Introduction to

Computational Physics

University of Maribor,
Summer Term 2003

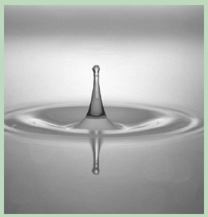
Franz J. Vesely
University of Vienna

www.ap.univie.ac.at/users/Franz.Vesely/

DIFFERENTIAL EQUATIONS EVERYWHERE

The laws and relations of physics are often formulated in terms of DEs. Usually, analytical solutions are hard to come by, therefore numerical integration schemes are wanted:

Section 4. Ordinary Differential Equations



Section 5. Partial Differential Equations

Euler's integration scheme:
respect it, but don't use it!

Ordinary Differential Equations (ODE):
Find the solution $y(x)$ of

$$L(x, y, y', y'', \ldots y^{(n)}) \;=\; 0$$

($y' \equiv dy/dx$ etc.)

In physics:

– mostly first or second order
– usually given in explicit form, $y' = f(x, y)$ or $y'' = g(x, y)$

Second order DE may be written as 2 DEs of first order: $y' = z(x,y)$; $z' = g(x,y)$.

Example:
Harmonic oscillator: Instead of $d^2x/dt^2 = -\omega_0^2 x$, write

$$\frac{dx}{dt} = v\,; \quad \frac{dv}{dt} = -\omega_0^2 x$$

or

$$\frac{d\mathbf{y}}{dt} = \mathbf{L}\cdot\mathbf{y}\,, \quad \text{where } \mathbf{y} \equiv \begin{pmatrix} x \\ v \end{pmatrix} \text{ and } \mathbf{L} = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{pmatrix}$$

– If the values of y, y' etc. are all given at x_0:
$\Longrightarrow$*Initial Value Problem* (IVP).

– If y, y' etc. are given at several points $x_0, x_1, \ldots$:
$\Longrightarrow$*Boundary Value Problem* (BVP).

Typical IVP: *equations of motion* $d^2x/dt^2 = K/m$; $x(0)$ and $x'(0)$ given

Typical BVP: potential equation $d^2\phi/dx^2 = \rho(x)$; $\phi(x)$ given at boundary points

Initial Value Problems of First Order

2 guinea pigs will often be used:

(1) Relaxation equation

$$\frac{dy}{dt} = -\lambda\, y\,, \quad \text{with} \;\; y(t=0) = y_0$$

(2) Harmonic oscillator in linear form

$$\frac{d\mathbf{y}}{dt} = \mathbf{L}\cdot\mathbf{y}\,, \quad \text{where} \;\; \mathbf{y} \equiv \begin{pmatrix} x \\ v \end{pmatrix} \;\; \text{and} \;\; \mathbf{L} = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{pmatrix}$$

Euler-Cauchy Algorithm

Apply DNGF approximation

$$\left.\frac{d\mathbf{y}}{dt}\right|_{t_n} = \frac{\Delta \mathbf{y}_n}{\Delta t} + O[(\Delta t)]$$

to the linear DE and find the *Euler-Cauchy* (EC) formula

$$\frac{\Delta \mathbf{y}_n}{\Delta t} = f_n + O[(\Delta t)]$$

or

$$\boxed{\mathbf{y}_{n+1} = \mathbf{y}_n + \mathbf{f}_n \Delta t + O[(\Delta t)^2]}$$

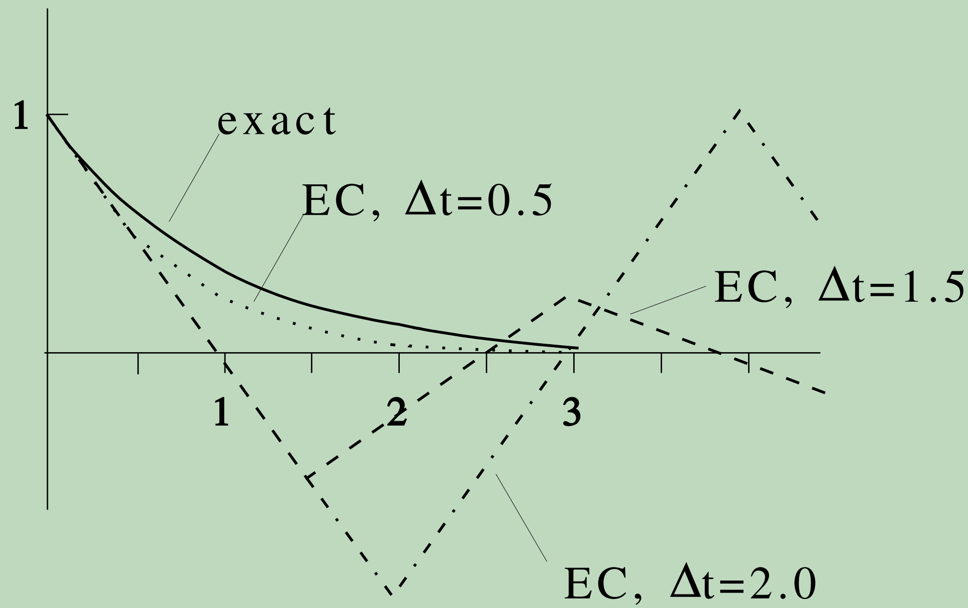
Algebraically and computationally simple, but useless:

– Only first order accuracy

– Unstable: small aberrations from the true solution tend to grow in the course of further steps. $\Longrightarrow$

