Efficient numerical and algorithmic realization of a pressure Poisson complement solver for the incompressible Navier-Stokes equations in FEAT3

Mirco Arndt

Institut für Angewandte Mathematik (LS3)
Fakultät für Mathematik
TU Dortmund
mirco.arndt@math.tu-dortmund.de

20. November 2018
Overview

1. Introduction
   - weak formulation and discretization
   - matrix formulation
   - structure of (almost) all solvers

2. PP formulation
   - PP formulation - pressure Poisson problem
   - construction of globally defined additive preconditioning operators

3. PP Algorithm
   - PP solver configurations

4. Benchmark results
Introduction

non-stationary incompressible Navier-Stokes equations

\[ u_t + u \cdot \nabla u - \nu \Delta u + \nabla p = f, \quad (-) \nabla \cdot u = 0 \quad \text{in } \Omega \times [0, T] \]

with initial and boundary conditions

- domain \( \Omega \subset \mathbb{R}^d, \ d = 2, 3 \)
- time limit \( T \)
- velocity \( u \)
- pressure \( p \)
- kinematic viscosity \( \nu \)
- rhs \( f \) (external source)
Introduction

weak formulation

\[ \int_{\Omega} \partial_t u \cdot v + \int_{\Omega} (u \cdot \nabla u) \cdot v - \nu \int_{\Omega} \Delta u \cdot v + \int_{\Omega} \nabla p \cdot v = \int_{\Omega} f \cdot v \]

partial derivative

\[ -\nu \int_{\Omega} \Delta u \cdot v = \nu \int_{\Omega} \nabla u : \nabla v - \nu \int_{\partial\Omega} (\nabla u \cdot n) \cdot v \]

\[ \int_{\Omega} \nabla p \cdot v = -\int_{\Omega} p \nabla \cdot v + \int_{\partial\Omega} pv \cdot n \]

discretization

\[ (\partial_t u_h, v_h) + (u_h \cdot \nabla u_h, v_h) + \nu(\nabla u_h, \nabla v_h) - (p_h, \nabla \cdot v_h) = (f, v_h) \]
\[ (-)(q_h, \nabla \cdot u_h) = 0 \]
Introduction

boundary conditions

Given area \( \Omega \subset \mathbb{R}^d \), \( d = 2, 3 \) with boundary \( \partial \Omega = \partial \Omega_{\text{in}} \cup \partial \Omega_0 \cup \partial \Omega_{\text{out}} \).

- On \( \partial \Omega_{D} := \partial \Omega_{\text{in}} \cup \partial \Omega_0 \) dirichlet boundary are given:
  \[ u|_{\partial \Omega_{\text{in}}} := u_{\text{in}} \quad \text{and} \quad u|_{\partial \Omega_0} := 0 \]

- On \( \partial \Omega_{\text{out}} \) „do nothing“
  \[ \Rightarrow -\nu \int_{\partial \Omega} (\nabla u \cdot n) \cdot v + \int_{\partial \Omega} pv \cdot n = 0 \]

„do nothing“
leave the solution and the test space free on that portion of the boundary
Introduction

matrix formulation

\[
\begin{bmatrix}
M & 0 \\
0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
\partial_t u_h \\
0 \\
\end{bmatrix}
+ \begin{bmatrix}
K(u_h) + \nu L & B \\
(+) \quad -B^T & 0 \\
\end{bmatrix}
\begin{bmatrix}
u_h \\
p_h \\
\end{bmatrix}
= \begin{bmatrix}
F \\
0 \\
\end{bmatrix}
\]

mass matrix \quad \mathbf{M}_{ij} := (v_h^i, v_h^j)
transport matrix \quad \mathbf{K}(u)_{ij} := (u_h \cdot \nabla v_h^j, v_h^i)
laplacian matrix \quad \mathbf{L}_{ij} := (\nabla v_h^j, \nabla v_h^i)
gradiant matrix \quad \mathbf{B}_{ij} := -(q_h^j, \nabla \cdot v_h^i) \quad \text{(divergence matrix } \mathbf{B}^T)\nright hand side \quad \mathbf{F}_i := (f, v_h^i)

time stepping techniques (theta-scheme)

\[
\begin{bmatrix}
S(u^l) & kB \\
B^T & 0 \\
\end{bmatrix}
\begin{bmatrix}
u^l \\
p^l \\
\end{bmatrix}
= \begin{bmatrix}
g \\
0 \\
\end{bmatrix}
\]

- step size \( k := \Delta t \)
- \( S(u^l) := \alpha M + k\theta(K(u^l) + \nu L) \)
- \( g := Mu^{l-1} - k(1 - \theta)(K(u^{l-1}) + \nu L) + k\theta F^l + k(1 - \theta)F^{l-1} \)
Structure of (almost) all solvers

\[
\begin{bmatrix}
S(u^l) & kB \\
B^T & 0
\end{bmatrix}
\begin{bmatrix}
u^l \\
p^l
\end{bmatrix} =
\begin{bmatrix}
g \\
0
\end{bmatrix} \iff
S(u)u + kBp = g \\
B^Tu = 0
\]

