Adaptive Lie-integration

5th Austrian-Hungarian Workshop
on Trojans and Related Topics

9 – 10 April, 2010

Ordinary differential equations

Problem: let's solve the equation

$$\dot{x}_i = f_i(\mathbf{x}) \tag{1}$$

where $\mathbf{x} : \mathbb{R} \to \mathbb{R}^N$ $(\mathbf{x} = (x_1, \ldots, x_N))$. Note: all non-autonomous equation can be transformed into such a form by introducing a new variable. Numerical methods for solving such equations:

- classic explicit methods (e.g. RKn, MMID, BS);
- symplectic mappings (for special /Hamiltonian/ problems, e.g. Leap-Frog);
- implicit methods (e.g. modified Euler);
- Lie-integration: the power series expansion of the solution is computed and the coefficients are then summed appropriately.

Adatptive integration methods

Basic problem: the numerical solution is performed with a given stepsize, however, it is not obvious what is the "optimal" stepsize in order to obtain a certain (relative or absolute) precision:

- analytic estimations for this optimal stepsize; or
- direct variations (until the desired precision is obtained).

With the exception of the Euler method, all of the explicit methods must compute the right-hand side of the ODE in instances that depend on the stepsize $\Rightarrow$ if it turns out to be too small or too large, stepsize variation yields CPU time loss.

Lie-integration

Formally, the solution of the differential equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{2}$$

(where $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$) can be written as

$$\mathbf{x}(t + \Delta t) = \exp(\Delta t L)\mathbf{x}(t), \tag{3}$$

where $L = \sum_{i=1}^{N} f_i D_i$ and $D_i = \frac{\partial}{\partial x_i}$ (L is the so-called Lie-operator). The exponential function can be expanded as:

$$\exp(\Delta t L) = \sum_{k=0}^{\infty} \frac{\Delta t^k}{k!} L^k. \tag{4}$$

The Lie-integration is the finite approximation of the sum in equation (4) (see, e.g., Hanslmeier & Dvorak, 1984, A&A).

Properties of the Lie-integration

Advantages:

- yields the coefficients of the Taylor-expansion (of course, these can be exploited for other purposes as well, example: transit light curve asymmetries due to an eccentric orbit: $M \leq 3 \ldots 5$);
- the coefficients are computed using recurrence relations: the derivatives $L^{n+1}x_i$ are written as the functions of the derivatives $L^k x_j$ ($0 \leq k \leq n$);
- if the coefficients are known $\Rightarrow$ the computation of the sum is extremely fast, for arbitrary values of Δt
- all in all: a very fast method

Disadvantages:

- For each problem (differential equation), we need a different set of recurrence relations that should be derived independently. It is highly not obvious and such a derivation requires some sort of intuition.

All in all, the Lie-integration is not a widespread method, although it is definitely more effective than the other techniques.

Linearized equations

The original set of ODEs ($\mathbf{x} : \mathbb{R} \to \mathbb{R}^N$) and its linearized ($\xi : \mathbb{R} \to \mathbb{R}^N$):

$$\begin{aligned} \dot{x}_i &= f_i(\mathbf{x}), \\ \dot{\xi}_i &= \sum_{m=1}^{N} \xi_m \frac{\partial f_i(\mathbf{x})}{\partial x_m}. \end{aligned}$$

Using the notations introduced earlier:

$$L = L_0 + L_\ell = f_i D_i + \xi_m D_m f_i \partial_i, \tag{5}$$

where $D_m = \frac{\partial}{\partial x_m}$ and $\partial_i = \frac{\partial}{\partial \xi_i}$ (thus, $L_0 = f_i D_i$ and $L_\ell = \xi_m D_m f_i \partial_i$). This extension of the original ODE does not modify the formal solution of equation (3), since $L_0 \xi_i \equiv 0$ for all $i = 1, \ldots, N$.

Solving the linearized equations

We can write the solution similarly to the original equations (see Pál & Süli 2007, MNRAS):

