

A Review of Solution Techniques for Unsteady Incompressible Flow

David Silvester
School of Mathematics
University of Manchester

Outline

- PDEs
- Review : 1967 – 1999
- Update : 2000 – 2010

Outline

- PDEs



$$\left. \begin{aligned} \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} - \nu \nabla^2 \vec{u} + \nabla p &= 0 \\ \nabla \cdot \vec{u} &= 0 \end{aligned} \right\} \text{Navier–Stokes}$$

Outline

- PDEs



$$\left. \begin{aligned} \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} - \nu \nabla^2 \vec{u} + \nabla p &= 0 \\ \nabla \cdot \vec{u} &= 0 \end{aligned} \right\} \text{Navier–Stokes}$$



$$\left. \begin{aligned} \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} - \nu \nabla^2 \vec{u} + \nabla p &= \vec{j}T \\ \nabla \cdot \vec{u} &= 0 \\ \frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T - \nu \nabla^2 T &= 0 \end{aligned} \right\} \text{Boussinesq}$$

Navier-Stokes Equations

$$\begin{aligned}\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} - \nu \nabla^2 \vec{u} + \nabla p &= 0 & \text{in } \mathcal{W} \equiv \Omega \times (0, T) \\ \nabla \cdot \vec{u} &= 0 & \text{in } \mathcal{W}\end{aligned}$$

Boundary and Initial conditions

$$\begin{aligned}\vec{u} &= \vec{g} & \text{on } \Gamma_D \times [0, T]; \\ \nu \nabla \vec{u} \cdot \vec{n} - p \vec{n} &= \vec{0} & \text{on } \Gamma_N \times [0, T]; \\ \vec{u}(\vec{x}, 0) &= \vec{u}_0(\vec{x}) & \text{in } \Omega.\end{aligned}$$

Finite element matrix formulation

Introducing the basis sets

$$\begin{aligned} \mathbf{X}_h &= \text{span}\{\vec{\phi}_i\}_{i=1}^{n_u}, & \text{Velocity basis functions;} \\ M_h &= \text{span}\{\psi_j\}_{j=1}^{n_p}, & \text{Pressure basis functions.} \end{aligned}$$

gives the method-of-lines discretized system:

$$\begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{u}}{\partial t} \\ \frac{\partial p}{\partial t} \end{pmatrix} + \begin{pmatrix} N(\vec{u}) + \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} \vec{f} \\ 0 \end{pmatrix}$$

with associated matrices

$$\begin{aligned} N_{ij} &= (\vec{u} \cdot \nabla \vec{\phi}_i, \vec{\phi}_j), & \text{convection} \\ A_{ij} &= (\nabla \vec{\phi}_i, \nabla \vec{\phi}_j), & \text{diffusion} \\ B_{ij} &= -(\nabla \cdot \vec{\phi}_j, \psi_i), & \text{divergence .} \end{aligned}$$

Spatial Discretization

The method-of-lines discretized system is a semi-explicit system of DAEs:

$$\begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{u}}{\partial t} \\ \frac{\partial p}{\partial t} \end{pmatrix} + \begin{pmatrix} N(\vec{u}) + \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} \vec{f} \\ 0 \end{pmatrix}$$

- The DAEs have index equal to **two**
- The discrete problem is nonlinear $N(\vec{u}) + \nu A =: F(\vec{u})$

Spatial Discretization

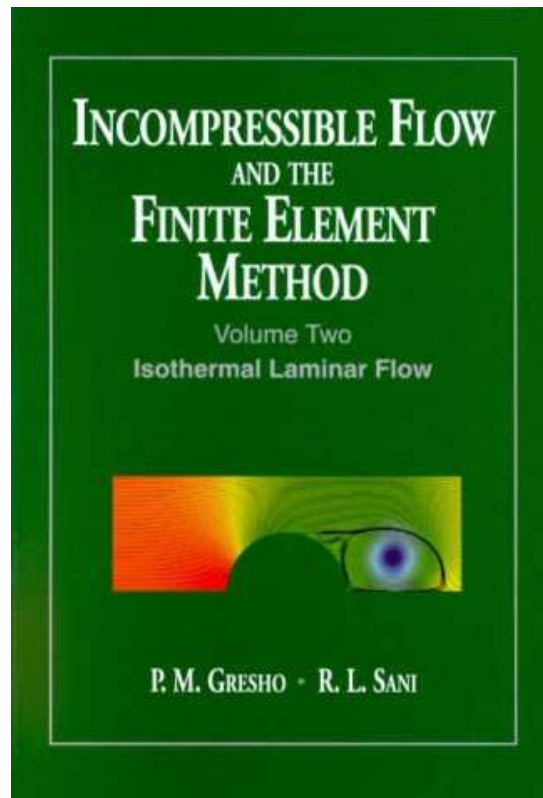
The method-of-lines discretized system is a semi-explicit system of DAEs:

$$\begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{u}}{\partial t} \\ \frac{\partial p}{\partial t} \end{pmatrix} + \begin{pmatrix} N(\vec{u}) + \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} \vec{f} \\ 0 \end{pmatrix}$$

- The DAEs have index equal to **two**
- The discrete problem is nonlinear $N(\vec{u}) + \nu A =: F(\vec{u})$
- Implicit approximation in time does not look attractive!

Outline

- PDEs
- Review : 1967 – 1999
 - 1967
 - 1982
 - 1991
 - 1999



$$\begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{u}}{\partial t} \\ \frac{\partial p}{\partial t} \end{pmatrix} + \begin{pmatrix} N(\vec{u}) + \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} \vec{f} \\ 0 \end{pmatrix}$$

- 1967

A simple splitting/projection approach — I

$$\begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{u}}{\partial t} \\ \frac{\partial p}{\partial t} \end{pmatrix} + \begin{pmatrix} F(\vec{u}) & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} \vec{f} \\ 0 \end{pmatrix}$$

Given a time step dT and a parameter $0 < \gamma \leq 2$

Algorithm: Chorin, 1967; Temam, 1969

for $k = 0, 1, \dots$

solve $M \frac{\partial \vec{u}_*}{\partial t} + F(\vec{u}_k) \vec{u}_* = \vec{f} - B^T p_k$

solve $BM^{-1}B^T \phi = B \vec{u}_*$

compute $\vec{u}_{k+1} = \vec{u}_* - M^{-1}B^T \phi$

compute $p_{k+1} = p_k + (\gamma/dT) \phi$

end

A simple splitting/projection approach — II

Key Question: Is the projection/splitting **consistent**?

A simple splitting/projection approach — II

Key Question: Is the projection/splitting **consistent**?

Practitioners say **yes** ...

- Chorin (1967), Temam(1969), Kim & Moin (1985), Van Kan (1986), Bell, Colella & Glaz (1989), **Gresho & Chan (1984, 1990)**, **Perot (1993)**, Turek (1997).

A simple splitting/projection approach — II

Key Question: Is the projection/splitting **consistent**?

Practitioners say **yes** ...

- Chorin (1967), Temam(1969), Kim & Moin (1985), Van Kan (1986), Bell, Colella & Glaz (1989), **Gresho & Chan (1984, 1990), Perot (1993), Turek (1997).**

Mathematicians say **sometimes** ...

- Guermond (1984), Rannacher (1992), E & Liu (1995, 1996), **Shen (1992, 1996), Prohl (1997, 2008).**

$$\begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{u}}{\partial t} \\ \frac{\partial p}{\partial t} \end{pmatrix} + \begin{pmatrix} N(\vec{u}) + \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} \vec{f} \\ 0 \end{pmatrix}$$

- 1967
- 1982

A Stokes splitting/projection approach — I

$$\begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{u}}{\partial t} \\ \frac{\partial p}{\partial t} \end{pmatrix} + \begin{pmatrix} N(\vec{u}) + \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Given a time step dT , discretise the **total** derivative:

Algorithm: Pironneau, 1982; Hansbo, 1982

for $k = 0, 1, \dots$

compute $X^m(\vec{x})$ the solution at $\tau_k = kdT$ of

$$\frac{dX}{d\tau} = \vec{u}_k(X, \tau), \quad X(\tau_{k+1}) = \vec{x};$$

interpolate $\vec{u}_k^* = \vec{u}_k(X^m(\vec{x}))$

solve
$$\begin{pmatrix} \frac{1}{dT}M + \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u}_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} \frac{1}{dT}M \vec{u}_k^* \\ 0 \end{pmatrix}$$

end

A simple splitting/projection approach — II

Key¹ Question: Is the total derivative approximation **stable** numerically ?

A simple splitting/projection approach — II

Key¹ Question: Is the total derivative approximation **stable** numerically ?

Mathematicians say **yes**, but care is needed ...

- Hansbo (1982), Pironneau (1982), **Morton, Priestley & Suli (1988, 1989, 1994)**, Bermejo & Staniforth (1991, 1992).

A simple splitting/projection approach — II

Key¹ Question: Is the total derivative approximation **stable** numerically ?

Mathematicians say **yes**, but care is needed ...

- Hansbo (1982), Pironneau (1982), **Morton, Priestley & Suli (1988, 1989, 1994)**, Bermejo & Staniforth (1991, 1992).

Key² Question: Adaptive time stepping ?

$$\begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{u}}{\partial t} \\ \frac{\partial p}{\partial t} \end{pmatrix} + \begin{pmatrix} N(\vec{u}) + \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} \vec{f} \\ 0 \end{pmatrix}$$

- 1967
- 1982
- 1991



A clever splitting/projection approach — I

$$\begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{u}}{\partial t} \\ \frac{\partial p}{\partial t} \end{pmatrix} + \begin{pmatrix} N(\vec{u}) + \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} \vec{f} \\ 0 \end{pmatrix}$$

Given dT and parameters $\theta = 1 - 1/\sqrt{2}$, $\alpha = \frac{1-2\theta}{1-\theta}$, $\beta = 1 - \alpha$.