\textbf{Galerkin schemes}

\textit{local MPSC}

\textbf{Outer:} \(N\) nonlinear steps

\textbf{Inner:} \(L\) DPM \textbf{(Discrete Projection Method) steps} (Oseen)

\[L = 1\]

\begin{align*}
\text{CC} & \quad \text{Coupled solution by Coupled solver} \\
\text{CP} & \quad \text{Coupled solution by Projection solver}
\end{align*}

\textbf{Projection schemes}

\textit{global MPSC}

\textbf{Outer:} 1 DPM steps

\textbf{Inner:} \(N\) nonlinear steps (Burgers)

\[L > 1\]

\begin{align*}
\text{PP} & \quad \text{Projection solution by Projection solver}
\end{align*}

All versions of CC, CP, PP lead to the „same“ solutions.
Essential difference between the solvers

**key ideas of MPSC approaches**

*Re-interpretation of Navier-Stokes solvers (Chorin, VanKan, Uzawa, etc.) as „incomplete solvers“ for discrete saddle-point problems.*

<table>
<thead>
<tr>
<th>Galerkin schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>local MPSC</strong> <em>(Multilevel Pressure Schur Complement)</em></td>
</tr>
<tr>
<td>■ <strong>Fully coupled</strong> Newton-like solver as outer nonlinear procedure</td>
</tr>
<tr>
<td>■ Solve exactly on <strong>subsets/patches</strong> and perform an outer Block–Gauß-Seidel/Jacobi iteration as smoother</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Projection schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>global MPSC</strong> <em>(Multilevel Pressure Schur Complement)</em></td>
</tr>
<tr>
<td>■ Outer <strong>decoupling</strong> of velocity and pressure</td>
</tr>
<tr>
<td>■ Newton-like schemes for Momentum equations</td>
</tr>
<tr>
<td>■ Multigrid solver for all scalar subproblems</td>
</tr>
</tbody>
</table>
Essential difference between the solvers

**Galerkin schemes**

**CC** (Coupled solution by Coupled solver)
- „direct solvers“ for stationary (generalized) Navier-Stokes equations
- fully implicit character
  - ⇒ the **most accurate and robust** time stepping schemes
  - ⇒ only variants which allow a rigorous **a posteriori error control**
- **large time steps** to reach a desired accuracy
- **very expensive costs** for one time step

**CP** (Coupled solution by Projection solver)
- **cost can be diminished** by
  - weakening the threshold parameters
  - applying only a fixed small number of nonlinear steps
  - ⇒ **accuracy and robustness behaviour may be weakened**
Essential difference between the solvers

Projection schemes

**PP** (Projection solution by Projection solver)

- applied to nonstationary flows only
- rigorous error control in time is not clear
- "exact" treatment of the nonlinearity depending on the last pressure iterate ($\rightarrow$ smaller time steps)
- **smaller and much cheaper time steps** (compared to Galerkin schemes)
- resulting solutions satisfy the continuity equation, but the discrete momentum equation only approximately

for fully nonstationary flows with dominating convective term and on complex domains, this approach is a favoured one
PP formulation

- PP (Projection solution by Projection solver)
  global MPSC (Multilevel Pressure Schur Complement)
  1. a decoupling step for \( u \) and \( p \) as outer iteration
  2. compute a velocity field without taking into account incompressibility
  3. perform a pressure correction, which is a projection back to the subspace of divergence free vector fields

pressure Schur complement:

\[
S(u)u + kBp = g \\
B^T u = 0
\]

\[
0 = B^T S^{-1}(u)g - kB^T S^{-1}(u)Bp
\]

pressure Schur complement

A scalar equation that contains the pressure:

\[
\underbrace{B^T S^{-1}(u)Bp}_{=:\tilde{P}} = \frac{1}{k} \underbrace{B^T S^{-1}(u)g}_{=:f_p}
\]
PP formulation - perform only once per time step

**Equation for \( u \) (‘Burgers’)**

Solve for \( u^l \):

\[
S(u^l)u^l = g - kBp^{l-1}
\]

**Equation for \( p \)**

Pressure correction with a suitable preconditioner \( C \):

\[
p^l = p^{l-1} + C^{-1} \left( \frac{1}{k} BT S^{-1}(u^l)g - B^T S^{-1}(u^l)Bp \right)
\]

residual (pressure Schur complement)

\[
Cp^l = Cp^{l-1} + \frac{1}{k} BT \left( S^{-1}(u^l)g - kS^{-1}(u^l)Bp^l \right)
\]

definition of \( u^l \)

\[
= Cp^{l-1} + \frac{1}{k} BT u^l
\]

