Manchester Numerical Analysis

ADAPTIVE TIME STEPPING for PARABOLIC PDEs

David Silvester

University of Manchester, United Kingdom

Key reference:

Philip Gresho & David Griffiths & David Silvester

Adaptive time-stepping for incompressible flow; part I: scalar advection-diffusion,

SIAM J. Scientific Computing, **30**: 2018–2054, 2008.

https://doi.org/10.1137/070688018

TIME INTEGRATION – I

For the simple ODE

$$\dot{u} = f(u)$$

we use the Trapezoidal Rule TR: $u_n \approx u(t_n)$ is computed so that for $n = 0, 1, \ldots$

$$u_{n+1} - u_n = \tfrac{1}{2}\Delta t_n \left(f_{n+1} + f_n\right)$$

The Local Truncation Error LTE is

$$u_n - u(t_n) \equiv T_n = \tfrac{1}{12}\Delta t_n^3 \, \dddot{u}(\bar{t_n})$$

TIME INTEGRATION – II

For the simple ODE

$$\dot{u} = f(u)$$

we can estimate the LTE in TR using the explicit Adams–Bashforth method AB2: $u_n^* \approx u(t_n)$ is computed so that for $n = 1, 2, \ldots$

$$u_{n+1}^* - u_n^* = \Delta t_n \, f_n + \tfrac{1}{2} \Delta t_n^2 \left(\tfrac{f_n - f_{n-1}}{\Delta t_{n-1}} \right)$$

The Local Truncation Error LTE* is

$$u_n^* - u(t_n) \equiv T_n^* = -(2 + 3 \tfrac{\Delta t_{n-1}}{\Delta t_n})\rangle \tfrac{1}{12} \Delta t_n^3 \, \dddot{u}(t_n^*)$$

TIME INTEGRATION – III

Manipulating the truncation error terms for TR and AB2 gives the estimate

$$T_n = \frac{u_{n+1} - u^*_{n+1}}{3(1 + \frac{\Delta t_{n-1}}{\Delta t_n})}$$

Given some user-prescribed error tolerance `tol`, the following time step is selected to be the biggest possible such that $\|T_{n+1}\| \leq \texttt{tol} \times u_{\max}$.

This criterion leads to

$$\Delta t_{n+1} := \Delta t_n \left(\frac{\texttt{tol} \times u_{\max}}{\|T_n\|} \right)^{1/3}$$

Key point

- Implementation is "delicate"—it is very sensitive to round-off.

To address the instability issue

- We rewrite the AB2–TR algorithm to compute updates v_n and w_n scaled by the time-step:

$$u_{n+1} - u_n = \tfrac{1}{2}\Delta t_n v_n; \quad u^*_{n+1} - u^*_n = \Delta t_n w_n.$$

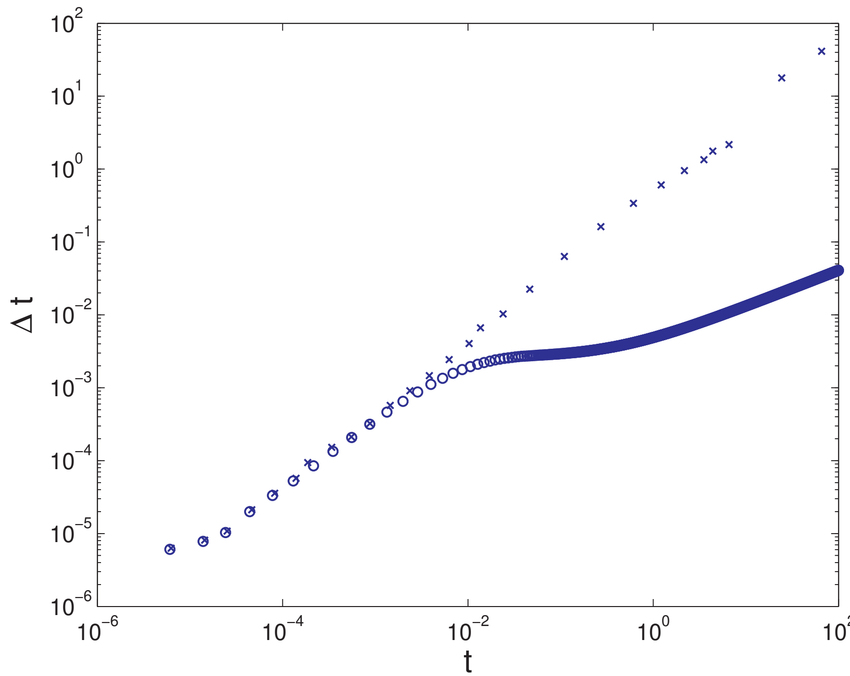
- We perform time-step averaging every n^* steps:

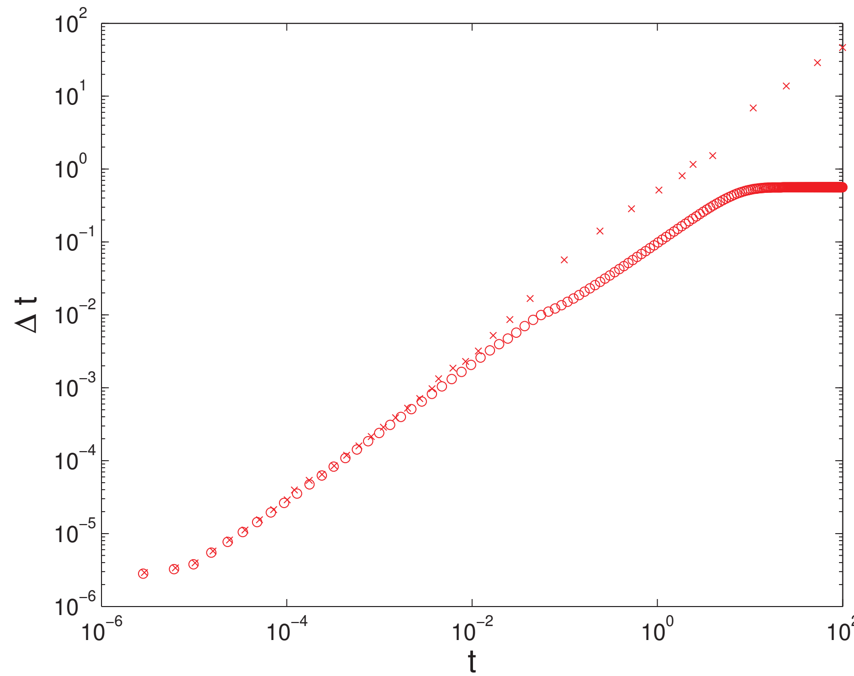
$$u_n := \tfrac{1}{2}(u_n + u_{n-1}); \quad u_{n+1} := u_n + \tfrac{1}{4}\Delta t_n v_n; \quad \dot{u}_{n+1} := \tfrac{1}{2} v_n.$$

Contrast this with the standard acceleration obtained by "inverting" the TR formula:

$$\dot{u}_{n+1} = \frac{2}{\Delta t_n}(u_{n+1} - u_n) - \dot{u}_n$$

STABILIZED INTEGRATOR – II





`tol` $= 10^{-3}$

`tol` $= 10^{-4}$

Advection–diffusion of step profile

HEAT EQUATION – I

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0, \qquad 0 < x < 1$$

$$u(0,t) = 1, \quad u(1,t) = 0 \qquad\qquad BC$$

$$u(x,0) = 1, \quad 0 \le x < 1, \quad u(1,0) = 0 \qquad IC$$

solution

$$u(x,t) = \begin{cases} \operatorname{erf}\left(\frac{1-x}{\sqrt{4t}}\right) \\ (1-x) + \sum_{j=1}^{\infty} \frac{2}{j\pi} e^{-j^2\pi^2 t} \sin j\pi x \end{cases}$$

HEAT EQUATION – II

spatial discretization

using linear FEM gives the ODE system

$$M\dot{\mathbf{u}} + A\mathbf{u} = \mathbf{f}$$

with M and A both symmetric positive definite matrices.

discrete solution

$$\mathbf{u}(t) = (1 - x) + \sum_{k=1}^{n_u} a_k e^{-\lambda_k t} \mathbf{v}_k$$

where $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{n_u}$ and $\{\lambda_k, \mathbf{v}_k\}$ satisfy