$$\xi(t + \Delta t) = \exp(\Delta t L)\xi(t). \tag{6}$$

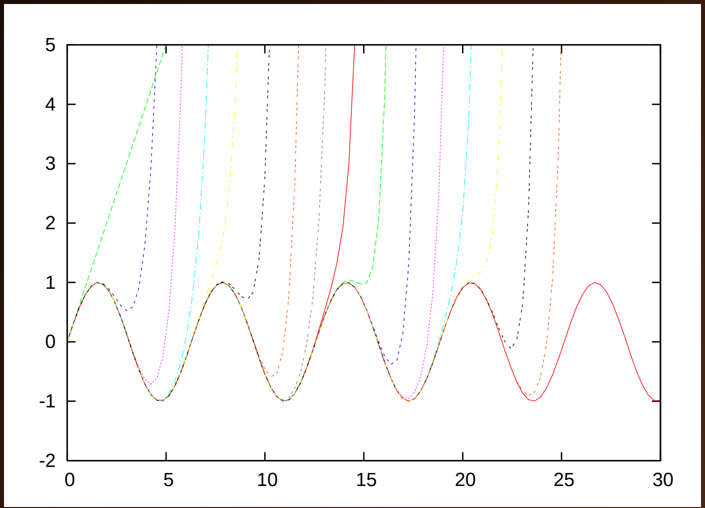
It has been proven that the derivatives $L^n \xi_k = (L_0 + L_\ell)^n \xi_k$ can be computed in a simpler manner, namely:

$$L^n \xi_k = \xi_m D_m L^n x_k = \xi_m D_m L_0^n x_k. \tag{7}$$

On the right-hand side, there are only functions of the D_m derivatives (in practice, in the form of $D_m L_0^n$).

Adaptive integration – I.

An example: the Taylor-expansion of a periodic function:



The expansion of the sine function up to the order of 61.

To obtain a certain precision, the integration order is roughly proportional to the integration stepsize.

Adaptive integration – II.

Simple algorithm: let us define a minimal and maximal integration (polynomial) order: $M_{\min}$ and $M_{\max}$

1. the integration (i.e. the computation of the coefficients and the summation) is performed for a given Δt stepsize.

2. if the desired precision (δ) is reached earlier (so, $M < M_{\min}$), then Δt is multiplied by $M_{\max}/M_{\min}$ and the sum of the power series is calculated again (including the computation of the subsequent Lie-derivatives). This step might have to be repeated until M reaches $M_{\min}$.

3. If the precision δ is not obtained before the order $M_{\max}$, then Δt is multiplied (decreased) by $M_{\min}/M_{\max}$ and the sum is computed (like above, this step is repeated until convergence).

4. If the given precision is reached between $M_{\min}$ and $M_{\max}$, we proceed with the next integration step.

In practice, even the machine precision ($\delta \approx 2 \cdot 10^{-16}$, for IEEE 64 bit numbers, `double` types) can be reached without any additional tricks!

Adaptive integration – III.

Some hints:

- Choices for $M_{\rm min}$ and $M_{\rm max}$: make the integration as fast as possible.
- Of course, it depends on the problem, the actual implementation and the value of δ. In practice, $M_{\rm min} \approx 16$ and $M_{\rm max} \approx 20$ is a good choice for the N-body problem and for machine precision.

Applications:

- Time series analysis: the model function can be threated as an analytical function even if it can be derived only as a solution of an ODE;
- For this analysis, one needs: time series <u>and</u> the parametric derivatives (see, e.g.: linear regression, nonlinear Levenberg-Marquardt fit, error propagation and estimation of the uncertainties using Fisher analysis).

Adaptive integration – IV – the N-body problem

Propertites of a “regular” planetary system: almost circular orbits; no orbital intersections and regular motion on shorter timescales.
How can the adaptive Lie-integration be made more efficient:

- the integration order is not the same for the bodies;
- inner planets: higher orders for a certain stepsize (the orbital curveture is larger, see the figure about the sine function), outer planets: a smaller order is adequate;
- “crosstalk” between the coefficients: terms related to the interaction between the central body and the given planet have to be computed up to a higher order than the terms related to mutual interactions.
- $1 \ll N$-body systems: although the initialization of the integration requires $\mathcal{O}(N^2)$ operation, we might save CPU time during the computation of the $1 \leq k$ Lie-coefficients by employing such an algorithm, thus such an implementation might be an $\mathcal{O}(N^p)$ one (where $1 \leq p < 2$).

To be done, under construction, ongoing study, etc ...

Applications

Analytical investigations of ODE solutions: there is a quantity Q that depends on the solution itself: $Q \equiv Q(\mathbf{x}(t))$. Problem: what are the parametric derivatives of Q with respect to the initial conditions ($\mathbf{x}^0 \equiv \mathbf{x}|_{t=0}$)? These are:

$$\frac{\partial Q}{\partial x_\ell^0} = \mathcal{Z}_{\ell k} \frac{\partial Q}{\partial x_k}, \tag{8}$$

while $\mathcal{Z}_{\ell k}$ is the solution of the full linearized set of equations:

$$\dot{\mathcal{Z}}_{\ell k} = \mathcal{Z}_{\ell m} \frac{\partial f_k(\mathbf{x})}{\partial x_m}, \tag{9}$$

with the initial conditions of $\mathcal{Z}_{\ell k}|_{t=0} = \delta_{\ell k}$.

