Recent advances with PETSc in FreeFEM

Pierre Jolivet — CNRS

December 18, 2019

11th FreeFEM days

Inputs from: T. Mathew, Y. Favennec (LTEN), M. A. Badri (CEA)
J. Moulin, O. Marquet, V. Mons (ONERA)
A. Suzuki (Cybermedia Center)
L. Audibert, H. Girardon (EDF)
J. Sierra, D. Fabre (IMFT)
F. Feppon (CMAP)
L. Hao (Kyoto University)
G. Sadaka (LMRS)
F. Hecht (LJLL)

Introduction

1. Introduction
2. Operator
3. Linear solvers
4. Eigensolvers
5. Nonlinear solvers
6. Applications

Introduction

PETSc

- Portable, Extensible Toolkit for Scientific Computation
- suite of data structures
  - `Vec` and `Mat`
  - `PC` and `KSP`
  - `SNES`, `TS`, and `Tao`

PETSc

- Portable, Extensible Toolkit for Scientific Computation
- suite of data structures
  - `Vec` and `Mat`
  - `PC` and `KSP`
  - `SNES`, `TS`, and `Tao`
- useful for compiling other libraries (MUMPS, *hypre*)

PETSc

- Portable, Extensible Toolkit for Scientific Computation
- suite of data structures
    - `Vec` and `Mat`
    - `PC` and `KSP`
    - `SNES`, `TS`, and `Tao`
- useful for compiling other libraries (MUMPS, *hypre*)
- `https://www.mcs.anl.gov/petsc`
- `https://gitlab.com/petsc/petsc`

Integration in FreeFEM

- available in the Docker image

Integration in FreeFEM

- available in the Docker image
- may be compiled from the sources
- currently install the following packages
  - MUMPS
  - SuperLU
  - SuiteSparse
  - *hypre*
  - Metis
  - SCOTCH
  - TetGen
  - SLEPc
  - HPDDM

Integration in FreeFEM

- available in the Docker image
- may be compiled from the sources
- currently install the following packages
  - MUMPS
  - SuperLU
  - SuiteSparse
  - *hypre*
  - Metis
  - SCOTCH
  - TetGen
  - SLEPc
  - HPDDM
- `load "PETSc"` or `load "PETSc-complex"`
- `load "SLEPc"` or `load "SLEPc-complex"`

Operator

Mat

- same typename as in PETSc
- list of operations in the `Mat` manual pages
- same operations as with a `matrix`
    - `A * x`
    - `A' * x`
    - `A^-1 * x`
    - `A'^-1 * x`

Mat

- same typename as in PETSc
- list of operations in the `Mat` manual pages
- same operations as with a `matrix`
  - `A * x`
  - `A' * x`
  - `A^-1 * x`
  - `A'^-1 * x`
- as easy as
```
matrix a = ...
Mat A(a)
```

Block matrix

- matrix nested with submatrices or vectors
- exactly same structure with a `Mat`
- useful for multiphysics, e.g.,
```
Mat St = [[A , B],
          [B', 0]]
```
- underyling PETSc MatNest

Distributed numbering

- all this works in parallel
- FreeFEM and PETSc have their own numberings
- `changeNumbering` to go from one to the other

Linear solvers

Attaching a KSP

- for a `matrix`, `set(solver = sparsesolver)`

Attaching a KSP

- for a `matrix`, `set(solver = sparsesolver)`
- for a `Mat`, `set(sparams = "-pc_type lu")`

Attaching a KSP

- for a `matrix`, `set(solver = sparsesolver)`
- for a `Mat`, `set(sparams = "-pc_type lu")`
- large choice of preconditioners and Krylov solvers
  - KSP manual pages
  - PC manual pages
- may be used for exact factorizations, DDM, MG...