$$M\mathbf{v}_k = \lambda_k A\mathbf{v}_k.$$

HEAT EQUATION – III

$$\mathbf{u}(t) = (1-x) + \sum_{k=1}^{n_u} a_k e^{-\lambda_k t} \mathbf{v}_k$$

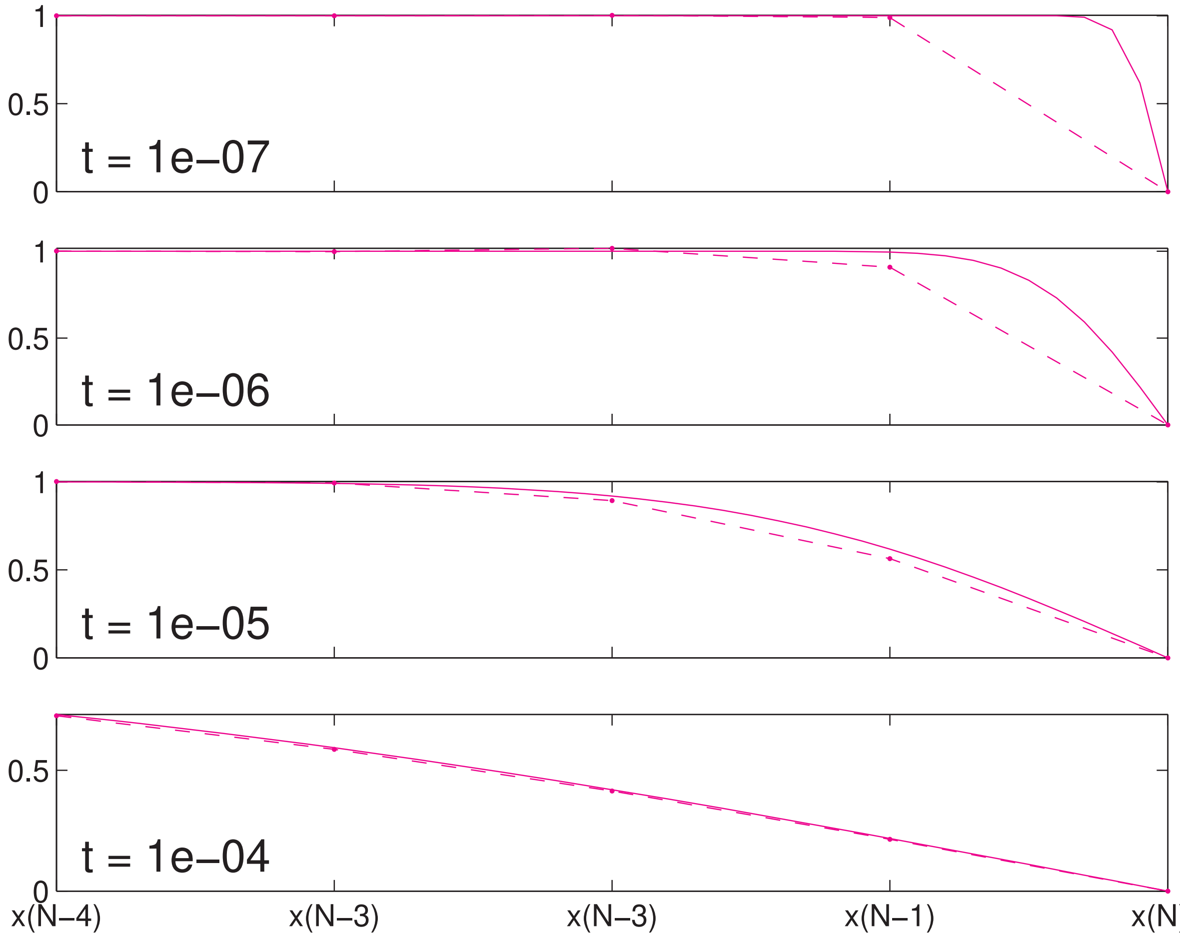
... suggests two asymptotic extremes ...

- For $t < \frac{1}{\lambda_{n_u}} =: \tau_{\mathtt{mtb}}$ there is a fast transient:
  $\mathbf{u}(t) \sim a_{n_u} e^{-\lambda_{n_u} t} \mathbf{v}_{n_u} +$ slowly varying terms
- For $t \gg 1$ there is a slow transient:
  $\mathbf{u}(t) \sim (1-x)\rangle + a_1 e^{-\lambda_1 t} \mathbf{v}_1$

$\tau_{\mathtt{mtb}} \approx \frac{h^2}{4}$ is the "minimum time of believability" for spatially discretized convection-diffusion problems — it is the time for discontinuities in IC to grow to size h.