Apply EC to the relaxation equation $\frac{dy(t)}{dt} = -\lambda y(t)$:

$$y_{n+1} \;=\; (1 - \lambda\,\Delta t)\, y_n$$



EC applied to the equation $dy/dt = -\lambda y$, with $\lambda = 1$ and $y_0 = 1$: unstable for $\lambda \Delta t > 2$

Stability and Accuracy of Difference Schemes:

Let $\mathbf{y}(t)$ be the exact solution of a DE, and $\mathbf{e}(t)$ an error: at time t_n, the algorithm produces $\mathbf{y}_n + \mathbf{e}_n$.
$\Longrightarrow t_{n+1}$? For EC, $\mathbf{y}_{n+1} + \mathbf{e}_{n+1} = \mathbf{y}_n + \mathbf{e}_n + \mathbf{f}(\mathbf{y}_n + \mathbf{e}_n)\Delta t$. Generally,

$$\mathbf{y}_{n+1} + \mathbf{e}_{n+1} \quad = \quad T(\mathbf{y}_n + \mathbf{e}_n)$$

Expand around the correct solution:

$$T(\mathbf{y}_n + \mathbf{e}_n) \quad \approx \quad T(\mathbf{y}_n) + \left.\frac{dT(\mathbf{y})}{d\mathbf{y}}\right|_{\mathbf{y}_n} \cdot \mathbf{e}_n$$

or

$$\mathbf{e}_{n+1} \quad \approx \quad \left.\frac{dT(\mathbf{y})}{d\mathbf{y}}\right|_{\mathbf{y}_n} \cdot \mathbf{e}_n \equiv \mathbf{G} \cdot \mathbf{e}_n$$

The matrix $\mathbf{G}$ is called *amplification matrix*. All its eigenvalues must be within the unit circle:

$$|g_i| \quad \leq \quad 1\,, \quad \text{for all } i$$

Example: *EC + Relaxation equation*

$$T(y_n) \equiv (1-\lambda\Delta t)\, y_n$$

$\Longrightarrow$

$$|1-\lambda\Delta t| \leq 1$$

For $\lambda = 1$ this condition is met whenever $\Delta t \leq 2$. $\Longrightarrow$Check the previous figure!

Example: *EC + Harmonic oscillator*

$$\mathbf{y}_{n+1} = [\mathbf{I} + \mathbf{L}\Delta t] \cdot \mathbf{y}_n \equiv T(\mathbf{y}_n)$$

The amplification matrix is

$$\mathbf{G} \equiv \left.\frac{dT(\mathbf{y})}{d\mathbf{y}}\right|_{\mathbf{y}_n} = \mathbf{I} + \mathbf{L}\Delta t$$

with eigenvalues $g_{1,2} = 1 \pm i\omega_0\Delta t$, so that

$$|g_{1,2}| = \sqrt{1 + (\omega_0\, \Delta t)^2} > 1 \quad \text{always!!}$$

$\Longrightarrow$EC applied to the harmonic oscillator is never stable.

Explicit Methods

– Euler-Cauchy (from DNGF; see above)

– Leapfrog algorithm (from DST):

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_{n-1} + \mathbf{f}_n \, 2\Delta t + O[(\Delta t)^3] \\ \mathbf{y}_{n+2} &= \mathbf{y}_n + \mathbf{f}_{n+1} \, 2\Delta t + O[(\Delta t)^3] \end{aligned}$$

Example: *Relaxation equation*

$$y_{n+1} = y_{n-1} - 2\Delta t \lambda y_n + O[(\Delta t)^3]$$

Always unstable!

Example: *Harmonic oscillator*

$$\mathbf{y}_{n+1} = 2\Delta t \, \mathbf{L} \cdot \mathbf{y}_n + \mathbf{y}_{n-1}$$

Marginally stable for all Δt and ω_0^2.

Implicit Methods

Much more stable!

– First order scheme (from DNGB): Insert

$$\left.\frac{d\mathbf{y}}{dt}\right|_{n+1} = \frac{\nabla \mathbf{y}_{n+1}}{\Delta t} + O[\Delta t]$$

in $d\mathbf{y}/dt = f[\mathbf{y}(t)]$ to find

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \mathbf{f}_{n+1}\Delta t + O[(\Delta t)^2]$$

If $\mathbf{f}(\mathbf{y})$ is *linear*, $\mathbf{f}_{n+1} = \mathbf{L}\cdot\mathbf{y}_{n+1}$:

$$\boxed{\mathbf{y}_{n+1} = [\mathbf{I} - \mathbf{L}\Delta t]^{-1}\cdot\mathbf{y}_n + O[(\Delta t)^2]}$$

Always stable for relaxation equation and harmonic oscillator.

