{-100t}(y_1(0) - \frac{10}{999}y_2(0)) + e^{-\frac{t}{10}}\frac{10}{999}y_2(0) \\ y_2(t) = e^{-\frac{t}{10}}y_2(0) \end{cases} \tag{18}$$

Question: Could the problem be solved by using Euler's method? Could the step size be chosen arbitrarily? (All gucci, if h → 0.)

Component 2:

$$\begin{aligned} y_{2,k+1} &= (1 - \frac{h}{10})y_{2,k} \\ \Rightarrow \quad y_{2,k} &= (1 - \frac{h}{10})^k y_2(0) \end{aligned} \tag{19}$$

Component 1:

$$\begin{aligned} y_{1,k+1} &= (1-100h)y_{1,k} + hy_{2,k} \\ &= (1-100h)y_{1,k} + h(1-\frac{h}{10})^k y_2(0) \\ &= (1-100h)^2 y_{1,k-1} + h[(1-100h)(1-\frac{h}{10})^{k-1} + (1-\frac{h}{10})^k]y_2(0) \\ &\dots \\ &= (1-100h)^{k+1}y_1(0) + h(1-\frac{h}{10})^k[\sum_{l=0}^{k}(\frac{1-100h}{1-\frac{h}{10}})^l]y_2(0) \end{aligned} \tag{20}$$

which leads us to:

$$y_{1,k+1} = (1-100h)^{k+1}[y_1(0) - \frac{10}{999}y_2(0)] + (1-\frac{h}{10})^{k+1}\frac{10}{999}y_2(0) \tag{21}$$

We notice immediately that if $h > \frac{1}{50}$, then $|1-100h| > 1$ and $(1-100h)^{k+1}$ grows geometrically. Even if the initial conditions guaranteed that $y_1(0) - \frac{10}{999}y_2(0) = 0$, the rounding error grows unbounded.

In that case, Euler's method is unstable, when $h > \frac{1}{50}$.

Absolute stability

General problem: $y' = \lambda y \quad \Rightarrow \quad y = e^{\lambda t}y(0), \quad \lambda \in \mathbb{C}$

We know that $y(t) \to 0$, when $t \to \infty$ only if Re λ < 0.
System: $y' = Ay$; A is $n \times n$-matrix
Let us assume that A is diagonalizable.

$$A = V\Lambda V^{-1}$$

where Λ is a diagonal matrix of eigenvalues and the columns of V are eigenvectors.

With a variable change $\tilde{y} = V^{-1}y$, we obtain:

$$\tilde{y}' = \Lambda\tilde{y} \quad \text{thus} \quad \tilde{y}_i = \lambda_i\tilde{y}_i, \quad i = 1, ..., n$$

Modified system converges in modified coordinates, which is not always simple to interpret.

However, the next definition is reasonable:

Definition: The region of absolute stability is the set $\{h\lambda \in \mathbb{C} \mid y_k \to 0, \text{ when } k \to \infty\}$, where y_k is the solution of the general problem and h is a constant step size, h > 0.

Definiton: A-stability

A method is A-stable if its region of absolute stability contains entire left half plane.

NB: On the region of absolute stability, it holds that if $z_{k+1} = (1 + h\lambda)z_k, \quad z_k \neq y_k$, then

$$\begin{aligned} & z_{k+1} - y_{k+1} = (1 + h\lambda)(z_k - y_k) \\ \Rightarrow \quad & |z_{k+1} - y_{k+1}| \leq |z_k - y_k| \end{aligned} \tag{22}$$

Example: The backward Euler method

$$y_{k+1} = y_k + h\lambda y_{k+1} \Rightarrow y_{k+1} = \frac{1}{1 - h\lambda}y_k = ... = \frac{1}{(1 - h\lambda)^{k+1}}y_0 \tag{23}$$

Absolute stability: $\{h\lambda \mid |1 - h\lambda| > 1\}$

$$|1 - h\lambda| = \sqrt{(1 - h\,Re\lambda)^2 + (Im\lambda)^2} > 1, \quad \text{when}\, Re\lambda < 0 \tag{24}$$

The backward Euler method is A-stable.

One can prove that there are no explicit A-stable linear multistep methods.

Theorem: The highest order of an A-stable multispe method is two.

Depressing. Nevertheless, it is possible to form a high-order methods with the region of absolute stability "almost" the entire left half plane.

Particular methods

BDF-methods: Backward Differentiation Formulas

m-step method with m-order: $\sum_{l=0}^{m} a_l y_{k+1} = hb_m f(t_{k+m}, y_{k+m}$

All implicit.

