The need for accuracy and smoothness in numerical simulations

Carl Christian Kjelgaard Mikkelsen Lorién López-Villellas

Department of Computer Science, Umeå University, Sweden,
spock@cs.umu.se

Department of Computer Science, Zaragoza University, Spain,
lorien.lopez@unizar.es

Visit to FSHMN
Pristina, Kosovo, September 12th 2024

UMEÅ UNIVERSITY

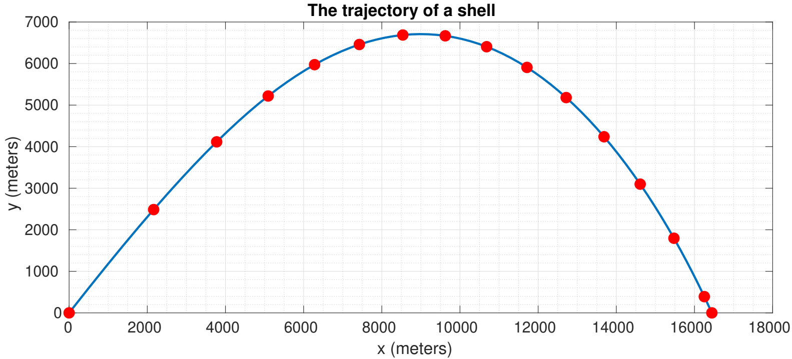
Universidad Zaragoza

THE e-SCIENCE COLLABORATION

Introduction

- Danish nationality
- Ass. Prof. of Comp. Sci. at Umeå University, Sweden
- Ph.D in mathematics from Purdue University, Indiana, USA
- MSc. in mathematics from Aarhus University, Denmark
- Research interests
    - Structured linear system
    - Nonsymmetric eigenvalue problems
    - Solution of nonlinear constraint equations
    - High performance scientific computing
- Software
    - Coauthor of StarNEIG
    - Coauthor of ILVES

External ballistics



The total force $\boldsymbol{F}$ acting on the shell

$$\boldsymbol{F} = m\boldsymbol{g} - \frac{1}{2}\rho(y) A C_D(\nu, y) \|\boldsymbol{v} - \boldsymbol{w}\|_2 (\boldsymbol{v} - \boldsymbol{w}) \tag{1}$$

is the combination of gravity and aerodynamic drag.

External ballistics

Key questions

Does it matter

1. if the drag coefficient
$$\nu \to C_D(\nu)$$
is smooth or not?
2. if the event equation
$$y(t) = 0$$
is solved accurately or not?

Molecular dynamics with constraints

The system of differential algebraic equations

$$\begin{aligned} \boldsymbol{q}'(t) &= \boldsymbol{v}(t), && (2) \\ \boldsymbol{M}\boldsymbol{v}'(t) &= \boldsymbol{f}(\boldsymbol{q}(t)) - \boldsymbol{G}(\boldsymbol{q}(t))^T \boldsymbol{\lambda}(t), && (3) \\ \boldsymbol{g}(\boldsymbol{q}(t)) &= \boldsymbol{0}. && (4) \end{aligned}$$

is solved using the SHAKE algorithm

$$\begin{aligned} \boldsymbol{v}_{n+1/2} &= \boldsymbol{v}_{n-1/2} + \boldsymbol{h}\boldsymbol{M}^{-1}\left(\boldsymbol{f}(\boldsymbol{q}_n) - \boldsymbol{G}(\boldsymbol{q}_n)^T \boldsymbol{\lambda}\right), && (5) \\ \boldsymbol{q}_{n+1} &= \boldsymbol{q}_n + h\boldsymbol{v}_{n+1/2}, && (6) \\ \boldsymbol{g}(\boldsymbol{q}_{n+1}) &= \boldsymbol{0}. && (7) \end{aligned}$$

Molecular dynamics

Key questions

Does it matter

1. if the force-field

$$\boldsymbol{q} \to \boldsymbol{f}(\boldsymbol{q})$$

   is smooth or not?