Attaching a KSP

- for a `matrix`, `set(solver = sparsesolver)`
- for a `Mat`, `set(sparams = "-pc_type lu")`
- large choice of preconditioners and Krylov solvers
  - KSP manual pages
  - PC manual pages
- may be used for exact factorizations, DDM, MG...
- deprecated plugins in v4, available through PETSc

Fieldsplit

- separate preconditioners for different fields
- `fespace Vh(Pk,Pq,Pt)` ⇒ `Vh [u,v,w] = [1,2,3]`
- new KSP prefixed by `-fieldsplit_%d_`

Optional parameters

- prefixes customizable with a `string[int]`
- approximate Schur complement

Fieldsplit

- separate preconditioners for different fields
- `fespace Vh(Pk,Pk,Pq)` $\Rightarrow$ `Vh [u,v,p] = [1,1,2]`
- new KSP prefixed by `-fieldsplit_%d_`

Optional parameters

- prefixes customizable with a `string[int]`
- approximate Schur complement

Examples for Stokes equations [Knepley 2013]

$$A = \begin{bmatrix} C & B \\ B^T & 0 \end{bmatrix} \approx \mathrm{pc}_A = \begin{bmatrix} C^{-1} & 0 \\ 0 & I \end{bmatrix}$$

-ksp_type gmres -pc_fieldsplit_type additive
-fieldsplit_velocity_pc_type lu
-fieldsplit_pressure_pc_type jacobi

Fieldsplit

- separate preconditioners for different fields
- `fespace Vh(Pk,Pk,Pq)` ⇒ `Vh [u,v,p] = [1,1,2]`
- new KSP prefixed by `-fieldsplit_%d_`

Optional parameters

- prefixes customizable with a `string[int]`
- approximate Schur complement

Examples for Stokes equations [Knepley 2013]

$$A = \begin{bmatrix} C & B \\ B^T & 0 \end{bmatrix} \approx \mathrm{pc}_A = \begin{bmatrix} \mathrm{ksp}_C & 0 \\ 0 & I \end{bmatrix}$$

-ksp_type fgmres -pc_fieldsplit_type additive
-fieldsplit_velocity_pc_type gamg
-fieldsplit_pressure_pc_type jacobi

Fieldsplit

- separate preconditioners for different fields
- `fespace Vh(Pk,Pk,Pq)` $\Rightarrow$ `Vh [u,v,p] = [1,1,2]`
- new KSP prefixed by `-fieldsplit_%d_`

Optional parameters

- prefixes customizable with a `string[int]`
- approximate Schur complement

Examples for Stokes equations [Knepley 2013]

$$A = \begin{bmatrix} C & B \\ B^T & 0 \end{bmatrix} \approx \mathrm{pc}_A = \begin{bmatrix} \mathrm{ksp}_C & B \\ 0 & I \end{bmatrix}$$

-ksp_type fgmres -pc_fieldsplit_type multiplicative
-fieldsplit_velocity_pc_type gamg
-fieldsplit_pressure_pc_type jacobi

Fieldsplit

- separate preconditioners for different fields
- `fespace Vh(Pk,Pk,Pq)` $\Rightarrow$ `Vh [u,v,p] = [1,1,2]`
- new KSP prefixed by `-fieldsplit_%d_`

Optional parameters

- prefixes customizable with a `string[int]`
- approximate Schur complement

Examples for Stokes equations [Knepley 2013]

$$A = \begin{bmatrix} C & B \\ B^T & 0 \end{bmatrix} \approx \mathrm{pc}_A = \begin{bmatrix} \mathrm{ksp}_C & 0 \\ 0 & -\mathrm{ksp}_S \end{bmatrix}$$

-ksp_type fgmres -pc_fieldsplit_type schur
-fieldsplit_velocity_pc_type gamg
-fieldsplit_pressure_pc_type jacobi
-pc_fieldsplit_schur_factorization_type diag

Fieldsplit

- separate preconditioners for different fields
- `fespace Vh(Pk,Pk,Pq)` $\Rightarrow$ `Vh [u,v,p] = [1,1,2]`
- new `KSP` prefixed by `-fieldsplit_%d_`

