Introduction to Scientific Computing

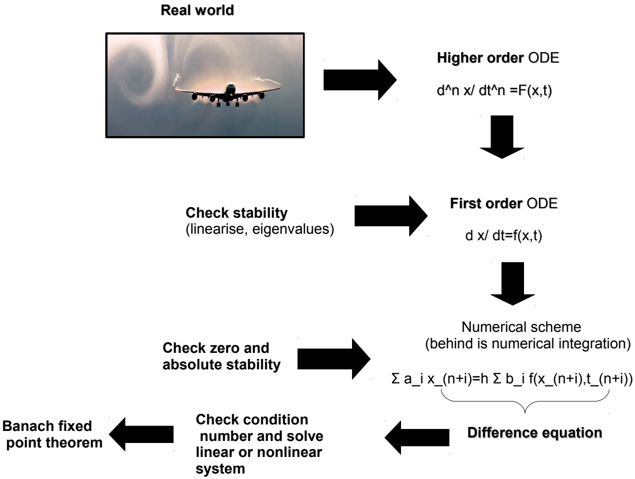
(Lecture 11: Overview)

Bojana Rosić

Institute of Scientific Computing

January 31, 2018

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)

Check stability
(linearise, eigenvalues)

Numerical scheme
(behind is numerical integration)

Σ a_i x_(n+i)=h Σ b_i f(x_(n+i),t_(n+i))

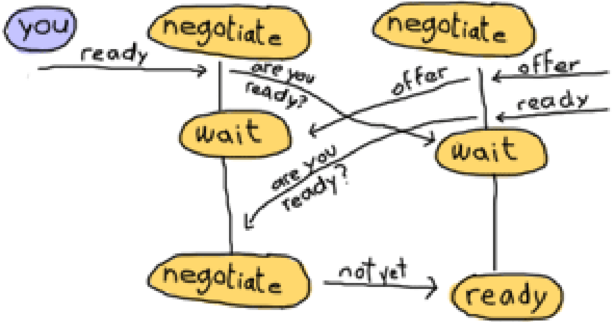
Check zero and absolute stability

Difference equation

Check condition number and solve linear or nonlinear system

Banach fixed point theorem

CONDITION NUMBER



http://learnyousomeerlang.com/

Condition number

This number predicts the relative error in output y if the relative error in input x is known. Or better to say represents sensitivity of the output on perturbations of input data.

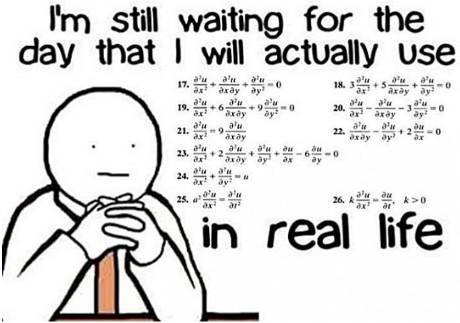
$$k(x) := \frac{\partial f}{\partial x}(\mathbf{x})\frac{x}{f(\mathbf{x})}$$

$$\epsilon_y = \frac{\Delta y}{y} = k(x)\epsilon_x = k(x)\frac{\Delta x}{x}$$

Example: substruction is ill-conditioned operation, e.g.

$$\frac{1}{x_1 - x_2}, \quad x_1 \approx x_2$$

DIFFERENCE AND DIFFERENTIAL EQUATIONS


I'm still waiting for the day that I will actually use
17. $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2} = 0$
18. $3\frac{\partial^2 u}{\partial x^2} + 5\frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2} = 0$
19. $\frac{\partial^2 u}{\partial x^2} + 6\frac{\partial^2 u}{\partial x \partial y} + 9\frac{\partial^2 u}{\partial y^2} = 0$
20. $\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial x \partial y} - 3\frac{\partial^2 u}{\partial y^2} = 0$
21. $\frac{\partial^2 u}{\partial x^2} = 9\frac{\partial^2 u}{\partial x \partial y}$
22. $\frac{\partial^2 u}{\partial x \partial y} - \frac{\partial^2 u}{\partial y^2} + 2\frac{\partial u}{\partial x} = 0$
23. $\frac{\partial^2 u}{\partial x^2} + 2\frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial x} - 6\frac{\partial u}{\partial y} = 0$
24. $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = u$
25. $a^2\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2}$
26. $k\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}, \quad k > 0$
in real life

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
$x_n = C_1 \rho_1^n + n C_2 \rho_1^n$	$x(t) = C_1 e^{\rho_1 t} + t C_2 e^{\rho_1 t}$

