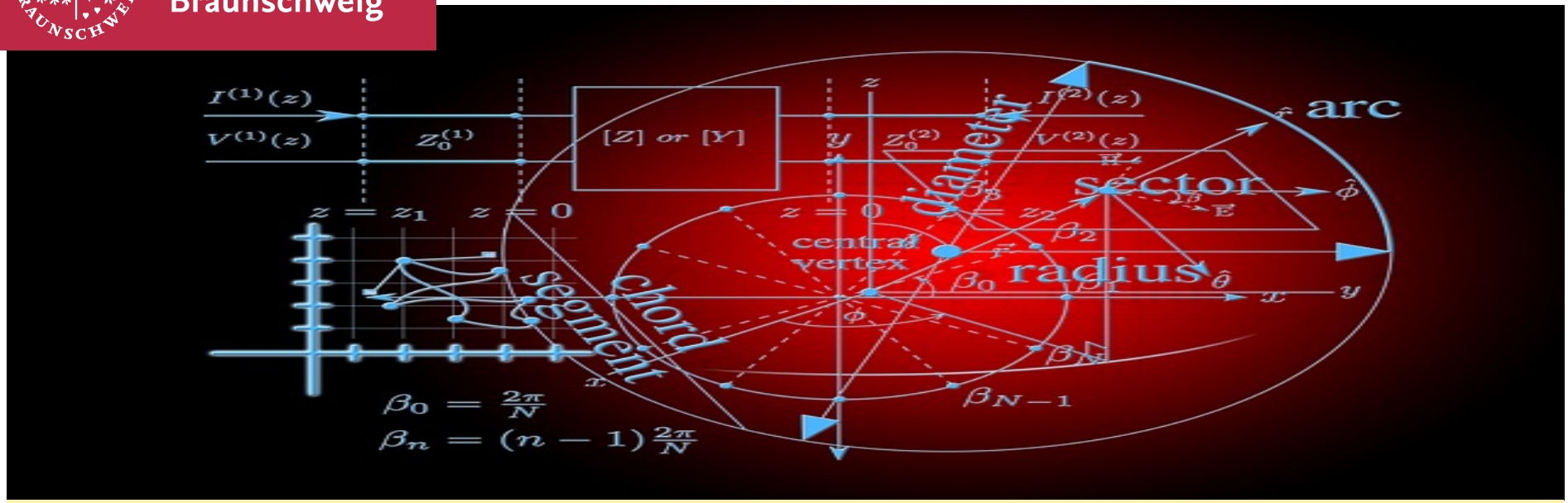
Technische Universität Braunschweig



Introduction to Scientific Computing

Overview

Bojana Rosić, 10. Februar 2016

What we have learned?

Real world

Higher order ODE

$d^n x/ dt^n = F(x,t)$

First order ODE

$d x/ dt = f(x,t)$

Numerical scheme
(behind is numerical integration)

$\Sigma a_i x_{(n+i)} = h \Sigma b_i f(x_{(n+i)}, t_{(n+i)})$

Difference equation

Check condition number and solve linear or nonlinear system

Banach fixed point theorem

Check zero and absolute stability

Check stability
(linearise, eigenvalues)

Real world $\Rightarrow$ ODEs

Felix's free fall

$$m\frac{d^2y}{dt^2} = \frac{1}{2}\rho C_{\mathrm{D}} A \left(\frac{dy}{dt}\right)^2 - mg$$

where m is mass, C_{D} drag coefficient, ρ air density and g is gravitational constant

Here, the maximal order of derivative is 2, so we say that the ODE is of the **second** order.

Higher order ODE $\Rightarrow$ first order ODE

Higher order ODEs can be solved by using the method of transformation to the system of lower order ODEs.

$$m\frac{d^2y}{dt^2} = \frac{1}{2}\rho C_{\mathrm{D}} A \left(\frac{dy}{dt}\right)^2 - mg$$

Take $x_1 := y, x_2 = \frac{dy}{dt}$ as a new variable, then

$$\frac{dx_1}{dt} = \frac{dy}{dt}, \quad \frac{dx_2}{dt} = \frac{d^2y}{dt^2}$$

which further leads to the system of first order ODEs:

$$\dot{\mathbf{x}} = \begin{pmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{pmatrix} = \begin{pmatrix} x_2 \\ \frac{1}{2m}\rho C_{\mathrm{D}} A\,(x_1)^2 - g \end{pmatrix} = \mathbf{f}(\mathbf{x})$$

