SIAM Conference on
Computational Science and Engineering
February 25-March 1, 2013
The Westin Boston Waterfront
Boston, Massachusetts, USA

2013 SIAM Conference
On Computational Science and Engineering
Boston, 27th February 2013

PALADINS:
Scalable Time-Adaptive Algebraic Splitting and Preconditioners for the Navier-Stokes Equations

U. Villa, A. Veneziani
Department of Mathematics and Computer Science
EMORY UNIVERSITY
Atlanta, GA, US

Outline:

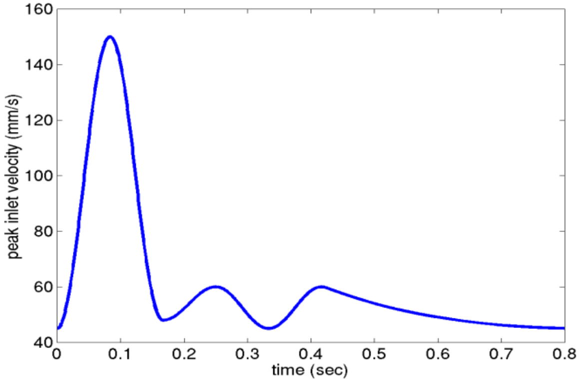
- **Introduction**
  - Motivations
  - Basic settings
  - Incremental pressure schemes
- **Algebraic Splittings**
  - Inexact LU block factorizations
  - High Order Yosida Schemes
- **Adaptivity**
  - Local splitting error analysis
  - A posteriori error estimators
  - An application to blood flow problems
- **Scalability**
  - Strong scaling test
  - Weak scaling test
- **Conclusions**

Motivations:

Some applications of INS feature *sequences of fast and slow transients*

Blood flow dynamics:

- fast transients during **systole**
- slower dynamics during **diastole**



☑ Time adaptivity can reduce CPU times in these applications…

☑ An effective a-posteriori error estimator is however mandatory

Standard a-posteriori error estimator requires:

- Complex *space-time error estimator* or
- The *comparison of two numerical solutions* obtained with different accuracy time discretizations (eg. Adams–Bashforth/BDF2, as in Kay, Gresho, Griffiths, Silvester, 2008)

Algebraic splittings of velocity/pressure can provide effective estimator as a by product of the computations.

Basic settings:

Incompressible Unsteady Navier-Stokes Equations

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = \mathbf{f}(\mathbf{x},t) \\ \nabla\cdot\mathbf{u} = 0 \end{cases}$$

Discretization

Space: **Galerkin methods** *LBB conditions fulfilled*

Time: **BDF*q*** schemes ($q \leq 3$)

At each time level $t = t^{n+1}$ we need to solve the system:

$$\mathcal{A}\mathbf{y}^{n+1} = \mathbf{b}^{n+1}$$

$$\begin{bmatrix} C & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_p \end{bmatrix}$$

$C = \dfrac{\alpha_0}{\Delta t} M + A$ being A the stiffness matrix (viscous stresses + convection terms)

Incremental Pressure Schemes:

At each time step we write $p^n = \delta p^n + \sigma_p^n$, where

σ_p^n is the pressure extrapolated from previous time steps, and

$$\|\delta p^n\| = \|p^n - \sigma_p^n\| = \mathcal{O}(\Delta t^s)$$

For example if s=1 then $\sigma_p^n = p^{n-1}$ and if s = 2 then $\sigma_p^n = 2p^{n-1} - p^{n-2}$

The incremental pressure formulation reads $\begin{bmatrix} C & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^n \\ \delta p^n \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u - D^T \sigma_p^n \\ f_p \end{bmatrix}$

- **For velocity/pressure splitting:**
  - Incremental pressure schemes **improve the accuracy** in time.
  - High order extrapolation in time might **reduce the stability**.

- **For Schur-Complement/Monolithic solutions:**
  - Incremental pressure provides a **good initial guess**.
  - High order extrapolation in time **does not affect stability**.

A. Prohl – *Projection and quasi-compressibility methods for solving the incompressible Navier-Stokes equations*, Wiley Teubner Advances in Num. Math., 1997

Outline:

- **Introduction**
  - Motivations
  - Basic settings
  - Incremental pressure schemes
- **Algebraic Splittings**
  - Inexact LU block factorizations
  - High Order Yosida Schemes
- **Adaptivity**
  - Local splitting error analysis
  - A posteriori error estimators
  - An application to blood flow problems
- **Scalability**
  - Strong scaling test
  - Weak scaling test
- **Conclusions**

Inexact LU block factorization:

$$\mathcal{A} = \begin{bmatrix} C & 0 \\ D & -DC^{-1}D^T \end{bmatrix} \begin{bmatrix} I_{N_u} & C^{-1}D^T \\ 0 & I_{N_p} \end{bmatrix} = LU \qquad \Rightarrow$$

$$\hat{\mathcal{A}} = \begin{bmatrix} C & 0 \\ D & -DFD^T \end{bmatrix} \begin{bmatrix} I_{N_u} & GD^T \\ 0 & Q \end{bmatrix} = \begin{bmatrix} C & CGD^T \\ D & DGD^T - DFD^T Q \end{bmatrix}$$

F and G *appropriate approximations* of C^{-1}, Q is such that $DGD^T - DFD^T Q$ is small

Neumann expansion: $\quad C^{-1} = \frac{\Delta t}{\alpha_0} \sum_{k=1}^{\infty} \left(\frac{-\Delta t}{\alpha_0} \right)^{k-1} (M^{-1}A)^{k-1} M^{-1} \simeq \frac{\Delta t}{\alpha_0} M^{-1} \equiv H$

provided Δt is small enough.

Mass preserving scheme: F=G=H, $Q=I_{Np}$ (*Algebraic Chorin Temam*, Perot '93)

Momentum preserving sch.: $G=C^{-1}$, F=H ($Q=I_{Np}$: *Yosida*, Quarteroni, Saleri, Veneziani,'99)

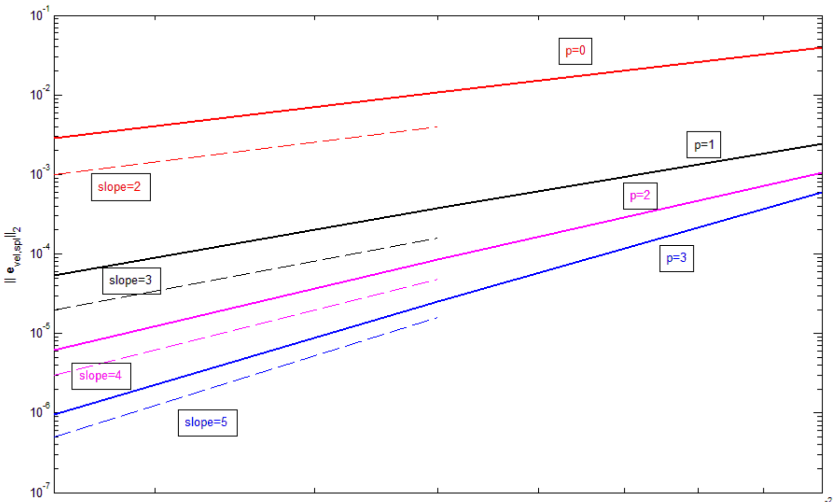
Higher order schemes build a sequence of Q_q such that: $|||\Sigma - SQ_q||| = \mathcal{O}(\Delta t^{q+2})$

being $\Sigma = -DC^{-1}D^T$ and $S = -DHD^T$

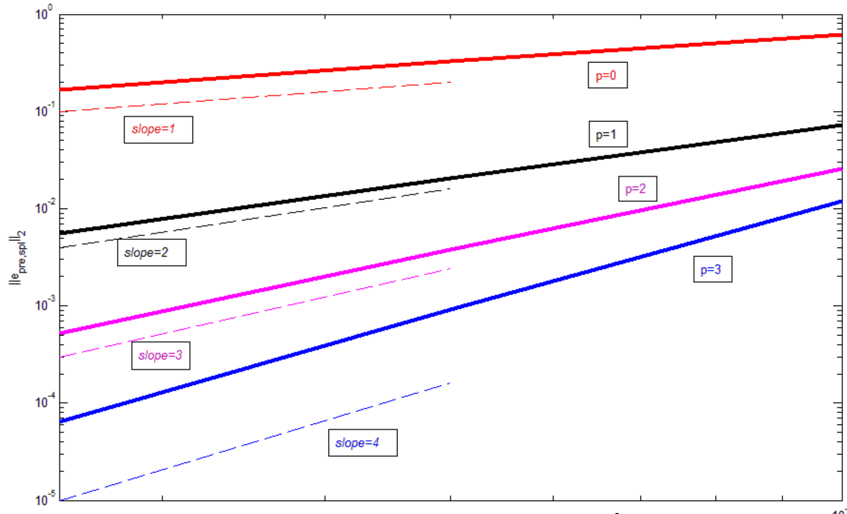
F. Saleri, A. Veneziani – *Pressure correction algebraic splitting methods for the incompressible Navier-Stokes equations* – SIAM J. Num. Anal. (2006)

