Numerical Maths

isakhammer

A20

Contents

1 Lecture 1

1.1 Practical Information

- Brynjulf Owren, room 1350, Sentralbygg 2, brynjulf.owren@ntnu.no
- Alvar Lindell, room 1201, Sentralbygg 2, alvar.lindell@ntnu.no

There will be a total of 6 assignment where 4 should be approved. It should be delivered in blackboard as a jupyter notebook file including some control questions.

- **Project 1** It counts 10 procent on the final grade, relativaly small work, but somewhat large assignment. Every student submits her own separate .ipynb file. Discuss problem if you like, but make your own write-up. Likely to be a topic of algebra. Deadline. 10-15 September.
- **Project 2** Counts 20 procent on the final grade. Group project 1-3 students. Numerical ODE and may some optimization.

Lecture contents of the course

- Introduction 3.6%
- Numerical linear algebra 21.4%
- Numerical ODE 28.6%
- Nonlinear Systems and Numerical Optization 7.1%

May be jupyter programming on the exam.

1.2 M2 Basic Linear Algebra

1.2.1 Background summary

Vectors. Most of the time we think of vectors as n-plets of real numbers.

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Vecotrs are columns vectors if row vectors are needed use.

$$v^T = \begin{bmatrix} v_1 & v_2 & v_3 & \dots & v_n \end{bmatrix}$$

Linear Transformations are given by $A : \mathbb{R}^n \to \mathbb{R}^m$. These are represented ass $m \times n$ matrices. $A = ((a_{ij}))$ such that $1 \leq i \leq m$ and $1 \leq j \leq n$. Notation $A \in \mathbb{R}^{m \times n}$

$$(Av)_i = \sum_{j=1}^{n} a_{ij} v_j, \quad i = 1, \dots, m.$$

If $A = ((a_{ij}))$, B $((b_{ij}))$ then $A + B = C$, $C = ((c_{ij}))$, $c_{ij} = a_{ij} + b_{ij}$.
Given to matrices, $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{k \times n}$

$$\mathbb{R}^n \to \mathbb{R}^k \to \mathbb{R}^m$$

$$\mathbb{R}^n \to \mathbb{R}^m$$

$$(A \cdot B)_{ij} = \sum_{r=1}^{k} a_{ir} b_{ri}$$

Fix a way to have notation on top of arrow and a better snippet for the summation. Might also train making quick vector notations.

1.2.2 Linear Independence

Let assume that we have $v_1, \ldots, v_k$ be vectors in $\mathbb{R}^n$ and let $\alpha_1, \alpha_2, \ldots, \alpha_k$ be scalar if

$$\sum_{n=1}^{k} \alpha_i v_i = 0 \quad \text{then is} \quad \alpha_1 = \alpha_2 = \ldots = 0$$

Then $v_1, v_2, \ldots, v_k$ is linear independent.

1.2.3 Inverse of an matrix

If there is a matrix $B \in \mathbb{R}^{n \times n}$ such that

$$A \cdot B = B \cdot A = I$$

Then B is the inverse of A. B is denoted $B = A^{-1}$ Basis of $\mathbb{R}^n$. Any set of n linearly independent vectors in $\mathbb{R}^n$ is called a basis.

1.2.4 Permutation Matrix

Permuation Matrix. Let $I \in \mathbb{R}^{n \times n}$ be the identity matrix. I has columns $e_1, e_2, \ldots, e_n$ where e_i is the i-th canonical unit vector

$$\begin{bmatrix} 0 & 0 & \ldots 1 \ldots 0 \end{bmatrix} = e^T$$

Let $p = \begin{bmatrix} i_1, i_2, \ldots, i_n \end{bmatrix}^T$ Be a permutation of the set $\{1, \ldots, n\}$ then

$$P = \begin{bmatrix} e_1 & e_2 & e_2 \end{bmatrix}$$

The permutation matrix.

Implement example snippet

The inverse of a permutation matrix in $P^{-1} = P^T$ and $\left(P^{-1}\right)_{ij} = P_{ji}$.

1.2.5 Types of Matrices

- Symmetric: $A^T = A$
- Skew symmetric: $A^T = -A$
- Orthogonal. $A^T A = I$

2 Lecture 3 - August 25 - 2020

2.1 Continuation of previous lecture

Lets find a practical computation of $p^{(0)}, p^{(1)}, \ldots$. Always start with $p^{(0)} = r^{(0)} = b - Ax^{(0)}$. Suppose that $p^{(0)}, \ldots, p^{(k)}$ have been found. Set $p^{(k+1)} = r^{(k+1)} - b_k p^{(k)}$. Require that

$$0 = \left\langle p^{(k)}, p^{(k+1)} \right\rangle_A = \left\langle p^{(k)}, r^{(k+1)} \right\rangle - \beta_k \left\langle p^{(k)}, p^{(k)} \right\rangle$$

$$\text{so} \quad \beta_k = \frac{\left\langle p^{(k)}, r^{(k+1)} \right\rangle_A}{\left\langle p^{(k)}, p^{(k)} \right\rangle_A}$$