First order ODE $\Rightarrow$ stability

Equilibria points are obtained by solving

$$\mathbf{f}(\mathbf{x}_*) = 0$$

which possibly requires Newton method:

$$\mathbf{x}_*^{(k+1)} = \mathbf{x}_*^{(k)} - (\mathbf{f}'(\mathbf{x}^{(k)}))^{-1}\mathbf{f}(\mathbf{x}^{(k)})$$

Stability is determined by

$$\dot{\mathbf{x}} = D\mathbf{f}(\mathbf{x}_*)\mathbf{x}$$

where $\mathbf{J} := D\mathbf{f}(\mathbf{x}_*)$ is the Jacobi matrix.

ODE stability

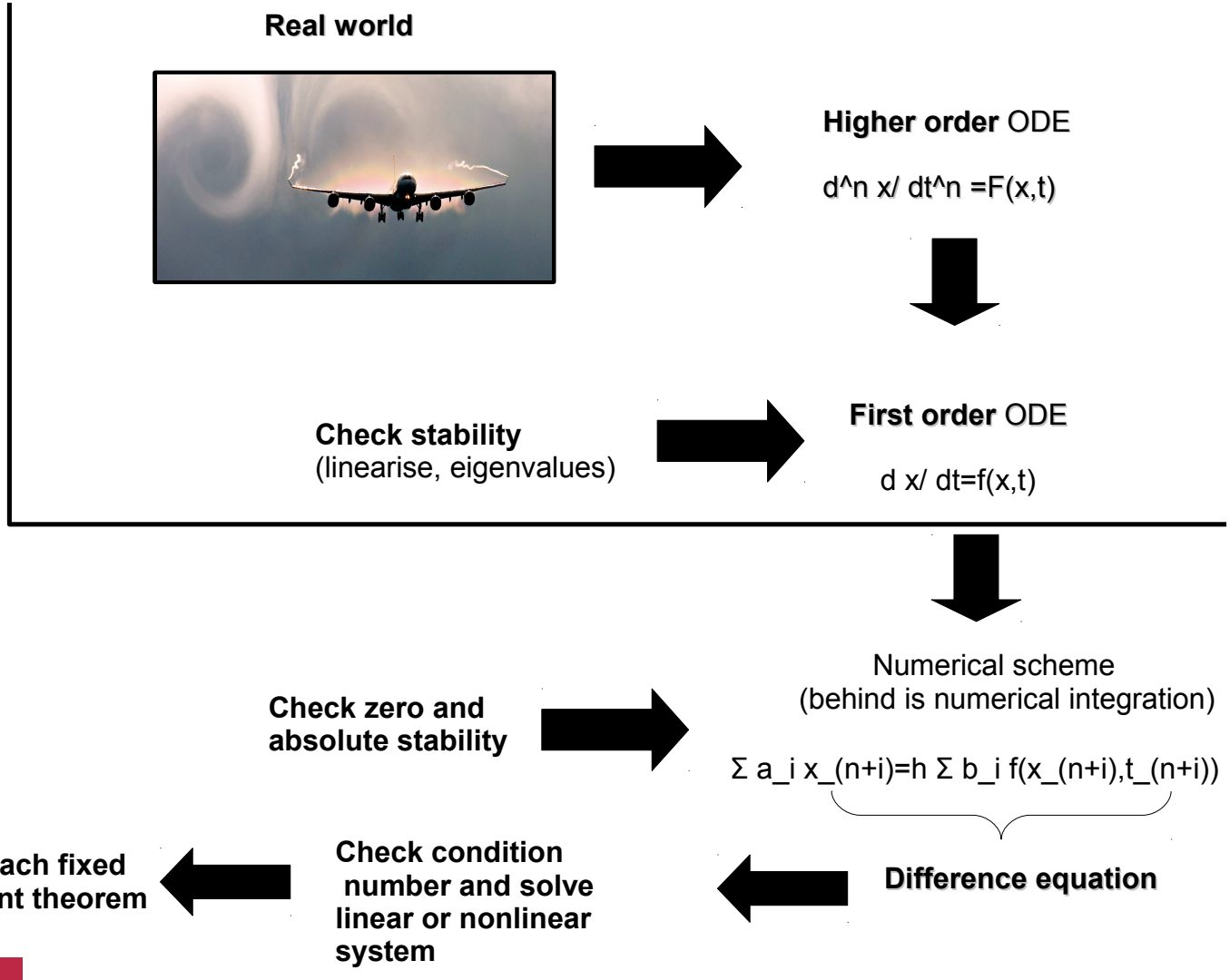
$$\mathbf{J} = \mathbf{Q} e^{t\Lambda} \mathbf{Q}^{-1}$$