Inexact LU block factorization:

$$\hat{\mathcal{A}} = \begin{bmatrix} \mathrm{C} & 0 \\ \mathrm{D} & -\mathrm{DFD} \end{bmatrix} \begin{bmatrix} \mathrm{I}_{N_u} & \mathrm{GD}^T \\ 0 & \mathrm{Q} \end{bmatrix} = \begin{bmatrix} \mathrm{C} & \mathrm{CGD}^T \\ \mathrm{D} & \mathrm{DGD}^T - \mathrm{DFD}^T\mathrm{Q} \end{bmatrix}$$



Velocity splitting errors



Pressure splitting errors

Higher order schemes build a sequence of Q_q such that: $|||\Sigma - SQ_q||| = \mathcal{O}(\Delta t^{q+2})$

being $\Sigma = -DC^{-1}D^T$ and $S = -DHD^T$

A. Veneziani, *A Note on the Consistency and Stability Properties of Yosida Fractional Step Schemes for the Unsteady Stokes Equations*, SIAM J. Numer. Anal.,2009

High order Yosida schemes:

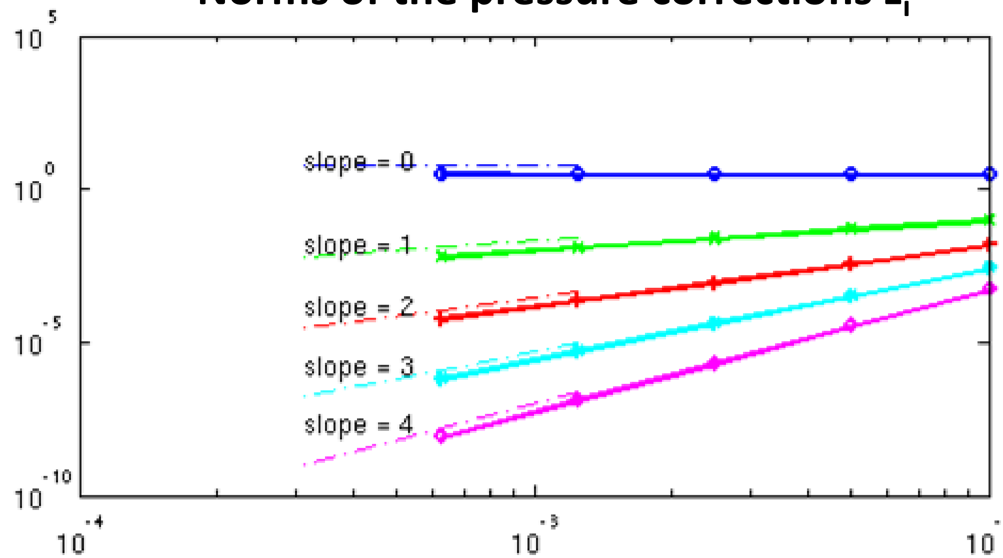
Algorithm to apply $(SQ_q)^{-1}$ to a vector

```
//Pressure corrections
z(i)    = ZeroVector(dim_P);
//Temporanely data structures
zz(i,j) = ZeroVector(dim_U);
dzz(i,j)= ZeroVector(dim_P);

Solve: S z(0) = rhs;
for(i=0; i<q; ++i)
    zz(i,0) = -H A H D^T z(i);
    dzz(i,0) = D zz(i,0);
    cc = dzz(i,0);
    for(j=1; j<1+i; ++j)
        zz(i-j,j) = - H A zz(i-j,j-1);
        dzz(i-j,j) = D zz(i-j,j);
        cc += dzz(i-j,j);
    Solve: S z(i+1) = cc;

P = sum(z);
```

Norms of the pressure corrections z_i

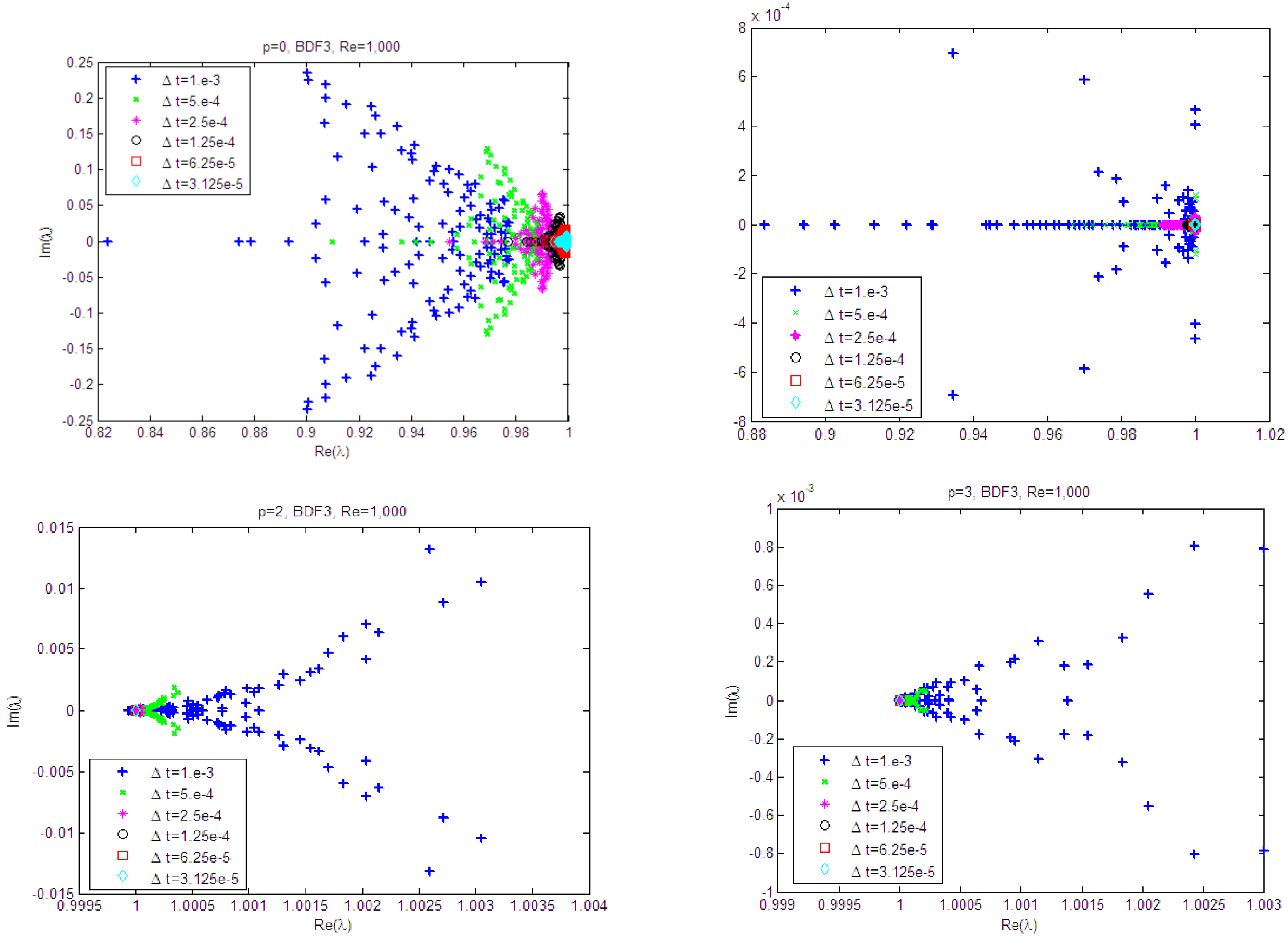


Computational cost for each correction step:

- Three mat-vec in the velocity space
- One linear solve with the spd matrix S

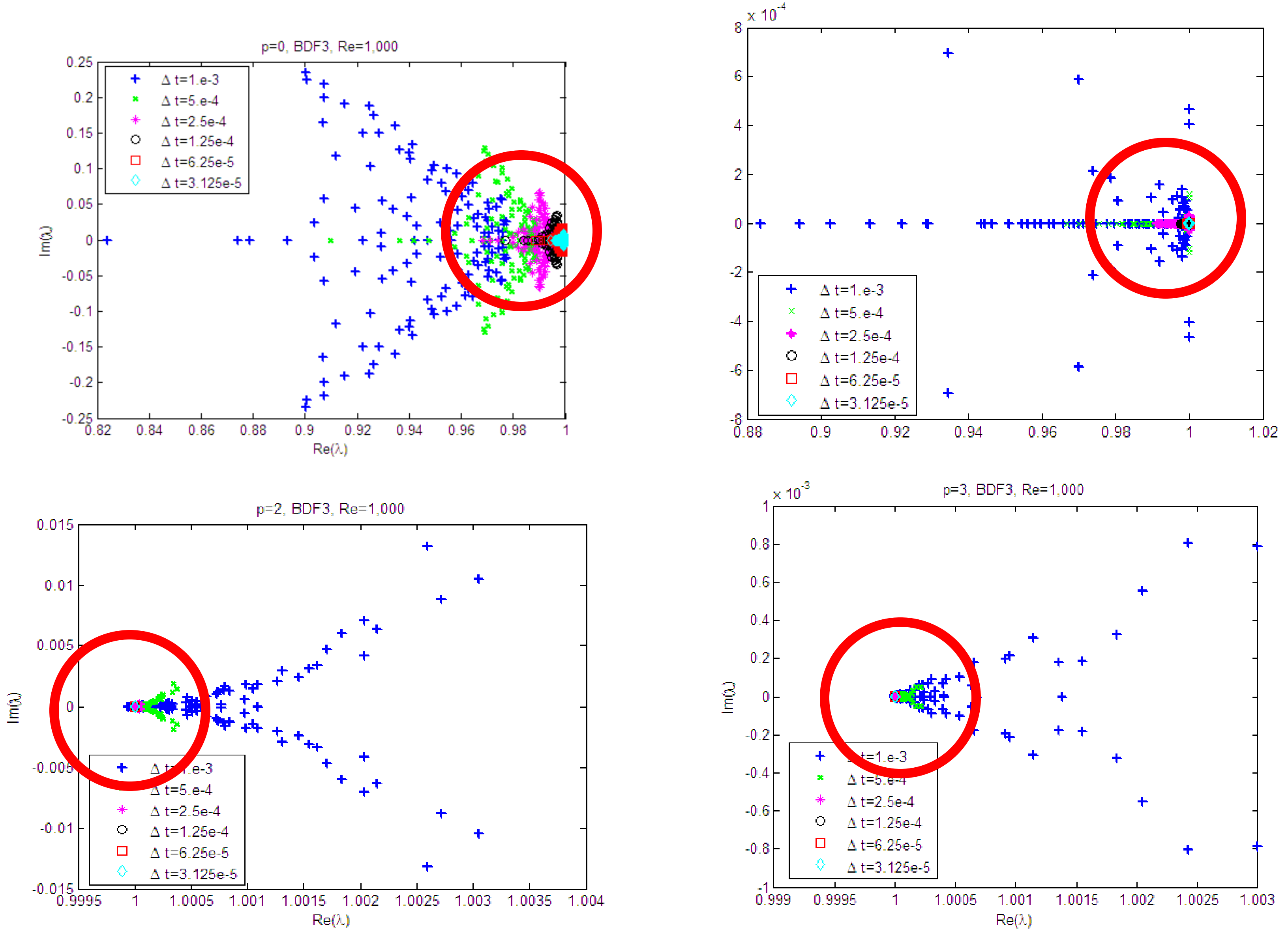
A. Veneziani, U. Villa – *ALADINS: an ALgebraic splitting time ADaptive solver for the Incompressible Navier-Stokes equations*, J. Comput. Phys. (2013)

High Order Yosida as Preconditioner:



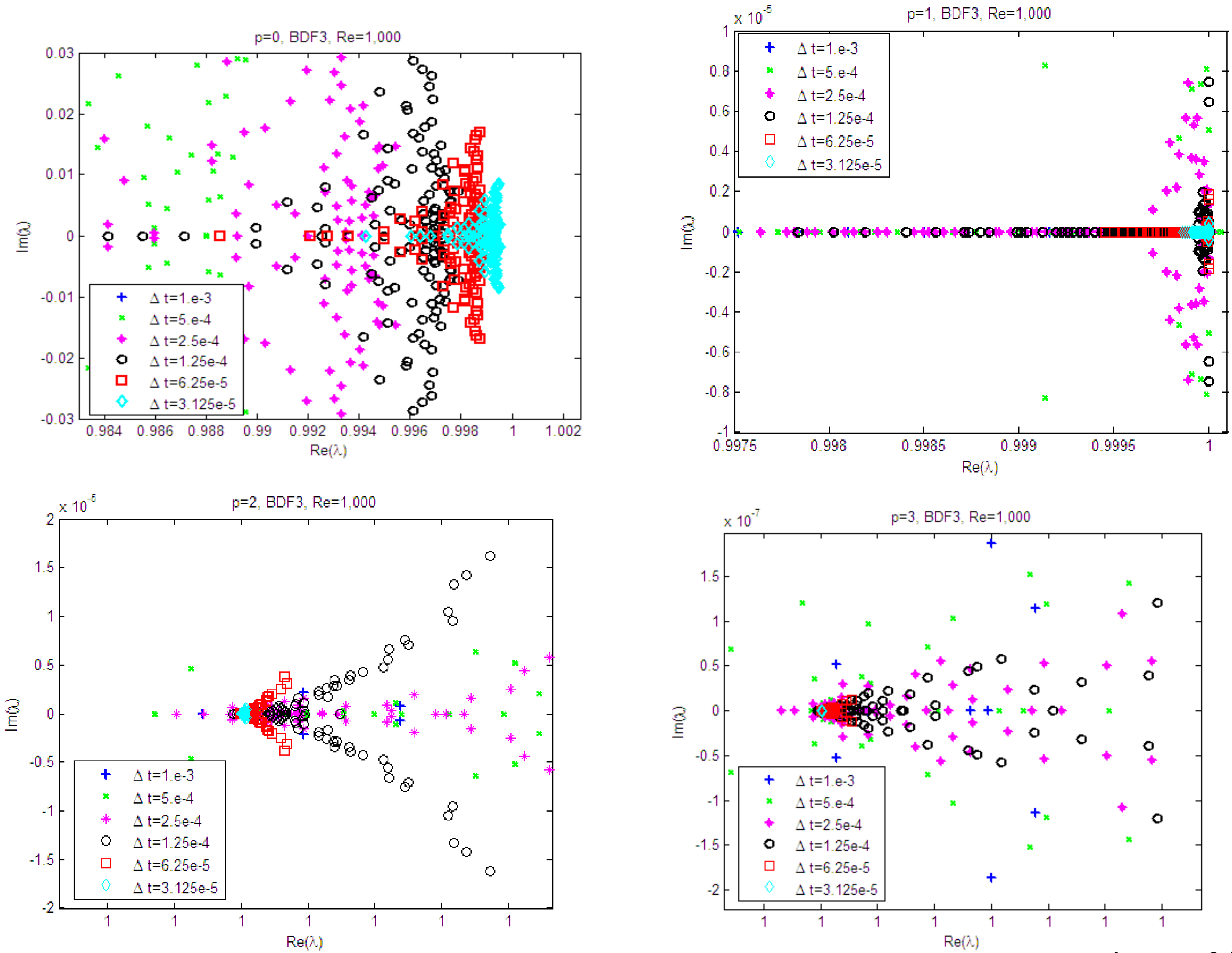
Eigenvalues of $(SQ_p)^{-1}\Sigma$

High Order Yosida as Preconditioner:



Eigenvalues of $(SQ_p)^{-1}\Sigma$

High Order Yosida as Preconditioner:



Eigenvalues of $(SQ_p)^{-1}\Sigma$

Outline:

- **Introduction**
  - Motivations
  - Basic settings
  - Incremental pressure schemes
- **Algebraic Splittings**
  - Inexact LU block factorizations
  - High Order Yosida Schemes
- **Adaptivity**
  - Local splitting error analysis
  - A posteriori error estimators
  - An application to blood flow problems
- **Scalability**
  - Strong scaling test
  - Weak scaling test
- **Conclusions**