Optional parameters

- prefixes customizable with a `string[int]`
- approximate Schur complement

Examples for Stokes equations [Knepley 2013]

$$A = \begin{bmatrix} C & B \\ B^T & 0 \end{bmatrix} \approx \mathrm{pc}_A = \begin{bmatrix} \mathrm{ksp}_C & 0 \\ B^T & \mathrm{ksp}_S \end{bmatrix}$$

-ksp_type fgmres -pc_fieldsplit_type schur
-fieldsplit_velocity_pc_type gamg
-fieldsplit_pressure_pc_type jacobi
-pc_fieldsplit_schur_factorization_type lower

Fieldsplit

- separate preconditioners for different fields
- `fespace Vh(Pk,Pk,Pq)` ⇒ `Vh [u,v,p] = [1,1,2]`
- new KSP prefixed by `-fieldsplit_%d_`

Optional parameters

- prefixes customizable with a `string[int]`
- approximate Schur complement

Examples for Stokes equations [Knepley 2013]

$$A = \begin{bmatrix} C & B \\ B^T & 0 \end{bmatrix} \approx \mathrm{pc}_A = \begin{bmatrix} C^{-1} & B \\ 0 & \mathrm{ksp}_S \end{bmatrix}$$

-ksp_type gmres -fieldsplit_pressure_ksp_max_its 1
-pc_fieldsplit_type schur -fieldsplit_velocity_pc_type lu
-fieldsplit_pressure_ksp_type richardson
-pc_fieldsplit_schur_factorization_type upper

New multigrid solvers

Geometric multigrid

- FreeFEM generates a hierarchy of grids
- corresponding prolongation operators
- `PCMG` manual pages
- `diffusion-mg-3d-PETSc.edp`

New multigrid solvers

Geometric multigrid

- FreeFEM generates a hierarchy of grids
- corresponding prolongation operators
- `PCMG` manual pages
- `diffusion-mg-3d-PETSc.edp`

For problems in $H(\mathrm{curl})$

- AMS from *hypre* [Hiptmair and Xu 2007]
- edge to node correspondance in `mat_edgeP1`
- `maxwell-3d-PETSc.edp`

Eigensolvers

EPSSolve

- `EigenValue` uses ARPACK
- not much versatility

EPSSolve

- `EigenValue` uses ARPACK
- not much versatility
- `EPSSolve(A)`, `EPSSolve(A,B)`
- `EPSSolve(func)`

EPSSolve

- `EigenValue` uses ARPACK
- not much versatility
- `EPSSolve(A)`, `EPSSolve(A,B)`
- `EPSSolve(func)`
- spectral transformations relies on a `KSP`
  - `EPSSolve(A, sparams = "...")`
  - `set(A, sparams = "...")`

    $\Rightarrow$ object composability
- `EPS` manual pages

Nonlinear solvers

SNESSolve

- easy-to-use interface to (quasi-)Newton methods
- solve $\mathcal{F}(u) = b$
- handle variational inequalities on u

SNESSolve

- easy-to-use interface to (quasi-)Newton methods
- solve $\mathcal{F}(u) = b$
- handle variational inequalities on u
- must provide two `func`s
    - to evaluate residuals stored as `K[int]`
    - to update a Jacobian stored as `Mat`
      $\Rightarrow$ object composability, `KSP` attached to the `Mat`
- `SNES` manual pages

Timesteppers and optimizers

`TSSolve` [Abhyankar et al. 2018]

- for DAE and ODE s.t. $F(t, u, \dot{u}) = G(u, t)$
- implicit, explicit, and IMEX methods
- `advection-TS-2d-PETSc.edp`
- `TS` manual pages

Timesteppers and optimizers

`TSSolve` [Abhyankar et al. 2018]

- for DAE and ODE s.t. $F(t, u, \dot{u}) = G(u, t)$
- implicit, explicit, and IMEX methods
- `advection-TS-2d-PETSc.edp`
- `TS` manual pages