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: mfl0298

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
$[\lambda, v] = \mathrm{eig}\ DF(x_*)$	$[\lambda, v] = \mathrm{eig}\ Df(x_*)$
$(\lambda)^n \Rightarrow \mathrm{abs}\ (\lambda) \leq 1$ (real)	$e^{\lambda t} \Rightarrow (\lambda) \leq 0$ (real)
$\mathrm{abs}\ (\mu) \leq 1$ (complex)	$(Re(\lambda)) \leq 0$ (complex)

Stability

If there is just one root equal to 1 (i.e. 0) ⇒ stable but not asympt.

$$(1)^n = 1, \quad e^{0t} = 1, \quad n \to \infty \quad \& \quad t \to \infty$$

If there are more roots equal to 1(i.e. 0) ⇒ not stable

$$c_1(1)^n + c_2 n(1)^n \to \infty, \quad c_1 e^{0t} + c_2 t e^{0t} \to \infty, \quad n \to \infty \quad \& \quad t \to \infty$$

Zero-stability

Stability with respect to the intial conditions. Apply numerical method to the ODE

$$\dot{x} = 0$$

such as for example Euler:

$$x_{n+1} - x_n = 0$$

then test stability of this difference equation.

Zero stability is important due to the process of selecting initial conditions. They are not always correct.

COMPUTING THE ROOTS—SOLVING SYSTEMS OF EQUATIONS

$$F(X) = 0$$

Solving linear system of equations

System:

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

Gauss-Seidel method:

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

Jacobi method:

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

Successive over-relaxation (SOR) method:

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

Fixed-point iterations

We want to solve

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{x}$$

such that

- $\boldsymbol{F}$ has fixed point (solution)
- the fixed point is unique
- can be obtained by iterative process

$\boldsymbol{F}$ is also called mapping.

Fixed-point iterations

Banach fixed point

Lipshitz continuity (for $q \geq 0$):

$$\|\boldsymbol{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})\| \leq q\|\mathbf{x} - \mathbf{y}\|$$

If this holds when $0 \leq q < 1$ then

- $\boldsymbol{F}(\mathbf{x}_*) = \mathbf{x}_*$ has unique solution
- for any initial value sequence $\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n)$ converges to solution $\mathbf{x}_*$ (this means that iterative method is convergent and gives the solution)
- speed of convergence are given by apriori and aposteriori estimates

Error estimates

We have studied two error estimates

- A posteriori error estimate:

$$\|\mathbf{x}_* - \mathbf{x}_n\| \leq \frac{q}{1-q}\|\mathbf{x}_n - \mathbf{x}_{n-1}\|$$

- A priori error estimate:

$$\|\mathbf{x}_* - \mathbf{x}_n\| \leq \frac{q^n}{1-q}\|\mathbf{x}_1 - \mathbf{x}_0\|.$$

Newton method

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

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

Jacobian:

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

with elements:

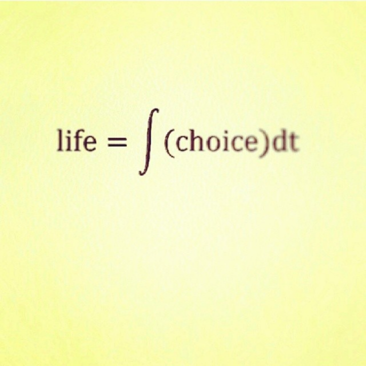
$$J_{ij} = \frac{\partial f_i}{\partial x_j}$$

Newton iteration:

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

NUMERICAL INTEGRATION

$$\int_a^b f(t)dt = ...$$



$$\text{life} = \int (\text{choice})\text{dt}$$

Picards iteration

To compute the solution of integral equation one may use the fixed point iteration (also called Picard-Lindelöf iteration):

$$x^{(k+1)}(t) = x_0 + \int_{t_0}^{t} f(x^{(k)}(s), s)ds = F(x^{(k)}, s)$$

by starting from $x^{(0)} = x_0$.

Newton-Cotes integration

Approximate function (rhs of ODE) by a polynomial

$$\int_{t_0}^{T} f(t)\;dt \approx \int_{t_0}^{T} P(t)\;dt = \int_{t_0}^{T} \sum_{j=0}^{n} a_j t_i^j dt$$

Possible polynomial approximations (global or local):

- piecewise constant interpolation (rectangle rule)
- linear interpolation (trapezoidal rule)
- quadtratic interpolation (Simpsons rule)

Gauss integration

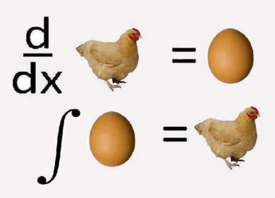
In Newton-Cotes formula one chooses the points t_i in which the value of function will be evaluated. This is not optimal as the result may lead to large errors. The Gauss quadrature has for a goal to vary the placements t_i such that the integration is more accurate. In general Gauss formula approximates:

$$\int_{-1}^{1} g(t)dt \approx \int_{-1}^{1} P_n(t)dt = \sum_{i=0}^{n} g(t_i)w_i, \quad w_i = \int_{-1}^{1} \ell_i(t)dt$$

in the same way as Newton-Cotes. But, t_i are unknown and have to be found. Additionaly to them are also unknown the coefficients of a polynomial $P_n(x)$.