Note that $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$ and

$$b - Ax^{(k+1)} = b - Ax^{(k)} - \alpha_k A p^{(k)}$$

$$\underbrace{r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}}_{\text{essential}}$$

Let $V_k = span\left\{p^{(0)}, \ldots, p^{(k)}\right\}$ and since $r^{(0)} = p^{(s)}, \quad r^{(k+1)} = p^{(k+1)} - \alpha_k A p^{(k)}$, it happens that $Ap^{(k)} \in V_{k+1}$, we have

$$V_k = span\left\{r^{(0)}, \ldots, r^{(k)}\right\}$$

We want to prove that $\left\langle p^{(k+1)}, p^{(j)} \right\rangle = 0$ for $j = 0, \ldots, k-1$

$$\left\langle r^{(k+1)} - \beta_k p^{(k)}, p^{(j)} \right\rangle_A = \left\langle r^{(k+1)}, p^{(j)} \right\rangle - \beta_k \left\langle p^{(k)}, p^{(j)} \right\rangle_A$$

We know that

$$Ap^{(j)} \in V_{j+1}, \quad Ap^{(j)} = \sum_{e=0}^{j+1} c_e p^{(e)}$$

$$\left\langle r^{(k+1)}, p^{(j)} \right\rangle_A = \sum_{e=0}^{j=1} \left\langle r^{(k+1)}, c_e p^{(e)} \right\rangle$$

Chosing the search directions like this is corresponding to the Conjugate gradient method.

2.2 Conjugate Gradient Method Algorithm

$$
\begin{aligned}
&x^{(0)} \quad \text{is given} \\
&\quad r^{(0)} = b - A \cdot x^{(0)} \\
&\quad p^{(s)} = r^{(s)} \\
&\text{For } k = 0, 1, 2, \ldots \\
&\qquad \begin{cases}
\alpha & = \frac{p^{(k)T} r^{(k)}}{p^{(k)T} A p^{(k)}} \\
x^{(k+1)} & = x^{(k)} + \alpha p^{(k)} \\
r^{(k+1)} & = r^{(k)} - \alpha_k A p^{(k)} \\
\beta_k & = \frac{\left(A p^{(k)}\right)^T r^{(k+1)}}{\left(A p^{(k)}\right)^T p^{(k)}} \\
p^{(k+1)} & = r^{(k+1)} - \beta_k p^{(k)}
\end{cases} .
\end{aligned}
$$

2.3 Simplification

We want to simplify the expression for α_k and β_k

$$
\begin{aligned}
p^{(k+1)} &= r^{(k+1)} - \beta_k p^{(k)} \\
p^{(k)} &= r^{(k)} - \beta_{k-1} p^{(k-1)} \quad \Longrightarrow \text{ multiply } \quad r^{(k)T} \\
r^{(k)T} p^{(k)} &= \|r^{(k)}\|_2^2 - \beta_{k-1} r^{(k)T} p^{(k-1)} \\
\text{So} \quad \alpha_k &= \frac{\|r^{(k)}\|_2^2}{p^{(k)} A p^{(k)}} \\
r^{(k+1)T} p^{(k+1)} &= \|r^{(k+1)}\|^2 - \beta_k r^{(k+1)T} p^{(k)} \\
r^{(k)T} p^{(k+1)} &= -\beta_k r^{(k)T} p^{(k)} = -\beta_k \|r^{(k)}\|^2 = \|r^{(k+1)}\|^2
\end{aligned}
$$

In the end is the results

$$
\beta_k = -\frac{\|r^{(k)}\|^2}{\|r^{(k+1)}\|^2} \quad \text{and} \quad \alpha_k = \frac{\|r^{(k)}\|_2^2}{p^{(k)} A p^{(k)}}
$$

2.4 Modified algorithm

$$
\begin{aligned}
p^{(0)} &= r^{(0)} \\
r_l &= \|r^{(0)}\|^2 \\
\text{For} \quad & k = 0, 1, 2, \ldots
\end{aligned}
$$

$$
\begin{cases}
v & = Ap^{(k)} & \\
t & = p^{(k)T} v & \rightarrow \text{saxpy} \\
\alpha_k & = \frac{r_l}{t} & \rightarrow \text{saxpy} \\
x^{(k+1)} & = x^{(k)} + \alpha_k p^{(k)} & \rightarrow \text{inner product} \\
r_c & = \|r^{(k+1)}\|^2 & \rightarrow \text{saxpy} \\
p^{(k+1)} & = r^{(k+1)} + \frac{rc}{r_l} p^{(k)} & \rightarrow \text{saxpy} \\
r_l & \leftarrow r_c &
\end{cases}
$$

Operations done in the numerical method

$$
A \times \text{vector} \quad \left[B\left(h^2\right) \quad \text{for full matrices} \right]
$$

2.5 Convergence of the Conjugate Gradient Algorithm

Since all search directions are mutually A- orthogonal. theu are linearly indepedent. After n iterations, they span all of $\mathbb{R}^n$.. Since the residuals $r^{(n)}$ is orthogonal to all of $r^{(0)}, \ldots, r^{(n-1)}$ must be 0 and therefore

$$
0 = r^{(0)n} = b - Ax^{(n)} \rightarrow Ax^{(n)} = b
$$

But then the algorithm is only competitive when it terminates in $k \ll n$ iterations with a sufficient accurate solution.