`TaoSolve`

- similar interface to Ipopt
- `minimal-surface-Tao-2d-PETSc.edp`
- `Tao` manual pages

Applications

BEM

Contribution from P. Marchand (UOB)

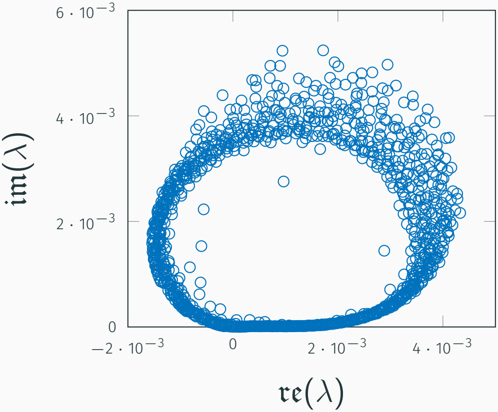
- preconditioning with a FE mass matrix
- 26 GHz on 256 MPI processes, 111k unknowns
- `helmholtz-3d-surf-PETSc-complex.edp`

Preconditioning	None	M^{-1}
Solve (seconds)	122	91
# of iterations	1,649	1,271

BEM

Contribution from P. Marchand (UOB)

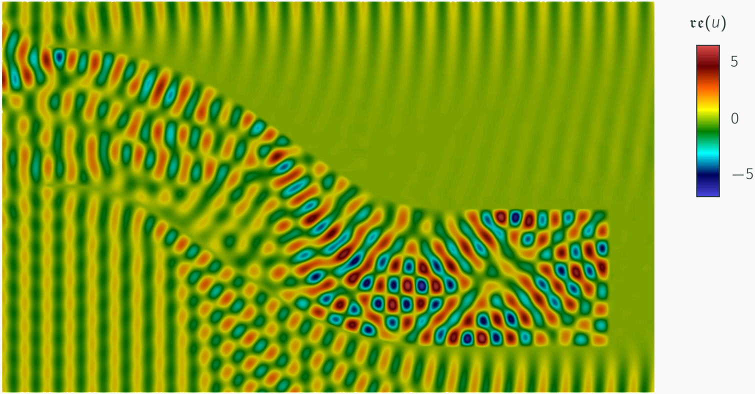
- preconditioning with a FE mass matrix
- 26 GHz on 256 MPI processes, 111k unknowns
- `helmholtz-3d-surf-PETSc-complex.edp`



BEM

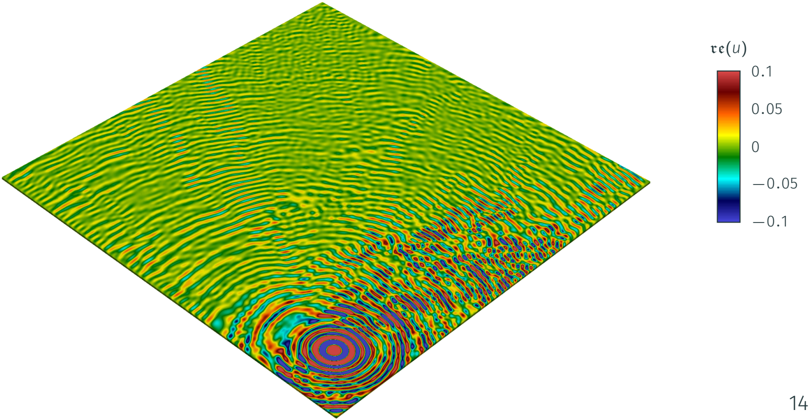
Contribution from P. Marchand (UOB)

- preconditioning with a FE mass matrix
- 26 GHz on 256 MPI processes, 111k unknowns
- `helmholtz-3d-surf-PETSc-complex.edp`



PCMG with optimized Schwarz methods

- overthrust model at 10 Hz on 512 MPI processes
- subdomains with $\approx$ 600k unknowns $\Rightarrow$ 344 seconds