Algorithm: Glowinski, 1986, 1991

for $k = 0, 1, \dots$

solve
$$\begin{pmatrix} \frac{1}{\theta dT} M + \alpha \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u}_{k+\theta} \\ p_{k+\theta} \end{pmatrix} = \begin{pmatrix} f_k \\ 0 \end{pmatrix}$$

solve
$$\left(\frac{1}{(1-2\theta)dT} M + N(\vec{u}_{k+1-\theta}) + \beta \nu A \right) \vec{u}_{k+1-\theta} = \vec{f}_{k+\theta}$$

solve
$$\begin{pmatrix} \frac{1}{\theta dT} M + \alpha \nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u}_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} f_{k+1-\theta} \\ 0 \end{pmatrix}$$

end

A clever splitting/projection approach — II

Key¹ Question: Is the algorithm over-complicated?

A clever splitting/projection approach — II

Key¹ Question: Is the algorithm over-complicated?

Mathematicians say **no**, ...

- **Glowinski & Dean (1984, 1993)** , Bristeau et al. (1985), Rannacher (1989, 1990), Turek (1996), **Smith & Silvester (1997)**.

A clever splitting/projection approach — II

Key¹ Question: Is the algorithm over-complicated?

Mathematicians say **no**, ...

- **Glowinski & Dean (1984, 1993)** , Bristeau et al. (1985), Rannacher (1989, 1990), Turek (1996), **Smith & Silvester (1997)**.

Key² Question: Adaptive time stepping ?

- Update : 2000 – 2010
- 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.
- David Kay & Philip Gresho & David Griffiths & David Silvester Adaptive time-stepping for incompressible flow; part II: Navier-Stokes equations. SIAM J. Scientific Computing, 32: 111–128, 2010.

“Smart Integrator” (SI) definition

- **Optimal time-stepping:** time-steps automatically chosen to “follow the physics”.
- **Black-box implementation:** few parameters that have to be estimated a priori.

“Smart Integrator” (SI) definition

- **Optimal time-stepping:** time-steps automatically chosen to “follow the physics”.
- **Black-box implementation:** few parameters that have to be estimated a priori.
- **Solver efficiency:** the linear solver convergence rate is robust with respect to the mesh size h and the Reynolds number $1/\nu$.

Trapezoidal Rule (TR) time discretization

We subdivide $[0, T]$ into time levels $\{t_i\}_{i=1}^N$. Given (\vec{u}^n, p^n) at time level t_n , $k_{n+1} := t_{n+1} - t_n$, compute (\vec{u}^{n+1}, p^{n+1}) via

$$\frac{2}{k_{n+1}} \vec{u}^{n+1} + \vec{w}^{n+1} \cdot \nabla \vec{u}^{n+1} - \nu \nabla^2 \vec{u}^{n+1} + \nabla p^{n+1} = \vec{f}^{n+1}$$

$$-\nabla \cdot \vec{u}^{n+1} = 0 \quad \text{in } \Omega$$

$$\vec{u}^{n+1} = \vec{g}^{n+1} \quad \text{on } \Gamma_D$$

$$\nu \nabla \vec{u}^{n+1} \cdot \vec{n} - p^{n+1} \vec{n} = \vec{0} \quad \text{on } \Gamma_N$$

with **second-order** linearization

$$\vec{f}^{n+1} = \frac{2}{k_{n+1}} \vec{u}^n + \nu \nabla^2 \vec{u}^n - \vec{u}^n \cdot \nabla \vec{u}^n - \nabla p^n$$

$$\vec{w}^{n+1} = \left(1 + \frac{k_{n+1}}{k_n}\right) \vec{u}^n - \frac{k_{n+1}}{k_n} \vec{u}^{n-1}$$

Saddle-point system

In \mathbb{R}^2 the discretized Oseen system (*) is:

$$\begin{pmatrix} F^{n+1} & 0 & B_x^T \\ 0 & F^{n+1} & B_y^T \\ B_x & B_y & 0 \end{pmatrix} \begin{bmatrix} \alpha^{x,n+1} \\ \alpha^{y,n+1} \\ \alpha^{p,n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{x,n+1} \\ \mathbf{f}^{y,n+1} \\ \mathbf{f}^{p,n+1} \end{bmatrix}$$

- $F^{n+1} := \frac{2}{k_{n+1}} M + \nu A + N(\vec{w}_h^{n+1})$
- The vector \mathbf{f} is constructed from the boundary data \vec{g}^{n+1} , the computed velocity \vec{u}_h^n at the previous time level and the acceleration $\frac{\partial \vec{u}_h^n}{\partial t}$
- The system can be efficiently solved using “appropriately” preconditioned **GMRES**...

Preconditioned system

$$\begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix} \mathcal{P}^{-1} \mathcal{P} \begin{pmatrix} \alpha^u \\ \alpha^p \end{pmatrix} = \begin{pmatrix} \mathbf{f}^u \\ \mathbf{f}^p \end{pmatrix}$$

A **perfect** preconditioner is given by

$$\begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix} \underbrace{\begin{pmatrix} F^{-1} & F^{-1} B^T S^{-1} \\ 0 & -S^{-1} \end{pmatrix}}_{\mathcal{P}^{-1}} = \begin{pmatrix} I & 0 \\ BF^{-1} & I \end{pmatrix}$$

with $F = \frac{2}{k_{n+1}}M + \nu A + N$ and $S = BF^{-1}B^T$.

For an **efficient** preconditioner we need to construct a sparse approximation to the “exact” Schur complement

$$S^{-1} = (BF^{-1}B^T)^{-1}$$

See Chapter 8 of

- Howard Elman & David Silvester & Andrew Wathen
Finite Elements and Fast Iterative Solvers: with applications in incompressible fluid dynamics
Oxford University Press, 2005.

Two possible constructions ...

Schur complement approximation – I

Introducing the diagonal of the velocity mass matrix

$$M_* \sim M_{ij} = (\vec{\phi}_i, \vec{\phi}_j),$$

gives the “least-squares commutator preconditioner”:

$$(BF^{-1}B^T)^{-1} \approx \underbrace{(BM_*^{-1}B^T)^{-1}}_{\text{AMG}} (BM_*^{-1}FB_*^{-1}B^T) \underbrace{(BM_*^{-1}B^T)^{-1}}_{\text{AMG}}$$

Schur complement approximation – II

Introducing associated pressure matrices

$$M_p \sim (\nabla\psi_i, \nabla\psi_j), \quad \text{mass}$$

$$A_p \sim (\nabla\psi_i, \nabla\psi_j), \quad \text{diffusion}$$

$$N_p \sim (\vec{w}_h \cdot \nabla\psi_i, \psi_j), \quad \text{convection}$$

$$F_p = \frac{2}{k_{n+1}} M_p + \nu A_p + N_p, \quad \text{convection-diffusion}$$

gives the “pressure convection-diffusion preconditioner”:

$$(BF^{-1}B^T)^{-1} \approx M_p^{-1} F_p \underbrace{A_p^{-1}}_{\text{AMG}}$$

Adaptive Time Stepping AB2–TR

Consider the simple ODE $\dot{u} = f(u)$

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

$$T_n = \frac{u_{n+1} - u_{n+1}^*}{3\left(1 + \frac{k_n}{k_{n+1}}\right)}$$

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

$$k_{n+2} := k_{n+1} \left(\frac{\text{tol} \times u_{\max}}{\|T_n\|} \right)^{1/3}$$

Adaptive Time Stepping AB2–TR

Consider the simple ODE $\dot{u} = f(u)$

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

$$T_n = \frac{u_{n+1} - u_{n+1}^*}{3\left(1 + \frac{k_n}{k_{n+1}}\right)}$$

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

$$k_{n+2} := k_{n+1} \left(\frac{\text{tol} \times u_{\max}}{\|T_n\|} \right)^{1/3}$$

But look out for “ringing” ...

Stabilized AB2–TR

To address the instability issues:

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

$$u_{n+1} - u_n = \frac{1}{2}k_{n+1}v_n; \quad u_{n+1}^* - u_n^* = k_{n+1}w_n.$$

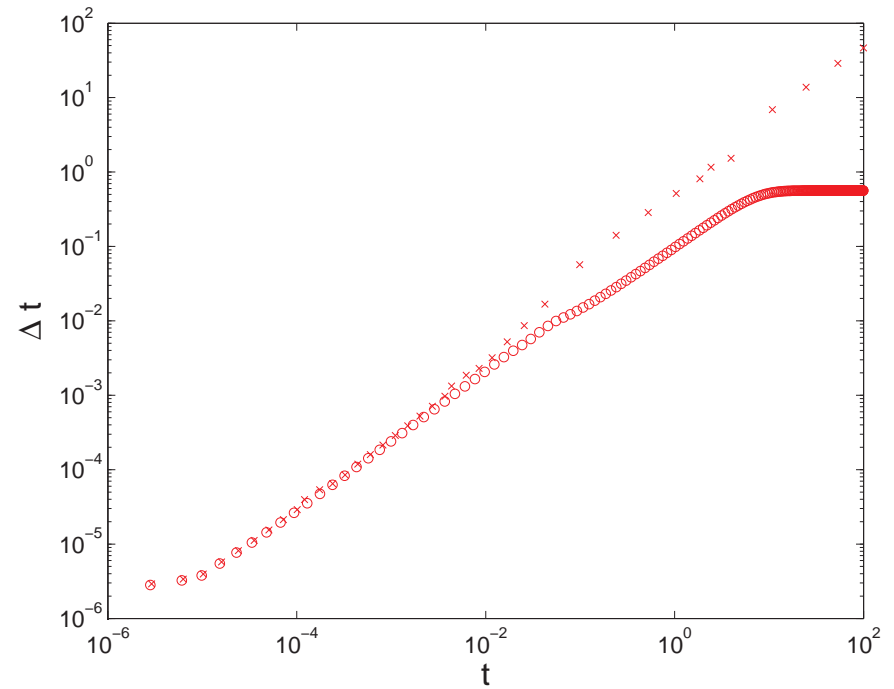
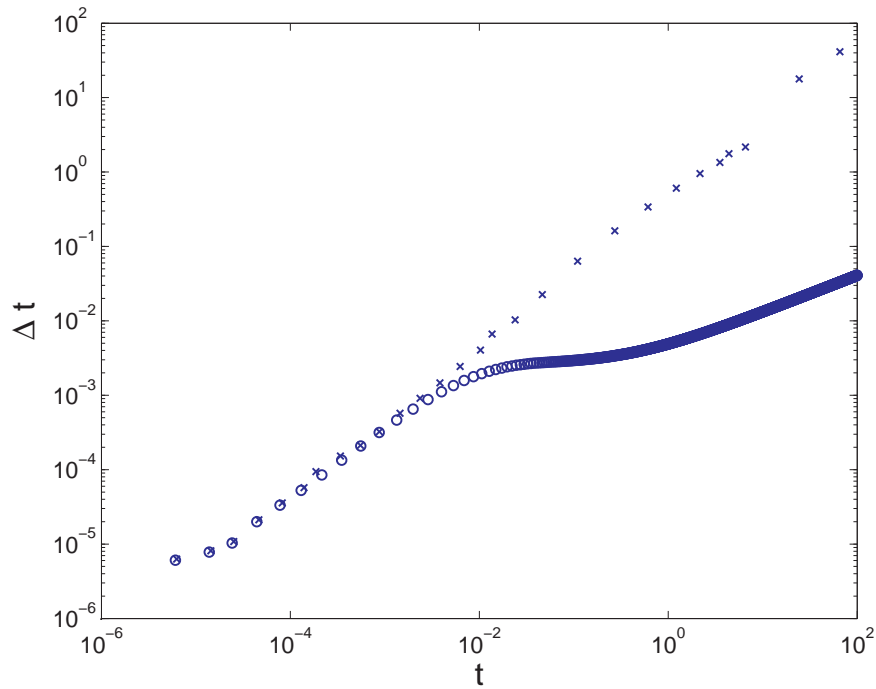
- We perform **time-step averaging** every n^* steps:

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

Contrast this with the standard acceleration obtained by “inverting” the **TR** formula:

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

Stabilized AB2-TR

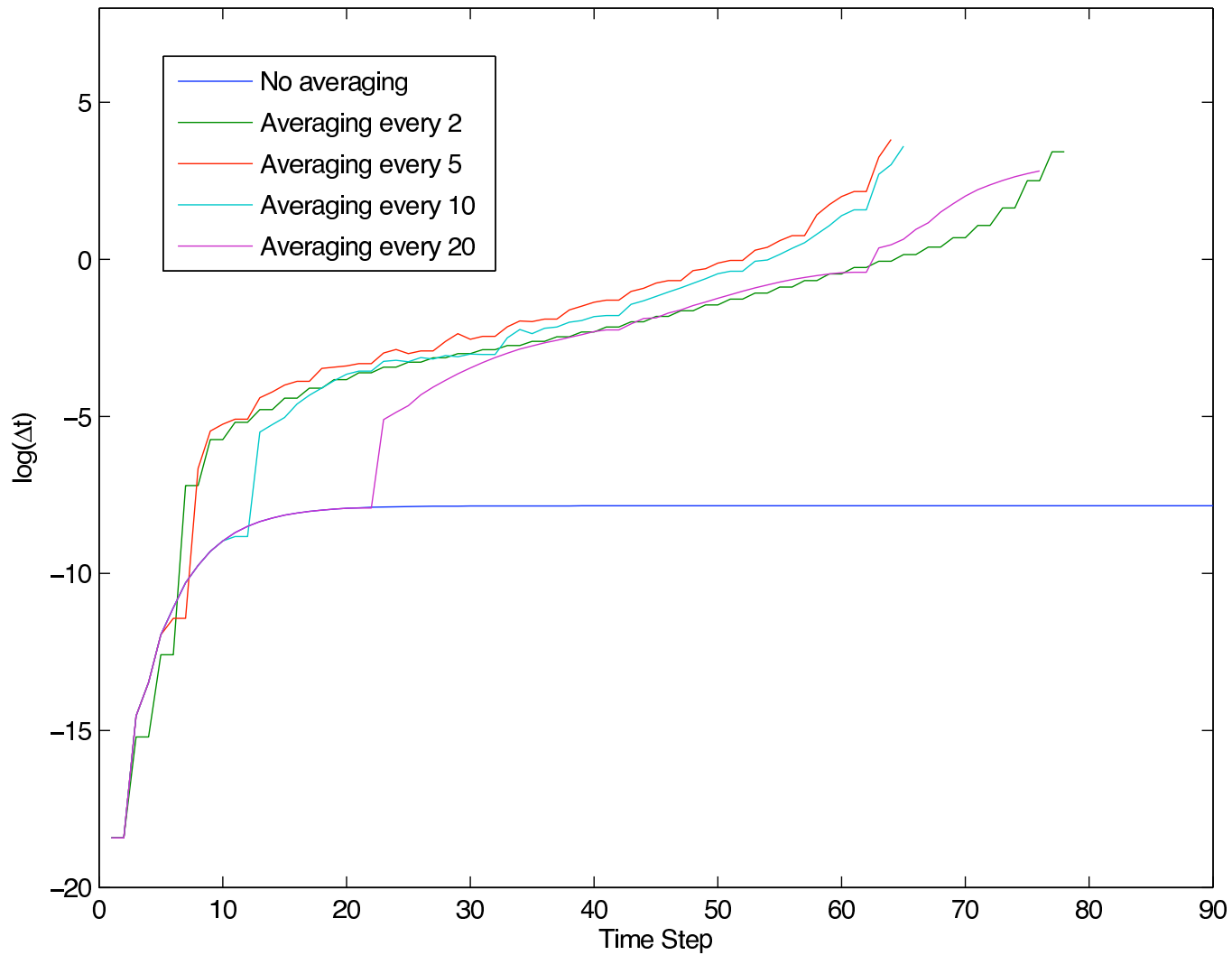


Advection-Diffusion of step profile on Shishkin grid.

$$\text{tol} = 10^{-3}$$

$$\text{tol} = 10^{-4}$$

Stabilized AB2-TR



“Spin up” driven cavity flow with $\nu = 1/100$.

Adaptive Time-Stepping Algorithm I

- The following parameters must be specified:

time accuracy tolerance	tol	(10^{-4})
GMRES tolerance	<code>itol</code>	(10^{-6})
GMRES iteration limit	<code>maxit</code>	(50)

Adaptive Time-Stepping Algorithm I

- The following parameters must be specified:

time accuracy tolerance **tol** (10^{-4})

GMRES tolerance `itol` (10^{-6})

GMRES iteration limit `maxit` (50)

- Starting from rest, $\vec{u}^0 = \vec{0}$, and given a steady state boundary condition $\vec{u}(\vec{x}, t) = \vec{g}$, we model the impulse with a time-dependent boundary condition:

$$\vec{u}(\vec{x}, t) = \vec{g}(1 - e^{-5t}) \quad \text{on } \Gamma_D \times [0, T].$$

Adaptive Time-Stepping Algorithm I

- The following parameters must be specified:

time accuracy tolerance **tol** (10^{-4})

GMRES tolerance `itol` (10^{-6})

GMRES iteration limit `maxit` (50)

- Starting from rest, $\vec{u}^0 = \vec{0}$, and given a steady state boundary condition $\vec{u}(\vec{x}, t) = \vec{g}$, we model the impulse with a time-dependent boundary condition:

$$\vec{u}(\vec{x}, t) = \vec{g}(1 - e^{-5t}) \quad \text{on } \Gamma_D \times [0, T].$$

- We specify the frequency of averaging, typically $n_* = 10$. We also choose a very small initial timestep, typically, $k_1 = 10^{-8}$.

Adaptive Time-Stepping Algorithm II

- Setup the Oseen System (*) and compute $[\alpha^{x,n+1}, \alpha^{y,n+1}]$ using **GMRES**(maxit, itol).