Theorem 2.1. *Let A be SPD. The error after k iterations is bounded as*

$$
\|e^{(j)}\|_A \leq \frac{2c^k}{1 + c^{2k}}, \quad c = \frac{\sqrt{K_2(A)} - 1}{\sqrt{K_2(A)} + 1}
$$

Remark. $\|v\|_A = \sqrt{v^T A v}$

2.6 Next Lecture Hint

Next lecture will be about precondition. We solve $Ax = b$. An equivalent formulation is to pick an invertible P and solve

$$
\begin{aligned}
P^{-1}Ax &= xP^{-1} \\
\hat{A}x &= \hat{b}
\end{aligned}
$$

Criteria

1. Let P approximate A
2. Should be cheap to solve systems

$$P^{-1}y = c$$

3 Lecture 01/09/20

Well posedness of the inital value problem.

$$\dot{y} = f(t, y), \quad y(0) = y_0$$

Is stable on $[0, T]$ if for any sufficiently small $\varepsilon > 0$ m there are $(\delta_0, \delta(t))$ s.t.

$$\|\delta_0\| < \varepsilon, \quad \|\delta(t)\| < \varepsilon, \quad t \in [0, T]$$

Such that $\|y(t) - z(t)\| < C \cdot \varepsilon \quad \forall t \in [0, T]$ for some constant C. $z(t)$ solves the IVP

$$\dot{z}(t) = f(t, z(t)) + \delta(t), \quad z_0 = y_0 + \delta_0$$

One can prove that if f is Lipschitz (constant L), then the solution is stable $C = (1 + T)e^{tL}$.

3.1 Flow of a vector field

For us a vector field is a continuous map

$$f : \mathbb{R}^m \to \mathbb{R}^m$$

For a fixed value of t consider the map $\phi_{t,f}(y_0) = y(t)$ is called the t-flow of the vector field f. Its domain of definition may depend on t and f.

Suppose that

$$\phi_{t_1,f}(\phi_{t_2,f}(y_0)) = \phi_{t_1,t_2,f}(y_0)$$

and

$$\phi_{-t_1,f}(\phi_{t_1,f}(y_0)) = \phi_{0,f}(y_0) = y_0 \quad \Longrightarrow \quad (\phi_{t,f})^{-1} = \phi_{-t,f}$$

Typical notation:

$$\phi_{t,f}(y_0) = e^{tf} y_0$$

3.2 Numerical Integration of ODE

Always assume a finite time interval $[0, T]$.

Vector field (f) does not contain parameters. Split the finite interval into subintervals

$$t_0 < t_1 < \ldots < t^N = T$$

Often we assume $t_n = t_0 + nh$, where h is the stepsize. Also we may have $h = \frac{T - t_0}{N}$. For $t = t_j$, let $u_j \approx y(t_j) = y_j$ and $f_j f(t_j, y_j)$

Two classes of methods

1. One-step method , $u_{n+1} = \chi_{n+1}(u_n)$
2. Multistep methods, u_{n+1} depends on $u_n, u_{n-1}, \ldots, u_{nk+1}$ and also $f_{n+1}, f_n, \ldots, f_{n-k+1}$

3.2.1 The Simplest Schemes

1. **Euler** , $u_{n+1} = u_n + hf(t_n, u_n)$
2. **Backward Euler (Implicit Euler)**
$$u_{n+1} = u_n + hf(t_{n+1}, u_{n+1})$$
3. **Trapezoid rule** .
$$u_{n+1} = u_n + \frac{h}{2}(f())$$
4. Midpoint rule
$$u_{n+1} = u_n 0 h f\left(t_n + \frac{h}{2}, \frac{y_n + y_{n+1}}{2}\right)$$

3.2.2 The most important classes of one-step schemes

Taylor Series methods.

$$\dot{y} = f(t, y), \quad y(0) = y_0, \quad y(t) \in \mathbb{R}^m$$

If $y(t)$ is sufficiently smooth then we can compute the taylor expansion such that

$$y(t+h) = y(t) h\dot{y}(t) + \frac{1}{2}h^2\ddot{y}(t) + \ldots + \frac{1}{q!}h^q y^{(q)} + R_{q+1}.$$

where

$$R_{q+1} = \frac{y^{(q+1)}(\zeta)}{(q+1)!}, \quad \zeta \in (t, t+h)$$

Compute $y^{(k)}(t)$, $k > 1$ from diff eq.

$$\begin{aligned}\ddot{y}(t) &= \frac{d}{dt}f(t, y(t)) = \frac{\partial}{\partial t}f(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t))\dot{y}(t) \\ &= \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \cdot f(t, y(t))\end{aligned}$$

Similary we can compute

$$y^{(3)}(t), y^{(4)}(t), \ldots$$

and plug this into the taylor expansion and ignore the remainder term.