Contribution from M. A. Badri (CEA) and Y. Favennec (LTEN)

Semi-discretized RTE

$$\forall m \in [\![1; N_d]\!], \left(\vec{s}_m \cdot \nabla + (\kappa + \sigma)\right) I_m = \sigma \sum_{n=1}^{N_d} \omega_n \varphi_{m,n} \cdot I_n + \kappa B$$

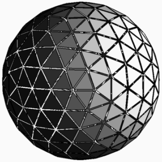
Radiative Transfer

Contribution from M. A. Badri (CEA) and Y. Favennec (LTEN)

Semi-discretized RTE

$$\forall m \in [\![1; N_d]\!], \left(\vec{s}_m \cdot \nabla + (\kappa + \sigma)\right) I_m = \sigma \sum_{n=1}^{N_d} \omega_n \varphi_{m,n} \cdot I_n + \kappa B$$

$N_d = 80$



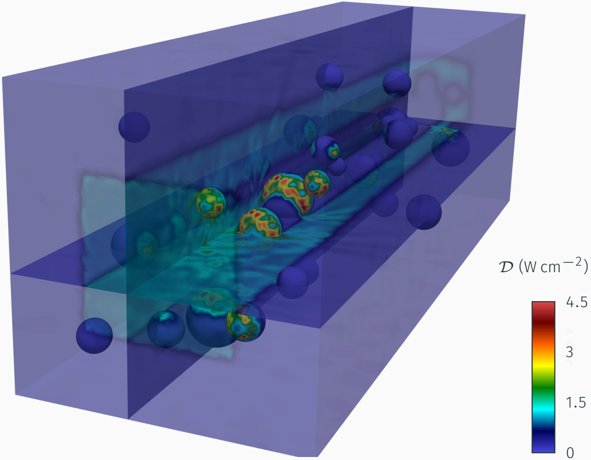
$N_d = 320$

$$\vec{s}_m = \begin{bmatrix} \sin\theta_m \cos\psi_m \\ \sin\theta_m \sin\psi_m \\ \cos\theta_m \end{bmatrix}$$

- κ (resp. σ) absorption (resp. scattering) coefficient
- φ phase scattering function
- B black body emissivity function

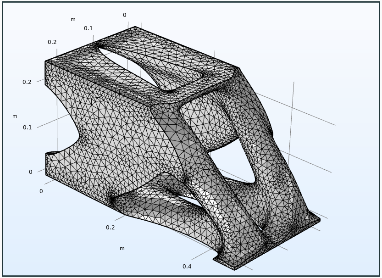
Radiative Transfer

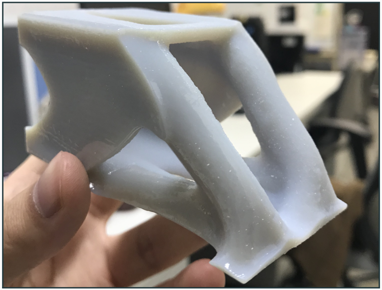
- matrix-free solver with BoomerAMG preconditioning
- 45×10^9 unknowns, 24k MPI processes
- 3.4×10^{15} nonzero entries



From design to manufacturing

Contribution from L. Hao, T. Yamada, S. Nishiwaki (Kyoto U.)





Augmented Lagrangian

grad–div stabilization [Benzi and Olshanskii 2006]

$$\begin{bmatrix} J_{uu} + \gamma J_{pu}^T W^{-1} J_{up} & J_{pu}^T \\ J_{pu} & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f_{u,\gamma} \\ f_p \end{bmatrix}$$

with $f_{u,\gamma} = f_u + J_{pu}^T W^{-1} f_p$

Augmented Lagrangian

grad–div stabilization [Benzi and Olshanskii 2006]

$$\begin{bmatrix} J_{uu} + \gamma J_{pu}^T W^{-1} J_{up} & J_{pu}^T \\ J_{pu} & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f_{u,\gamma} \\ f_p \end{bmatrix}$$

with $f_{u,\gamma} = f_u + J_{pu}^T W^{-1} f_p$

Augmented Lagrangian preconditioning

$$P_{\mathrm{AL}} = \begin{bmatrix} J_{uu,\gamma} & 0 \\ J_{pu} & S_p \end{bmatrix}$$

with $S_p^{-1} = -\nu M_p^{-1} - \gamma W^{-1}$ [Farrell et al. 2019]