2. if the nonlinear constraint equation

$$\boldsymbol{g}(\boldsymbol{q}_{n+1}(\boldsymbol{\lambda})) = \boldsymbol{0}$$

   is solved accurately with respect to $\boldsymbol{\lambda}$ or not?

The primary point of this talk

If one of the following is true:

1. the central functions are not smooth enough
2. the central equations are not solved accurately enough

then we will almost certainly lose the ability to

1. assert that rounding errors are irrelevant
2. estimate the discretization error
3. estimate the modelling error

The key terms of this talk

1. P: physical quantity that can be measured
2. T: approximation of P predicted by our model
3. A_h: approximation of T returned by our algorithm
4. $\hat{A}_h$: the computed value of A_h.

The three different error terms

1. $P - T$ is the modelling error and

$$P - T \neq 0 \tag{8}$$

because our model is simpler than the real world.

2. $T - A_h$ is the discretization error and

$$T - A_h \neq 0 \tag{9}$$

because we cannot solve most differential equations exactly.

3. $A_h - \hat{A}_h$ is the computational error and

$$A_h - \hat{A}_h \neq 0 \tag{10}$$

due to rounding errors and truncation errors.

Why should we care?

We validate our models by demonstrating that

$$P - T \approx 0 \tag{11}$$

is a good approximation, but

$$P - T \approx P - \hat{A}_h \tag{12}$$

is not necessarily a good approximation. We have

$$\underbrace{P - T}_{\text{modelling error}} = (P - \hat{A}_h) - \underbrace{(T - A_h)}_{\text{discretization error}} - \underbrace{(A_h - \hat{A}_h)}_{\text{computational error}} \tag{13}$$

so we need to assert that

$$|T - A_h| \ll |P - \hat{A}_h| \tag{14}$$

$$|A_h - \hat{A}_h| \ll |P - \hat{A}_h| \tag{15}$$

Practical error estimation

It is frequently possible to simultanously

1. assert that the computational error

$$A_h - \hat{A}_h$$

is irrelevant, and

2. estimate the discretization error

$$T_h - A_h$$

accurately

Practical error estimation

The key is to have an asymptotic error expansion (AEX)

$$E_h = T - A_h = \alpha h^p + \beta h^q + O(h^r), \quad h \to 0_+ \tag{16}$$

where

$$0 < p < q < r \tag{17}$$

are not necessarily integers and α, β are independent of h.

Basic definitions

We define Richardson's error estimate by

$$R_h := \frac{A_h - A_{2h}}{2^p - 1} \tag{18}$$

and Richardson's fraction by

$$F_h := \frac{A_{2h} - A_{4h}}{A_h - A_{2h}} \tag{19}$$

Elementary results

If

$$E_h = T - A_h = \alpha h^p + \beta h^q + O(h^r), \quad h \to 0_+ \tag{20}$$

then

$$\frac{E_h - R_h}{h^q} \to \text{constant} \tag{21}$$

and

$$\frac{2^p - F_h}{h^{q-p}} \to \text{constant} \tag{22}$$

Implications

1. We can determine p from

$$F_h \to 2^p \tag{23}$$

2. We can determine q from

$$\log |2^p - F_h| \approx \log |\text{constant}| + (q - p) \log h \tag{24}$$

3. We can then compute R_h and estimate

$$E_h \approx R_h \tag{25}$$

when h is *sufficiently* small.

Numerical integration `rint_mwe1`

Our target value is

$$T = \int_0^1 f(x)dx, \quad f(x) = \exp(x) \tag{26}$$

Our approximation is the composite trapezoidal rule

$$A_h = \frac{h}{2} \sum_{j=0}^{n-1} \left(f(x_i) + f(x_{i+1})\right), \quad x_i = ih, \quad nh = 1 \tag{27}$$