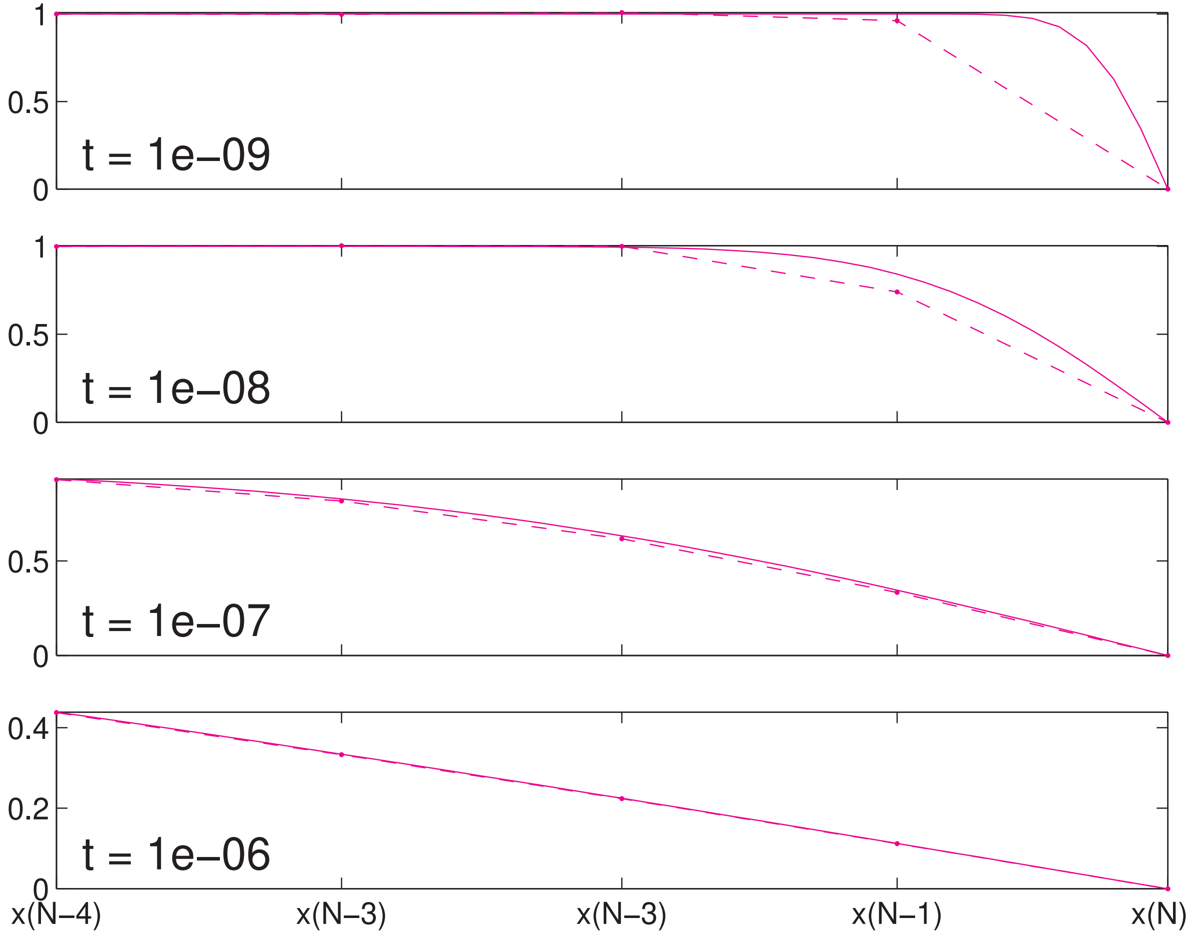
SPATIAL DISCRETIZATION – I

- Uniform: $n_u = 255, h = 1/256, \tau_{\mathtt{mtb}} \sim 4 \times 10^{-6}$



SPATIAL DISCRETIZATION – II

- Geometric: $h_{\min} = 2 \times 10^{-4}, n_u = 255, \tau_{\texttt{mtb}} \sim 10^{-8}$



HEAT EQUATION – IV

$$\mathbf{u}(t) = (1-x) + \sum_{k=1}^{n_u} a_k e^{-\lambda_k t} \mathbf{v}_k; \qquad \Delta t_n^3 = \frac{12\texttt{tol}}{\|\dddot{\mathbf{u}}\|}$$

- For $t < \tau_{\texttt{mtb}}$ there is a fast transient:
  $\mathbf{u}(t) \sim a_{n_u} e^{-\lambda_{n_u} t} \mathbf{v}_{n_u} +$ slowly varying terms
  $\Delta t_n \sim e^{\lambda_{n_u} t/3}$

- For $t \gg 1$ there is a slow transient:
  $\mathbf{u}(t) \sim (1-x)\rangle + a_1 e^{-\lambda_1 t} \mathbf{v}_1$
  $\Delta t_n \sim e^{\lambda_1 t/3}$

HEAT EQUATION – IV

$$\mathbf{u}(t) = (1-x) + \sum_{k=1}^{n_u} a_k e^{-\lambda_k t} \mathbf{v}_k; \qquad \Delta t_n^3 = \frac{12\mathtt{tol}}{\|\ddot{\mathbf{u}}\|}$$