Augmented Lagrangian

grad–div stabilization [Benzi and Olshanskii 2006]

$$\begin{bmatrix} J_{uu} + \gamma J_{pu}^T W^{-1} J_{up} & J_{pu}^T \\ J_{pu} & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f_{u,\gamma} \\ f_p \end{bmatrix}$$

with $f_{u,\gamma} = f_u + J_{pu}^T W^{-1} f_p$

Augmented Lagrangian preconditioning

$$P_{\mathrm{mAL}} = \begin{bmatrix} \begin{bmatrix} J_{u_x u_x,\gamma} & J_{u_x u_y,\gamma} \\ J_{u_y u_x,\gamma} & J_{u_y u_y,\gamma} \end{bmatrix} & 0 \\ J_{pu} & S_p \end{bmatrix}$$

with $S_p^{-1} = -\nu M_p^{-1} - \gamma W^{-1}$ [Farrell et al. 2019]

Augmented Lagrangian

grad–div stabilization [Benzi, Olshanskii, and Wang 2011]

$$\begin{bmatrix} J_{uu} + \gamma J_{pu}^T W^{-1} J_{up} & J_{pu}^T \\ J_{pu} & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f_{u,\gamma} \\ f_p \end{bmatrix}$$

with $f_{u,\gamma} = f_u + J_{pu}^T W^{-1} f_p$

Modified augmented Lagrangian preconditioning

$$P_{\mathrm{mAL}} = \begin{bmatrix} \begin{bmatrix} J_{u_x u_x,\gamma} & 0 \\ J_{u_y u_x,\gamma} & J_{u_y u_y,\gamma} \end{bmatrix} & 0 \\ J_{pu} & S_p \end{bmatrix}$$

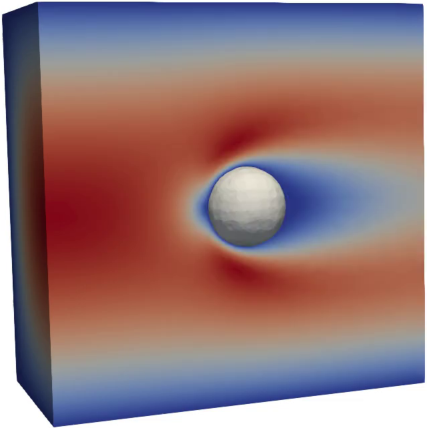
with $S_p^{-1} = -\nu M_p^{-1} - \gamma W^{-1}$ [Moulin et al. 2019]

$\Longrightarrow$ `https://github.com/prj-/moulin2019al`

Shape optimization

Contribution from F. Feppon, G. Allaire (CMAP), C. Dapogny (LJK)

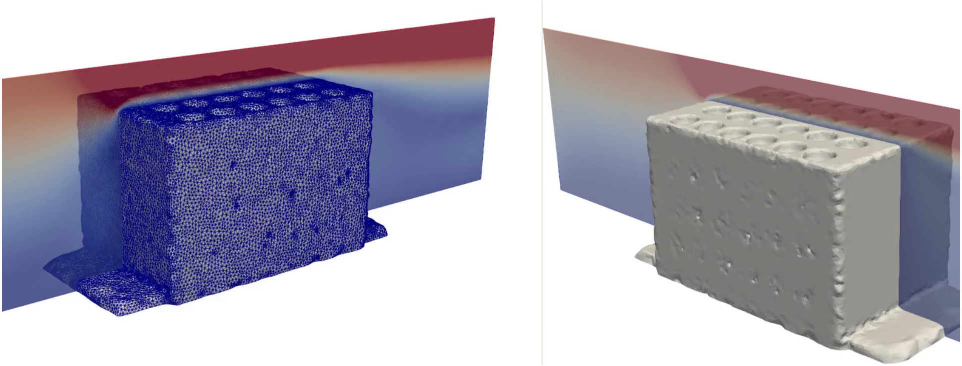
- maximizing the lift of a structure



Shape optimization

Contribution from F. Feppon, G. Allaire (CMAP), C. Dapogny (LJK)