Thus,

- if **any** $\mathrm{Re}\ (\lambda_j) > 0$ then $x \to \infty$ when $t \to \infty$ (unstable)
- if for **almost all** $\mathrm{Re}\ (\lambda_j) < 0$ besides **at least** one $\mathrm{Re}\ (\lambda_j) = 0$ then it is stable but not attractive in case of linear system, otherwise it is difficult to say based only on the first order Taylor expansion
- if **all** $\mathrm{Re}\ (\lambda_j) < 0$ then the state x_* is stable and attractive

So find eigenvalues of J and check if you are on the left side of complex plane!!

where we are


Real world
Higher order ODE
d^n x/ dt^n =F(x,t)
Check stability
(linearise, eigenvalues)
First order ODE
d x/ dt=f(x,t)
Numerical scheme
(behind is numerical integration)
Check zero and
absolute stability
Σ a_i x_(n+i)=h Σ b_i f(x_(n+i),t_(n+i))
Difference equation
Check condition
number and solve
linear or nonlinear
system
Banach fixed
point theorem

First order ODE $\Rightarrow$ solving

Algebraic solving is possible for linear system of ODEs. Solution is of the form

$$\mathbf{x} = c_1 e^{\lambda t}\mathbf{v} + c_2(te^{\lambda t}\mathbf{v} + e^{\lambda t}\mathbf{w})$$

in which $\mathbf{v}$ is eigenvector and $\mathbf{w}$ is **the generalised eigenvector**. These are obtained by solving

$$(A - \lambda I)\mathbf{v} = 0, \quad (A - \lambda I)\mathbf{w} = \mathbf{v}.$$

However, the algebraic solution is not something that is used in practice.

First order ODE $\Rightarrow$ numerical integration

$$\dot{x} = f(t) \Rightarrow x(t_n) = x(t_m) + \int_{t_m}^{t_n} f(t)dt$$

and thus includes the numerical integration

$$\int_{t_m}^{t_n} f(t)dt = \int_{t_m}^{t_n} P_n(t)dt = \sum_{i=1}^{N} w_i f(t_i)$$

in which P_n is n-th order interpolation polynomial satisfying interpolation conditions

$$P_n(t_i) = f(t_i), \quad i = 1, \ldots n+1$$

The interpolation points t_i are chosen to be equidistant or are assumed to be unknown (in this case they are computed from the error estimates).

ODE $\Rightarrow$ integration $\Rightarrow$ difference equation

Solution

$$x(t_n) = x(t_m) + \int_{t_m}^{t_n} f(t)dt$$

is difference equation. If $f(t)$ is approximated by constant, then one may distinguish explicit from implicit Euler method:

$$x(t_{n+1}) = x(t_n) + hf(x_n, t_n) \quad \text{and} \quad x(t_{n+1}) = x(t_n) + hf(x_{n+1}, t_{n+1})$$

These are one step methods described by interpolation points $t = t_n$ or $t = t_{n+1}$. Multistep methods use approximation of n-th order to model the right hand side $f \approx P_n$ or the state itself $x \approx P_n$, $\frac{dP_n}{dt} = f$. Hence, they need several interpolation points and thus several starting points.

Numerical scheme $\Rightarrow$ convergence

Convergence implies consistency and zero stability. Consistency does not imply convergence.