Example. Let $\dot{y} = y^2$ and $y(0) = y_0$ such that

$$\ddot{y} = 2y\dot{y} = 2y^3, \quad y^{(3)} = 6y^2\dot{y} = 6y^4, \quad y^{(k)} = k!y^{(k+1)}$$

This can be plugged into the taylor expansion such that

$$y(t+h) = y(t) + hy(t)^2 + \frac{1}{2}h^2 2y(t)^2 + \ldots = y(t) + hy(t)^2 + \ldots$$

which ends up with the method

$$y_{n+1} = \sum_{k=1}^{q} h^{k-1} y_n^k$$

3.2.3 Runga kutta methods

Generalization class of methods. COntains Euler, Backwards Euler, Trapozoidal Rule, Midpoint and more. The format is described like this

$$K_i = f\left(t_n + c_i h, u_n + \sum_{j=1}^{s} a_{ij} K_j\right) \quad i = 1, \ldots, s$$

$$u_{n+1} = u_n + h \sum_{i=1}^{s} b_i K_i$$

- **Explicit RK Methods** . $a_{ij} = 0, \quad j \geq i$
- **Butcher Tablaeuz**

Table 1: Butcher Tablaeuz

c_1	a_{11}	$\ldots$	a_{1s}
$\vdots$			
c_s	a_{s1}	$\ldots$	a_{ss}
	b_1	$\ldots$	b_s

3.3 Analysis of one-step methods

The exact solution does not obey the numerical formula. We write the formula as

$$u_{n+1} = u_n + h\phi_{h,f}\left(t_n, u_n\right)$$
$$y_{n+1} = y_n + h\phi_{h,f}\left(t_n, y_n\right) + \varepsilon_{n+1}$$

We set $\varepsilon_{n+1} = h\tau_{n+1}\left(h\right)$ where $\tau_{n+1}\left(h\right)$ is the local trunction error.

Define also

$$\tau\left(h\right) = \max_n \|\tau_{n+1}\left(h\right)\|$$

A method is called **consistent** if $\lim_{h\to 0} \tau\left(h\right) = 0$. If $\tau\left(h\right) = O\left(h^p\right)$ as $h \to 0$ then ot has order (of consistency) p.

4 Lecture 2020-09-14

Local Error and order conditions

Multistep methods - k-step.

$$\alpha_k u_{n+k} + \alpha_{k-1} u_{n+k} + \ldots + \alpha_0 u_n = h\left(\beta_k f_{n+k} + \ldots + \beta_0 f_n\right) \qquad (1)$$
$$\alpha_0 \text{ or } \beta_0 \neq 0, \alpha_k \neq 0 \quad \text{often } \alpha_k = 1$$

We have the local arror $y_k - u_k$. where we assume $u_j = y_j, j = 0, \ldots, k-1$ (exact input values). For any multistep method define

$$L\left(y, t, h\right) = \sum_{i=0}^{k} \left(\alpha_i y\left(t + ih\right) - h\beta_i y'\left(t + ih\right)\right)$$

From the formula (1) we get

$$\sum_{i=0}^{k-1} \left(\alpha_i y_i - h\beta_i f\left(t_i, y_i\right) + \alpha_k u_k - h\beta_k f\left(t_k, u_k\right)\right) = 0$$
$$L\left(y, t_0, h\right) = \alpha_k\left(y_k - u_k\right) - h\beta_k\left(f\left(t_k, y_k\right) f\left(t_k, u_k\right)\right)$$

By taylor expansion

$$L\left(y, t_0, h\right) = \left(\alpha_k I - h\beta_k \frac{\partial f}{\partial y}\left(t_k, \mu\right)\right)\left(y_k - u_k\right)$$
$$y_k - u_k = \left(\alpha_k I - h\beta_k \frac{\partial f}{\partial y}\left(t_k, \mu\right)\right)^{-1} L\left(y, t_0, h\right)$$
$$\mu = \theta u_k + \left(1 - \theta\right) y_k, \quad \theta \in (0, 1)$$

For the local error : $y_k - u_k = \frac{1}{\alpha_k} L\left(y, t_0, h\right)\left(1 + O\left(h\right)\right)$

4.1 Order of linear multistep methods.

Two equivalent definition of order

- For any sufficuent smooth function $y\left(t\right)$, we have $L\left(y, t, h\right) = O\left(h^{p+1}\right)$
- The local error is $y_k - u_k = U\left(h^{p+1}\right)$.