- maximizing the lift of a structure
- minimizing the compliance of a structure



Contribution from J. Moulin and O. Marquet (ONERA)

$$J = \begin{bmatrix} J_{ss} & 0 & 0 & J_{s\lambda} \\ J_{es} & J_{ee} & 0 & 0 \\ 0 & J_{qe} & J_{qq} & J_{q\lambda} \\ 0 & 0 & J_{\lambda q} & 0 \end{bmatrix} \Longrightarrow$$

```
Mat J = [[Jss,0  ,0  ,Jsl],
         [Jes,Jee,0  ,  0],
         [0  ,Jqe,Jqq,Jql],
         [0  ,0  ,Jlq,0  ]];
```

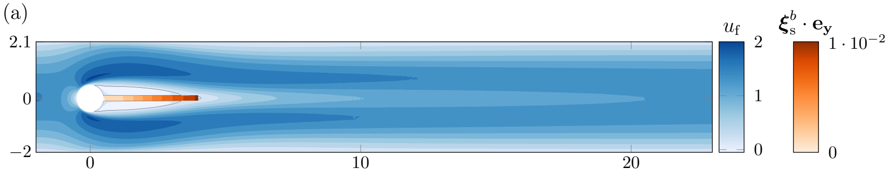
Contribution from J. Moulin and O. Marquet (ONERA)

$$J = \begin{bmatrix} J_{ss} & 0 & 0 & J_{s\lambda} \\ J_{es} & J_{ee} & 0 & 0 \\ 0 & J_{qe} & J_{qq} & J_{q\lambda} \\ 0 & 0 & J_{\lambda q} & 0 \end{bmatrix} \Longrightarrow$$

```
Mat J = [[Jss,0  ,0  ,Jsl],
         [Jes,Jee,0  ,  0],
         [0  ,Jqe,Jqq,Jql],
         [0  ,0  ,Jlq,0  ]];
```

- fieldsplit for the outer PC [Deparis et al. 2016]
- `set(sparams = "...")` for the diagonal blocks
- nested fieldsplit for `Jqq`

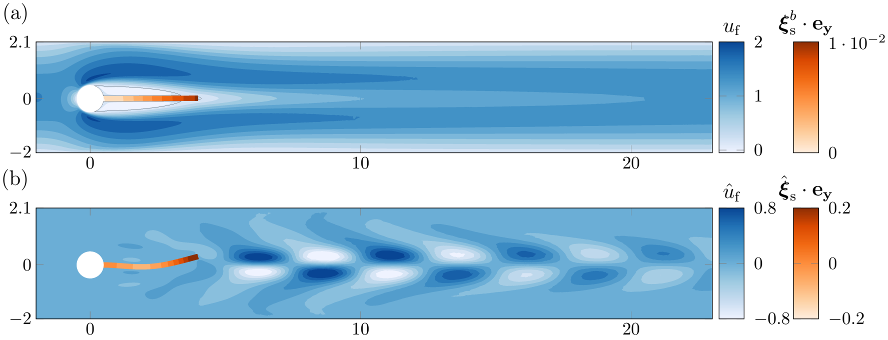
Contribution from J. Moulin and O. Marquet (ONERA)



Baseflow

[Turek and Hron 2006]