– Second order implicit scheme (from adding the DNGF formulae at t_n and t_{n+1}, respectively):

$$\mathbf{y}_{n+1} \quad = \quad \mathbf{y}_n + \frac{\Delta t}{2}[\mathbf{f}_n + \mathbf{f}_{n+1}] + O[(\Delta t)^3]$$

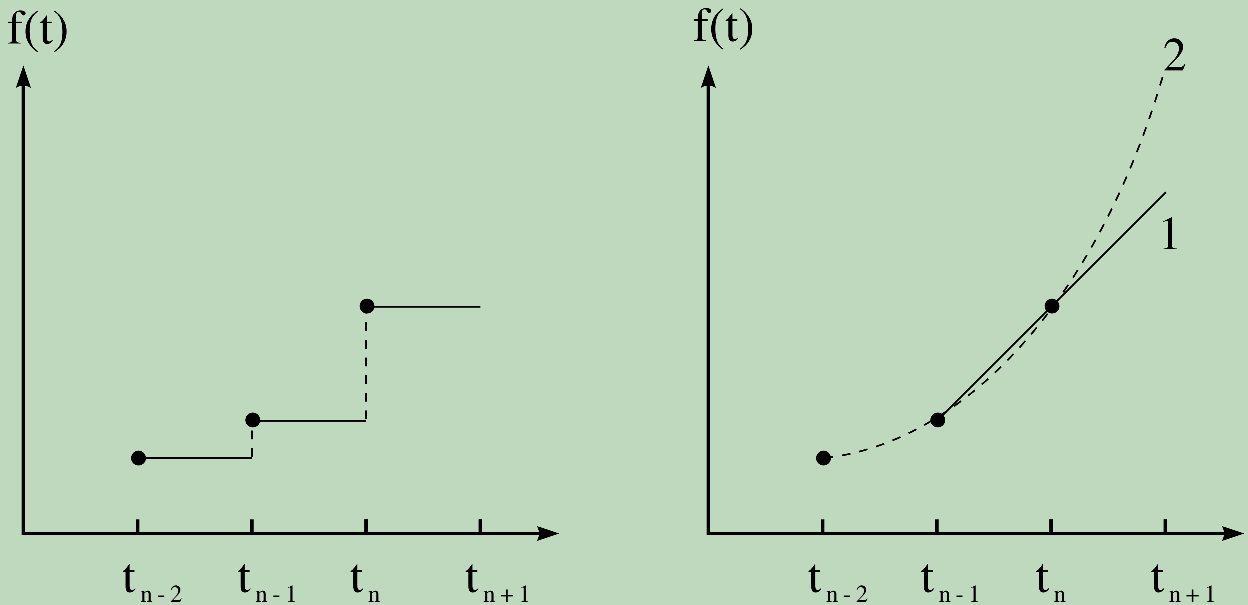
If $\mathbf{f}_n = \mathbf{L} \cdot \mathbf{y}_n$ etc.:

$$\boxed{\mathbf{y}_{n+1} = [\mathbf{I} - \mathbf{L}\tfrac{\Delta t}{2}]^{-1} \cdot [\mathbf{I} + \mathbf{L}\tfrac{\Delta t}{2}] \cdot \mathbf{y}_n + O[(\Delta t)^3]}$$

Always stable for relaxation equation and harmonic oscillator.

Predictor-Corrector Method:

Explicit predictor / *implicit* corrector



(a) (b)

PC method: a) EC ansatz: step function for $f(t)$; b) general predictor-corrector schemes: 1... linear NGB extrapolation; 2... parabolic NGB extrapolation

Predictor step:

– Extrapolate the function $f(t)$, using an NGB polynomial, into $[t_n, t_{n+1}]$.

– Formally integrate the r.h.s. in $dy/dt = f(t)$:
$\Longrightarrow$ *Adams-Bashforth predictor*:

$$y_{n+1}^P = y_n + \Delta t \left[f_n + \frac{1}{2}\nabla f_n + \frac{5}{12}\nabla^2 f_n + \frac{3}{8}\nabla^3 f_n + \ldots \right]$$

– Truncate at some term to obtain the various predictors in the table.

Predictors for first order differential equations:

$$
\begin{aligned}
y_{n+1}^P = y_n \quad &+ \quad \Delta t\, f_n + O[(\Delta t)^2] \quad (\text{Euler} - \text{Cauchy!}) \\
\ldots \quad &+ \quad \frac{\Delta t}{2}[3 f_n - f_{n-1}] + O[(\Delta t)^3] \\
\ldots \quad &+ \quad \frac{\Delta t}{12}[23 f_n - 16 f_{n-1} + 5 f_{n-2}] + O[(\Delta t)^4] \\
\ldots \quad &+ \quad \frac{\Delta t}{24}[55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3}] \\
& \qquad + O[(\Delta t)^5] \\
\vdots
\end{aligned}
$$

Adams-Bashforth predictors

Evaluation step:

As soon as the predictor y^P_{n+1} is available, insert it in $f(y)$:

$$f^P_{n+1} \equiv f[y^P_{n+1}]$$

Corrector step:

– Again back-interpolate the function $f(t)$, using NGB, but now starting at t_{n+1}.

– Formally re-integrate the r.h.s. in $dy/dt = f(t)$:
$\Longrightarrow$ *Adams-Moulton corrector*:

$$y_{n+1} = y_n + \Delta t \left[f_{n+1} - \frac{1}{2}\nabla f_{n+1} - \frac{1}{12}\nabla^2 f_{n+1} - \frac{1}{24}\nabla^3 f_{n+1} - \ldots \right]$$

Correctors for first order differential equations:

$$
\begin{aligned}
y_{n+1} = y_n \quad &+ \quad \Delta t f^P_{n+1} + O[(\Delta t)^2] \\
\dots \quad &+ \quad \frac{\Delta t}{2}[f^P_{n+1} + f_n] + O[(\Delta t)^3] \\
\dots \quad &+ \quad \frac{\Delta t}{12}[5 f^P_{n+1} + 8 f_n - f_{n-1}] + O[(\Delta t)^4] \\
\dots \quad &+ \quad \frac{\Delta t}{24}[9 f^P_{n+1} + 19 f_n - 5 f_{n-1} + f_{n-2}] \\
& \qquad\qquad\qquad\qquad + O[(\Delta t)^5] \\
\vdots \quad &
\end{aligned}
$$