Definition 4.1. *The two polynomials* $\rho\left(z\right), \sigma\left(z\right)$

$$\rho\left(z\right) = \alpha_k z^k + \alpha_{k+1} z^{k+1} + \ldots + \alpha_0$$
$$\sigma\left(z\right) = \beta_k z^k + \beta_{k-1} z^{z-1} + \ldots + \beta_0$$

Theorem 4.1. *The method has order p if one of the following equivalent conditions is satisfied.*

(i) $\sum_{i=0}^{k} \alpha_i = 0$ *and* $\sum_{i=0}^{k} \alpha_i i^q = q \sum_{i=0}^{k} \beta_i i^{q-1}, \quad q = 1, \ldots, p$

(ii) $\rho\left(e^h\right) - h\sigma\left(e^h\right) = O\left(h^{p+1}\right)$ *as* $h \to 0$

(iii) $\dfrac{\rho(z)}{\ln z} - \sigma(z) = O\left((z-1)^p\right)$ *as* $z \to 1$

Proof. (i)

$$
\begin{aligned}
L(y,t,h) &= \sum_{i=0}^{k} \left(\alpha_i \sum_{q \geq 0} \frac{i^q h^q}{q!} y^{(q)}(t) - h\beta_i \sum_{r \geq 0} \frac{i^r h^r}{r!} y^{(r+1)}(t) \right) \\
&= \sum_{i=0}^{k} \alpha_i y(t) + \sum_{q \geq 1} \frac{h^q}{q!} y^{(q)}(t) \left(\sum_{i=0}^{k} i^q - q \sum_{i=0}^{k} \beta_i i^{q-1} \right) \\
&= O\left(h^{p+1}\right) \quad \to \sum_{i=0}^{k} \alpha_i = 0 \text{ and} \\
&\qquad\qquad \sum_{i=0}^{k} \alpha_i i^q - q \sum_{i=0}^{k} \beta_i i^{q-1} = 0, \quad q = 1, \ldots, p
\end{aligned}
$$

□

Example. 2-step explicit Adams

$$u_{n+2} - u_{n+1} = h\left(\frac{3}{2}f_{n+1}\frac{1}{2}f_n\right)$$

$$\alpha_2 = 1, \quad \alpha_1 = -1, \alpha_0 = 0,, \beta_2 = 0, \beta_1 = \frac{3}{2}, \beta_0 = -\frac{1}{2}$$

- $q = 0$

$$\sum_{i=0}^{2} \alpha_i = 0 - 1 + 1 = 0$$

- $q = 1$

$$\sum_{i=0}^{2} i\alpha_i - 1\sum_{i=0}^{2} \beta_i i^0 = (-1+2) - \left(\frac{3}{2} - \frac{1}{2}\right) = 0$$

- $q = 2$

$$\sum_{i=0}^{2} i^2\alpha_i - 2\sum_{i=0}^{2} i\beta_i = \left[-1 + 2^2 \cdot 1\right] - 2\left[0 \cdot \left(-\frac{1}{2}\right) + 1 \cdot \frac{3}{2}\right] = 0$$

- $q = 3$

$$\sum_{i=0}^{2} i^3\alpha_i - 3\sum_{i=0}^{2} i^2\beta_i = \left[-1 + 2^3 \cdot 1\right] - 3\left[1 \cdot \frac{3}{2}\right] = \frac{5}{2} \neq 0$$

The method has order $p = 2$

Definition 4.2. ***Error constant.*** *We found in the proof*

$$L(y, t, h) = \sum_{q \geq 0} Cqh^q y^{(q)}(t)$$

and $p \implies C_0 = C_1 = \ldots = C_p = 0$, *where* C_{p+1} *is called* ***the error constant*** .

Definition 4.3. ***Consistency.*** *(11.6). A multistep method is consistent if* $p \geq 1$. $C_0 = C_1 = 0$. *Can be formulated as*

$$P(1) = 0, \quad P'(1) = \sigma(1)$$

Usual assumption: $P(z)$ and $\sigma(z)$ have no common factors

$$p(z) = \alpha_k (z - r) \dots (z - r_k), \quad \sigma(z) = \beta_k (z - s_1) \dots (z - s_k)$$

Then $r_i \neq s_j$ for all i, j.

4.2 Difference equations and (zero) stability

Consider the multistep method

$$u_{n+2} + 4u_{n+1} - 5u_n = h(4f_{n+1} + 2f_n)$$

Check order

$$c_0 = \sum \alpha_i = 0$$

$$\begin{aligned} c_1 &= (4 + 2) - (4 + 2) = 0 \\ c_2 &= (4 + 4) - 2(1 \cdot 4) = 0 \\ c_3 &= (4 + 8) - 3 \cdot (1 \cdot 4) = 0 \\ c_4 &= (4 + 16) - 4 \cdot (1 \cdot 4) = 4 \end{aligned}$$

Apply method to the problem $y' = y$ with $y(0) = 1$

$$\begin{aligned} u_{n+2} + 4u_{n+1} - 5u_n &= h(4u_{n+1} + 2u_n) \\ u_{n+2} + 4(1 - h)u_{n+1} - (3 + 2h)u_n &= 0 \end{aligned}. \tag{2}$$

Try to insert solution $y_j = z^j$ in the equation into (2)

$$z^{n+2} + 4(1 - h)z^{n+1} - (3 + 2h)z^n = 0.$$

Ignore $z = 0$

$$z^2 + 4(1 - h)z - (5 + 2h) = 0$$

Two roots, z_1 and z_2. The general solution is

$$u_n = A \cdot z_1(h)^n + Bz_2(h)^n$$

The roots

$$z_{1,2} = -2 + 2h \pm \sqrt{4h^2 - 6h + 9}$$

Leads to $z_1 = 1 + h + O(h^2)$, $z_2 = -5 + 3h + O(h^2)$.

z_1 reflects the true solution, but z_2 is a spurious solution (falsk løsning).