- For $t < \tau_{\mathtt{mtb}}$ there is a fast transient:
  $\mathbf{u}(t) \sim a_{n_u} e^{-\lambda_{n_u} t} \mathbf{v}_{n_u} +$ slowly varying terms
  $\Delta t_n \sim e^{\lambda_{n_u} t/3}$
- What happens in between?
- For $t \gg 1$ there is a slow transient:
  $\mathbf{u}(t) \sim (1-x)\rangle + a_1 e^{-\lambda_1 t} \mathbf{v}_1$
  $\Delta t_n \sim e^{\lambda_1 t/3}$

HEAT EQUATION – V

$$u(t) = (1 - x) + \sum_{j=1}^{\infty} a_j e^{-j^2\pi^2 t} \sin j\pi x$$

Parabolic smoothing (Luskin & Rannacher)

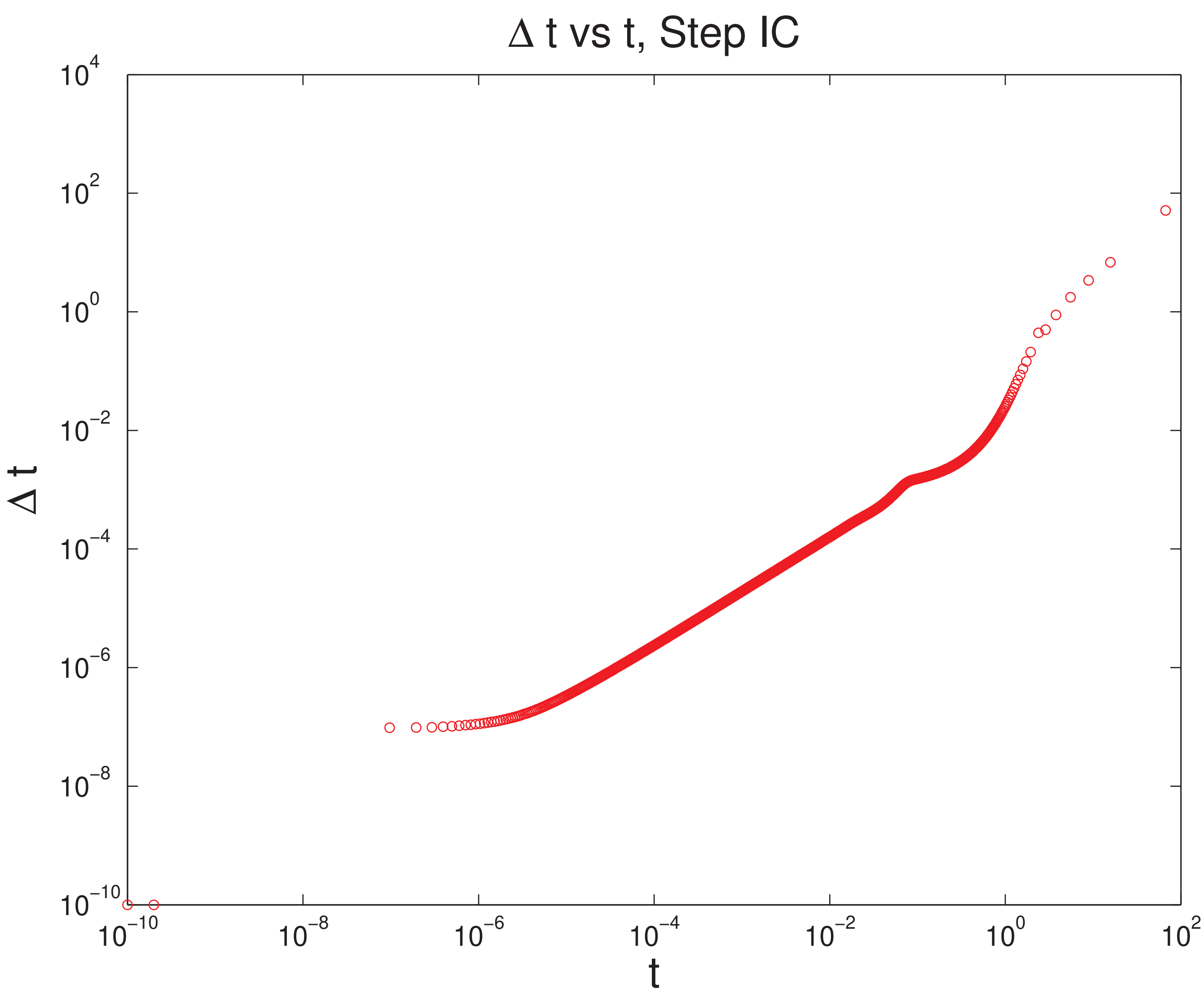
$$\|\ddot{\mathbf{u}}\|^2 \leq C\|\dddot{u}\|^2$$