Adaptive Time-Stepping Algorithm II

- Setup the Oseen System (*) and compute $[\alpha^{x,n+1}, \alpha^{y,n+1}]$ using **GMRES**(maxit, itol).
- Compute the **LTE** estimate $e^{v,n+1}$

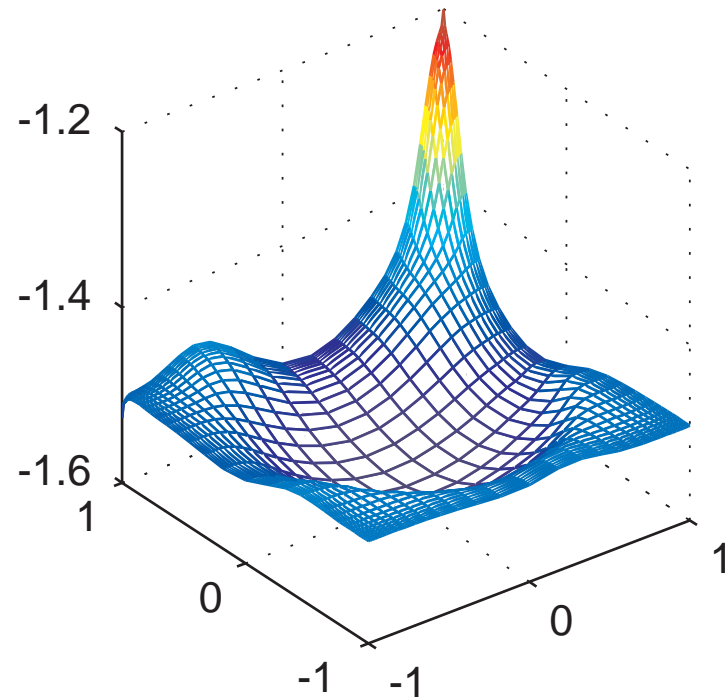
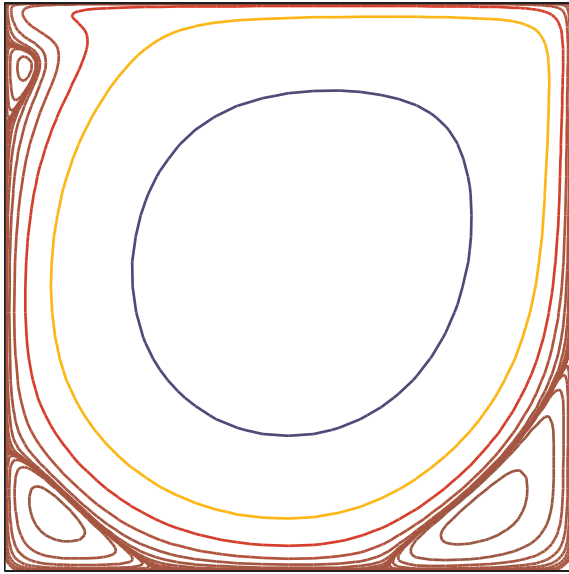
Adaptive Time-Stepping Algorithm II

- Setup the Oseen System (*) and compute $[\alpha^{x,n+1}, \alpha^{y,n+1}]$ using **GMRES**(maxit, itol).
- Compute the **LTE** estimate $e^{v,n+1}$
- If $\|e^{v,n+1}\| > (1/0.7)^3 \text{tol}$, we **reject** the current time step, and repeat the old time step with $k_{n+1} = k_{n+1} \left(\frac{\text{tol}}{\|e^{v,n+1}\|} \right)^{1/3}$.

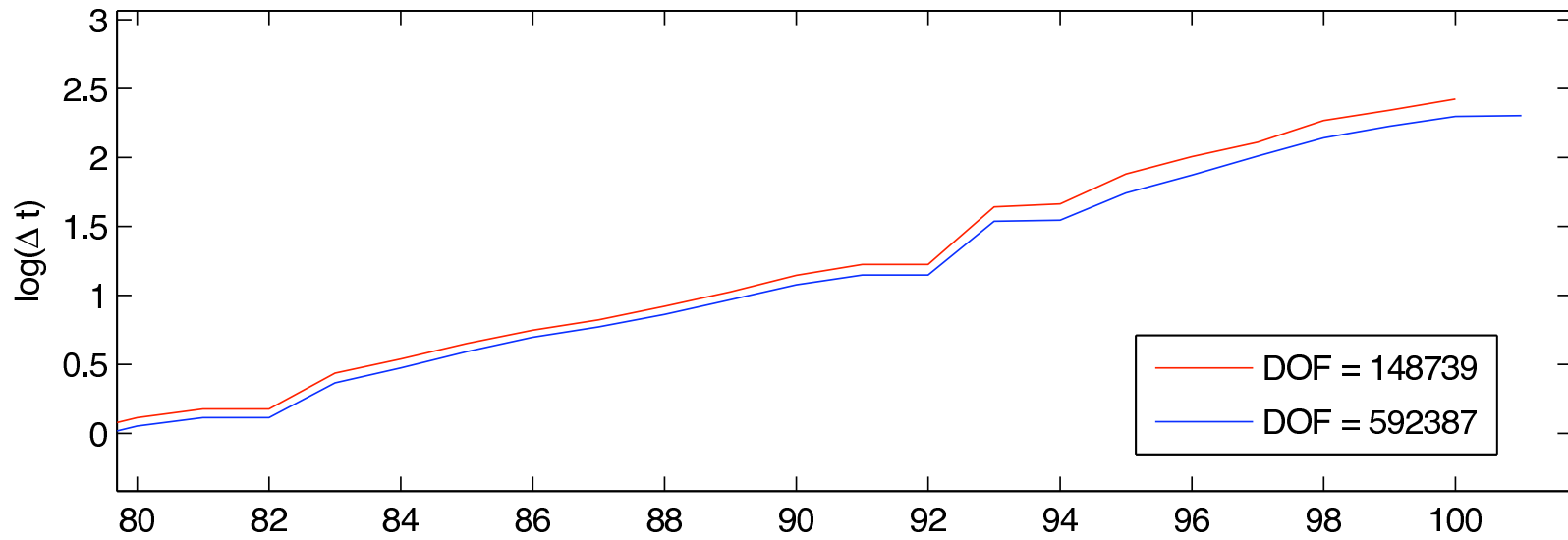
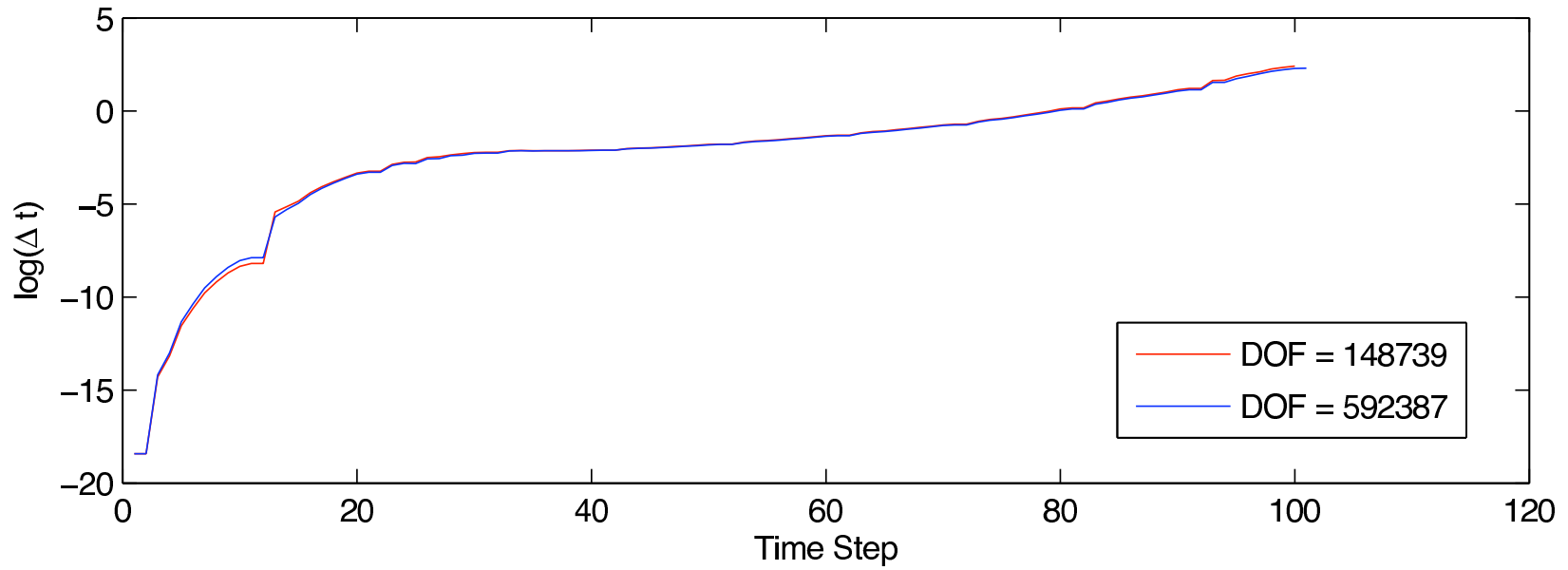
Adaptive Time-Stepping Algorithm II

- Setup the Oseen System (*) and compute $[\alpha^{x,n+1}, \alpha^{y,n+1}]$ using **GMRES**(maxit, itol).
- Compute the **LTE** estimate $e^{v,n+1}$
- If $\|e^{v,n+1}\| > (1/0.7)^3 \text{tol}$, we **reject** the current time step, and repeat the old time step with
$$k_{n+1} = k_{n+1} \left(\frac{\text{tol}}{\|e^{v,n+1}\|} \right)^{1/3}.$$
- Otherwise, **accept** the step and continue with $n = n + 1$ and k_{n+2} based on the **LTE** estimate and the accuracy tolerance **tol**.