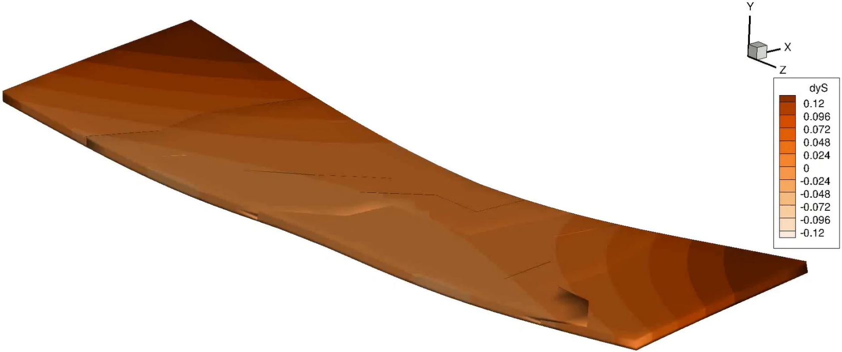
Contribution from J. Moulin and O. Marquet (ONERA)



Baseflow and leading eigenmode [Turek and Hron 2006]

Fluid–structure interaction

Contribution from J. Moulin and O. Marquet (ONERA)



Final words

- multiphysics solvers with PETSc and FreeFEM
  - high level of abstraction of FreeFEM
  - versatility of PETSc
- composability of solvers
- ongoing developments are user-driven
- `jolivet.perso.enseeiht.fr/FreeFem-tutorial`
- automatic differentiation?

Final words

- multiphysics solvers with PETSc and FreeFEM
  - high level of abstraction of FreeFEM
  - versatility of PETSc
- composability of solvers
- ongoing developments are user-driven
- `jolivet.perso.enseeiht.fr/FreeFem-tutorial`
- automatic differentiation?

Thank you!

References I

Abhyankar, Shrirang, Jed Brown, Emil M. Constantinescu, Debojyoti Ghosh, Barry F. Smith, and Hong Zhang (2018). *PETSc/TS: A Modern Scalable ODE/DAE Solver Library*. arXiv: `1806.01437 [math.NA]`.

Benzi, Michele and Maxim A. Olshanskii (2006). "An Augmented Lagrangian-Based Approach to the Oseen Problem". In: *SIAM Journal on Scientific Computing* 28.6, pp. 2095–2113.

Benzi, Michele, Maxim A. Olshanskii, and Zhen Wang (2011). "Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations". In: *International Journal for Numerical Methods in Fluids* 66.4, pp. 486–508.

Deparis, Simone, Davide Forti, Gwenol Grandperrin, and Alfio Quarteroni (2016). "FaCSI: A block parallel preconditioner for fluid–structure interaction in hemodynamics". In: *Journal of Computational Physics* 327, pp. 700–718.

Farrell, Patrick E., Lawrence Mitchell, and Florian Wechsung (2019). "An augmented Lagrangian preconditioner for the 3D stationary incompressible Navier–Stokes equations at high Reynolds number". In: *SIAM Journal on Scientific Computing* 41.5.

References II

Hiptmair, Ralf and Jinchao Xu (2007). "Nodal auxiliary space preconditioning in $H(\mathrm{curl})$ and $H(\mathrm{div})$ spaces". In: *SIAM Journal on Numerical Analysis* 45.6, pp. 2483–2509.

Knepley, Matthew (2013). *Nested and Hierarchical Solvers in PETSc*. URL: `https://www.caam.rice.edu/~mk51/presentations/SIAMCSE13.pdf`.

Moulin, Johann, Pierre Jolivet, and Olivier Marquet (2019). "Augmented Lagrangian Preconditioner for Large-Scale Hydrodynamic Stability Analysis". In: *Computer Methods in Applied Mechanics and Engineering* 351, pp. 718–743. URL: `https://github.com/prj-/moulin2019al`.

Turek, Stefan and Jaroslav Hron (2006). "Proposal for numerical benchmarking of fluid–structure interaction between an elastic object and laminar incompressible flow". In: *Fluid–Structure Interaction*. Springer, pp. 371–385.