Remark. $|z_2(h)|^n \to \infty$, even for small h.

This method cannot converge even though we have seen that $P = 3$.

Consider $y' = f(t,y) := 0$. In general for a multistep method applied to $y' = 0$ will give you

$$\sum_{j=0}^{k} \alpha_j u_{n+j} = 0$$

Solve by setting $u_i = z^i$

$$\sum_{j=0}^{k} \alpha_j z^j = 0 \quad \text{or} \quad P(z) = 0$$

First consistency condition says $P(1) = 0$. so $z_1 = 1$ is a root. We also get

$$z_2, \ldots, z_k \text{ some may be multiple roots}$$

$$P(z) = (z - \xi_1)^{m_1} (z - \xi_2)^{m_2} \ldots (z - \xi_\mu)^{m_\mu}$$

$$\sum_{i=1}^{\mu} m_i = k$$

The general solution is

$$u_n = p_1(n)\, \xi_1^n + \ldots + P_\mu(n)\, \xi_\mu^n$$

where p_i is a polynomial of degree $m_i - 1$.

Definition 4.4. *The method is zero-stable if* $P(z)$

(i) The roots ξ_i *satisfied* $|\xi_i| \leq 1$

(ii) for root ξ_i *such that* $|\xi_i| = 1$*, then* $m_i = 1$

5 Lecture 2020-09-15

Example. The Adams Methods. We have

$$u_{n+k} - u_{n+k-1} = h \sum_{j=0}^{k} \beta_k f_k$$

Which means

$$\rho(z) = z^k - z^{k-1} = z^{k-1}(z-1)$$
$$\zeta_1 = 1, \quad m_1 = 1, \quad \zeta_2 = 0, \quad m_2 = k-1$$

All Adams methods are zero-stable.
The first Dahlsquist barrier. We have in principle $2k+1$ parameters

$$\alpha_0, \ldots, \alpha_{k-1}, \beta_0, \ldots, \beta_k$$

For order p

$$p+1 \text{ conditions } c_0 = c_1 = \ldots = c_p = 0$$

Should be able to obtain order $2k$.

Curiously, this is not reconcilable with zero-stability.

Theorem 5.1. *The order p of a zero-stable k- step multistep-method satisfies*

$$p \leq k+2, \quad k \text{ is even}$$
$$p \leq k+1, \quad k \text{ is odd}$$
$$p \leq k, \quad \frac{\beta_k}{\alpha_k} \leq 0 \text{ (explicit methods included.)}$$

The proof is rather involved and is not included in this course.

5.1 Convergence of multistep methods.

Assume f is contionious and Lipschitz continuous.

$$y' = f(t, y)$$

on some domain

$$\mathcal{D} = \{(t, y) : \quad t \in [t_0, T], \quad \|y(t) - y\| \leq b\}$$

here is $y(t)$ the exact solution. Applying a method to this problem, results in a sequence $\{u_i\}, i \geq 0$ of approximations. For a fixed $t \in [t_0, T]$ and h such that

$\frac{t-t_0}{h} = n$, define

$$u_h(t) = u_n$$

Definition 5.1. ***Converges.***

(i) The multistep method converges if for all f as defined

$$y(t) - u_n(t) \to 0, \quad t \in [t_0, T]$$

whenever the starting values satisfy

$$y(t_0 + ih) - u_n(t_0 + ih) \to 0, \quad i \in 0, \ldots, k-1$$

(ii) The methods is convergent of order p, isf for any sufficenttly smooth f, there is h_0 such that the behaviour

$$\|y(t) - u_h(t)\| \leq ch^p, \quad 0 < h \leq h_0$$

whenever

$$\|y(t_0 + ih) - u_n(t_0 + ih)\| \leq c_0 h^p, \quad i = 0, \ldots, k-1$$

Theorem 5.2. *A multistep method converges*

$$\downarrow$$

It is zero stable and consistent.