The method is said to be consistent if the local (truncation) error converges to zero when the step–size becomes arbitrarily small, i.e.

$$\lim_{h \to 0} T_h = \lim_{h \to 0} \frac{\epsilon_{loc}}{h} = 0$$

The scheme is consistent of order p if

$$\max \| \frac{\epsilon_{loc}}{h} \| \leqslant C h^p$$

The local error ϵ_{loc} is the difference between the exact $x_a(t_{n+1})$ and numerical $x(t_{n+1})$ solutions at t_{n+1} where $x(t_{n+1})$ is computed by a numerical scheme starting from $x_a(t_n)$.

Numerical scheme $\Rightarrow$ convergence

Linear multistep method is said to be convergent if

$$\max_{t_n \in T} \|\epsilon_{glob}(t_n, h)\| \to 0, \quad h \to 0$$

and it has the order of convergence q if

$$\max_{t_n \in T} \|\epsilon_{glob}(t_n, h)\| \leqslant Ch^q$$

where the global error is $\epsilon_{glob} = x_a(t) - x(t)$ in which $x_a(t)$ is the exact solution of the differential equation $\dot{x} = f(t, x)$ and $x(t)$ is the approximate solution. C is a constant independent of h

Numerical scheme $\Rightarrow$ step size

The difference equation $\sum_{j=0}^{k} a_j x_{n+j} = h \sum_{j=0}^{k} b_j f(t_{n+j}, x_{n+j})$ is stable if roots

$$\rho(\xi) - z\sigma(\xi) = 0$$

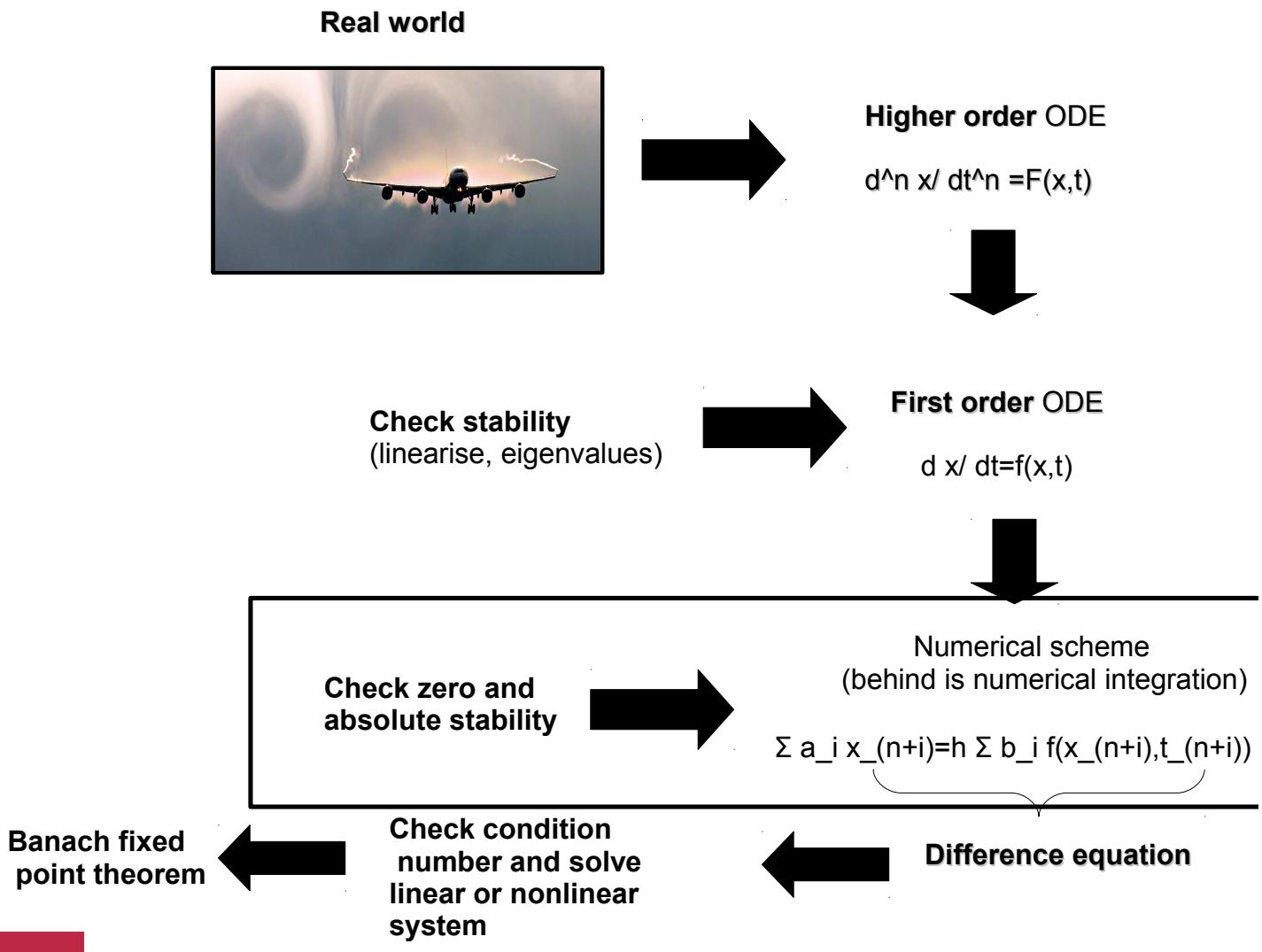
are smaller by amplitude than 1, or eventually only one of them is equal to 1. With respect to this one defines