Local Splitting error analysis:

Stokes System, incremental pressure approach*

Unsplit formulation

$$\begin{cases} \frac{\alpha_0}{\Delta t} M\mathbf{u}^k + \nu K \mathbf{u}^k + D^T p^k = \mathbf{f}_u^k + \frac{M}{\Delta t}\sum_{i=1}^{p} \alpha_i \mathbf{u}^{k-i} \\ D\mathbf{u}^k = f_p^k \end{cases}$$

-

Split formulation

$$\begin{cases} \frac{\alpha_0}{\Delta t} M\hat{\mathbf{u}}_{q,s}^k + \nu K \hat{\mathbf{u}}_{q,s}^k + D^T \hat{p}_{q,s}^k = \mathbf{f}_u^k + \frac{M}{\Delta t}\sum_{i=1}^{p} \alpha_i \hat{\mathbf{u}}_{q,s}^{k-i} \\ D\hat{\mathbf{u}}_{q,s}^k - (\Sigma - SQ_q)\delta\hat{p}_{q,s}^k = f_p^k \end{cases}$$

= Locality assumption $\hat{\mathbf{u}}_{q,s}^i = \mathbf{u}^i$ and $p_{q,s}^i = p^i$ for $i = k-1, \dots, k-p$

Local splitting error

$$\mathbf{e}^{k,*} = \mathbf{u}^k - \hat{\mathbf{u}}_{q,s}^k$$
$$e^{k,*} = p^k - \hat{p}_{q,s}^k$$

$$\begin{cases} \frac{\alpha_0}{\Delta t} M\mathbf{e}^{k,*} + \nu K \mathbf{e}^{k,*} + D^T e^{k,*} = 0 \\ \mathbf{D}\mathbf{e}^{k,*} - (\Sigma - SQ_q) e^{k,*} = -(\Sigma - SQ_q)\delta p^k \end{cases}$$

$$\begin{aligned} \|\mathbf{e}^{k,*}\|_0 &\leq C\Delta t^{q+s+2} \\ \|\mathbf{e}^{k,*}\|_1 &\leq C\Delta t^{q+s+3/2} \end{aligned}$$

and

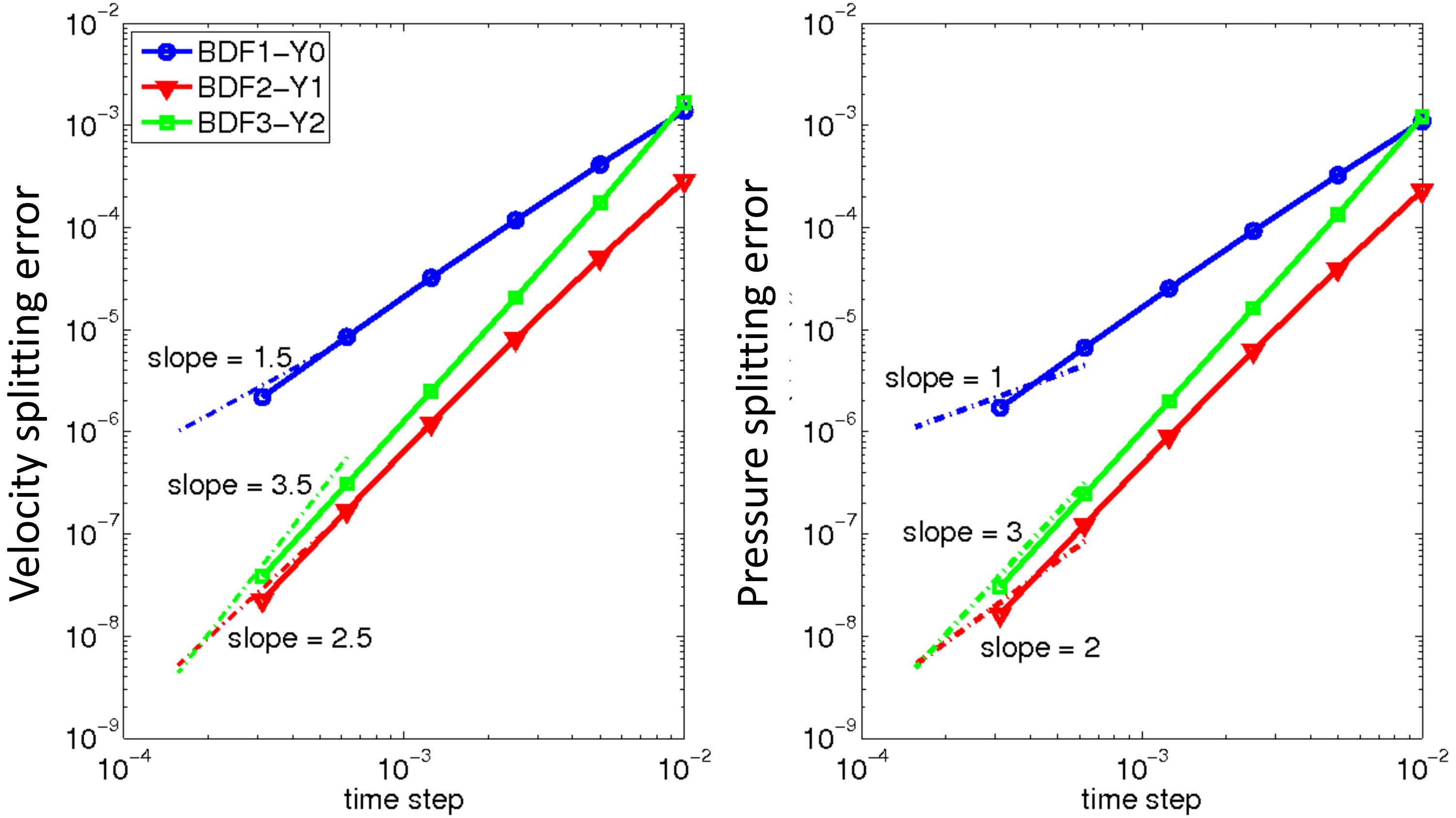
$$\|e^{k,*}\|_0 \leq C\Delta t^{q+s+1}$$

*The non incremental approach has been analyzed in P. Gervasio. *SIAM J. Numer. Anal., 2008*

A. Veneziani, U. Villa – *ALADINS: an ALgebraic splitting time ADaptive solver for the Incompressible Navier-Stokes equations*, J. Comput. Phys. (2013)

Local Splitting error analysis:

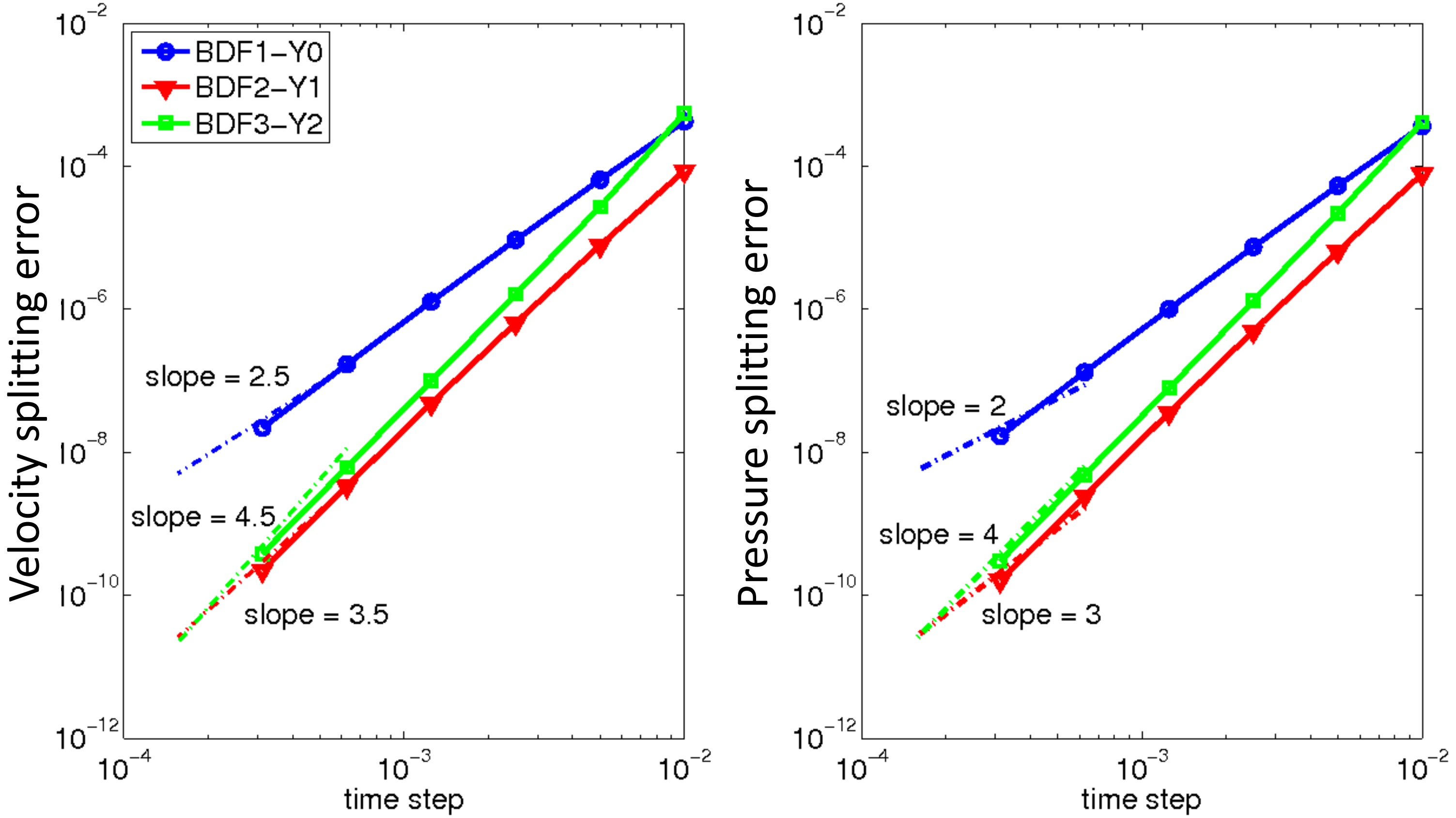
Non incremental method



Womersley analytical solution

Local Splitting error analysis:

Incremental method (s=1)



Womersley analytical solution

A posteriori error estimators:

1. **Yosida(q) – Yosida(q-1):**
   - Splitting based adaptivity (***conditionally stable***)
   - The last pressure increment z_q provides the error estimator.

$$\|p_{ex} - \hat{p}^{(s)}_{q-1}\| \leq \|\hat{p}^{(s)}_q - \hat{p}^{(s)}_{q-1}\| + \|p_{ex} - \hat{p}^{(s)}_q\| \longrightarrow z = \|\hat{p}^{(s)}_q - \hat{p}^{(s)}_{q-1}\| = \|z^{(s)}_q\| = \mathcal{O}(\Delta t^{q+s})$$

2. **Monolithic-Yosida(q-1) :**
   - Preconditioning based adaptivity (***unconditionally stable***)
   - The difference between the split and unsplit solution provides the error estimator

$$\|p_{ex} - \hat{p}^{(s)}_{q-1}\| \leq \|p^{(s)} - \hat{p}^{(s)}_{q-1}\| + \|p_{ex} - p^{(s)}\| \longrightarrow z = \|p^{(s)} - \hat{p}^{(s)}_{q-1}\| = \mathcal{O}(\Delta t^{q+s})$$

A posteriori error estimators:

1. **Yosida(q) – Yosida(q-1):**
   - Splitting based adaptivity (***conditionally stable***)
   - The last pressure increment z_q provides the error estimator.

$$\|p_{ex} - \hat{p}_{q-1}^{(s)}\| \leq \|\hat{p}_q^{(s)} - \hat{p}_{q-1}^{(s)}\| + \text{h.o.t.} \longrightarrow z = \|\hat{p}_q^{(s)} - \hat{p}_{q-1}^{(s)}\| = \|z_q^{(s)}\| = \mathcal{O}(\Delta t^{q+s})$$

2. **Monolithic-Yosida(q-1) :**
   - Preconditioning based adaptivity (***unconditionally stable***)
   - The difference between the split and unsplit solution provides the error estimator

$$\|p_{ex} - \hat{p}_{q-1}^{(s)}\| \leq \|p^{(s)} - \hat{p}_{q-1}^{(s)}\| + \text{h.o.t.} \longrightarrow z = \|p^{(s)} - \hat{p}_{q-1}^{(s)}\| = \mathcal{O}(\Delta t^{q+s})$$

Assume we require an accuracy τ for the absolute pressure error $\|p_{ex} - \hat{p}_{q-1}\| \leq \tau$

then we pick $\Delta t_{new} = \chi \Delta t_{old}$ where $\chi = \left(\frac{\tau \Delta t}{z}\right)^{\frac{1}{q+s-1}}$ and

A. If **χ < 1 reject** the time step

B. If **χ ≥ 1 accept** the time step

Monolithic – Yosida(q-1) Adaptivity:

At each time step we solve the coupled system in the velocity **u** and the pressure increment δp.

$$\begin{bmatrix} C & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ \delta p_s^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u - D^T \sigma_p^{n+1} \\ f_p \end{bmatrix}$$

As a **left preconditioner** we use the lower triangular part of the Yosida(q-1) splitting

$$P = \begin{bmatrix} C & 0 \\ D & SQ_{q-1} \end{bmatrix}$$

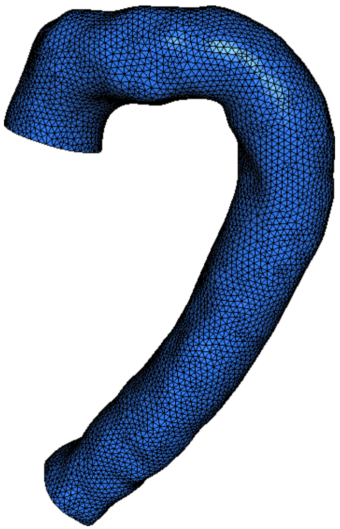
Let $\delta \hat{p}_{q-1,s}^{n+1}$ the first preconditioned residual.

→ a posteriori error estimator: $z = \|\delta p_s^{n+1} - \delta \hat{p}_{q-1,s}^{n+1}\| = \mathcal{O}(\Delta t^{q+s})$

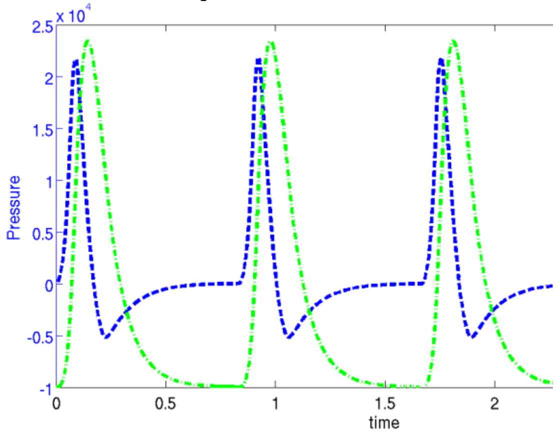
→ $\chi = \left(\frac{\tau \Delta t}{z}\right)^{\frac{1}{q+s-1}}$

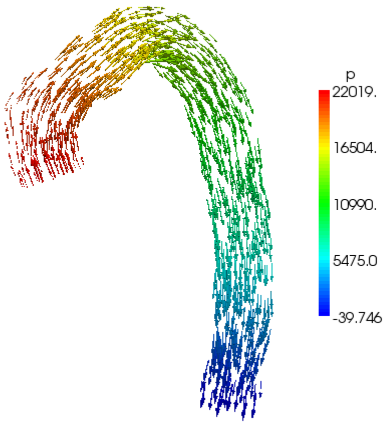
Note: the ***High Order Yosida*** Preconditioner ***SQ₁*** is equivalent to the ***Least Square Commutator*** preconditioner ***by Elman*** (SIAM J. Sci. Comput., 1999)

Blood flow application:

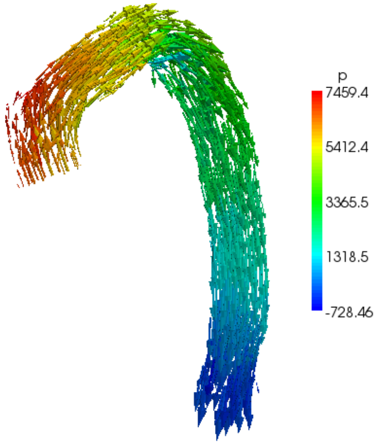


Inflow **pressure** and **flux**





Pressure peak



Flux peak

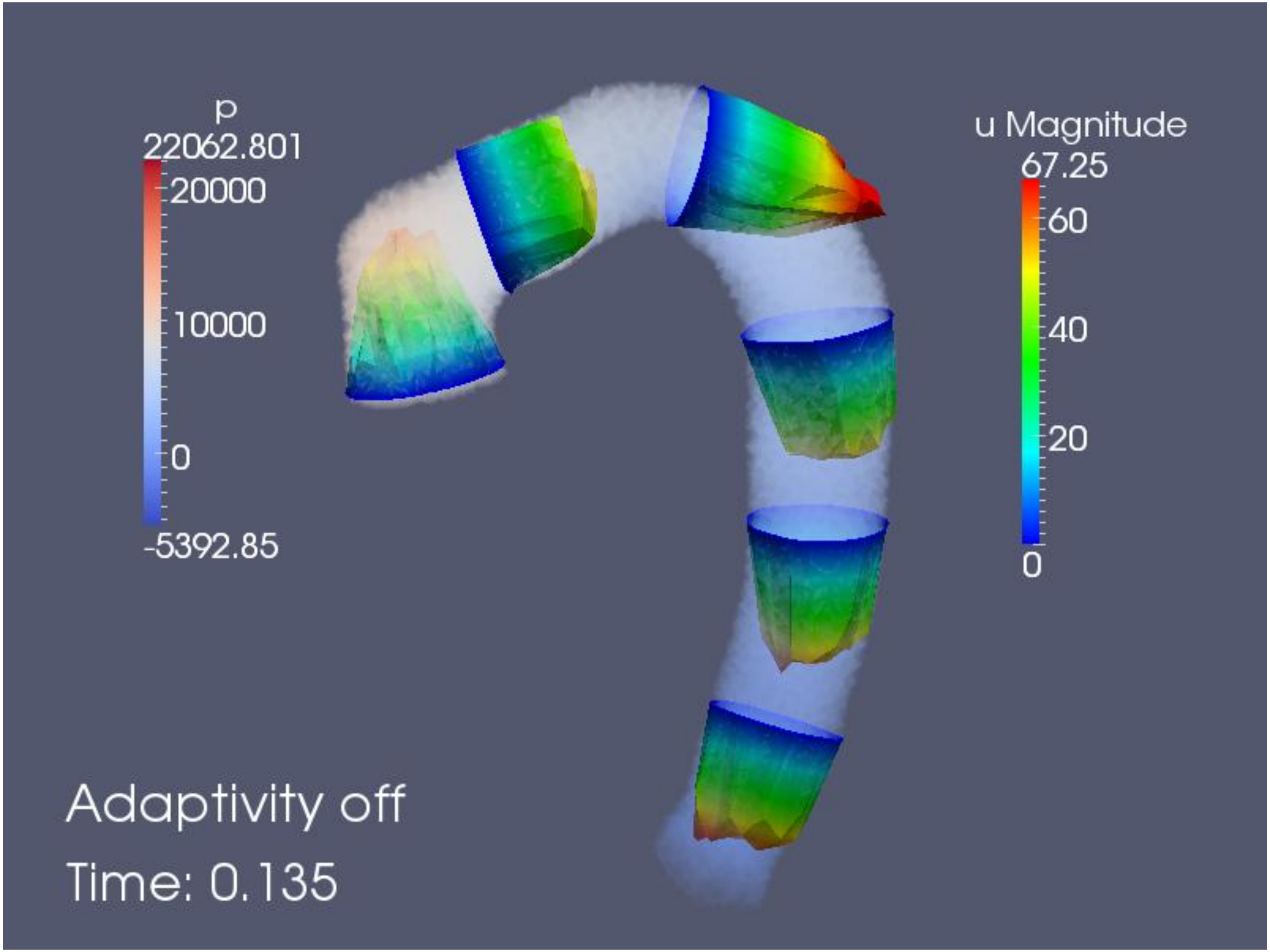
Adaptivity:
Monolithic – Yosida 1 error estimator
Second order error estimator

Discretization:
TIME: BDF2 with incremental pressure (s=1)
SPACE: Inf-sup compatible P1Bubble-P1 FE

Real geometry, physiological conditions

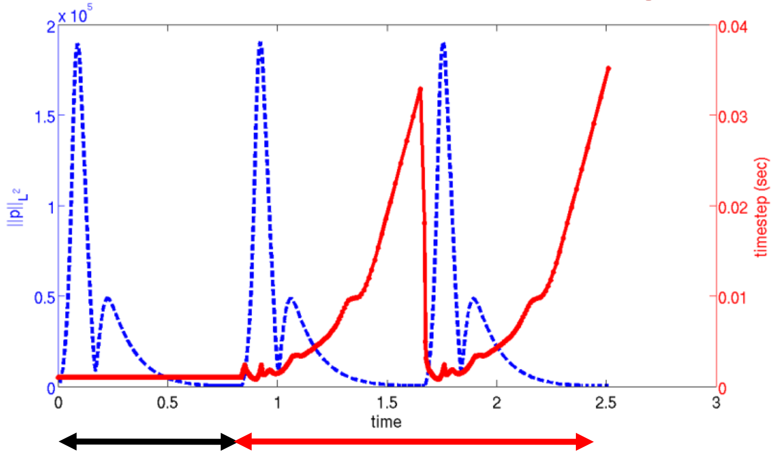
Reynolds	Womersley
300	21

Blood flow application:

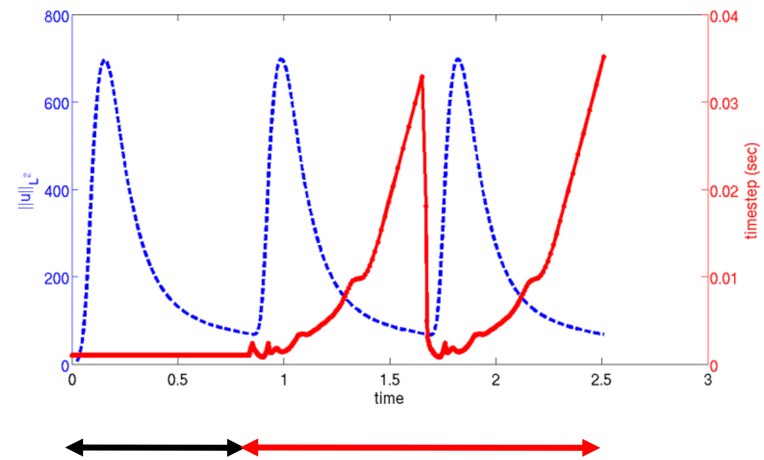


Blood flow application:

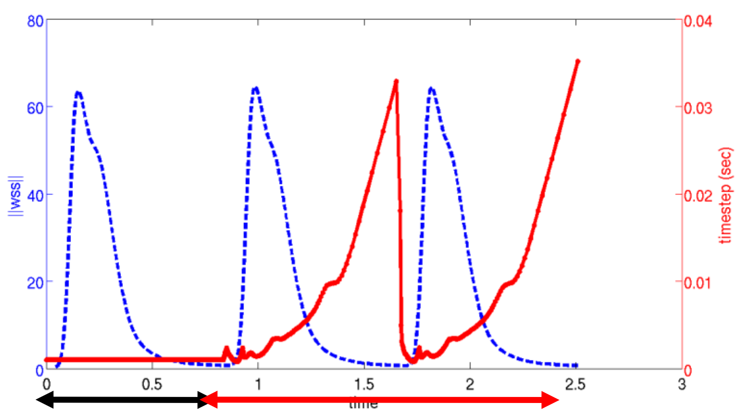
Pressure norm; Time Step



Velocity norm; Time Step



Wall shear stress; Time Step



	Steps per heart beat
Non adaptive	834
Adaptive	221
Speed-up	3.75

Outline:

- **Introduction**
  - Motivations
  - Basic settings
  - Incremental pressure schemes
- **Algebraic Splittings**
  - Inexact LU block factorizations
  - High Order Yosida Schemes
- **Adaptivity**
  - Local splitting error analysis
  - A posteriori error estimators
  - An application to blood flow problems
- **Scalability**
  - Strong scaling test
  - Weak scaling test
- **Conclusions**

Solution of the saddle-point system:

At each time step we solve the coupled system in the velocity **u** and the pressure p

$$\begin{bmatrix} C & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_p \end{bmatrix}$$

with preconditioned **GMRES iterations** (**Belos**)