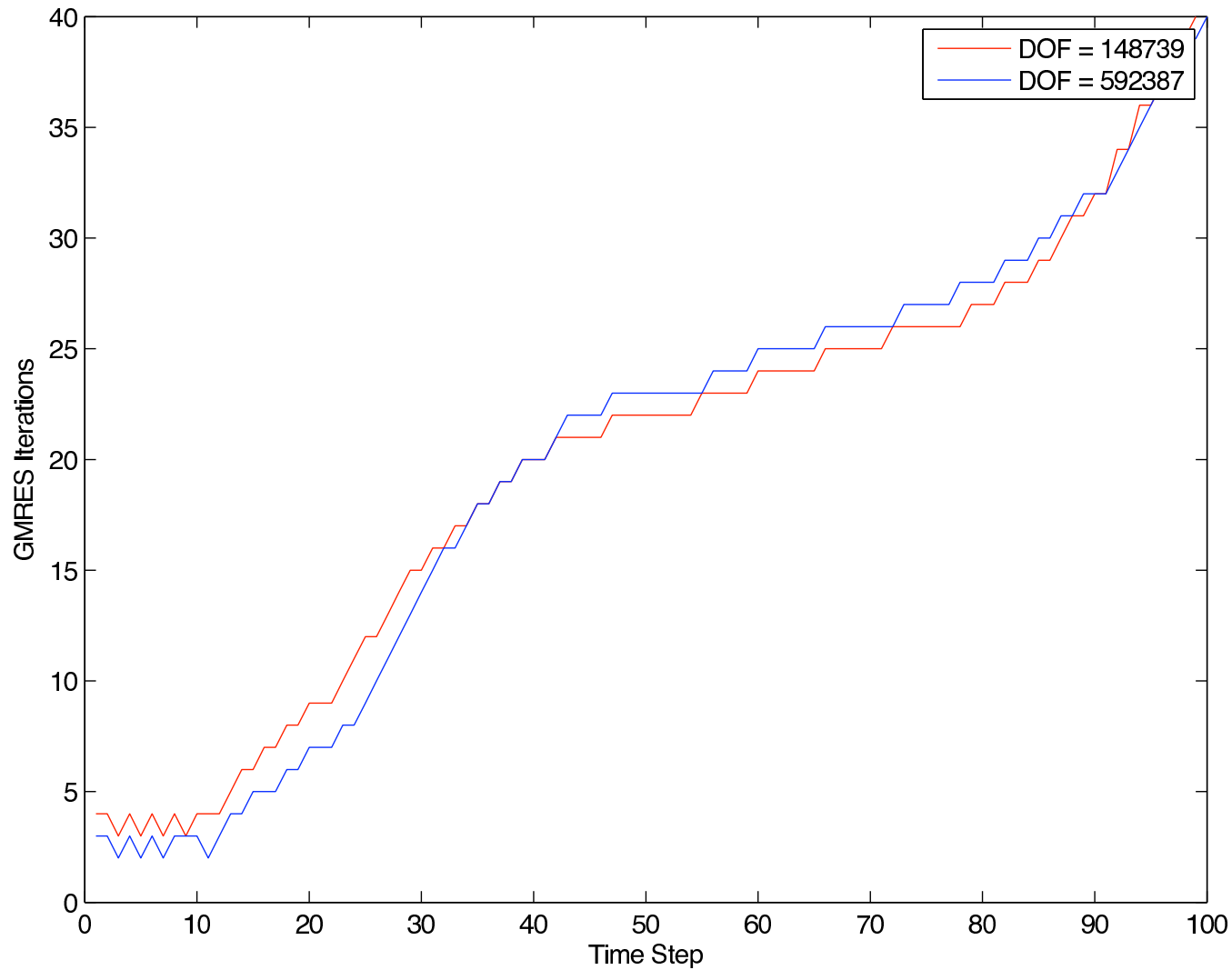
Example Flow Problem – I ($\nu = 1/1000$)



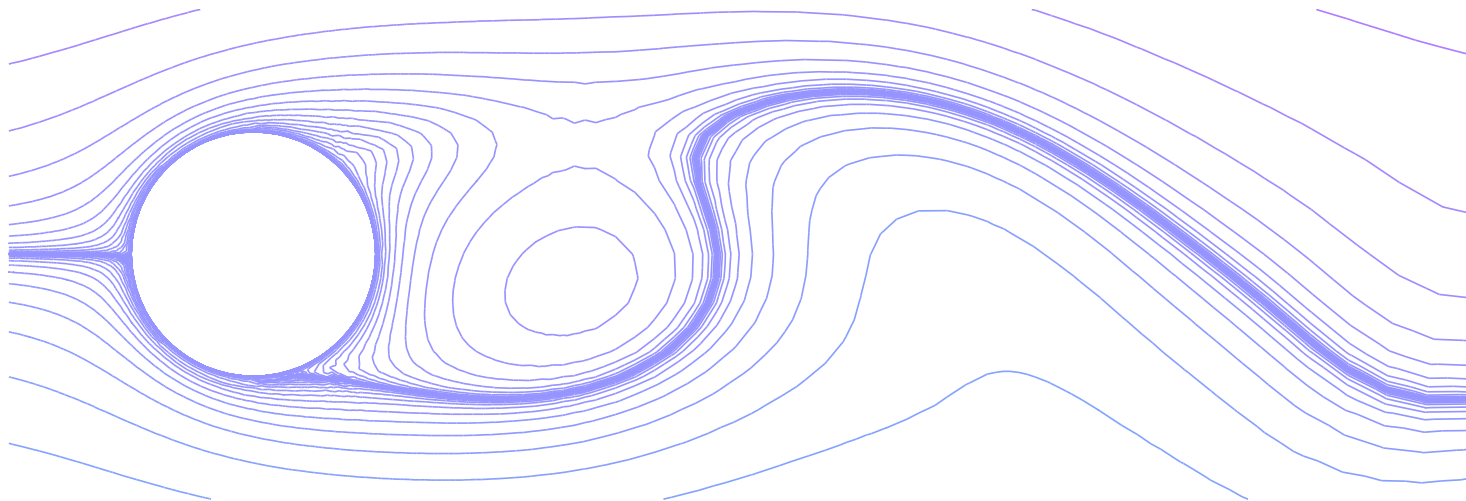
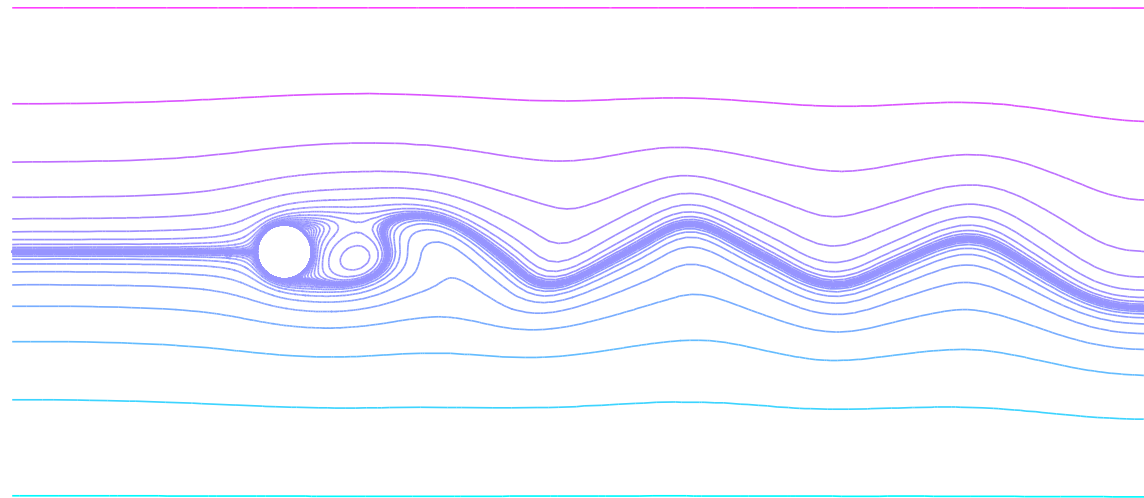
Time step evolution



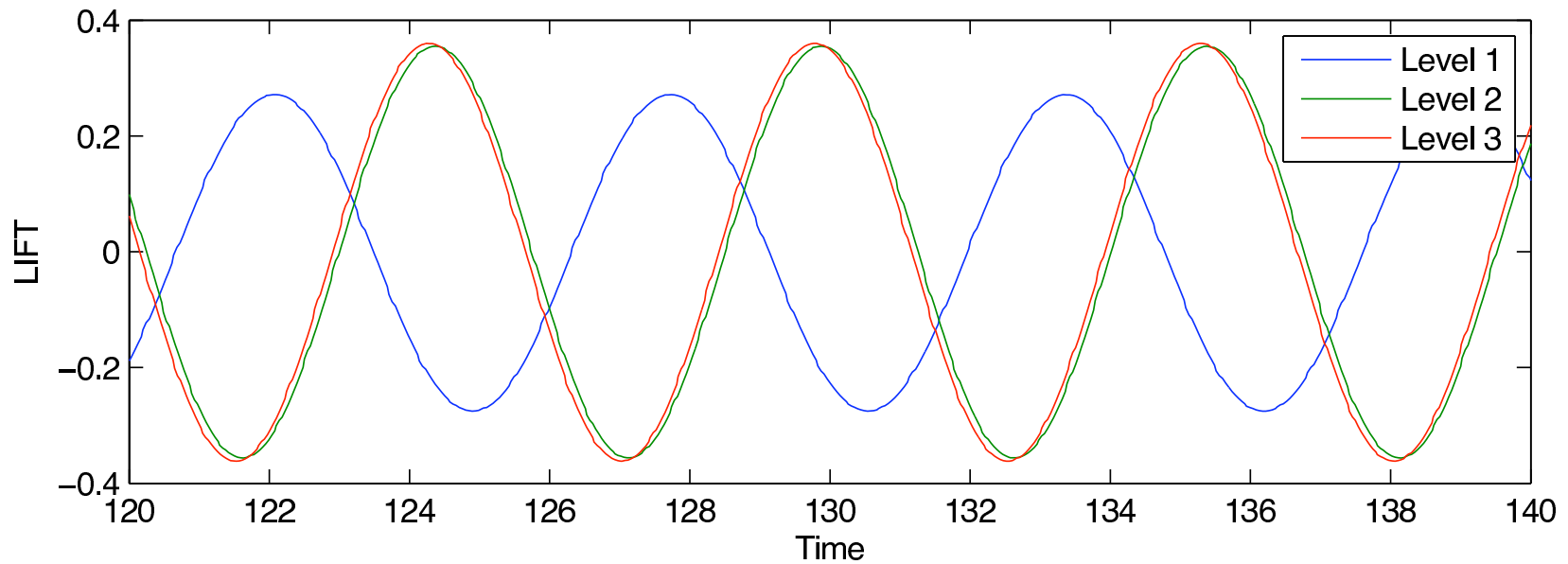
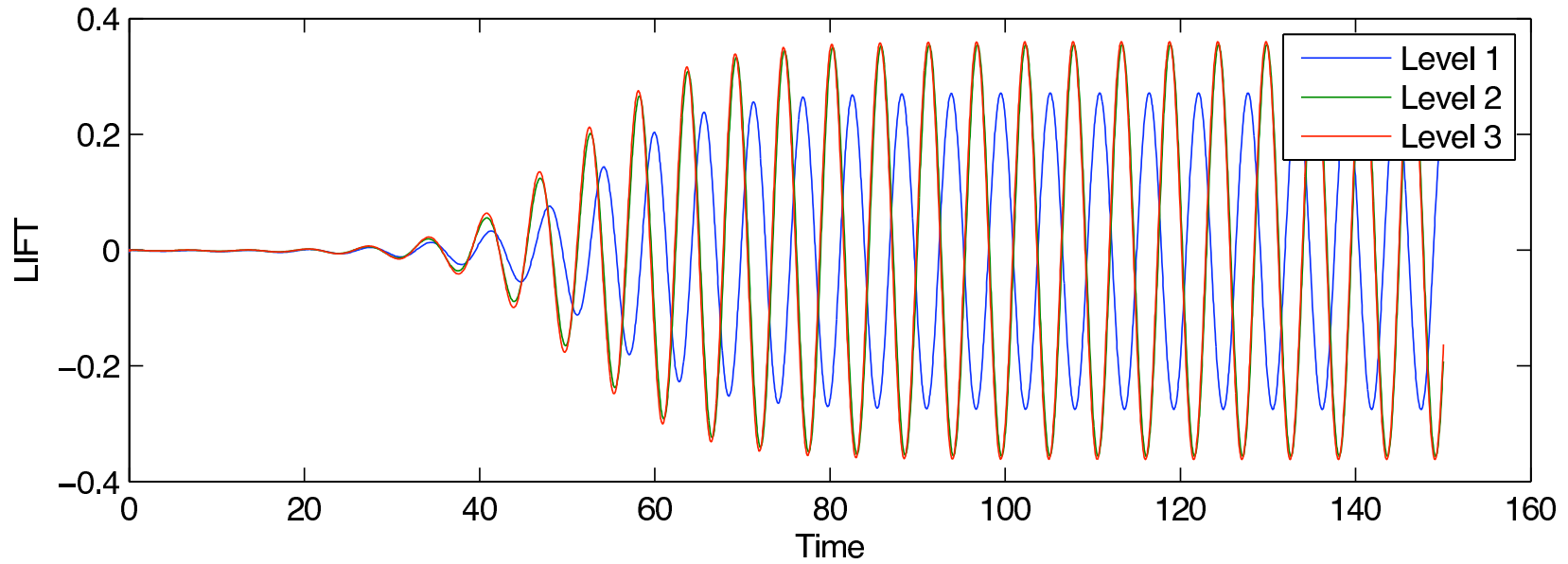
Linear solver performance