Definition

The region of absolute stability for the LMM is the set of points $z = h\lambda$ in the complex plane for which the polynomial $\rho(\xi) - z\sigma(\xi) = 0$ satisfies the root condition.

$$G_s := \{z \in \mathbb{C} \ : \ |\xi(z)| \leqslant 1\}$$

where we are


Real world
Higher order ODE
d^n x/ dt^n =F(x,t)
Check stability
(linearise, eigenvalues)
First order ODE
d x/ dt=f(x,t)
Check zero and
absolute stability
Numerical scheme
(behind is numerical integration)
Σ a_i x_(n+i)=h Σ b_i f(x_(n+i),t_(n+i))
Banach fixed
point theorem
Check condition
number and solve
linear or nonlinear
system
Difference equation

Numerical scheme $\Rightarrow$ solving system

Applying implicit numerical scheme onto ODE

$$\sum_{j=0}^{k} a_j x_{n+j} = h \sum_{j=0}^{k} b_j f(t_{n+j}, x_{n+j})$$

such as for example implicit Euler method

$$x_{n+1} = x_n + hf(x_{n+1}, t_{n+1})$$

one ends up with the linear/nonlinear system of equations

$$g(x_{n+1}) = x_n + hf(x_{n+1}, t_{n+1}) - x_{n+1} = 0$$

in which x_{n+1} is unknown.

Solving system $\Rightarrow$ fixed point iteration

The solution of

$$g(z) = 0$$

is fixed point z_* in interval $[z_a, z_b]$ which satisfies

$$z_* = \varphi(z_*)$$

The solution is unique and exists if **Banach fixed point** theorem is valid, i.e.

- $\varphi(z) : [z_a, z_b] \rightarrow [z_a, z_b]$
- the mapping $\varphi(z)$ is Lipshitz continuous

$$\|\varphi(z) - \varphi(\hat{z})\| \leqslant q\|z - \hat{z}\|$$

- the mapping is contractive $q < 0$

Solving system $\Rightarrow$ fixed point iteration

We have studied two error estimates

- A posteriori error estimate:

$$\|\mathbf{z}_* - \mathbf{z}_k\| \leqslant \frac{q}{1-q}\|\mathbf{z}_k - \mathbf{z}_{k-1}\|$$

- A priori error estimate:

$$\|\mathbf{z}_* - \mathbf{z}_k\| \leqslant \frac{q^k}{1-q}\|\mathbf{z}_1 - \mathbf{z}_0\|.$$

Here, index k is not time step, but iteration number.

Solving system $\Rightarrow$ linear system

System:

$$\mathbf{Az} = \mathbf{b}$$

Gauss-Seidel method:

$$\mathbf{L}\mathbf{z}^{(k+1)} = \mathbf{b} - \mathbf{U}\mathbf{z}^{(k)}$$

Jacobi method:

$$\mathbf{D}\mathbf{z}^{(k+1)} = \mathbf{b} - \mathbf{R}\mathbf{z}^{(k)}$$

Successive over-relaxation (SOR) method:

$$\mathbf{z}^{(k+1)} = (\mathbf{D} + \omega\mathbf{L})^{-1}\left(\omega\mathbf{b} - [\omega\mathbf{U} + (\omega - 1)\mathbf{D}]\mathbf{z}^{(k)}\right).$$