Adams-Moulton correctors

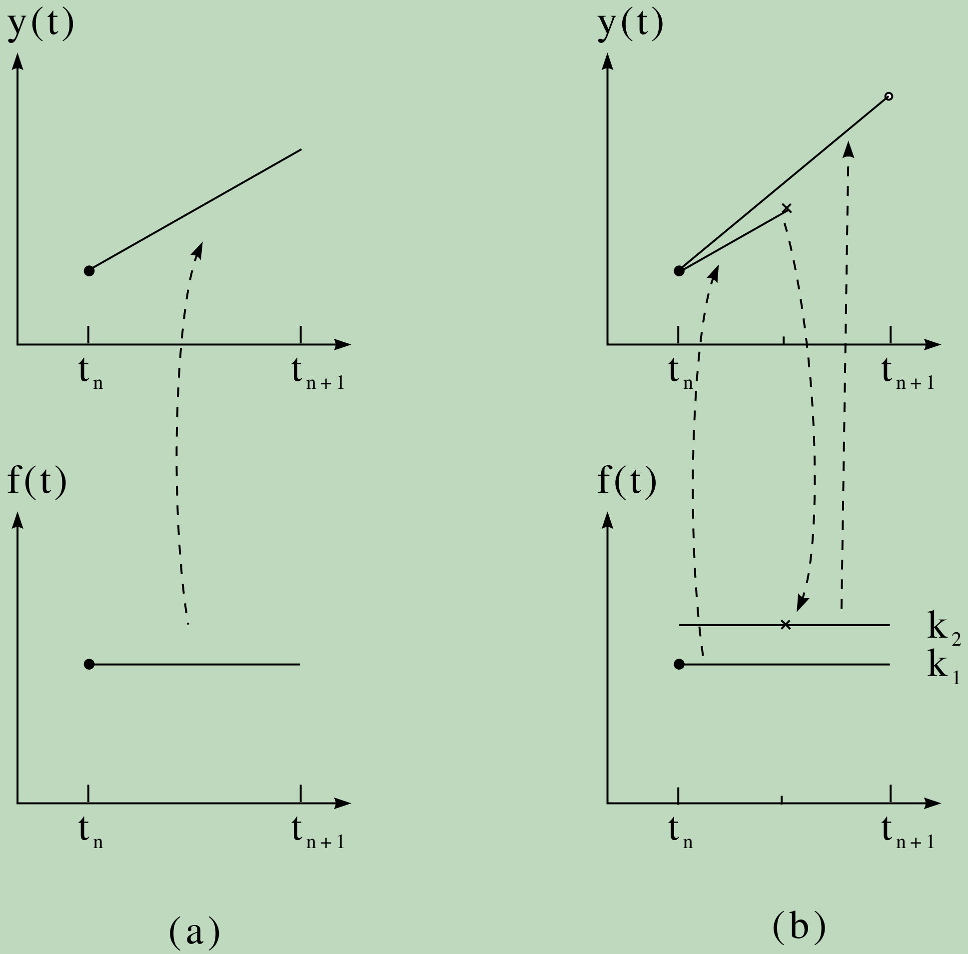
Finally: *evaluate* $f_{n+1} \equiv f(y_{n+1})$.

Stability of PC schemes:

Intermediate between the lousy explicit and the excellent implicit methods.

Example: 2nd order PC + relaxation equation: stable for $\Delta t \leq 2/\lambda$. (The bare predictor would have $\Delta t \leq 1/\lambda$.)

Runge-Kutta Method:



a) EC formula (= RK of first order); b) RK of second order

Runge-Kutta of order 2:

$$\begin{aligned} k_1 &= \Delta t\, f(y_n) \\ k_2 &= \Delta t\, f(y_n + \frac{1}{2}k_1) \\ y_{n+1} &= y_n + k_2 + O[(\Delta t)^3] \end{aligned}$$

(Also called *half-step method*, or *Euler-Richardson algorithm*.)

A much more powerful method that has found wide application is the RK algorithm of order 4, as described in the table.

Runge-Kutta of order 4 for first-order ODE:

$$
\begin{aligned}
k_1 &= \Delta t\, f(y_n) \\
k_2 &= \Delta t\, f(y_n + \frac{1}{2}k_1) \\
k_3 &= \Delta t\, f(y_n + \frac{1}{2}k_2) \\
k_4 &= \Delta t\, f(y_n + k_3) \\
y_{n+1} = y_n &+ \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4] + O[(\Delta t)^5]
\end{aligned}
$$

Advantages of RK:

– Self-starting (no preceding $y_{n-1}\dots$ needed)
– Adjustable Δt

But:

– Several evaluations of $f(y)$ per step; may be too expensive

Stability of RK:

Half-step + relaxation equation: $\Delta t \le 2/\lambda$.

Exercise: Test various algorithms by applying them to an analytically solvable problem, as the harmonic oscillator or the 2-body Kepler problem. Include in your code tests that do not rely on the existence of an analytical solution (energy conservation or such.) Finally, apply the code to more complex problems such as the anharmonic oscillator or the many-body Kepler problem.