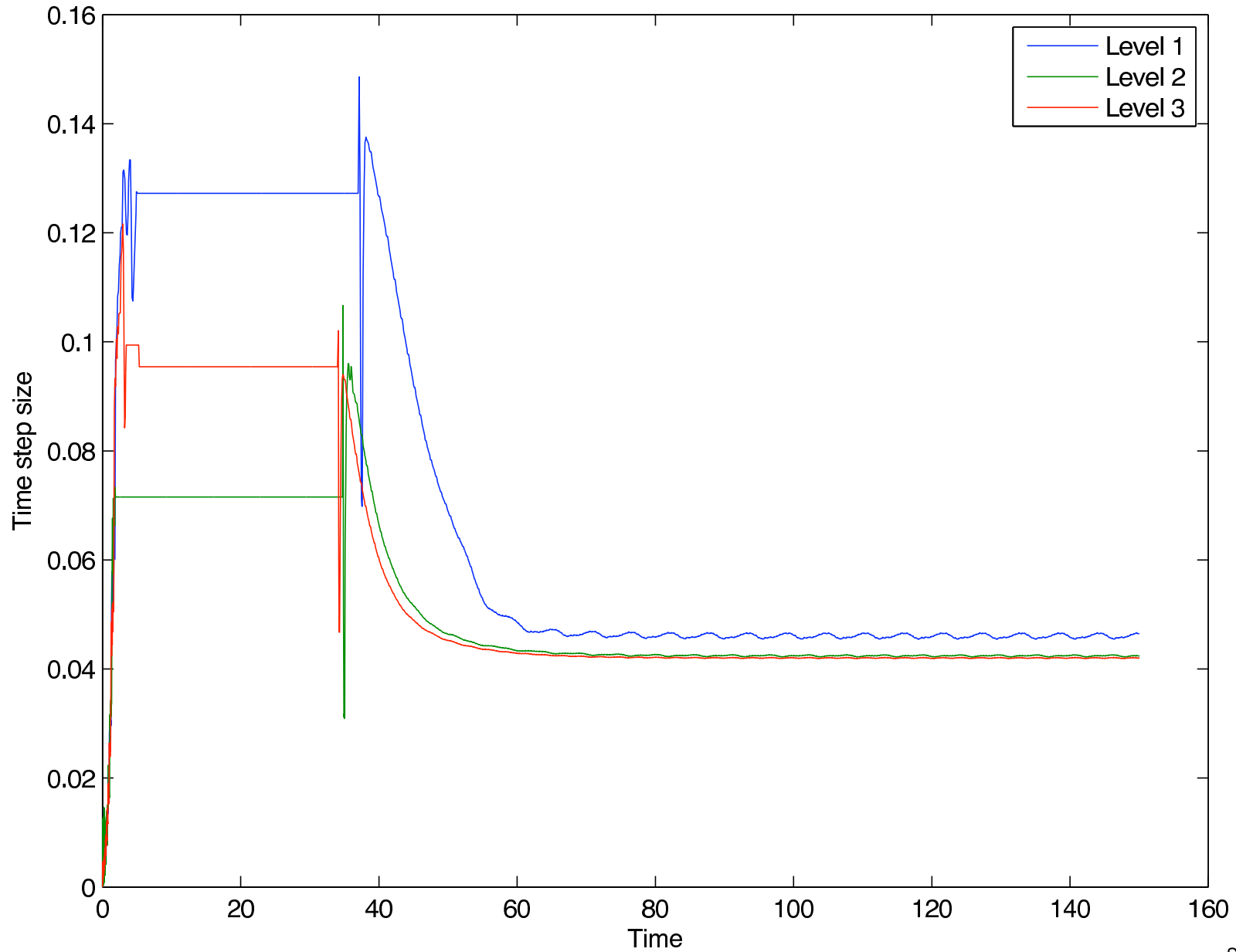
Example Flow Problem – II ($\nu = 1/100$)



Lift Coefficient



Time step evolution



- Update : 2000 – 2010
- 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.
- David Kay & Philip Gresho & David Griffiths & David Silvester
Adaptive time-stepping for incompressible flow; part II: Navier-Stokes equations. SIAM J. Scientific Computing, 32: 111–128, 2010.
- Howard Elman & Milan Mihajlović & David Silvester
Fast iterative solvers for buoyancy driven flow problems. MIMS Eprint 2010.75.

Bouyancy driven flow

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} - \nu \nabla^2 \vec{u} + \nabla p = \vec{j}T \quad \text{in } \mathcal{W} \equiv \Omega \times (0, T)$$

$$\nabla \cdot \vec{u} = 0 \quad \text{in } \mathcal{W}$$

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T - \nu \nabla^2 T = 0 \quad \text{in } \mathcal{W}$$

Boundary and Initial conditions

$$\vec{u} = \vec{0} \quad \text{on } \Gamma \times [0, T]; \quad \vec{u}(\vec{x}, 0) = \vec{0} \quad \text{in } \Omega.$$

$$T = T_g \quad \text{on } \Gamma_D \times [0, T]; \quad \nu \nabla T \cdot \vec{n} = 0 \quad \text{on } \Gamma_N \times [0, T];$$

$$T(\vec{x}, 0) = T_0(\vec{x}) \quad \text{in } \Omega.$$

Finite element matrix formulation

Introducing the basis sets

$$\begin{aligned} \mathbf{X}_h &= \text{span}\{\vec{\phi}_i\}_{i=1}^{n_u}, & \text{Velocity basis functions;} \\ M_h &= \text{span}\{\psi_j\}_{j=1}^{n_p}, & \text{Pressure basis functions.} \\ T_h &= \text{span}\{\phi_k\}_{k=1}^{n_T}, & \text{Temperature basis functions;} \end{aligned}$$

gives the method-of-lines discretized system:

$$\begin{pmatrix} M & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & M \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{u}}{\partial t} \\ \frac{\partial p}{\partial t} \\ \frac{\partial T}{\partial t} \end{pmatrix} + \begin{pmatrix} F & B^T & -\overset{\circ}{M} \\ B & 0 & 0 \\ 0 & 0 & F \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \\ T \end{pmatrix} = \begin{pmatrix} \vec{0} \\ 0 \\ g \end{pmatrix}$$

with a (vertical–) **mass** matrix:

$$\left(\frac{\circ}{M}\right)_{ij} = ([0, \phi_i], \phi_j)$$

Preconditioning strategy

$$\begin{pmatrix} F & B^T & -\frac{\circ}{M} \\ B & 0 & 0 \\ 0 & 0 & F \end{pmatrix} \mathcal{P}^{-1} \mathcal{P} \begin{pmatrix} \alpha^u \\ \alpha^p \\ \alpha^T \end{pmatrix} = \begin{pmatrix} \mathbf{f}^u \\ \mathbf{f}^p \\ \mathbf{f}^T \end{pmatrix}$$