NUMERICAL SCHEMES TO SOLVE ODE

$$\frac{dx}{dt} = f(x,t) \Rightarrow x(t_{n+1}) = ...$$



Multistep methods

Numerical methods for solving ODE

$$\dot{x} = f(x, t)$$

can be classified as

- **one step methods** : these methods use only information from one time point to compute the next

$$x_{n+1} = x_n + hf(t_n, x_n) \quad \text{requires knowledge on} \quad x_n$$

- **multistep methods**: these methods require knowledge on more than one time point

$$x_{n+1} = x_n + h\left(\frac{3}{2}f_n - \frac{1}{2}f_{n-1}\right) \quad \text{requires knowledge on} \quad x_n, x_{n-1}$$

Multistep method

Linear multistep methods of the form (only linear combination of x's and f's)

$$\sum_{l=0}^{k} a_l x_{n+l} = h \sum_{l=0}^{k} b_l f(t_{n+l}, x_{n+l})$$

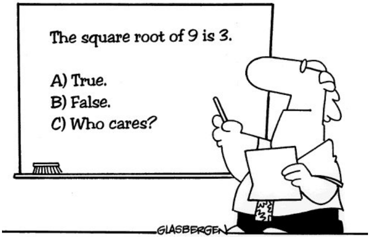
One step methods

Constraining ourselves to one step, one obtains

- explicit Euler method $x_{n+1} = x_n + hf(t_n, x_n)$
- implicit Euler method $x_{n+1} = x_n + hf(t_{n+1}, x_{n+1})$
- trapezoidal rule $x_{n+1} = x_n + \frac{h}{2}(f(t_n, x_n) + f(t_{n+1}, x_{n+1}))$
- explicit midpoint rule $x_{n+1} = x_n + hf(t_n + \frac{h}{2}, x(t_n) + \frac{h}{2}f(t_n, x_n))$
- implicit midpoint rule $x_{n+1} = x_n + hf(t_n + \frac{h}{2}, \frac{1}{2}(x_n + x_{n+1}))$

ACCURACY OF NUMERICAL SCHEMES

Copyright 1996 Randy Glasbergen. www.glasbergen.com



Many students actually look forward to Mr. Atwadder's math tests.

Consistency of numerical methods for ODE

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

$$\max \| \frac{\epsilon_{loc}}{h} \| \leq Ch^p$$

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}

$$\epsilon_{loc} = x_a(t_{n+1}) - x(t_{n+1})$$

where $x(t_{n+1})$ is computed by a numerical scheme starting from $x_a(t_n)$.

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

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

where

- the global error is

$$\epsilon_{glob} = x_a(t) - x(t)$$

- $x_a(t)$ is the exact solution of the differential equation $\dot{x} = f(t, x)$
- $x(t)$ is the approximate solution
- C is a constant independent of h

Convergence

Convergence implies consistency. Consistency does not imply convergence.

STABILITY OF NUMERICAL SCHEMES

Global error

Global error is defined as:

$$\epsilon_{glob}^{n+1} = x_a(t_{n+1}) - x(t_{n+1})$$

in which x_a is analytical and $x(t)$ full numerical solution. Having in mind that

$$x(t_{n+1}) = x(t_n) + hf(t_n, x_n) = x(t_n) + h\lambda x_n, \quad n = 0, 1, 2, ...$$

with λ being -1 or -100, one may write

$$\epsilon_{glob}^{n+1} = x_a(t_{n+1}) - x(t_n) - h\lambda x_n$$

Absolute Stability

To judge if the numerical method produces desired numerical accuracy for given $h > 0$, one introduces the notion of absolute stability. Absolute stability is defined as a stability of numerical scheme applied on the Dahlquist problem:

$$\dot{x} = \lambda x, \quad x(0) = 1$$

whose exact solution is given by

$$x = \exp(\lambda t)$$

and depends on the value of λ

$$\lim_{t \to \infty} |x(t)| = \begin{cases} 0, & \text{if } \lambda < 0 \Rightarrow \text{stable ODE} \\ 1, & \text{if } \lambda = 0 \Rightarrow \text{not considered} \\ \infty, & \text{if } \lambda > 0 \Rightarrow \text{not considered} \end{cases}$$

Absolute Stability

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.

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

Absolute Stability

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.

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

Thats all folks

“Any fool can know. The point is to understand.”
Albert Einstein

http://learnyousomeerlang.com/