**Convergence**

\[
Cp = Cp + \frac{1}{k} BT u^l \quad \Rightarrow \quad B^T u = 0
\]
Construction of globally defined additive preconditioning operators

$$A := B^T S^{-1}(u) B$$

with

$$S(u) = \alpha M + k\theta(\nu L + K(u))$$

**additive approach:** construct „optimal“ operators for the limit cases

$$C^{-1} := \alpha_R A_R^{-1} + \alpha_D A_D^{-1} + \alpha_K A_K^{-1}$$

- $A_R$ is an „optimal“ (reactive) preconditioner for $B^T M^{-1} B$
  (divergence-free $L^2$-projection)
- $A_D$ is an „optimal“ (diffusive) preconditioner for $B^T L^{-1} B$
  (Stokes-equation)
- $A_K$ is an „optimal“ (convective) preconditioner for $B^T K^{-1}(u) B$
  (incompressible Euler equation)

„optimal“

- partial preconditioners were direct solvers with respect to the underlying subproblem
- resulting convergence behaviour is independent of
  - outer parameters
  - underlying mesh
The „reactive“ preconditioner for $B^T M^{-1} B$

- $M$ is already diagonal by construction
  - finite difference approach
  - nonconforming triangular finite elements
- Otherwise lumping

$$A_R := P := B^T M_l^{-1} B$$

- An (almost) exact solver / preconditioner $A_R$ for small time steps
  $$S(u) = \alpha M_l + k\theta(\nu L + K(u)) \quad \rightarrow \quad \alpha M_l \quad \text{for} \quad k \rightarrow 0$$
- A flexible treatment of pressure boundary conditions on discrete level
- Highly efficient multigrid solvers for applying $A_R^{-1}$
- Very compact matrices $A_R$ in independence of the spatial discretization
The "reactive" preconditioner - Pressure Poisson problem

Poisson problem

\[-\Delta q = rhs\]

matrix formulation

- \(rhs = -\nabla \cdot \nabla q = \nabla \cdot v\)
- \(v = -\nabla q\)

\[
\begin{bmatrix}
I & \nabla \\
\nabla \cdot & 0
\end{bmatrix}
\begin{bmatrix}
v \\
q
\end{bmatrix}
= \begin{bmatrix}
0 \\
rhs
\end{bmatrix}
\sim
\begin{bmatrix}
M_l & B \\
B^T & 0
\end{bmatrix}
\begin{bmatrix}
v \\
q
\end{bmatrix}
= \begin{bmatrix}
0 \\
f_p
\end{bmatrix}
\]

Pressure Poisson problem

solve for \(q\): \((P := B^T M_l^{-1} B)\)

\[Pq = f_p\]

\(-\Delta q = rhs\)

- \(P\) calculated only once in a preprocessing step
  (or if the spatial mesh has changed)
- \(P\) arises from a mixed formulation
  \(\Rightarrow\) even piecewise constant ansatz functions are allowed
The „diffusive“ preconditioner for $B^T L^{-1} B$

The inverse discrete Laplacian $L^{-1}$ and also $B^T L^{-1} B$ are full matrices!
(At least for all finite difference/element/volumen approaches.)

Idea:

$$\nabla \cdot \Delta^{-1} \nabla \sim I$$

In the finite element context:

$$B^T L^{-1} B \sim M_p$$

$A_D := M_p$

- All numerical tests show, that indeed $A_D := M_p$ is sufficient.
- Absolutely robust against all variations of parameters and the shape of the mesh in the pure Stokes case.
- Leads to an improved convergence rate in the pressure update!
The ,,convective“ preconditioner for $B^T K^{-1}(u) B$

The inverse transport matrix $K^{-1}(u)$ and $B^T K^{-1}(u) B$ are full matrices!

- **continuous construction**
  \[
  \nabla \cdot (U \cdot \nabla)^{-1} \nabla \sim (\tilde{U} \cdot \nabla)\
  \]

- **discrete construction**
  \[
  \begin{bmatrix}
  ILU(S) & B \\
  B^T & 0
  \end{bmatrix} \iff B^T ILU(S)^{-1} B \quad \text{instead of} \quad \begin{bmatrix}
  S & B \\
  B^T & 0
  \end{bmatrix} \iff B^T S^{-1} B
  \]

- **poor condition number**: $O(h^{-1}) - O(h^{-2})$
- **sensitive to mesh anisotropies**
- **complete solution process is almost so expensive as for the original system**
- **new techniques . . .**
PP Algorithm

Start with: \( u_0 := 0 \). Given: Iterate \( p^{l-1} \).
Perform:

1. (**Burgers**) **Solve:** for an intermediate velocity \( \tilde{u} \):
   \[
   S(\tilde{u})\tilde{u} = g - kBp^{l-1} \tag{VanKan}
   \]
   \[
   S(\tilde{u})\tilde{u} = g \tag{Chorin}
   \]

2. **Calculate:** the right hand side \( f_p \) for the pressure Poisson problem:
   \[
   f_p = \frac{1}{k}B^T\tilde{u} \quad \left( = \frac{1}{k}B^T S^{-1}[g - kBp^{l-1}] = \text{residual} \ (p^{l-1}) \right)
   \]

3. (**Pressure Poisson**) **Solve:** for \( q \):
   \[
   Pq = f_p \quad (P := B^T M^{-1}_l B) \tag{VanKan}
   \]

4. **Update:** new pressure \( p^l \):
   \[
   p^l = p^{l-1} + \alpha_R q + \alpha_D M^{-1}_{pl} f_p \tag{VanKan}
   \]
   \[
   p^l = \alpha_R q + \alpha_D M^{-1}_{pl} f_p \tag{Chorin}
   \]

5. **Update:** new velocity \( u^l \) to satisfy the incompressibility constraint:
   \[
   u^l = \tilde{u} - kM^{-1}_l Bq \tag{VanKan}
   \]
PP Algorithm - Burgers equation

**Burgers equation** with given Iterate \( p^{l-1} \) and \( u^{l-1} \)

\[
S(\tilde{u})\tilde{u} = g - kBp^{l-1} \quad \text{(VanKan)} \]

or

\[
S(\tilde{u})\tilde{u} = g \quad \text{(Chorin)}
\]

**fixed point iteration**

1. Calculate nonlinear residual \( d^n \):

\[
d^n = f - S(\tilde{u}^n)\tilde{u}^n
\]

2. Solve an auxiliary subproblem for \( y^n \):

\[
S(\tilde{u}^n)y^n = d^n
\]

3. Update \( \tilde{u} \) via the auxiliary solution \( y^n \):

\[
\tilde{u}^{n+1} = \tilde{u}^n + y^n
\]

**full fixed point iteration**

- set \( \tilde{u}^0 := u^{l-1} \)
- use \( N \) nonlinear steps

**extrapolate previous time step**

- linear extrapolation of solution in time:

\[
\tilde{u}^0 := 2u^{l-1} - u^{l-2}
\]

- one nonlinear step
### PP solver configurations

#### fixed point loop
- relative tolerance ($\sim 10^{-2}$)

#### velocity solver - solver a
- Richardson with Jacobi Smoother
- Richardson-Multigird with Jacobi-Smoother (or SOR)
  - Coarse-Grid Solver: Richardson-Multigird with Jacobi-Smoother (50 iterations)
- max. iteration (50)
- relative tolerance ($\sim 10^{-2}$)
- smooth steps (2)
- smooth damp (0.7)
  

#### pressure solver - solver s
- PCG with Jacobi Smoother
- PCG-Multigird with Jacobi-Smoother
  - Coarse-Grid Solver: UMFPACK (exact solver)
- max. iteration (100)
- absolute tolerance ($\sim 10^{-10}$)
- relative tolerance ($\sim 10^{-2}$)
- smooth steps (16, 32)
- smooth damp ($\sim 0.9$)

#### scaling parameters
- $\alpha \in [0, 1]$ (1)
- $\alpha_R \in [0, 1]$ (1)
- $\alpha_D \leq k\theta\nu$ ($k\theta\nu$)

---

**non newtonian fluid** (0.3)
DFG flow around cylinder benchmark 2D-1, laminar case

Figure 1: Velocity and pressure profile for $Re = 20$.

Drag: 5.57953523384
Lift: 0.01061894815
Pressure difference: 0.11752016697

post processing - pressure

- do nothing
- midpoint
- linear extrapolation

$$p_{\text{post}} = 0.5(p^l + p^{l-1})$$
$$p_{\text{post}} = p^l + 0.5(p^l - p^{l-1})$$
DFG flow around cylinder benchmark 2D-2 and 2D-3

Figure 2: Velocity and pressure profile for Re = 100.

Figure 3: Velocity and pressure profile for Re ∈ [0, 100].
Efficient numerical and algorithmic realization of a pressure Poisson complement solver for the incompressible Navier-Stokes equations in FEAT3 – Mirco Arndt –

DFG flow around cylinder benchmark 2D-3, fixed time interval

- Drag and lift plots for different levels and values of parameter $k$.
- Comparison with reference featflow data.
Convergence history

Burgers: ($\sim 10^{-3}$)

Pressure Poisson: ($\sim 0.1$)
References

S. Turek: „Efficient solvers for incompressible flow problems: An algorithmic approach in view of computational aspects“, 1999


M. Schäfer and S. Turek: „Benchmark Computations of Laminar Flow Around a Cylinder“, 1996