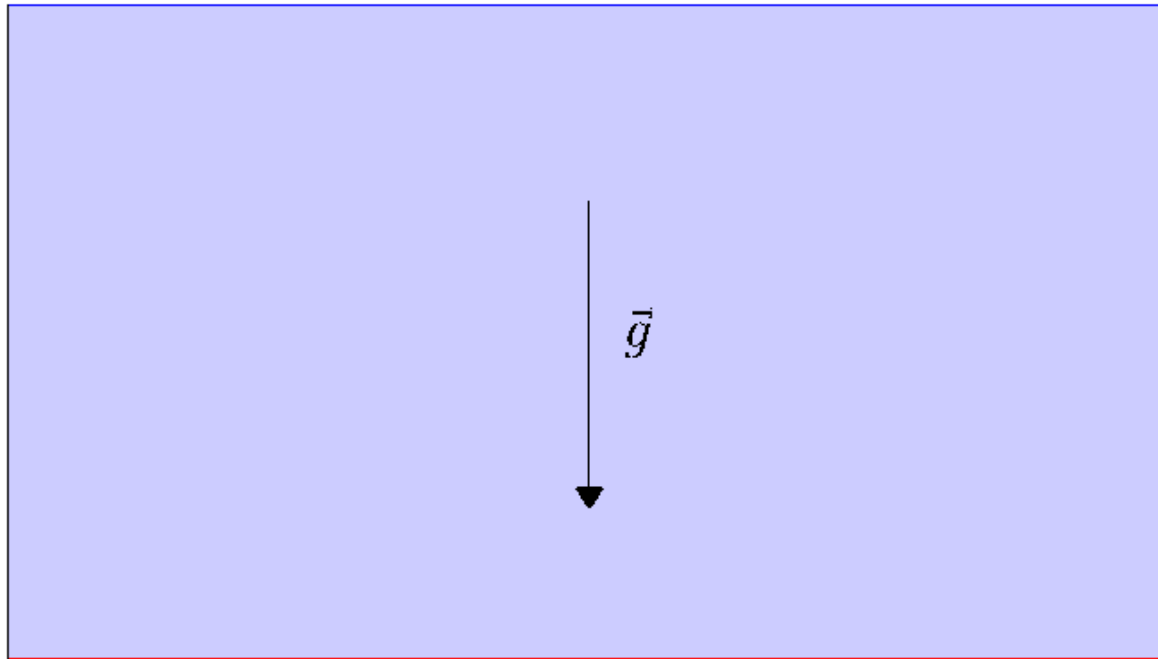
Given $S = BF^{-1}B^T$, a **perfect** preconditioner is given by

$$\begin{pmatrix} F & B^T & -\frac{\circ}{M} \\ B & 0 & 0 \\ 0 & 0 & F \end{pmatrix} \underbrace{\begin{pmatrix} F^{-1} & F^{-1}B^T S^{-1} & F^{-1}\frac{\circ}{M}F^{-1} \\ 0 & -S^{-1} & 0 \\ 0 & 0 & F^{-1} \end{pmatrix}}_{\mathcal{P}^{-1}} = \begin{pmatrix} I & 0 & 0 \\ BF^{-1} & I & BF^{-1}\frac{\circ}{M}F^{-1} \\ 0 & 0 & I \end{pmatrix}$$

Example: Natural Convection in 1:16 cavity

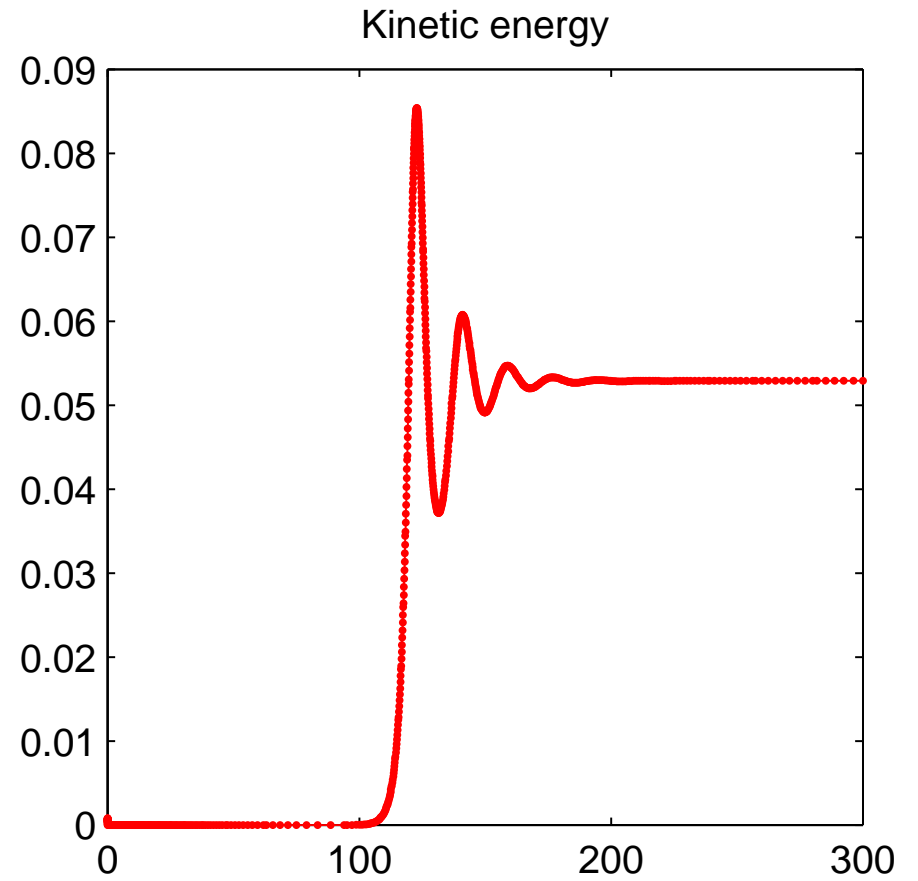
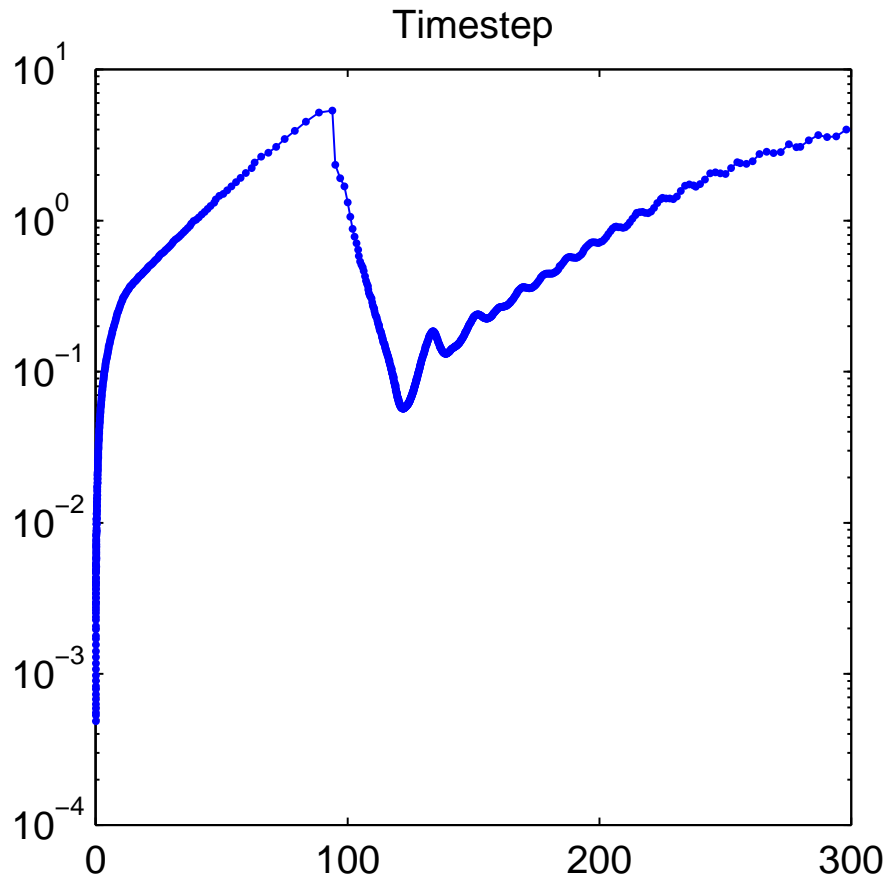
Rayleigh-Bernard convection