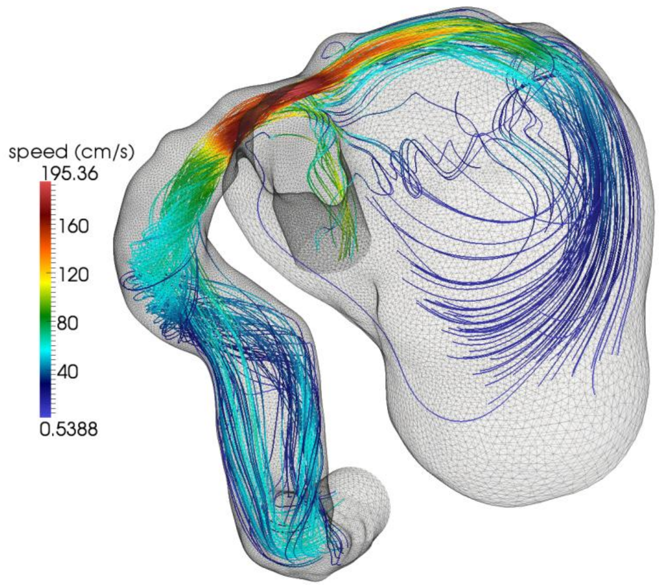
The block upper-triangular variant of the High Order Yosida Preconditioner

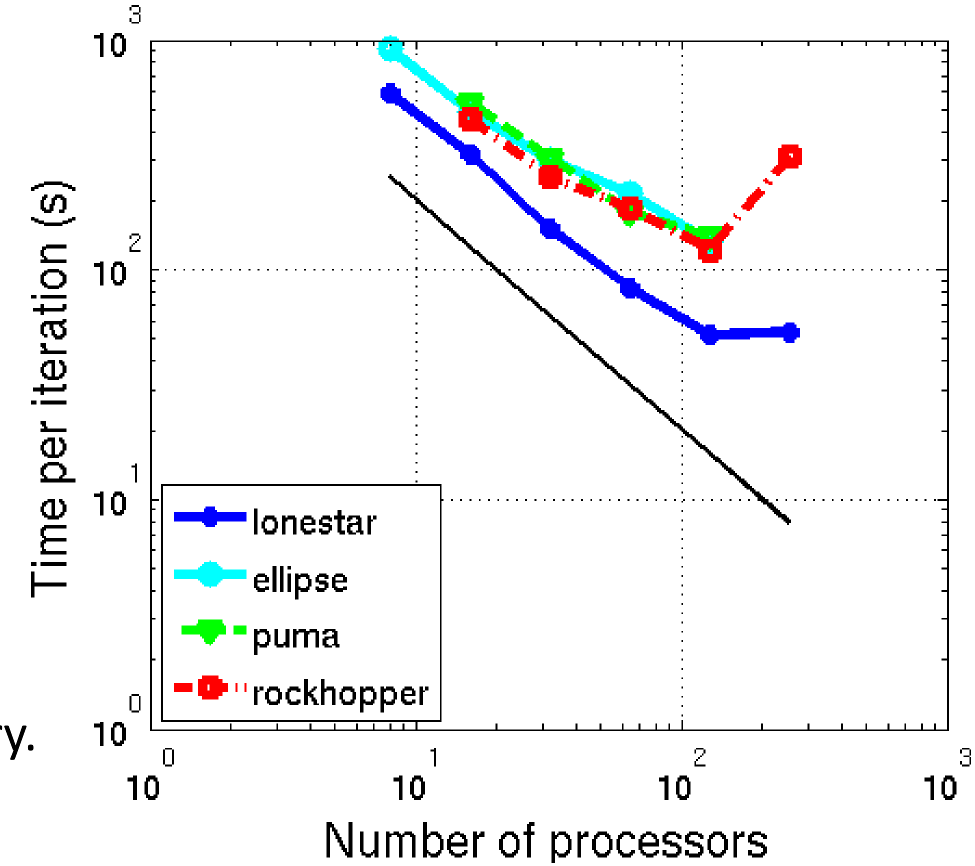
$$P = \begin{bmatrix} C & D^T \\ 0 & SQ_q \end{bmatrix}$$

is applied inexactly using the **AMG preconditioners** available in **ML** (Smoothed Aggregation and Symmetric Gauss-Seidel smoothers) for C and S

Energy minimization prolongation and **unsmoothed aggregation** are used to cope with the non-symmetry of C

Strong Scalability Test:



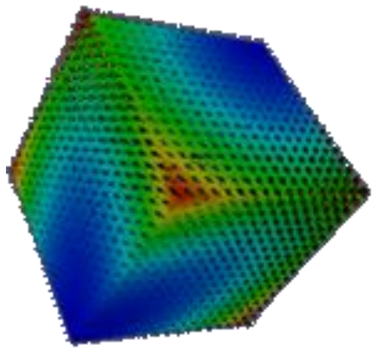


Simulation of blood flow in a giant aneurysm on the internal carotid artery. Benchmark proposed in the CFD Challenge Workshop at ASME 2012.

Space discretization: P1Bubble-P1 elements (≈ **3M unknowns**).
Time discretization: BDF2 (timestep 0.01s).

T. Passerini, J. Slawinski, U. V., A. Veneziani, V. Sunderam – *Experiences with a computational fluid dynamics code on clouds, grids, and on-premise resources.* (submitted to JPDC 2012)

Weak Scalability Test:



Ethier-Steinman Benchmark (1994)
Unstructured Tetrahedral Mesh (Netgen)
Low Reynolds number (approx 100)

Space discretization	
Taylor Hood P2-P1 FE	**Mini Element P1B-P1 FE**
Second order approx of velocity	First order approx of velocity
Denser FE matrices	Sparser matrices
No mass lumping	Accurate mass lumping

Convective term treatment	
Semi-implicit	**Explicit**
Non symmetric momentum matrix	Symmetric momentum matrix
Add grad-div stabilization	Block diagonal momentum matrix
Time-step proportional to the mesh diameter (*accuracy* and stability)	

Weak Scalability Test:

CONSORZIO INTERUNIVERSITARIO
CALCOLO ALTE PRESTAZIONI
TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

P2-P1 Finite Elements (Consistent Velocity Mass Matrix)

		Semi-implicit convective term				Explicit convective term			
n_p	N_{dof}	n_{it}	t_{solve}	t_{prec}	t_{tot}	n_{it}	t_{solve}	t_{prec}	t_{tot}
1	29K	114	3.30	0.28	3.90	70	1.08	0.12	1.36
2	57K	110	3.42	0.34	4.20	71	1.23	0.17	1.59
4	113K	106	4.40	0.39	5.29	67	1.44	0.20	1.85
8	216K	105	6.39	0.48	7.41	66	2.12	0.24	2.58
16	428K	103	6.97	0.55	8.13	65	2.46	0.30	3.02
32	860K	102	7.33	0.59	8.57	64	2.70	0.33	3.32
64	1.66M	99	8.43	0.65	9.77	62	3.33	0.40	4.02
128	3.33M	91	9.08	0.70	10.55	61	4.35	0.45	5.16
256*	6.71M	80	13.98	1.16	16.29	57	6.78	0.79	8.25

* 12 processes per node instead of 8. (One mpi process per core)

n_p: number of processes	N_{dof} number of unknowns (DOFs)	n_{it} average number of iterations
t_{solve}: average linear solver time	t_{prec} average preconditioner setup	t_{tot} average time per timestep

timings in seconds using *gettimeofday* function

Stopping criterion: relative norm of the residual less than 10^{-9}

Weak Scalability Test:

CONSORZIO INTERUNIVERSITARIO
CALCOLO ALTE PRESTAZIONI
TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

P2-P1 Finite Elements (Consistent Velocity Mass Matrix)

		Semi-implicit convective term				Explicit convective term			
n_p	N_{dof}	n_{it}	t_{solve}	t_{prec}	t_{tot}	n_{it}	t_{solve}	t_{prec}	t_{tot}
1	29K	114	3.30	0.28	3.90	70	1.08	0.12	1.36
2	57K	110	3.42	0.34	4.20	71	1.23	0.17	1.59
4	113K	106	4.40	0.39	5.29	67	1.44	0.20	1.85
8	216K	105	6.39	0.48	7.41	66	2.12	0.24	2.58
16	428K	103	6.97	0.55	8.13	65	2.46	0.30	3.02
32	860K	102	7.33	0.59	8.57	64	2.70	0.33	3.32
64	1.66M	99	8.43	0.65	9.77	62	3.33	0.40	4.02
128	3.33M	91	9.08	0.70	10.55	61	4.35	0.45	5.16
256*	6.71M	80	13.98	1.16	16.29	57	6.78	0.79	8.25