5.2 Backwards difference formulas (BDF)

Idea. We use indices $u_{n-k+1}, \ldots, u_{n+1}$. Construct a polynomial $q(t)$ that interpolates the

$$(t_{n-k+1}, u_{n-k+1}), \ldots, (t_{n+1}, u_{n+1}).$$

Then the set

$$q'(t_{n+1}) \approx f(t_{n+1}, u_{n+1})$$

Keeping in mind that $y'(t_{n+1}) = f(t_{n+1}, y_{n+1})$. This defines the BDF-methods. We can write $q(t)$ as follows

$$q(t) = q(t_n + sh) \approx \sum_{j=0}^{k} (-1)^j \binom{-s+1}{j} \nabla^j u_{n+j}$$

Here is

$$\begin{aligned}
&\nabla^0 u_{n+1} = u_{n+1}, \quad \nabla u_{n+1} = u_{n+1} - u_n \\
&\nabla^r u_{n+1} = \nabla\left(\nabla^{r-1} u_{n+1}\right) \\
&\nabla^2 u_{n+1} = \nabla\left(\nabla u_{n+1}\right) = \nabla\left(u_{n+1} - u_n\right) \\
&\qquad = \nabla u_{n+1} - \nabla u_n = u_{n+1} - u_n = \left(u_n - u_{n-1}\right) \\
&\qquad = u_{n+1} - 2u_n + u_{n-1}
\end{aligned}$$

Note:

$$(-1)^j \binom{-s+1}{j} = \frac{1}{j!}(s+1)\, s\,(s-1)\ldots(s+j-2)$$

Differentiate $q(t)$ and set $s = 1$, then we get

$$\sum_{j=1}^{k} \frac{1}{j} \nabla^j u_{n+1} = h f_{n+1}, \text{BDF formula}$$

Example. Find BDF formula

(i) $k = 1$: Backward Euler Method

(ii) $k = 2$:

$$\begin{aligned}
\nabla u_{n+1} &= \frac{1}{2}\nabla^2 u_{n+1} = h f_{n+1} \\
\implies u_{n+1} - u_n + \frac{1}{2}\left(u_{n+1} - 2u_n + u_n\right) &= h f_{n+1} \\
\frac{3}{2} u_{n+1} - 2u_n + \frac{1}{2} u_n &= h f_{n+1}
\end{aligned}$$

BDF-methods are zero-stable only if $k \leq 6$.

5.3 Absolute stability of multistep methods

- Zero-stability is necessary for convergence. Fundamentally property and considers $h \to 0$.
- Absolute stability. For which stepsizes do we get a stable method.

Again, consider model problem

$$y' = \lambda y, \quad \lambda \in \mathbb{C}$$

We cen then solve it by

$$\sum_{j=0}^{k} \lambda_j u_{n+j} = h\lambda \sum_{j=0}^{k} \beta_j u_{n+j}, \quad z = h\lambda$$

$$\sum_{j=0}^{k} (\lambda_j - z\beta_j)\, u_{n+j} = 0$$

if we let $u_i = \zeta^i$, then is

$$\sum_{j=0}^{k} \lambda_j \zeta^j = z \sum_{j=0}^{k} \implies \rho(\zeta) = z\sigma(\zeta)$$

We need to understand for wich z the root condition holds for the roots $\zeta_1, \zeta_2, \ldots, \zeta_k$

$$|\zeta_i| \leq 1 \quad \forall i, \quad \text{and if} \quad |\zeta| = 1, \quad \text{then} \quad m_j = 1$$

Example. BDF-2-step

$$\frac{3}{2}u_{n+2} - 2u_{n+1} + \frac{1}{2}u_n = h\lambda u_{n+2}$$

$$\left(\frac{3}{2} + z\right) u_{n+2} - 2u_{n+1} + \frac{1}{2}u_n = 0$$

$$u_i = \zeta^i$$

$$\left(\frac{3}{2} - z\right)\zeta^2 - 2\zeta + \frac{1}{2} = 0$$

Digression. if $\rho(\zeta) = a\zeta^2 + b\zeta + c$ one can prove that roots ζ_1 and ζ_2 satisfy $|\zeta_i| < 1$, if and only if

(i) $|a| < c$

(ii) $|a + c| < |b|$

6 Lecture 2020-09-22

6.1 Splitting Methods

In physics get vector field $f(t,y)$ can arise from different types of phenomena.

Example. Reaction-diffusion.

$$\frac{\partial u}{\partial t} = M\Delta u + R(u)$$

u is a vector of concentrations, $u = u(x,t)$, M is a diagonal matrix (positive diagonal), and $R(u)$ are local reactions. Can be split into two parts:

$$\frac{\partial u}{\partial t} = M\Delta v, \quad v(t_n, x) = u_n(x) \tag{3}$$

$$\frac{\partial w}{\partial t} = R(w), \quad w(t_n, x) \text{ is given.} \tag{4}$$

For a given $u_n(x)$, solve (3) to obtain $\widetilde{v}(t_{n+1}, x)$. Then solve (4) , with $w(t_n, x) = \widetilde{v}(t_{n+1}, x)$ then obtain $w(t_{n+1}, x) = u_{n+1}(x)$. Even if (3) and (4) are solved exactly the splitting introduce an error. Assume from now on that we have autonomous problems. Recall flows of vector fields. Let $z(t) = \phi_{f,t}(y_0)$, i.e $z' = f(z), \quad z(0) = y_0$.