Additionally, the chain rule can be applied if it is neccessary.

Applications – analysis of RV curves

Radial velocity variations caused by multiple planetary companions:

- N-body problem;
- ordinary differential equations, Lie-series are known;
- parameters: orbital elements and the observed RV amplitude (chain rule: it is not so simple to apply)
- observed quantity (denoted as Q earlier): radial velocity: the classic solution is the linear combination of each component (thus, the derivatives $\partial Q/\partial x_k$ can be computed easily);

Therefore: the previously introduced algorithms and methods can be applied for RV analysis. In other words, an RV time series can be treated as simple as any well-known ordinary analytical function (linear function, trigonometric, etc.)

Applications – HD 73526 – I.

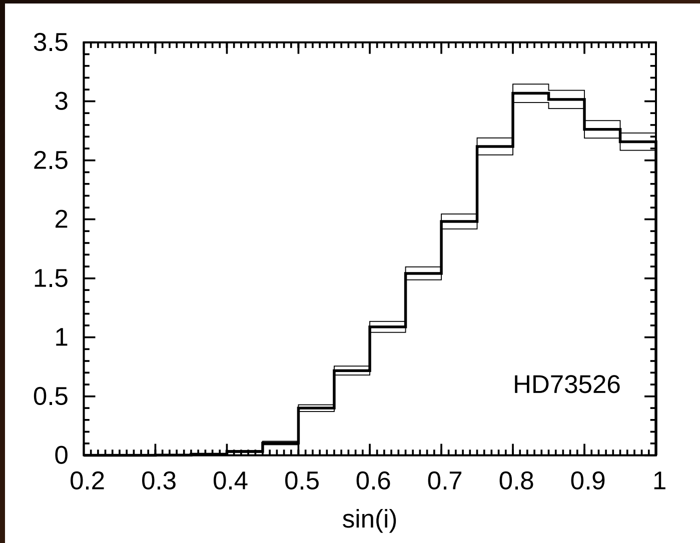
- Two planets, nearly 2:1 mean motion resonance.
- A simple question: can the orbital inclination be derived purely from RV data? Parameters:

$$N \times \{\mathcal{K}, n, \lambda_0, e\cos\omega, e\sin\omega\}, M_\star, \sin i.$$

- Small inclination $\Rightarrow$ larger masses (i.e. $m \sin i$ is given) $\Rightarrow$ stronger perturbations.
- The whole RV tiem series is treated and modelled as an analytical function. Important: it is independent from stability studies!
- It is good if the uncertainty of $\sin i$ is smaller than 1.
- Methods: fit of the orbital elements, RV amplitudes and $\sin i$, and uncertainty estimations: Monte-Carlo (MCMC). An independent estimation for the uncertainties: Fisher-analysis.

Applications – HD 73526 – II.

Monte-Carlo (MCMC) distribution for $\sin i$:



Fisher analysis: $\Delta(\sin i) \approx 0.19 \Rightarrow$ **the two methods yield the roughly the same values!**

Implementation – I.

Basic implementation on an UNIX-like system:

- "Normal" implementation: C code with ≈3.5 kloc (including a simple user interface that parses simple configuration files, some basic chaos detection algorithms, full implementation of the adaptive integration).
- Number of basic arithmetic operations (addition, subtraction, multiplication): $N^2\mathcal{O}(M^2)$, number of more complex operations (division, exponential and power, square root): N^2.
- Problems related to stability investigations: independent ODEs for each initial condition → parallel computations.

Implementation – II.

Implementation on a (GP)GPU architecture:

- Can be made very effective:
  - no need for complex operations at the most of the time; and
  - no interaction between the various initial conditions.
- Memory: although the Lie-integration requires “more” memory than a normal (RK, BS, . . .) integration:
  - data still fit into the registers(!) of the GPU;
  - additionally, no need for the global (DRAM) memory at all (only for communicating with the CPU and/or system DRAM); and
  - only minimal SRAM (higher level cache) is needed (for global constants /masses, physical constants/ and some initialization values required by the algorithm; such as ρ_{ij}^{-2} and so on).
- Non-trivial issues: computation on different threads yields different integration times and computing times as well.

To be done, under construction, ongoing study, etc . . .

Summary

- Lie-integration: very effective, can be applied easily and without losing (expensive) computation time as an adaptive integration scheme, but there is no general form (i.e. algorithm or implementation).
- Linearized equations: derived almost automatically.
- Possibilites for analytical investigations if the model function is a result of an ordinary differential equation.
- Applications: RV analysis, uncertainty estimations.
- Implementation: normal (CPU) code, GPU code; rather complex, but...

Thank you