* 12 processes per node instead of 8. (One mpi process per core)

n_p: number of processes — N_{dof} number of unknowns (DOFs) — n_{it} average number of iterations

t_{solve}: average linear solver time — t_{prec} average preconditioner setup — t_{tot} average time per timestep

timings in seconds using *gettimeofday* function

Stopping criterion: relative norm of the residual less than 10^{-9}

Weak Scalability Test:

P1B-P1 Finite Elements (Lumped Velocity Mass Matrix)

		Semi-implicit convective term				Explicit convective term			
n_p	N_{dof}	n_{it}	t_{solve}	t_{prec}	t_{tot}	n_{it}	t_{solve}	t_{prec}	t_{tot}
1	23K	12	0.15	0.11	0.62	11	0.10	0.06	0.46
2	46K	12	0.18	0.17	0.77	10	0.10	0.10	0.55
4	93K	13	0.25	0.20	0.91	11	0.13	0.11	0.59
8	181K	15	0.42	0.25	1.16	12	0.23	0.13	0.70
16	363K	15	0.51	0.31	1.33	12	0.28	0.17	0.82
32	734K	15	0.59	0.37	1.49	12	0.35	0.20	0.91
64	1.43M	17	0.86	0.43	1.83	13	0.52	0.25	1.15
128	2.87M	18	1.20	0.49	2.30	14	0.79	0.31	1.53
256*	5.82M	21	2.10	0.78	3.70	16	1.28	0.56	2.48

* 12 processes per node instead of 8. (One mpi process per core)

n_p: number of processes — N_{dof} number of unknowns (DOFs) — n_{it} average number of iterations

t_{solve}: average linear solver time — t_{prec} average preconditioner setup — t_{tot} average time per timestep

timings in seconds using *gettimeofday* function

Stopping criterion: relative norm of the residual less than 10^{-9}

Weak Scalability Test:

P1B-P1 Finite Elements (Lumped Velocity Mass Matrix)

		Semi-implicit convective term				Explicit convective term			
n_p	N_{dof}	n_{it}	t_{solve}	t_{prec}	t_{tot}	n_{it}	t_{solve}	t_{prec}	t_{tot}
1	23K	12	0.15	0.11	0.62	11	0.10	0.06	0.46
2	46K	12	0.18	0.17	0.77	10	0.10	0.10	0.55
4	93K	13	0.25	0.20	0.91	11	0.13	0.11	0.59
8	181K	15	0.42	0.25	1.16	12	0.23	0.13	0.70
16	363K	15	0.51	0.31	1.33	12	0.28	0.17	0.82
32	734K	15	0.59	0.37	1.49	12	0.35	0.20	0.91
64	1.43M	17	0.86	0.43	1.83	13	0.52	0.25	1.15
128	2.87M	18	1.20	0.49	2.30	14	0.79	0.31	1.53
256*	5.82M	21	2.10	0.78	3.70	16	1.28	0.56	2.48

* 12 processes per node instead of 8. (One mpi process per core)

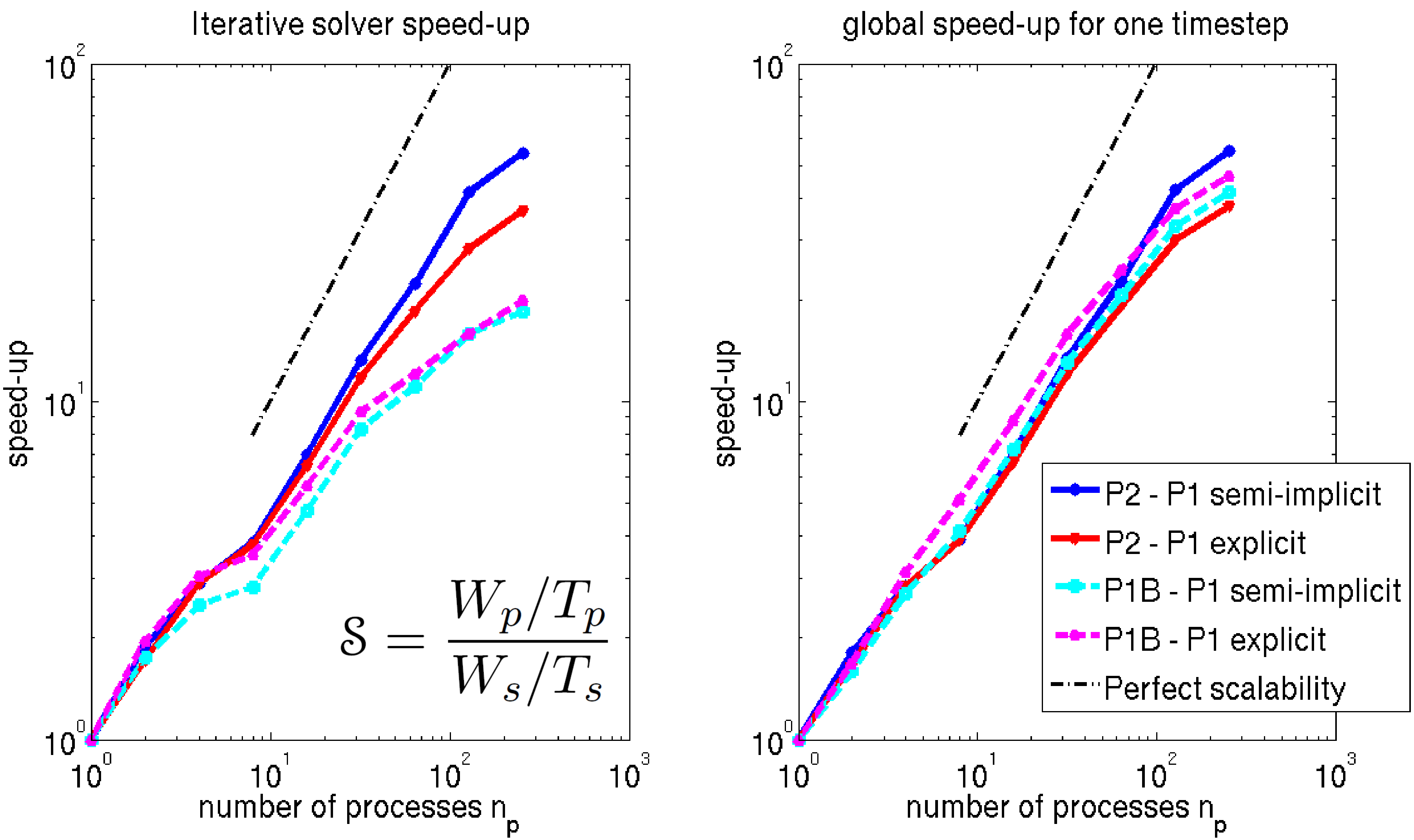
n_p: number of processes — N_{dof} number of unknowns (DOFs) — n_{it} average number of iterations

t_{solve}: average linear solver time — t_{prec} average preconditioner setup — t_{tot} average time per timestep

timings in seconds using *gettimeofday* function

Stopping criterion: relative norm of the residual less than 10^{-9}

Weak Scalability Speedup:



Outline:

- **Introduction**
  - Motivations
  - Basic settings
  - Incremental pressure schemes
- **Algebraic Splittings**
  - Inexact LU block factorizations
  - High Order Yosida Schemes
- **Adaptivity**
  - Local splitting error analysis
  - A posteriori error estimators
  - An application to blood flow problems
- **Scalability**
  - Strong scaling test
  - Weak scaling test
- **Conclusions**

Conclusions:

- **Incremental pressure methods** improve the accuracy of the splitting.
- High order Yosida splittings provide an effective **time adaptivity error estimator** as a by-product of the computation.
- Schur complement/Monolithic adaptive schemes allows selection of **larger time-step** due to their unconditionally stability.
- High order Yosida splittings are **optimal preconditioners** for the unsteady NSE.
- **(P)ALADINS** is a (**P**arallel) **AL**gebraic **AD**aptive **I**ncompressible **N**avier-**S**tokes Solver, based on algebraic splitting of velocity and pressure.
- **Good strong and weak scaling** properties in parallel when the local problem size is large enough using Trilinos (ML, Belos).