Conjugate gradients

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)}$$

Solving system $\Rightarrow$ nonlinear system

$$\mathbf{F}(\mathbf{z}) = \mathbf{0}$$

$$\mathbf{F}(\mathbf{z}) = \mathbf{F}(\mathbf{z}_0) + \mathbf{F}'(\mathbf{z}_0)(\mathbf{x} - \mathbf{z}_0) + h.o.t. = \mathbf{0}$$

Jacobian:

$$\mathbf{J}(\mathbf{z}) = \mathbf{F}'(\mathbf{z})$$

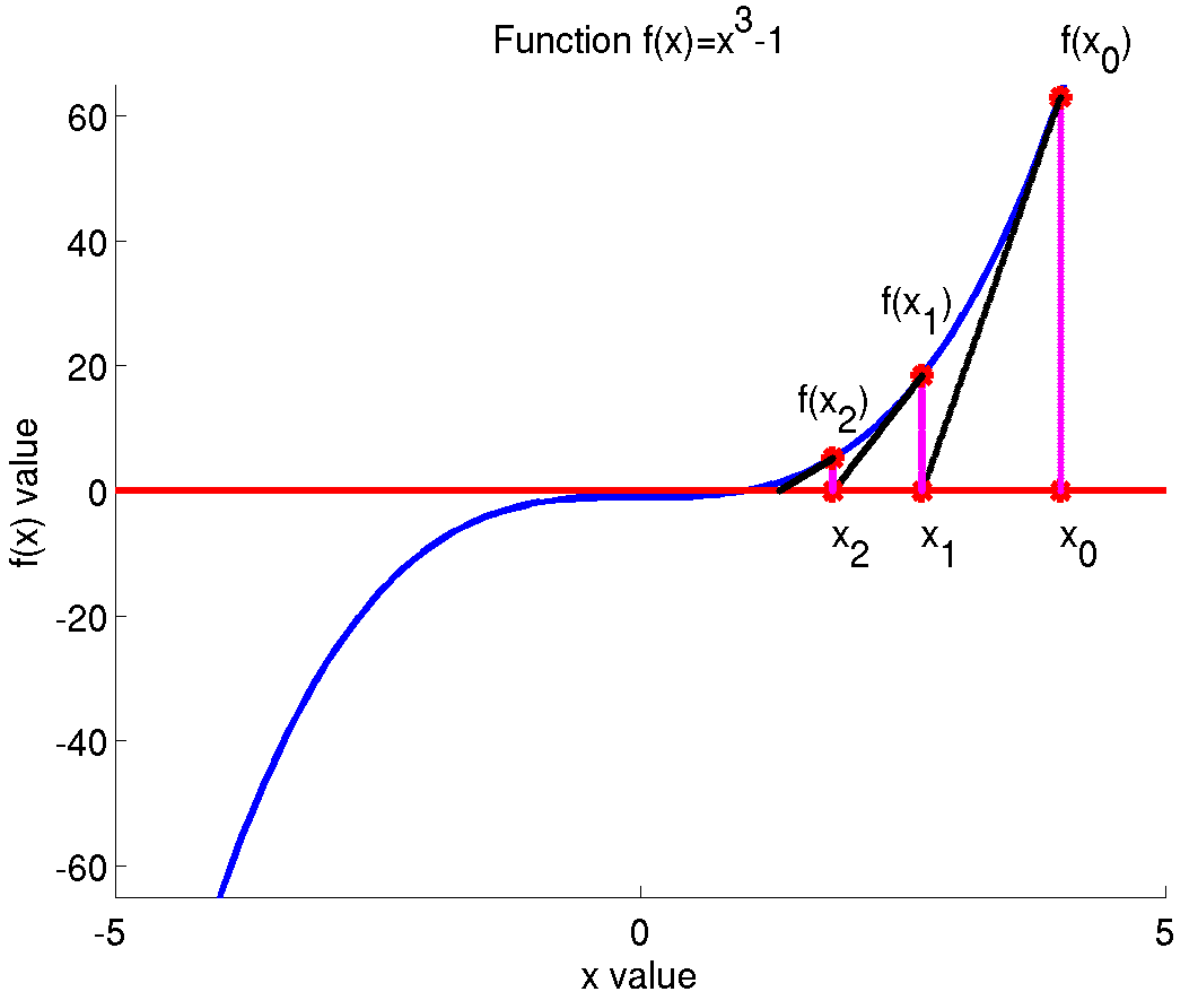
with elements:

$$J_{ij} = \frac{\partial f_i}{\partial z_j}$$

Newton iteration:

$$\mathbf{z}_k = \mathbf{z}_{k-1} - \frac{\mathbf{F}(\mathbf{z}_{k-1})}{\mathbf{J}(z_{k-1})}$$