Theorem: The truncation error of a multistep method is of order $p \geq 1$, if and only if

$$\sum_{l=0}^{m} a_l = 0 \,\text{and}\, \sum_{l=0}^{m} l^j a_l = j \sum_{l=0}^{m} l^{j-1} b_l,\ j = 1, ..., p$$

With this theorem, let us choose suitable coefficients:

m = 1: $a_0 + a_1 = 0$, $0 \cdot a_0 + 1 \cdot a_1 = b_1$

Let us (always) choose $a_1 = 1 \Rightarrow a_0 = -1$, $b_1 = 1$

Thus, we obtain: $y_{k+1} = y_k + hf(t_{k+1}, y_{k+1})$, which is the backward Euler.

We can continue this way, but when m = 3, the obtained method cannot be A-stable.

IRK-methods: Implicit Runge-Kutta

$$\xi_j = y_k + h \sum_{i=1}^{v} a_{ji} f(t_k + c_i h, \xi_i),\ j = 1, ..., v \tag{25}$$

$$y_{k+1} = y_k + h \sum_{j=1}^{v} b_j f(t_k + c_j h, \xi_j) \tag{26}$$

Arbitrary parameters: $a_j i$, b_j, c_j

Consistent: $\sum_{i=1}^{v} a_{ji} = c_j$, $j = 1, ..., v$

For every $v \leq 1$, there is a unique A-stable IRK method of order 2v.

Implicit systems

Multistep methods: $b_m \neq 0$

$$\boldsymbol{y}_{k+m} = hb_m \boldsymbol{f}(t_{k+m}, \boldsymbol{y}_{k+m}) + \gamma, \tag{27}$$

$$\text{where} \quad \gamma = h \sum_{l=0} *m - 1 b_l \boldsymbol{f}(t_{k+l}, \boldsymbol{y}_{k+l}) - \sum\bar{}l = 0^{m-1} a_l \boldsymbol{y}_{k+l} \tag{28}$$

is known.

IRK:

$$\begin{bmatrix} \boldsymbol{\xi}_1 \\ ... \\ \boldsymbol{\xi}_v \\ \boldsymbol{y}_{k+1} \end{bmatrix} = h \begin{bmatrix} \sum_{i=1}^{v} a_{1i} \boldsymbol{f}(t_k + c_i h, \boldsymbol{\xi}_i) \\ ... \\ \sum_{i=1}^{v} a_{vi} \boldsymbol{f}(t_k + c_i h, \boldsymbol{\xi}_i) \\ \sum_{j=1}^{v} b_i \boldsymbol{f}(t_k + c_j h, \boldsymbol{\xi}_j) \end{bmatrix} + \begin{bmatrix} \boldsymbol{y}_k \\ ... \\ \boldsymbol{y}_k \\ \boldsymbol{y}_k \end{bmatrix} \tag{29}$$

General format:

$$w = hg(w) + \gamma\,; \tag{30}$$
$$q(w) \equiv w - hg - \gamma = \mathbf{0} \tag{31}$$

Newton's method:
An initial guess $w^{(0)}$; Taylor's theorem for q in $w^{(0)}$:

$$\begin{bmatrix} q_1(w_1, ..., w_n) \\ ... \\ q_n(w_1, ..., w_n) \end{bmatrix} = \begin{bmatrix} q_1(w_1^{(0)}, ..., w_n^{(0)}) \\ ... \\ q_n(w_1^{(0)}, ..., w_n^{(0)}) \end{bmatrix} + \begin{bmatrix} \sum_{i=1}^{n} \frac{\partial q_1}{\partial w_i}(w^{(0)})(w_i - w_i^{(0)}) \\ ... \\ \sum_{i=1}^{n} \frac{\partial q_n}{\partial w_i}(w^{(0)})(w_i - w_i^{(0)}) \end{bmatrix} + \begin{bmatrix} O(\left|\left|w - w^{(0)}\right|\right|^2) \\ ... \\ O(\left|\left|w - w^{(0)}\right|\right|^2) \end{bmatrix} \tag{32}$$

or in the matrix format

$$q(w) = q(w^{(0)}) + J_q(w^{(0)})(w - w^{(0)}) + O(\left|\left|w - w^{(0)}\right|\right|^2) \tag{33}$$

where $J_q(w^{(0)})$ is the Jacobian evaluated at $w^{(0)}$.
Let us drop the quadratic term and solve $q(w) = 0$:

$$w^{(1)} = w^{(0)} - [J_q(w^{(0)})]^{-1} q(w^{(0)}) \tag{34}$$

We have obtained a step of Newton's method.
Observations:
a) Rotating the matrix means solving the system of equations.
b) The Jacobian must be non-singular.
c) The initial guess has to be good enough.
In this context: $q(w) = 0$; we obtain

$$w^{(j+1)} = w^{(j)} - [I - hJ_g(w^{(j)})]^{-1}(w^{(j)} - hg(w^{(j)})) - \gamma \tag{35}$$

where $[I - hJ_g(w^{(j)})]^{-1}$ is non-singular when h is sufficiently small.
Interpretation: The error of the predictor step can be around O(h).