We compute A_h for

$$h = h_k = 2^{-k} \tag{28}$$

The evolution of F_h and the quality of R_h

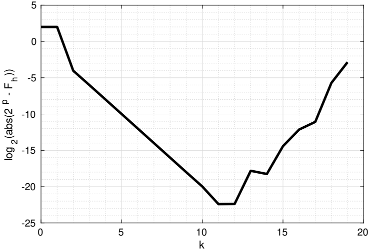


Figure: The evolution of F_h

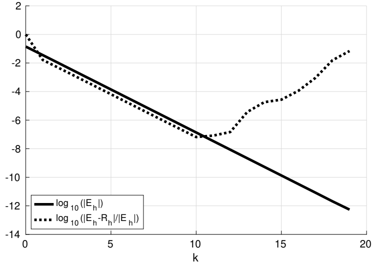


Figure: The accuracy of R_h

Strictly speaking we are not observing F_h and R_h

we are observing $\hat{F}_h$ and $\hat{R}_h$

The difference is controlled by the computational error

$$A_h - \hat{A}_h \tag{29}$$

Numerical integration `rint_mwe2`

Our target value is

$$T = \int_0^1 f(x)dx, \quad f(x) = \sqrt{x}. \tag{30}$$

Our approximation is the composite trapezoidal rule

$$A_h = \frac{h}{2}\sum_{j=0}^{n-1}\left(f(x_i) + f(x_{i+1})\right), \quad x_i = ih, \quad nh = 1 \tag{31}$$

We compute A_h for

$$h = h_k = 2^{-k} \tag{32}$$

The evolution of F_h and the quality of R_h

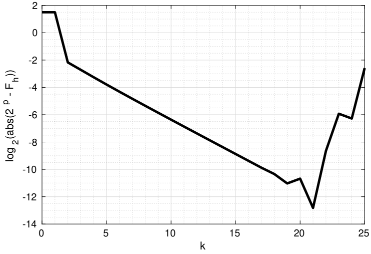


Figure: The evolution of F_h

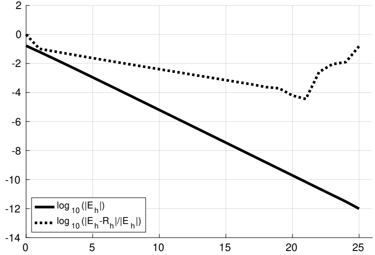


Figure: The accuracy of R_h

We observe that

1. the asymptotic range is much wider (good!)
2. the error estimate is less accurate (mostly harmless)

A spectacular failure

GROMACS simulation of hen egg white lysozyme in water:

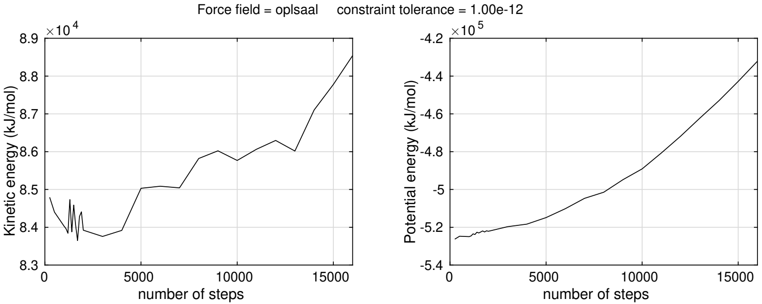
1. T: total energy of the simulation at the end
2. A_h: the approximation of T computed using SHAKE
3. $\hat{A}_h$: the value of A_h returned by the computer

Temporal matters:

1. The length of the simulations was 10^{-12} s (1 ps)
2. A common step size in MD is $10^{-15}s$ (1 fs) or $n = 1000$ steps
3. $n \in \{250, 500, 1000 : 100 : 2000, 3000 : 1000 : 16000\}$

What could possibly go wrong?