Initial Value Problems of Second Order

$$\frac{d^2y}{dt^2} = b[y, dy/dt]$$

Example: generic equation of motion,

$$\frac{d^2\mathbf{r}}{dt^2} = \frac{1}{m}\mathbf{K}[\mathbf{r}(t)]$$

Verlet Method:

Apply DDST,

$$\left.\frac{d^2y}{dt^2}\right|_n = \frac{\delta^2 y_n}{(\Delta t)^2} + O[(\Delta t)^2]$$

to find

$$\boxed{y_{n+1} = 2y_n - y_{n-1} + b_n(\Delta t)^2 + O[(\Delta t)^4]}$$

Note: velocity $v \equiv \dot{y}$ does not appear explicitly. Use the crude estimate

$$v_n = \frac{1}{2\Delta t}[y_{n+1} - y_{n-1}] + O[(\Delta t)^2]$$

Stability of the Verlet scheme:

Harmonic oscillator:

$$y_{n+1} = 2y_n - y_{n-1} - \omega_0^2 y_n (\Delta t)^2$$

leads to

$$g^2 - (2 - \alpha^2)g + 1 = 0$$

with $\alpha \equiv \omega_0 \Delta t$. The root

$$g = (1 - \frac{\alpha^2}{2}) \pm \sqrt{\frac{\alpha^4}{4} - \alpha^2}$$

is imaginary for $\alpha < 2$, with $|g|^2 = 1$.

Equivalent formulations: *Verlet leapfrog* and *Velocity Verlet*.

Verlet leapfrog:

$$\begin{aligned} v_{n+1/2} &= v_{n-1/2} + b_n \Delta t \\ v_n &= \frac{1}{2}(v_{n+1/2} + v_{n-1/2}) \quad \text{(if desired)} \\ y_{n+1} &= y_n + v_{n+1/2}\Delta t + O[(\Delta t)^4] \end{aligned}$$

Leapfrog version of the Verlet method

Velocity Verlet:

$$
\begin{aligned}
y_{n+1} &= y_n + v_n \Delta t + b_n \frac{(\Delta t)^2}{2} + O[(\Delta t)^4] \\
v_{n+1/2} &= v_n + b_n \frac{\Delta t}{2} \\
&\quad Evaluation\ \ step\ \ y_{n+1} \to b_{n+1} \\
v_{n+1} &= v_{n+1/2} + b_{n+1} \frac{\Delta t}{2}
\end{aligned}
$$

Swope's formulation of the Verlet algorithm

Predictor-Corrector Method for 2nd order ODE:

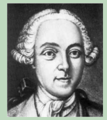
Predictor step:

In $d^2y/dt^2 = b(t)$ replace the function $b(t)$ by a NGB polynomial and integrate twice.

$$\begin{aligned} \dot{y}^P_{n+1}\Delta t - \dot{y}_n \Delta t &= (\Delta t)^2 \left[b_n + \frac{1}{2}\nabla b_n + \frac{5}{12}\nabla^2 b_n + \frac{3}{8}\nabla^3 b_n + \ldots \right] \\ y^P_{n+1} - y_n - \dot{y}_n \Delta t &= \frac{(\Delta t)^2}{2} \left[b_n + \frac{1}{3}\nabla b_n + \frac{1}{4}\nabla^2 b_n + \frac{19}{90}\nabla^3 b_n + \ldots \right] \end{aligned}$$

A specific predictor of order k is found by using terms up to order $\nabla^{k-2} b_n$. Thus the predictor of third order reads

$$\begin{aligned} \dot{y}^P_{n+1}\Delta t - \dot{y}_n \Delta t &= (\Delta t)^2 \left[\frac{3}{2} b_n - \frac{1}{2} b_{n-1} \right] + O[(\Delta t)^4] \\ y^P_{n+1} - y_n - \dot{y}_n \Delta t &= \frac{(\Delta t)^2}{2} \left[\frac{4}{3} b_n - \frac{1}{3} b_{n-1} \right] + O[(\Delta t)^4] \end{aligned}$$



For a compact notation we define the vector

$$\mathbf{b}_k \equiv \{b_n, b_{n-1}, \dots b_{n-k+2}\}^T$$

and the coefficient vectors $\mathbf{c}_k$ and $\mathbf{d}_k$. Then

Predictor of order k for second order DE:

$$\begin{aligned} \dot{y}^P_{n+1}\Delta t - \dot{y}_n \Delta t &= (\Delta t)^2 \mathbf{c}_k \cdot \mathbf{b}_k + O[(\Delta t)^{k+1}] \\ y^P_{n+1} - y_n - \dot{y}_n \Delta t &= \frac{(\Delta t)^2}{2} \mathbf{d}_k \cdot \mathbf{b}_k + O[(\Delta t)^{k+1}] \end{aligned}$$

The first few vectors $\mathbf{c}_k, \mathbf{d}_k$ are given by

$$\mathbf{c}_2 = 1 \qquad \mathbf{d}_2 = 1$$

$$\mathbf{c}_3 = \begin{pmatrix} 3/2 \\ -1/2 \end{pmatrix} \qquad \mathbf{d}_3 = \begin{pmatrix} 4/3 \\ -1/3 \end{pmatrix}$$

Evaluation step:

Insert the preliminary result $y^P_{n+1}, \dot{y}^P_{n+1}$ in the physical law for $b[y,\dot{y}]$:

$$b^P_{n+1} \quad \equiv \quad b\left[y^P_{n+1}, \dot{y}^P_{n+1}\right]$$

Corrector step:

Insert b^P_{n+1} in a NGB formula centered on t_{n+1} and re-integrate twice:

$$\begin{aligned}
\dot{y}_{n+1}\Delta t - \dot{y}_n \Delta t &= (\Delta t)^2 \left[b^P_{n+1} - \frac{1}{2}\nabla b_{n+1} - \frac{1}{12}\nabla^2 b_{n+1} - \frac{1}{24}\nabla^3 b_{n+1} - \ldots\right] \\
y_{n+1} - y_n - \dot{y}_n \Delta t &= \frac{(\Delta t)^2}{2} \left[b^P_{n+1} - \frac{2}{3}\nabla b_{n+1} - \frac{1}{12}\nabla^2 b_{n+1} - \frac{7}{180}\nabla^3 b_{n+1} - \ldots\right]
\end{aligned}$$

Defining the vector

$$\mathbf{b}_k^P \equiv \{b_{n+1}^P, b_n, \ldots b_{n-k+3}\}^T$$

and coefficient vectors $\mathbf{e}_k, \mathbf{f}_k$, we write

Corrector of order k for second-order DE:

$$\begin{aligned} \dot{y}_{n+1}\Delta t - \dot{y}_n \Delta t &= (\Delta t)^2 \mathbf{e}_k \cdot \mathbf{b}_k^P + O[(\Delta t)^{k+1}] \\ y_{n+1} - y_n - \dot{y}_n \Delta t &= \frac{(\Delta t)^2}{2} \mathbf{f}_k \cdot \mathbf{b}_k^P + O[(\Delta t)^{k+1}] \end{aligned}$$

The first few coefficient vectors are

$$\mathbf{e}_2 = 1 \qquad \mathbf{f}_2 = 1$$

$$\mathbf{e}_3 = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \qquad \mathbf{f}_3 = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix}$$

Nordsieck Formulation of the PC Method:

Instead of threading a NGB polynomial through preceding points, expand $y(t)$ about t_n. $\Longrightarrow$ "Taylor predictor" (e.g. of order 3):

$$
\begin{aligned}
y^P_{n+1} &= y_n + \dot{y}_n \Delta t + \ddot{y}_n \frac{(\Delta t)^2}{2!} + \dddot{y}_n \frac{(\Delta t)^3}{3!} + O[(\Delta t)^4] \\
\dot{y}^P_{n+1} \Delta t &= \dot{y}_n \Delta t + \ddot{y}_n (\Delta t)^2 + \dddot{y}_n \frac{(\Delta t)^3}{2!} + O[(\Delta t)^4] \\
\ddot{y}^P_{n+1} \frac{(\Delta t)^2}{2!} &= \ddot{y}_n \frac{(\Delta t)^2}{2!} + \dddot{y}_n \frac{(\Delta t)^3}{2!} + O[(\Delta t)^4] \\
\dddot{y}^P_{n+1} \frac{(\Delta t)^3}{3!} &= \dddot{y}_n \frac{(\Delta t)^3}{3!} + O[(\Delta t)^4]
\end{aligned}
$$

Defining the vector

$$\mathbf{z}_n \equiv \begin{pmatrix} y_n \\ \dot{y}_n \Delta t \\ \ddot{y}_n \frac{(\Delta t)^2}{2!} \\ \vdots \end{pmatrix}$$

and the (Pascal triangle) matrix

$$\mathbf{A} \quad \equiv \quad \begin{pmatrix} 1 & 1 & 1 & 1 & \ldots \\ 0 & 1 & 2 & 3 & \ldots \\ 0 & 0 & 1 & 3 & \ldots \\ & \ddots & \ddots & 1 & \ddots \\ & & & & \ddots \end{pmatrix}$$

we have

$$\boxed{\mathbf{z}_{n+1}^P = \mathbf{A} \cdot \mathbf{z}_n}$$

Now evaluate: $\Longrightarrow$

Evaluation step:

Insert $\mathbf{z}_{n+1}^P$ in the force law: $b_{n+1}^P \equiv b[y_{n+1}^P, \dot{y}_{n+1}^P]$

Corrector step:

Define the deviation $\gamma \equiv [b_{n+1}^P - \ddot{y}_{n+1}^P]\frac{(\Delta t)^2}{2}$ and write the *corrector* as

$$\boxed{\mathbf{z}_{n+1} = \mathbf{z}_{n+1}^P + \gamma \mathbf{c}}$$

with an optimized coefficient vector $\mathbf{c}$. The first few vectors are

$$\mathbf{c} = \begin{pmatrix} 1/6 \\ 5/6 \\ 1 \\ 1/3 \end{pmatrix}, \quad \begin{pmatrix} 19/120 \\ 3/4 \\ 1 \\ 1/2 \\ 1/12 \end{pmatrix}, \quad \begin{pmatrix} 3/20 \\ 251/360 \\ 1 \\ 11/18 \\ 1/6 \\ 1/60 \end{pmatrix}, \ldots$$

Advantages of Nordsieck PC:

– Self-starting
– Adjustable time steps

Stability: Again, between explicit (bad) and implicit (good).

