Numerical Integration ODEs II

S. Eggl

Numerical Integration
ODEs II

S. Eggl[1]

[1] Institute for Astronomy,
University of Vienna,
Vienna, Austria.

Practical in numerical Astronomy
Vienna
Mai 2010

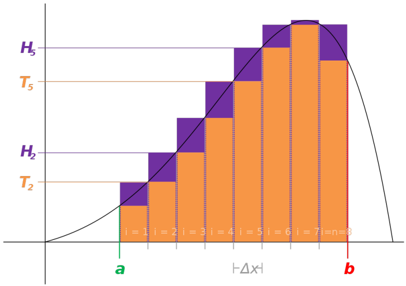
Outline

A long long time ago...

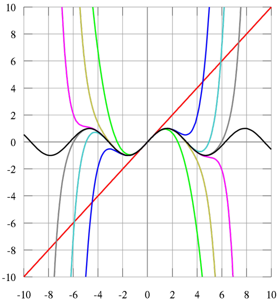
Two "philosophies"

Algorithm Types

Geometry-based (Collocation)



Taylor-based



Algorithm Types II

Collocation	**Taylor-based**
Newton-Cotes	Runge-Kutta

A few odd ideas... Extrapolation

Bulirsch-Stoer Method

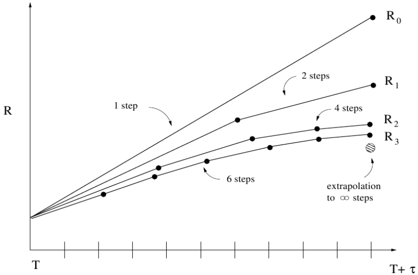
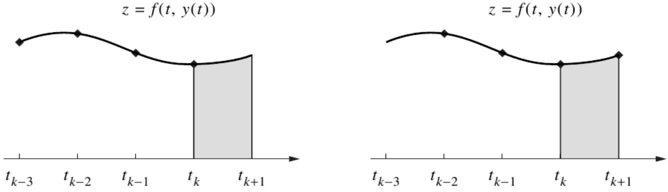


Fig. 2: Bulirsch-Stoer method. The results R_m after a time-step τ are sampled with different numbers of sub-steps $\frac{\tau}{n_m}$. These results are seen as a function of the number of sub-steps, and will finally be extrapolated to a value R_∞, that represents - in principle - the solution of a differential equation calculated with a (sub-)stepsize of $\tau_m = 0$.

A few odd ideas... Predictor Corrector

Adams - Bashforth - Moulton - Predictor - Corrector



(a) The four nodes for the Adams-Bashforth predictor (extrapolation is used)

(a) The four nodes for the Adams-Moulton corrector (interpolation is used)

Figure 9.10 Integration over $[t_k, t_{k-1}]$ in the Adams-Bashforth method.

ref: John H. Mathews

Predictor Corrector II

Adams-Bashforth-Moulton Method

The Adams-Bashforth-Moulton predictor-corrector method is a multistep method derived from the fundamental theorem of calculus:

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(t, y(t))\, dt. \tag{1}$$

The predictor uses the Lagrange polynomial approximation for $f(t, y(t))$ based on the points (t_{k-3}, f_{k-3}), (t_{k-2}, f_{k-2}), (t_{k-1}, f_{k-1}), and (t_k, f_k). It is integrated over the interval $[t_k, t_{k+1}]$ in (1). This process produces the Adams-Bashforth predictor:

$$p_{k+1} = y_k + \frac{h}{24}(-9 f_{k-3} + 37 f_{k-2} - 59 f_{k-1} + 55 f_k). \tag{2}$$

Predictor Corrector III

The corrector is developed similarly. The value p_{k+1} just computed can now be used. A second Lagrange polynomial for $f(t, y(t))$ is constructed, which is based on the points (t_{k-2}, f_{k-2}), (t_{k-1}, f_{k-1}), (t_k, f_k), and the new point $(t_{k+1}, f_{k+1}) = (t_{k+1}, f(t_{k+1}, p_{k+1}))$. This polynomial is then integrated over $[t_k, t_{k+1}]$, producing the Adams-Moulton corrector:

$$y_{k+1} = y_k + \frac{h}{24}(f_{k-2} - 5 f_{k-1} + 19 f_k + 9 f_{k+1}). \tag{3}$$

Figure 9.10 shows the nodes for the Lagrange polynomials that are used in developing formulas (2) and (3), respectively.