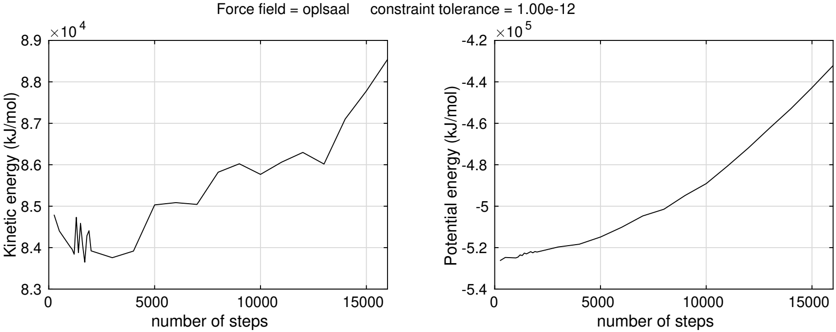
A spectacular failure



The rapid growth of the energy for large value of n

can likely be cured using compensated summation.

A spectacular failure



- The wiggles near $n = 1000$ steps are a great concern.
- If there is an AEX, then 1 fs is not inside the asymp. range.
- We cannot assert that rounding errors are irrelevant
- We cannot estimate the discretication error

When do we have an AEX?

1. Every AEX refers to the exact value of the A_h. If

$$\hat{A}_h \approx A_h$$

is not a good approximation, then

$$\hat{A}_h \text{ will not behave nicely}$$

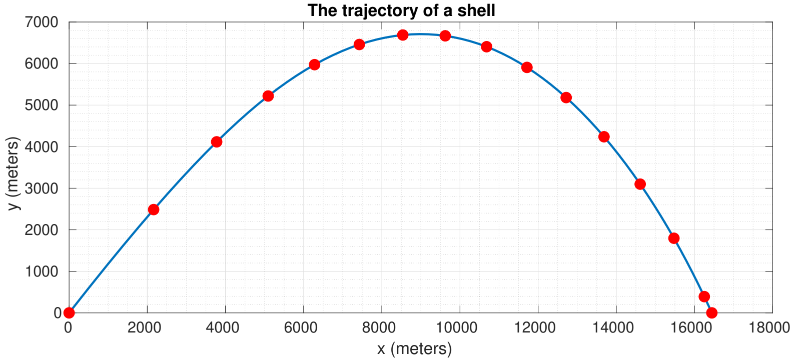
and we cannot identify the asymptotic range.

2. Deriving an AEX is an exercise in Taylor expansions. If

our functions are not many times differentiable

then the foundation crumbles.

External ballistics



The total force $\boldsymbol{F}$ acting on the shell

$$\boldsymbol{F} = m\boldsymbol{g} - \frac{1}{2}\rho(y)AC_D(\nu, y)\|\boldsymbol{v} - \boldsymbol{w}\|_2(\boldsymbol{v} - \boldsymbol{w}) \tag{33}$$

is the combination of gravity and aerodynamic drag.

Computing the optimal range of the a howitzer: Success

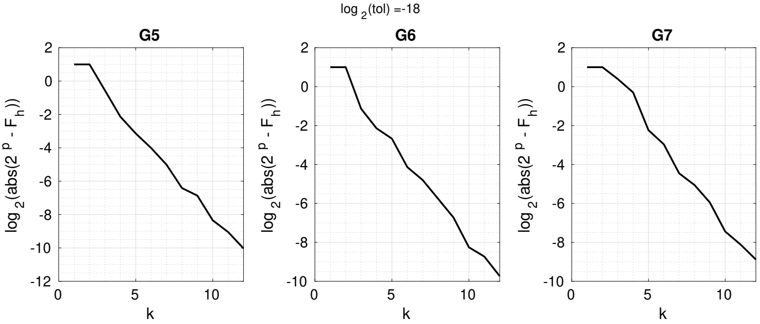


Figure: The evolution of F_h for 3 different drag coeffients, 1st order Runge-Kutta and sufficiently accurate event location.