Runge-Kutta Method for 2nd order ODE:

4th order RK for velocity independent forces:
Let $b = b(y)$; then

$$\begin{aligned}
b_1 &= b\left[y_n\right] \\
b_2 &= b\left[y_n + \dot{y}_n \frac{\Delta t}{2}\right] \\
b_3 &= b\left[y_n + \dot{y}_n \frac{\Delta t}{2} + b_1 \frac{(\Delta t)^2}{4}\right] \\
b_4 &= b\left[y_n + \dot{y}_n \Delta t + b_2 \frac{(\Delta t)^2}{2}\right] \\
\dot{y}_{n+1} = \dot{y}_n &+ \frac{\Delta t}{6}[b_1 + 2b_2 + 2b_3 + b_4] + O[(\Delta t)^5] \\
y_{n+1} = y_n &+ \dot{y}_n \Delta t + \frac{(\Delta t)^2}{6}[b_1 + b_2 + b_3] + O[(\Delta t)^5]
\end{aligned}$$

4th order RK for velocity dependent forces:

If $b = b(y, \dot{y})$, use

$$
\begin{aligned}
b_1 &= b[y_n, \dot{y}_n] \\
b_2 &= b\left[y_n + \dot{y}_n\frac{\Delta t}{2} + b_1\frac{(\Delta t)^2}{8}, \dot{y}_n + b_1\frac{\Delta t}{2}\right] \\
b_3 &= b\left[y_n + \dot{y}_n\frac{\Delta t}{2} + b_1\frac{(\Delta t)^2}{8}, \dot{y}_n + b_2\frac{\Delta t}{2}\right] \\
b_4 &= b\left[y_n + \dot{y}_n\Delta t + b_3\frac{(\Delta t)^2}{2}, \dot{y}_n + b_3\Delta t\right] \\
\dot{y}_{n+1} = \dot{y}_n &+ \frac{\Delta t}{6}[b_1 + 2b_2 + 2b_3 + b_4] + O[(\Delta t)^5] \\
y_{n+1} = y_n &+ \dot{y}_n\Delta t + \frac{(\Delta t)^2}{6}[b_1 + b_2 + b_3] + O[(\Delta t)^5]
\end{aligned}
$$

Advantages of RK:

– Self-starting
– Adjustable time steps

But: Repeated evaluation of the acceleration $b(y)$ in a single time step

Exercise: Write a code that permits to solve a given second-order equation of motion by various algorithms. Apply the program to problems of point mechanics and explore the stabilities and accuracies of the diverse techniques.

Boundary Value Problems

Examples:

– Poisson's and Laplace's equations, $d^2\phi/dx^2 = -\rho(x)$, or

$$\frac{d\phi}{dx} = -e$$
$$\frac{de}{dx} = \rho(x)$$

where $\rho(x)$ is a charge density. (Laplace: $\rho(x) = 0$). Another physical problem described by the same equation is the temperature distribution along a thin rod: $d^2T/dx^2 = 0$.

– Time independent Schroedinger equation for a particle of mass m in a potential $U(x)$:

$$\frac{d^2\psi}{dx^2} = -g(x)\psi\,, \quad \text{with} \quad g(x) = \frac{2m}{\hbar^2}[E - U(x)]$$

The general 1-dimensional BVP reads

$$\frac{dy_i}{dx} = f_i(x, y_1, \dots y_N)\,; \quad i = 1, \dots N$$

with N boundary values required. Typically there are

n_1 boundary values a_j $(j = 1, \dots n_1)$ at $x = x_1$, and
n_2 $\equiv N - n_1$ boundary values b_k $(k = 1, \dots n_2)$ at $x = x_2$.

The quantities y_i, a_j and b_k may simply be higher derivatives of a single solution function $y(x)$. Two methods are available:

– **Shooting method**
– **Relaxation technique**

Shooting Method:

– Transform the given *boundary* value problem into an *initial* value problem with estimated parameters

– Adjust the parameters iteratively to reproduce the given boundary values

First trial shot: Augment the n_1 boundary values given at $x = x_1$ by $n_2 \equiv N - n_1$ *estimated* parameters

$$\mathbf{a}^{(1)} \equiv \{a_k^{(1)};\ k = 1, \ldots n_2\}^T$$

to obtain an IVP. Integrate numerically up $x = x_2$. The newly calculated values of b_k at $x = x_2$,

$$\mathbf{b}^{(1)} \equiv \{b_k^{(1)};\ k = 1, \ldots n_2\}^T$$

will in general deviate from the given boundary values $\mathbf{b} \equiv \{b_k; \ldots\}^T$. The difference vector $\mathbf{e}^{(1)} \equiv \mathbf{b}^{(1)} - \mathbf{b}$ is stored for further use.

Second trial shot: Change the estimated initial values a_k by some small amount, $\mathbf{a}^{(2)} \equiv \mathbf{a}^{(1)} + \delta\mathbf{a}$, and once more integrate up to $x = x_2$. The values $b_k^{(2)}$ thus obtained are again different from the required values b_k: $\mathbf{e}^{(2)} \equiv \mathbf{b}^{(2)} - \mathbf{b}$.

Quasi-linearization: Assuming that the deviations $\mathbf{e}^{(1)}$ and $\mathbf{e}^{(2)}$ depend *linearly* on the estimated initial values $\mathbf{a}^{(1)}$ and $\mathbf{a}^{(2)}$, compute that vector $\mathbf{a}^{(3)}$ which would make the deviations disappear:

$$\mathbf{a}^{(3)} = \mathbf{a}^{(1)} - \mathbf{A}^{-1} \cdot \mathbf{e}^{(1)}, \text{ with } A_{ij} \equiv \frac{b_i^{(2)} - b_i^{(1)}}{a_j^{(2)} - a_j^{(1)}}$$

Iterate the procedure up to some desired accuracy.