T_c

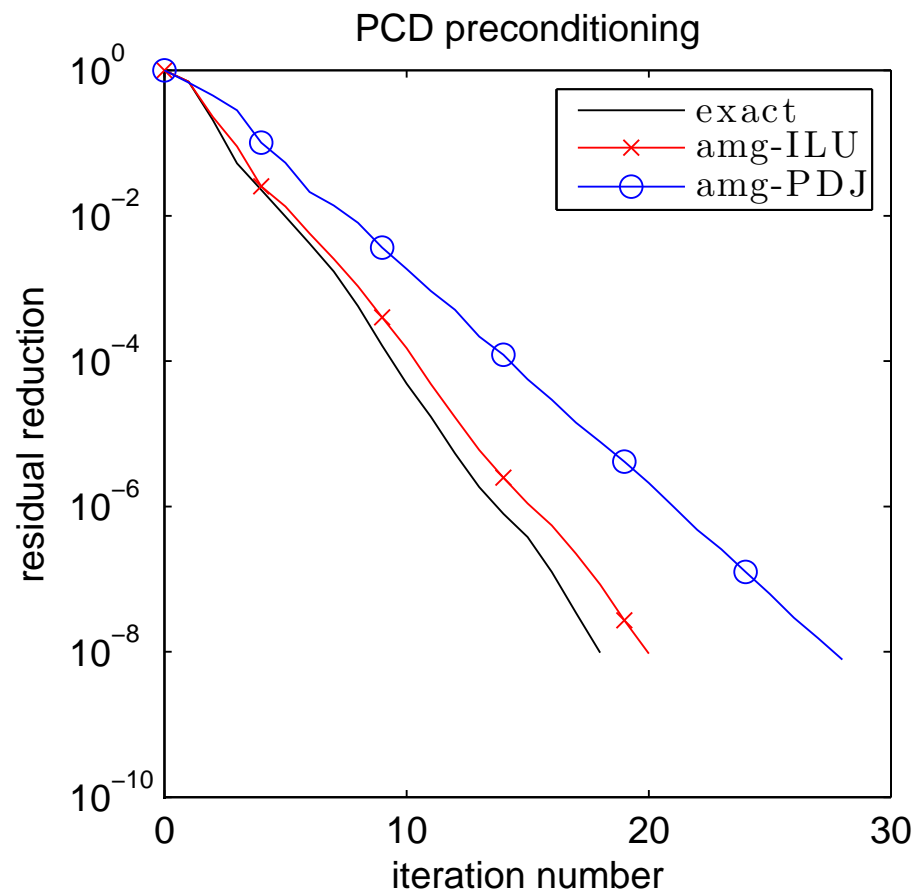
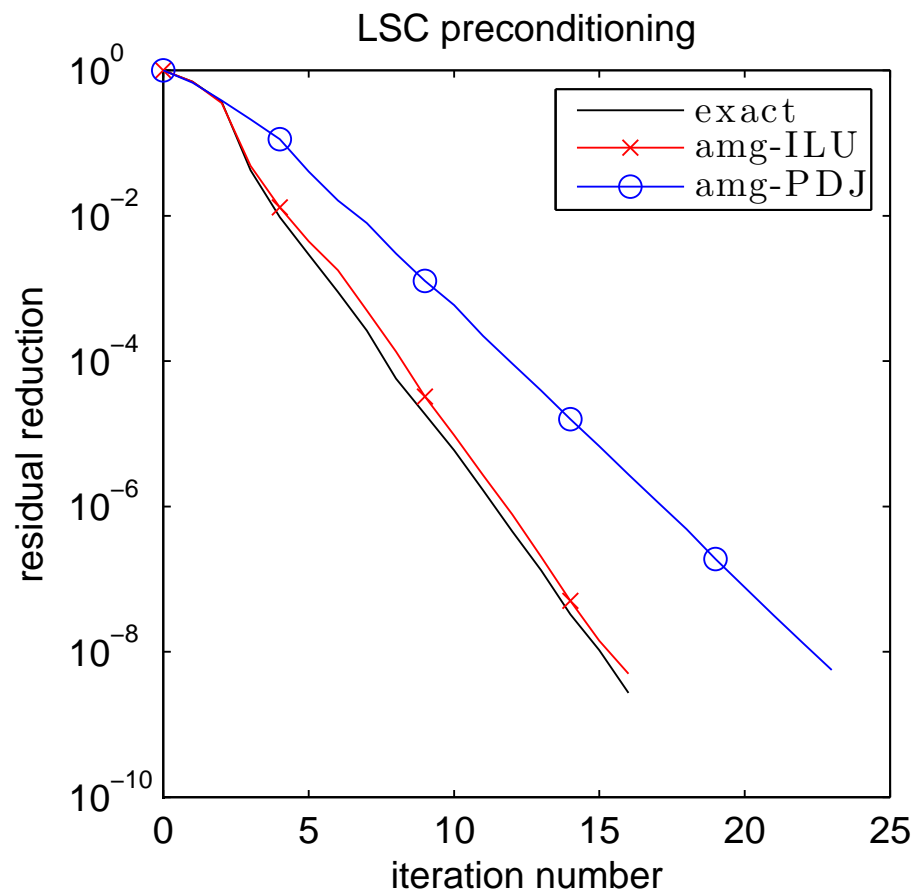


T_h

Adaptive timestepping ...



... with efficient linear algebra



What have we achieved?

- **Black-box implementation:** few parameters that have to be estimated a priori.
- **Optimal complexity:** essentially $O(n)$ flops per iteration, where n is dimension of the discrete system.
- **Efficient linear algebra:** convergence rate is (essentially) independent of h . Given an appropriate time accuracy tolerance convergence is also robust with respect to ν

References

Semi-Implicit, Fractional-Step, Pressure Projection — I

- A.J. Chorin
A numerical method for solving incompressible viscous flow problems, J. Comput. Phys., 2:12–26, 1967.
- J. Kim & P. Moin
Application of a fractional–step method to incompressible Navier–Stokes equations, J. Comput. Phys., 59:308–323, 1985.
- J. Van Kan
A second–order accurate pressure–correction scheme for viscous incompressible flow, SIAM J. Sci. Stat. Comput., 7(3):870–891, 1986.
- J.B. Bell, P. Colella & H.M. Glaz
A second–order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys., 85:(2)257–283, 1989.

Semi-Implicit, Fractional-Step, Pressure Projection — II

- P.M. Gresho, S.T. Chan, R.L. Lee & C.D. Upson
A modified finite element method for solving the time-dependent, incompressible Navier–Stokes equations. Part 1: Theory,
Int. J. Numer. Meth. Fluids, 4:557–598, 1984.
- P.M. Gresho & S.T. Chan
On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation,
Int. J. Numer. Meth. Fluids, 11(5):621-660, 1990.
- J.B. Perot
An analysis of the fractional step method,
J. Comput. Phys., 108:51–58, 1993.

Semi-Implicit, Fractional-Step, Pressure Projection — III

- J.–L. Guermond
Sur l'approximation des équations de Navier–Stokes instationnaires par une méthode de projection, C.R. Acad. Sci. Paris, 319:887–892, 1994. Serie I.
- R. Rannacher
The Navier–Stokes Equations II: Theory and Numerical Methods, Springer–Verlag, Berlin, Germany, 1992.
Chap. On Chorin's projection method for the incompressible Navier–Stokes equations; pp. 167–183; Lecture Notes in Mathematics, Vol. 1530.

Semi-Implicit, Fractional-Step, Pressure Projection — IV

- W. E & J.-G. Liu
Projection method I: Convergence and numerical boundary layers,
SIAM J. Numer. Anal., 32(4):1017–1057, 1995.
- W. E & J.-G. Liu
Vorticity boundary conditions and related issues for finite difference schemes,
J. Comput. Phys., 124:368–382, 1996.

Semi-Implicit, Fractional-Step, Pressure Projection — V

- J. Shen
On error estimates of some higher order projection and penalty–projection methods for Navier–Stokes equations, Numer. Math., 62:49–73, 1992.
- J. Shen
On error estimates of the projection methods for the Navier–Stokes equations: Second–order schemes, Math. Comput., 65, 215:1039–1065, 1996.
- A. Prohl
Analysis of Chorin’s projection method for solving the incompressible Navier–Stokes equations, Universität Heidelberg, Institut für Angewandte Mathematik, INF 294, D–69120 Heidelberg, Germany, 1996.

Transport Diffusion, Lagrange-Galerkin, Backward Method of Characteristics — I

- P. Hansbo
The characteristic streamline diffusion method for the time–dependent incompressible Navier–Stokes equations, *Comput. Meth. Appl. Mech. Eng.*, 99:171–186, 1992.
- O. Pironneau
On the transport–diffusion algorithm and its application to the Navier–Stokes equations, *Numer. Math.*, 38:309–332, 1982.
- K.W. Morton, A. Priestley & E. Süli
Stability of the Lagrange–Galerkin method with non–exact integration, *Math. Model. Numer. Anal.*, 22:(4)625–653, 1988.

Transport Diffusion, Lagrange-Galerkin, Backward Method of Characteristics — II

- **A. Priestley**
Exact projections and the Lagrange–Galerkin method: A realistic alternative to quadrature, J. Comput. Phys., 112:316–333, 1994
- **R. Bermejo**
Analysis of an algorithm for the Galerkin–characteristic method, Numer. Math., 60:163–194, 1991.

Fractional Step, le Θ scheme — I

- M.O. Bristeau, R. Glowinski, B. Mantel, J. Periaux & P. Perrier, **Numerical methods for incompressible and compressible Navier–Stokes problems**, Finite Elements in Fluids 6 (R.H. Gallagher, G. Carey, J.T. Oden & O.C. Zienkiewicz, eds), Chichester: Wiley, 1985. pp. 1–40.
- **E.J. Dean & R. Glowinski**
On some finite element methods for the numerical simulation of incompressible viscous flow, Incompressible Computational Fluid Dynamics., (M.D. Gunzburger & R.A. Nicolaides, eds), Cambridge University Press, 1993. pp. 17–65.

Fractional Step, ie Θ scheme — II

- R. Rannacher
Applications of mathematics in industry and technology, B.G. Teubner, Stuttgart, Germany, 1989. Chap. Numerical analysis of nonstationary fluid flow (a survey); pp. 34–53; (V.C. Boffi & H. Neunzert, eds).
- R. Rannacher
Navier–Stokes equations: theory and numerical methods, Springer–Verlag, Berlin, Germany, 1990. Chap. On the numerical analysis of the non–stationary Navier–Stokes equations; pp. 180–193; (J. Heywood et al., eds).

Fractional Step, ie Θ scheme — III

- S. Turek
A comparative study of some time–stepping techniques for the incompressible Navier–Stokes equations: From fully implicit nonlinear schemes to semi–implicit projection methods, *Int. J. Numer. Meth. Fluids*, 22(10):987–1012, 1996.
- A. Smith & D.J. Silvester
Implicit algorithms and their linearization for the transient incompressible Navier–Stokes equations, *IMA J. Numer. Anal.* 17: 527–545, 1997.