Solving system $\Rightarrow$ nonlinear system



Solving system $\Rightarrow$ nonlinear system

Besides classical Newton algorithm one may also use:

- simplified (stationary) Newton

$$\mathbf{z}_k = \mathbf{z}_{k-1} - \frac{\mathbf{F}(\mathbf{z}_{k-1})}{\mathbf{J}(z_0)}$$

- modified Newton

$$\mathbf{z}_k = \mathbf{z}_{k-1} - m\frac{\mathbf{F}(\mathbf{z}_{k-1})}{\mathbf{J}(z_{k-1})}$$

- Newton method with restart

$$\mathbf{z}_k = \mathbf{z}_{k-1} - \frac{\mathbf{F}(\mathbf{z}_{k-1})}{\mathbf{J}(z_m)}$$

- Broyden's method (updating Jacobian)

Solving system $\Rightarrow$ condition number

Modelling, as well as numerical solving need to satisfy the property of well-posedness:

- solution exists
- solution is unique
- small perturbations in data cause small pertubrations in the solution

Solving system $\Rightarrow$ condition number

Model

$$y = f(x)$$

has condition number

$$k(x) = \left| \frac{\partial f}{\partial x} \frac{x}{f(x)} \right|$$

The system

$$Ax = b$$

has condition number

$$k(A) = \|A\| \|A^{-1}\|$$

where we are

Real world

Higher order ODE

$d^n x/ dt^n = F(x,t)$

Check stability
(linearise, eigenvalues)

First order ODE

$d x/ dt = f(x,t)$

Numerical scheme
(behind is numerical integration)

$\Sigma a_i x_{(n+i)} = h \Sigma b_i f(x_{(n+i)}, t_{(n+i)})$

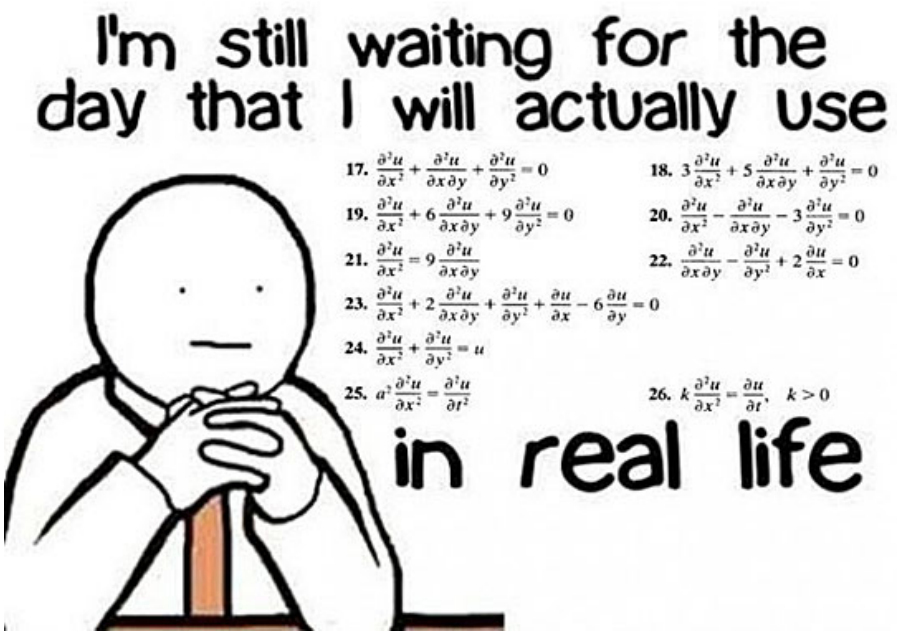
Check zero and absolute stability

Difference equation

Check condition number and solve linear or nonlinear system

Banach fixed point theorem

DIFFERENCE AND DIFFERENTIAL EQUATIONS



Difference vs. Differential equations

Discrete system	Continuous system
Difference	Differential
$x_{n+1} - x_n = 2h^2 n$	$\frac{dx}{dt} = 2t$
	Discretise in time
	$x_{n+1} - x_n = 2h^2 n$

Differential equations become difference when we apply numerical integration (time discretisation)!