Predictor Corrector IV

Error Estimation and Correction

The error terms for the numerical integration formulas used to obtain both the predictor and corrector are of the order $\boldsymbol{O}(h^5)$. The L.T.E. for formulas (2) and (3) are

$$(4) \qquad y(t_{k+1}) - p_{k+1} = \frac{251}{720} y^{(5)}(c_{k+1})h^5 \qquad \text{(L.T.E. for the predictor)},$$

$$(5) \qquad y(t_{k+1}) - y_{k+1} = \frac{-19}{720} y^{(5)}(d_{k+1})h^5 \qquad \text{(L.T.E. for the corrector)}.$$

Suppose that h is small and $y^{(5)}(t)$ is nearly constant over the interval; then the terms involving the fifth derivative in (4) and (5) can be eliminated, and the result is

$$(6) \qquad y(t_{k+1}) - y_{k+1} \approx \frac{-19}{270}(y_{k+1} - p_{k+1}).$$

The importance of the predictor-corrector method should now be evident. Formula (6) gives an approximate error estimate based on the two computed values p_{k+1} and y_{k+1} and does not use $y^{(5)}(t)$.

Structure Preserving Algorithms I

Hamiltonian Mechanics I

$$H(q,p,t) = T(q,p,t) + U(q,p,t)$$

$$\dot{q} = \frac{\partial H(q,p,t)}{\partial p} \qquad \dot{p} = -\frac{\partial H(q,p,t)}{\partial q}$$

Structure Preserving Algorithms II

Neat...

$$\frac{dH}{dt} = \frac{\partial H}{\partial t}$$

$$\frac{\partial H}{\partial t} = 0 \rightarrow \frac{dH}{dt} = 0 \rightarrow H = const$$

Structure Preserving Algorithms III

Hamiltonian Mechanics II

$$\vec{z} = \binom{\vec{q}}{\vec{p}}$$

$$\dot{\vec{z}} = J \cdot \vec{\nabla} H(\vec{z})$$

Structure Preserving Algorithms IV

Example: Symplectic Euler-Cromer

$$p_{n+1} = p_n - h\frac{\partial H}{\partial q}|_{p_{n+1},q_n}$$

$$q_{n+1} = q_n + h\frac{\partial H}{\partial p}|_{p_{n+1},q_n}$$

Structure Preserving Algorithms V

$$H(q,p) = \frac{p^2}{2} + k\frac{q^2}{2}$$

Example: Symplectic Euler-Cromer for the Harmonic Oscillator

$$p_{n+1} = p_n - h \cdot k \cdot q_n$$

$$q_{n+1} = q_n + h \cdot p_{n+1}$$

Structure Preserving Algorithms VI

Symplectic Structure

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

a symplectic algorithm keeps J intact

$$M^T J M = J$$

Structure Preserving Algorithms VII

M is the Jacobian of the flow $\varphi(\vec{z}, t)$ of an ODE

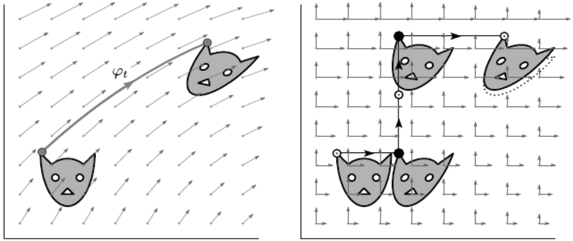


Fig. 2.2. Symplecticity of the Störmer/Verlet method for a separable Hamiltonian.

Structure Preserving Algorithms VIII

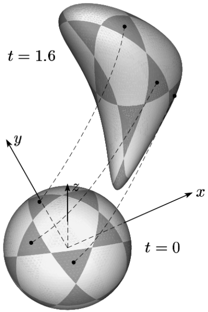
Numerical flow $\varphi(\vec{z}, [t_n, t_{n+1}])$ for a mapping

$$\vec{z}_n \ \rightarrow \vec{z}_{n+1} = \binom{q_n}{p_n} \rightarrow \binom{q_{n+1}}{p_{n+1}}$$

$$M = \begin{pmatrix} \frac{\partial q_{n+1}}{\partial q_n} & \frac{\partial q_{n+1}}{\partial p_n} \\ \frac{\partial p_{n+1}}{\partial q_n} & \frac{\partial p_{n+1}}{\partial p_n} \end{pmatrix}$$

Structure Preserving Algorithms IX

$det M = 1 \rightarrow$ phase-space volume conserved!