$$= C\sum_{j=1}^{\infty} j^6 a_j^2 e^{-2j^2\pi^2 t}$$

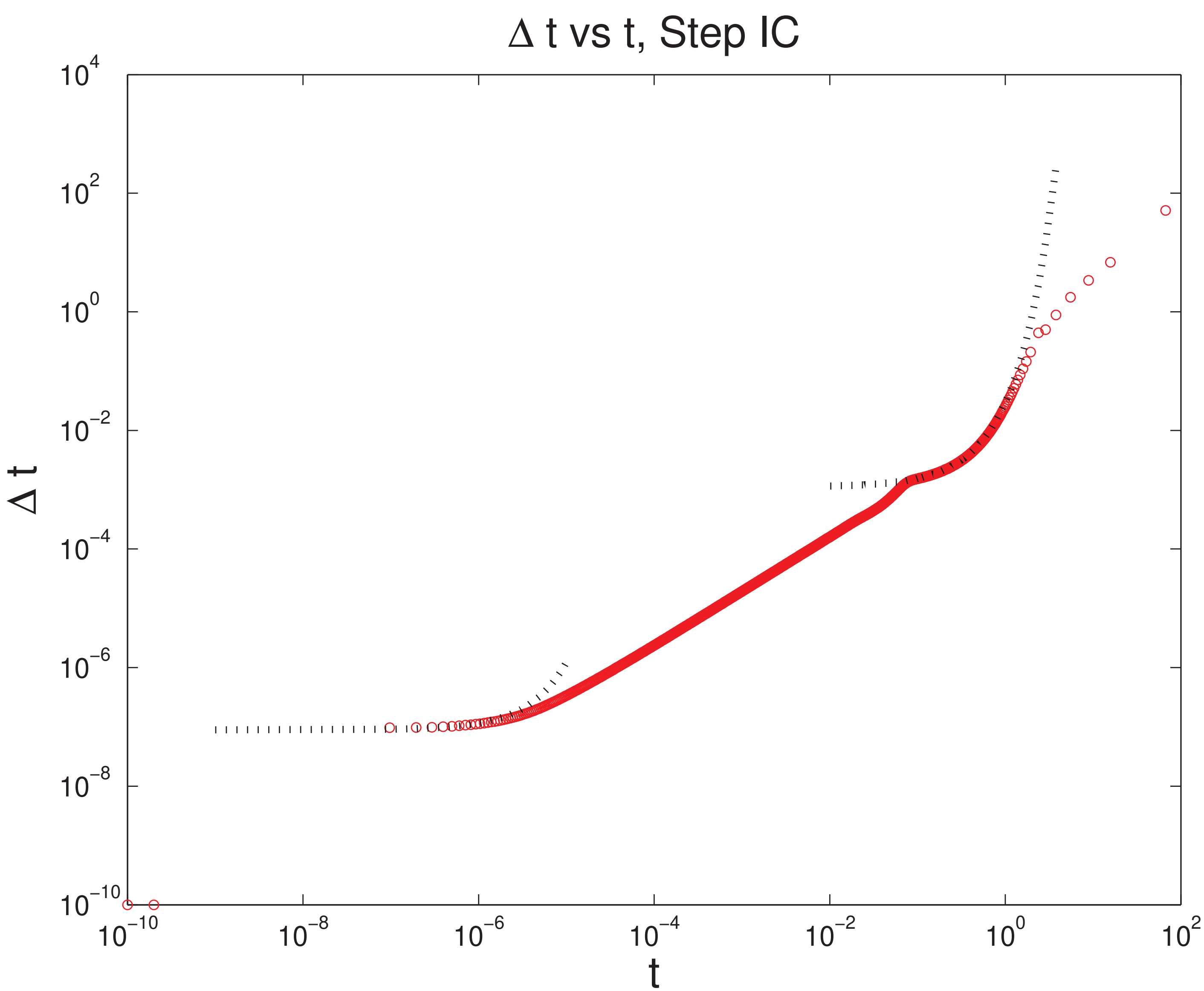
$$\leq C\max_j(j^{7+\epsilon}a_j^2 e^{-2j^2\pi^2 t})\sum_{j=1}^{\infty}\frac{1}{j^{1+\epsilon}} \leq \frac{C}{t^{11/2}}$$