Consider

$$y' = f(y) + g(y) = F(y)\,. \tag{5}$$

Then the Lie-Trotter Splitting is

$$y_{n+1} = \phi_{n,g}(\phi_{n,f}(y_n)) = \phi_{n,g} \circ \phi_{n,f}(y_n)\,. \tag{6}$$

Here we assume exact version of the flows. Compare $\phi_{F,h}(y_0)$ to $\phi_{n,g} \circ \phi_{n,f}$.

$$\begin{aligned} y_1 = y(h) &= y_0 + y'(0) + \frac{1}{2}h^2 y'' + O\left(h^3\right) \\ &= y_0 + hF(y_0) + \frac{1}{2}F'F(y_0) + O\left(h^3\right) \\ &= y_0 + h(f(y_0) + g(y_0)) + \frac{1}{2}h^2\left(f'f + f'g + g'g\right)(y_0) + O\left(h^2\right) \end{aligned}$$

And for the other one

$$
\begin{aligned}
&y' = f(y), \quad y(0) = y_0 \\
&\overline{u} = \phi_{n,f}(y_0) = y_0 h f(y_0) + \frac{1}{2}h^2 f' f(y_0) + O\left(h^3\right) \\
&z' = g(z), \quad z(0) = \overline{u} \\
&u_1 = \phi_{n,g} = \overline{u} + hg(\overline{u}) + \frac{1}{2}h^2 g' g(\overline{u}) + O\left(h^3\right) \\
&\quad = y_0 + hf(y_0) + \frac{1}{2}h^2 f' f(y_0)\, h \cdot g(y_0 + hf(y_0) + \ldots) + \frac{1}{2}h^2 g' g(y_0 + O(h)) \\
&\quad = y_0 + hf(y_0) + \frac{1}{2}h^2 + \frac{1}{2}^2 f' f(y_0) + h(g(y_0) + hg' f(y_0) + \ldots) + \frac{1}{2}h^2 g' g(y_0) + O\left(h^3\right) \\
&\quad = y_0 + h(f(y_0) + g(y_0)) \frac{1}{2}h^2 (f' f(y_0) + 2g' f(y_0)\, g' g(y_0)) + +O\left(h^3\right)
\end{aligned}
$$

We can the the difference such that

$$y_1 - u_1 = \frac{1}{2}h^2 (f' g - g' g)(y_0) + O\left(h^3\right)$$

Digression.

$$y_1 - u_1 = \varepsilon = h\tau$$

The order of Lie-Trotter is $p = 1$ since $y_1 - u_1 = O\left(h^3\right)$.

Let $u_1^* = \phi_{n,f} \circ \phi_{n,g}(y_0)$. Same order as u_1 and principal error term of opposite sign. So

$$\frac{1}{2}(\phi_{n,g} \circ \phi_{n,f} + \phi_{n,f} \circ \phi_{n,g})(y_0), \text{ has order } p = 2$$

An even more popular improvement to Lie-Trotter is the Strang splitting.

$$u_1^s = \phi_{\frac{h}{2},f} \circ \phi_{h,g} \circ \phi_{\frac{h}{2},f}(y_0)$$

A similar calculation shows that $y_1 - u_1^s = O\left(h^3\right)$.

Example.

$$y' = \sqrt{y} + y^2, \quad y(0) = 1$$

(i) $z' = \sqrt{z}, \quad z(0) = y_0 = 1,$

$$z(t) = y_0 + t\sqrt{y_0} + \frac{1}{4}t^2$$

(ii) $w' = w^2$, $w(0) = w_0$

$$w(t) = \frac{w_0}{1 - tw_0}$$

Using Strang Splitting, $\frac{h}{2}$ with i)

$$u^* = \left(\sqrt{y_0} + \frac{h}{4}\right)^2 = \left(1 + \frac{h}{4}\right)^2$$

and h with ii) where $u^* = \left(1 + \frac{h}{4}\right) = w_0$

$$u^{**} = \frac{u^*}{1 - hu^*}$$
$$u_1 = \left(\sqrt{u^{**}} + \frac{h}{4}\right)^2$$

Complicated to write out in terms of y_0.

Why use splitting methods?

(i) Each terms may be easier to integrate and the sum.

(ii) Software for each of the terms may be available (multi physics, PDEs).

(iii) Geometric arguments. Makes it easier to preserve certain properties of exact solution.May lead to better long time behaviour of the solution.

6.2 Nonlinear Systems and Numerical Optimization

Reading Material Quertoni and Ch 7.1, Ch 7.2. Duration 2-weeks.

6.2.1 Nonlinear Systems

Given a map $F : \mathbb{R}^n \to \mathbb{R}^n$, find $x^* \in \mathbb{R}^n$ such that $F(x^*) = 0$. Some notation. $DC \subseteq \mathbb{R}^n$: domain of a function. For $k \geq 0$, $C^k(D)$ is the set of functions whose k-th derivative is contionious.

Jacobian Matrix .

$$DF(x) = J_F(x) = F'(x)$$

is the $n \times n$ matrix, where

$$(J_F(x))_{ij} = \frac{\partial F_i}{\partial x_j}$$

7 References