Homogeneous equations (zero rhs)

Difference	Differential
$x_{n+1} - x_n = 0$	$\frac{dx}{dt} = 0$
Solution form	Solution form
$x_n = C\rho^n$	$x(t) = Ce^{\rho t}$
Characteristic	Characteristic
$\rho - 1 = 0$	$\rho = 0$
$x_n = C(1)^n = C$	$x = Ce^{0t} = C$

Homogeneous solution wrt roots

Difference	Differential
Single roots ρ_i	Single roots ρ_i
$x_n = \sum_i C_i \rho_i^n$	$x(t) = \sum_i C_i e^{\rho_i t}$
Double roots ρ_1	Double roots ρ_1
$x_n = C_1 \rho_1^n + nC_2 \rho_1^n$	$x(t) = C_1 e^{\rho_1 t} + tC_2 e^{\rho_1 t}$

Roots and eigenvalues are equal!!

Homogeneous solution wrt roots

Difference	Differential
Complex $\rho = \alpha \pm \beta i$	Complex $\rho = \alpha \pm \beta i$
$\mu := \sqrt{\alpha^2 + \beta^2}$	$e^{(\alpha \pm \beta i)t} = e^{\alpha t} e^{\beta i t}$
$\tan \phi = \beta / \alpha$	
$x_n = \mu^n (A\cos(n\phi) + iB\sin(n\phi))$	$x = e^{\alpha t}(A\cos(\beta t) + iB\sin(\beta t))$

Note that in one case is the modulus and in another real part of the root.

Non-homogeneous- Particular solution

Difference

$$x_n = x_{nh} + x_{np}$$

g_n	$x_n^{(p)}$
a^n	$d_1 a^n$
n^k	$d_0 + d_1 n + \cdots + d_k n^k$
$n^k a^n$	$(d_0 + d_1 n + \cdots + d_k n^k) a^n$

Differential

$$x = x_h + x_p$$
$$x_h = Ce^{pt},$$

variation of constant

$$x_p = c(t)e^{pt}$$

STABILITY


HOW ARE SALES GOING?
STEADY.
CARTOONSTOCK .com
Search ID: mf10298

Stability

Difference	Differential
$x_{n+1} = F(x_n)$	$\frac{dx}{dt} = f(x)$
$x_* = F(x_*)$	$f(x_*) = 0$
roots of charact. equation or	roots of charact. equation or
$[\lambda, v] = \mathrm{eig}\ DF(x_*)$	$[\lambda, v] = \mathrm{eig}\ Df(x_*)$
$(\lambda)^n \Rightarrow \mathrm{abs}\ (\lambda) \leqslant 1$ (real)	$e^{\lambda t} \Rightarrow (\lambda) \leqslant 0$ (real)
$\mathrm{abs}\ (\mu) \leqslant 1$ (complex)	$(Re(\lambda)) \leqslant 0$ (complex)

ODE 2 expectations

In this lecture we have only scratched the surface of the process of solving ordinary differential equations and stability properties of numerical schemes. Next semester will deal more seriously with the process of time discretisation of ordinary differential equations and time dependent partial differential equations. Special attention is put on the so-called stiff-systems which are very often arising in practical applications, as well as differential algebraic equations (differential equations with algebraic conditions).

Studienarbeit vs. Masterarbeit

for those interested in applied mathematics there are possiblities of offering student project (3months) or master (6 months) topics. The focus are ODEs/PDEs which describe

- plasticity, viscoplasticity, damage (description of cellulose, concrete, metals etc.)
- weather prediction models (different Lorenz models)
- simple biological models (modelling insulin change in body, drug transport through blood)

Topics are: uncertainty quantification, identification, control, adaptive algorithms, etc.

Thats all folks

Hints can never hurt you, except on exams. Then, I can mess with your mind.