This gives the lower bound: $\Delta t_n \geq Ct^{11/12}$

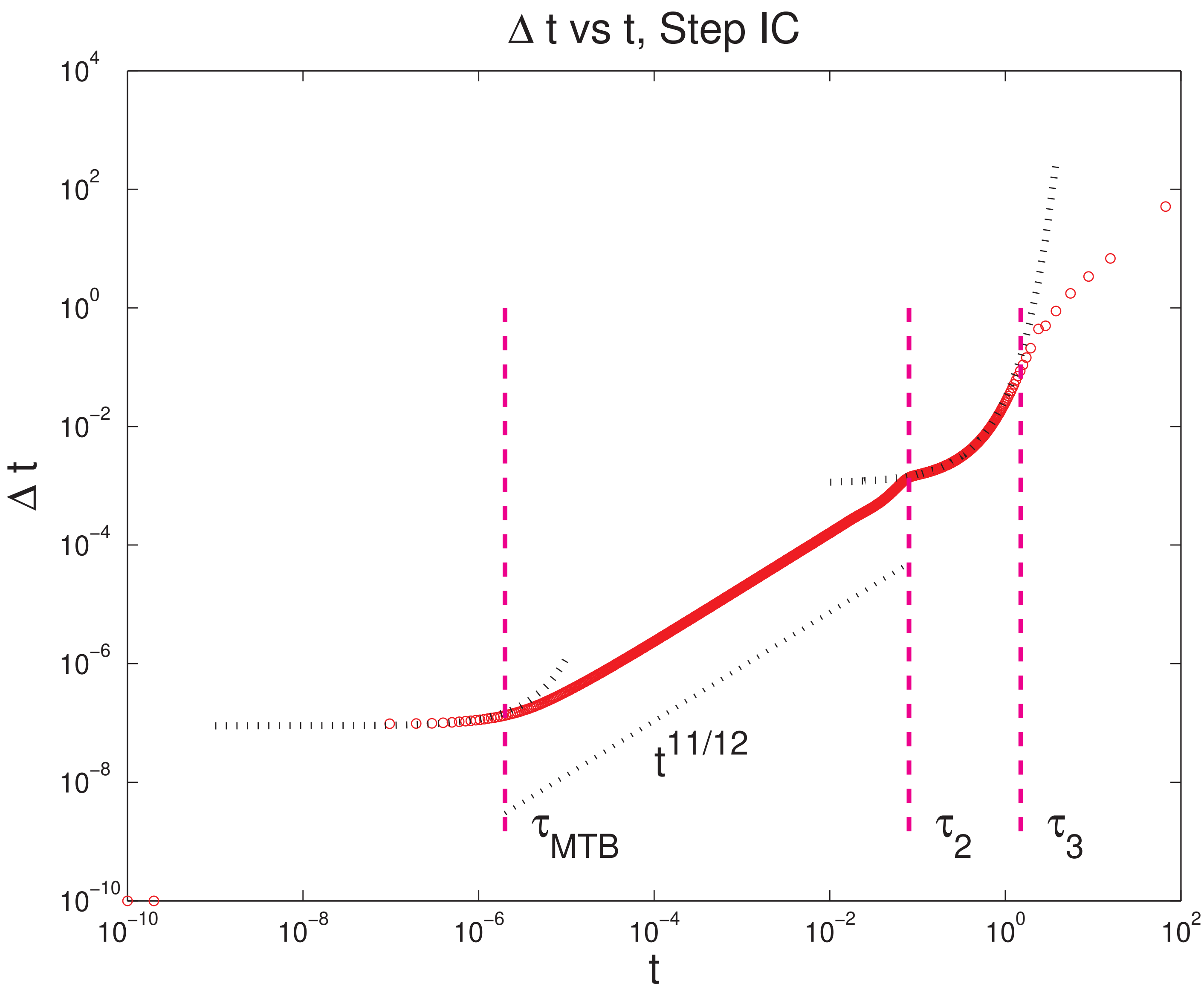
Uniform grid results – I



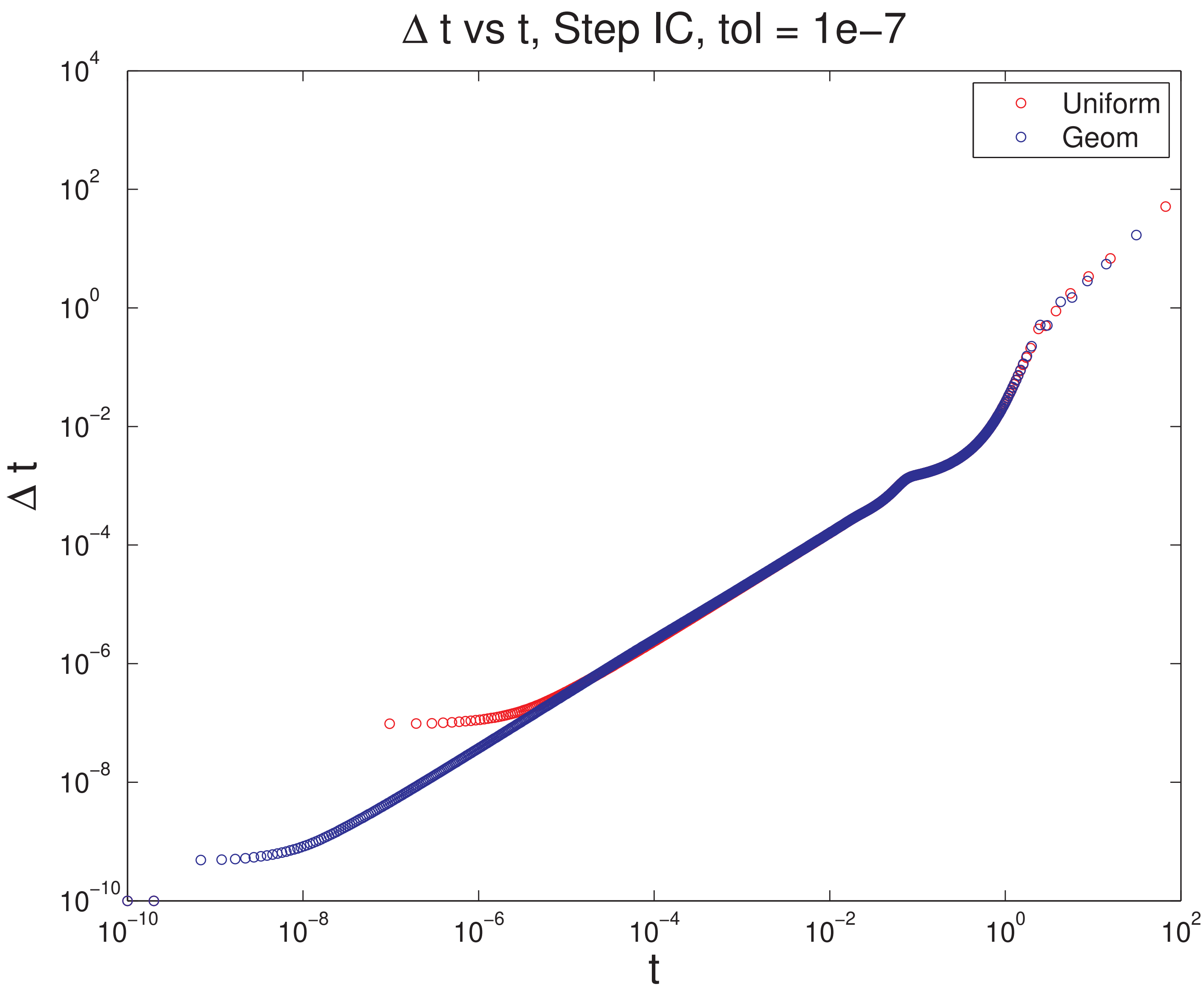
Uniform grid results – II



Uniform grid results – III



Geometric grid results – III



SUMMARY

Adaptive time stepping — what are the key ingredients?

- ♡ time steps automatically "follow the physics"
- ♡ no "tuning" parameters in the adaptive strategy

IFISS

HOME DOWNLOAD DOCUMENTATION PUBLICATIONS

Incompressible Flow & Iterative Solver Software

An open-source software package

Summary

IFISS is a graphical package for the interactive numerical study of incompressible flow problems which can be run under Matlab or Octave. It includes algorithms for discretization by mixed finite element methods and a posteriori error estimation of the computed solutions. The package can also be used as a computational laboratory for experimenting with state-of-the-art preconditioned iterative solvers for the discrete linear equation systems that arise in incompressible flow modelling.

Key Features

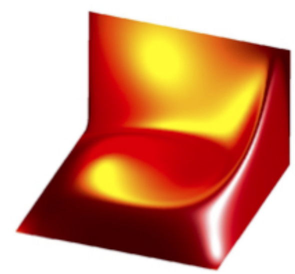
Key features include

- implementation of a variety of mixed finite element approximation methods;
- automatic calculation of stabilization parameters where appropriate;
- a posteriori error estimation for steady problems;
- a range of state-of-the-art preconditioned Krylov subspace solvers ;
- built-in geometric and algebraic multigrid solvers and preconditioners;
- fully implicit self-adaptive time stepping algorithms;
- useful visualization tools.

The developers of the IFISS package are David Silvester (School of Mathematics, University of Manchester), Howard Elman (Computer Science Department, University of Maryland), and Alison Ramage (Department of Mathematics and Statistics, University of Strathclyde).

Links

- Download
- Documentation
- Publications
- Overview
- Sample output
- Contact



The IFISS logo represents the solution of the *double glazing* convection-diffusion problem. It can be reproduced in IFISS via the function **ifisslogo**.

This webpage is based on a CSS template from Free CSS Templates.