Computing the optimal range of the a howitzer: Failure

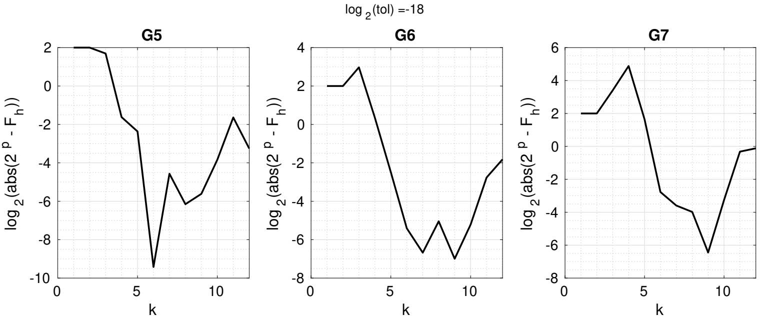


Figure: The evolution of F_h for 3 different drag coeffients, 2nd order Runge-Kutta and inaccurate event location.

Modelling ions: Setup

The total force on an ion is given

$$\boldsymbol{F}(\boldsymbol{r_i}) = -\alpha \sum_{j \neq i} \frac{1}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|^3}(\boldsymbol{r}_i - \boldsymbol{r}_j) - \beta \boldsymbol{r_i} - \gamma \boldsymbol{v}_i \tag{34}$$

where r_k is the position of the kth ion and v_k is it velocity.

- We wish to know the total kinetic energy T at a fixed time.
- We compute approximation A_{h_k} for $h_k = 2^{-k} h_0$.

Modelling ions: Success

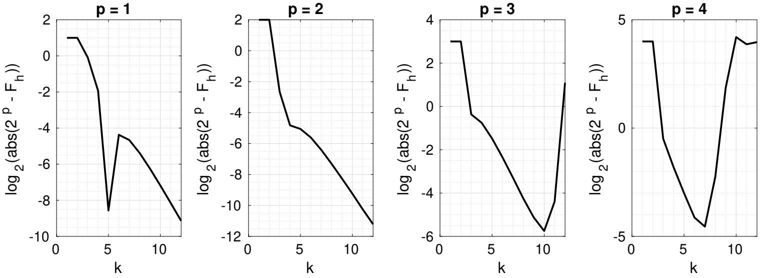


Figure: The evolution of F_h for Runge-Kutta methods of order $p \in \{1, 2, 3, 4\}$ and smooth force-fields with infinite range.

Modelling ions: Failure

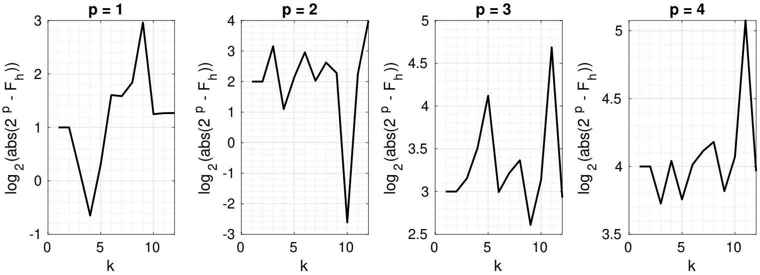


Figure: The evolution of F_h for Runge-Kutta methods of order $p \in \{1, 2, 3, 4\}$ and truncated force-fields with jump discontinuities.

Modelling ions: Success

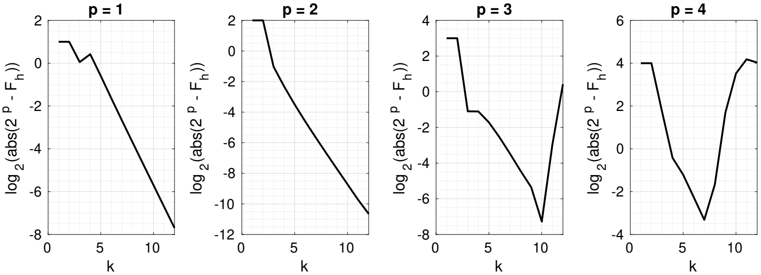


Figure: The evolution of F_h for Runge-Kutta methods of order $p \in \{1, 2, 3, 4\}$ and smoothly truncated force-fields.

Why should this concern the computational scientist?

We use every trick in the book to

1. reduce time-to-solution
2. increase the parallel efficicency
3. reduce energy-to-solution

We should not forget to ask the question:

Can we still validate our models against the real world?

Links

Thank your for your attention