Howto? Splitting Methods

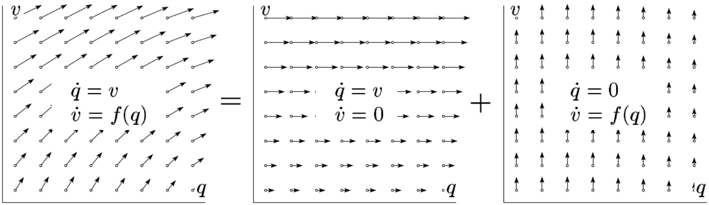
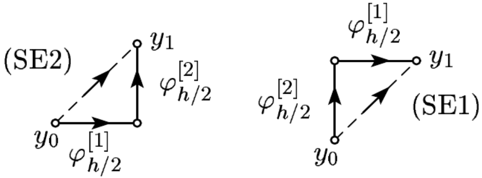


Fig. 1.3. The phase space vector field split into two fields

Howto? Splitting Methods II

$$\varphi_t^{[1]} : \begin{cases} q_1 = q_0 + t \cdot v_0 \\ v_1 = v_0 \end{cases} \qquad \varphi_t^{[2]} : \begin{cases} q_1 = q_0 \\ v_1 = v_0 + t \cdot f(q_0) . \end{cases}$$



Structure Preserving Algorithms V

$$H(q,p) = \frac{p^2}{2} + k\frac{q^2}{2}$$

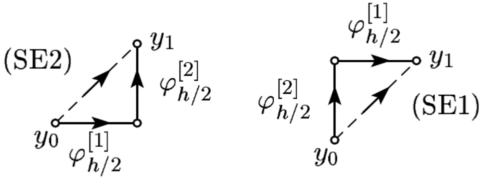
Example: Symplectic Euler-Cromer for the Harmonic Oscillator

$$p_{n+1} = p_n - h \cdot k \cdot q_n$$

$$q_{n+1} = q_n + h \cdot p_{n+1}$$

Howto? Splitting Methods II

$$\varphi_t^{[1]} : \begin{cases} q_1 = q_0 + t \cdot v_0 \\ v_1 = v_0 \end{cases} \qquad \varphi_t^{[2]} : \begin{cases} q_1 = q_0 \\ v_1 = v_0 + t \cdot f(q_0)\,. \end{cases}$$



What really happens... I

Hamilton's equations can be rewritten using Poisson's differential operator D_H.

$$\dot{\vec{z}} = \{\vec{z}, H(\vec{z})\}$$

$$\dot{\vec{z}} = D_H \vec{z}$$

with

$$\vec{z} = \begin{pmatrix} \vec{q} \\ \vec{p} \end{pmatrix}$$

$$D_H = \{\_, H\}$$

What really happens... II

Poisson brackets

$$\{F,G\} = \sum_{i=1}^{n} \left(\frac{\partial F}{\partial q_i} \frac{\partial G}{\partial p_i} - \frac{\partial F}{\partial p_i} \frac{\partial G}{\partial q_i} \right)$$

The formal solution

$$\vec{z}(h) = e^{hD_H} \vec{z}(0)$$

$$\vec{z}(h) = e^{h(D_T + D_U)} \vec{z}(0)$$

What really happens... III

$$e^{hD_T} e^{hD_U} = e^{(h(D_T+D_U)+\frac{h^2}{2}[D_T,D_U]+\frac{h^3}{12}([D_T,[D_T,D_U]]-...)}$$

with

$$[D_T, D_U] = D_T D_U - D_U D_T$$

expansion with coefficients a^i and b^i to cancel out unwanted terms containing commutators up to $O(h^{k+1})$.

$$e^{h(D_T+D_U)} \overset{O(h^{k+1})}{=} \prod_{i=1}^{k} e^{a^i h D_T} e^{b^i h D_U}$$

What really happens... IV

OOOPS....

$$e^{hD_H} = e^{h(D_T + D_U)}$$
$$H = T + U$$
$$e^{hD_T} e^{hD_U} = e^{hD_{\tilde{H}}}$$

solving not my origninal but a close by Hamiltonian...

$$\tilde{H} = T + U + \frac{h}{2}[T, U] + \frac{h^2}{12}([T, [T, U]] - [U, [T, U]]) + ...$$

Thank you for your attention

References:

Eggl,S., Dvorak,R. *An Introduction to Common Numerical Integration Codes Used in Dynamical Astronomy* , Lecture Notes in Physics Vol 790, Springer, *p. 431-477* (2010)

Hairer, E., Lubich, C., Wanner, G.: *Geometric Numerical Integration, Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer, 2nd edition (2006)

Mathews, J.H., Fink, K.D.: *Numerical Methods using Matlab*,Prentice-Hall Inc., 4th edition (2004)