Example:

$$\frac{d^2y}{dx^2} = -\frac{1}{(1+y)^2} \quad \text{with} \quad y(0) = y(1) = 0$$

* *First trial shot:* Choose $a^{(1)} \equiv y'(0) = 1.0$. Applying 4th order RK with $\Delta x = 0.1$ we find $b^{(1)} \equiv y_{calc}(1) = 0.674$. Thus $e^{(1)} \equiv b^{(1)} - y(1) = 0.674$.

* *Second trial shot:* With $a^{(2)} = 1.1$ we find $b^{(2)} = 0.787$, i.e. $e^{(2)} = 0.787$.

* *Quasi-linearization:* From

$$a^{(3)} = a^{(1)} - \frac{a^{(2)} - a^{(1)}}{b^{(2)} - b^{(1)}} e^{(1)}$$

we find $a^{(3)} = 0.405 \, (\equiv y'(0))$.

Iteration: The next few iterations yield the following values for $a \, (\equiv y'(0))$ and $b \, (\equiv y(1))$:

n	$a^{(n)}$	$b^{(n)}$
3	0.405	− 0.041
4	0.440	0.003
5	0.437	0.000

Relaxation Method:

Discretize x to transform a given DE into a set of algebraic equations. For example, applying DDST to

$$\frac{d^2y}{dx^2} = b(x,y)$$

we find

$$\frac{d^2y}{dx^2} \approx \frac{1}{(\Delta x)^2}[y_{i+1} - 2y_i + y_{i-1}]$$

which leads to the set of equations

$$y_{i+1} - 2y_i + y_{i-1} - b_i(\Delta x)^2 = 0\,, \quad i = 2, \ldots M-1$$

Since we have a BVP, y_1 and y_M will be given.

* Let $\mathbf{y}^{(1)} \equiv \{y_i\}$ be an inaccurate (estimated?) solution. The error components

$$e_i = y_{i+1} - 2y_i + y_{i-1} - b_i(\Delta x)^2\,, \;\; i = 2, \ldots M-1$$

together with $e_1 = e_M = 0$ then define an error vector $\mathbf{e}^{(1)}$.

* How to modify $\mathbf{y}^{(1)}$ to make $\mathbf{e}^{(1)}$ disappear? $\Longrightarrow$Expand e_i linearly:

$$\begin{aligned} e_i(y_{i-1} + \Delta y_{i-1}, y_i + \Delta y_i, y_{i+1} + \Delta y_{i+1}) \approx \\ \approx e_i + \frac{\partial e_i}{\partial y_{i-1}} \Delta y_{i-1} + \frac{\partial e_i}{\partial y_i} \Delta y_i + \frac{\partial e_i}{\partial y_{i+1}} \Delta y_{i+1} \\ \equiv e_i + \alpha_i \Delta y_{i-1} + \beta_i \Delta y_i + \gamma_i \Delta y_{i+1} \;\; (i = 1, \ldots M) \end{aligned}$$

This modified error vector is called $\mathbf{e}^{(2)}$. We want it to vanish, $\mathbf{e}^{(2)} = 0$:

$$\mathbf{A} \cdot \Delta \mathbf{y} = -\mathbf{e}^{(1)} \quad \text{with} \quad \mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & \ldots \\ \alpha_2 & \beta_2 & \gamma_2 & 0 \\ & \ddots & \ddots & \ddots \\ & & 0 & 1 \end{pmatrix}$$

Thus our system of equations is *tridiagonal*: $\Longrightarrow$*Recursion technique!*

Example:

$$\frac{d^2y}{dx^2} = -\frac{1}{(1+y)^2} \quad \text{with} \quad y(0) = y(1) = 0$$

DDST leads to $e_i = y_{i+1} - 2y_i + y_{i-1} + (\Delta x)^2/(1+y_i)^2$. Expand:

$$\alpha_i \equiv \frac{\partial e_i}{\partial y_{i-1}} = 1\,; \quad \gamma_i \equiv \frac{\partial e_i}{\partial y_{i+1}} = 1\,; \quad \beta_i \equiv \frac{\partial e_i}{\partial y_i} = -2\left[1 + \frac{(\Delta x)^2}{(1+y_i)^3}\right] \quad i = 2, \ldots M-1$$

Start the downwards recursion: $g_{M-1} = -\alpha_M/\beta_M = 0$ and $h_{M-1} = -e_M/\beta_M = 0$.

$$g_{i-1} = \frac{-\alpha_i}{\beta_i + \gamma_i g_i} = \frac{-1}{\beta_i + g_i}\,; \quad h_{i-1} = \frac{-e_i - h_i}{\beta_i + g_i}$$

brings us down to g_1, h_1. Putting

$$\Delta y_1 = \frac{-e_1 - \gamma_1 h_1}{\beta_1 + \gamma_1 g_1} = e_1 (= 0)$$

we take the upwards recursion

$$\Delta y_{i+1} = g_i \Delta y_i + h_i\,; \quad i = 1, \ldots M-1$$

Improve $y_i \longrightarrow y_i + \Delta y_i$ and iterate.