Goal-oriented error estimates

V. Dolejší
P. Tichý, O. Bartoš, F. Roskovec, M. Svärd, M. Vlasák

First progress meeting

November 26, 2020

Papers to be reported

- V. Dolejší, P. Tichý, On efficient numerical solution of linear algebraic systems arising in goal-oriented error estimates, Journal of Scientific Computing 83 (5) (2020). Task 1.2
- O. Bartoš, V. Dolejší, F. Roskovec, Goal-oriented mesh adaptation method for nonlinear problems including algebraic errors, Comput. Math. Appl. (submitted). Task 1.2
- V. Dolejší, F. Roskovec, Goal-oriented anisotropic *hp*-adaptive discontinuous Galerkin method for the Euler equations, Commun. Appl. Math. Comput. (accepted). Tasks 1.3, 4.1
- V. Dolejší, M. Svärd, Numerical study of two models for viscous compressible fluid flows, Journal of Computational Physics (under review). Task 1.3(??)
- V. Dolejší, F. Roskovec, M. Vlasák, A posteriori error estimates for parabolic problems, SIAM J. Numer. Anal., (under review). Task 1.4

Papers to be reported

- V. Dolejší, P. Tichý, On efficient numerical solution of linear algebraic systems arising in goal-oriented error estimates, Journal of Scientific Computing 83 (5) (2020). Task 1.2
- O. Bartoš, V. Dolejší, F. Roskovec, Goal-oriented mesh adaptation method for nonlinear problems including algebraic errors, Comput. Math. Appl. (submitted). Task 1.2
- V. Dolejší, F. Roskovec, Goal-oriented anisotropic *hp*-adaptive discontinuous Galerkin method for the Euler equations, Commun. Appl. Math. Comput. (accepted). Tasks 1.3, 4.1
- V. Dolejší, M. Svärd, Numerical study of two models for viscous compressible fluid flows, Journal of Computational Physics (under review). Task 1.3(??)
- V. Dolejší, F. Roskovec, M. Vlasák, A posteriori error estimates for parabolic problems, SIAM J. Numer. Anal., (under review). Task 1.4

Papers to be reported

- V. Dolejší, P. Tichý, On efficient numerical solution of linear algebraic systems arising in goal-oriented error estimates, Journal of Scientific Computing 83 (5) (2020). Task 1.2
- O. Bartoš, V. Dolejší, F. Roskovec, Goal-oriented mesh adaptation method for nonlinear problems including algebraic errors, Comput. Math. Appl. (submitted). Task 1.2
- V. Dolejší, F. Roskovec, Goal-oriented anisotropic *hp*-adaptive discontinuous Galerkin method for the Euler equations, Commun. Appl. Math. Comput. (accepted). Tasks 1.3, 4.1
- V. Dolejší, M. Svärd, Numerical study of two models for viscous compressible fluid flows, Journal of Computational Physics (under review). Task 1.3(??)
- V. Dolejší, F. Roskovec, M. Vlasák, A posteriori error estimates for parabolic problems, SIAM J. Numer. Anal., (under review). Task 1.4

Papers to be reported

- V. Dolejší, P. Tichý, On efficient numerical solution of linear algebraic systems arising in goal-oriented error estimates, Journal of Scientific Computing 83 (5) (2020). Task 1.2
- O. Bartoš, V. Dolejší, F. Roskovec, Goal-oriented mesh adaptation method for nonlinear problems including algebraic errors, Comput. Math. Appl. (submitted). Task 1.2
- V. Dolejší, F. Roskovec, Goal-oriented anisotropic *hp*-adaptive discontinuous Galerkin method for the Euler equations, Commun. Appl. Math. Comput. (accepted). Tasks 1.3, 4.1
- V. Dolejší, M. Svärd, Numerical study of two models for viscous compressible fluid flows, Journal of Computational Physics (under review). Task 1.3(??)
- V. Dolejší, F. Roskovec, M. Vlasák, A posteriori error estimates for parabolic problems, SIAM J. Numer. Anal., (under review). Task 1.4

Papers to be reported

- V. Dolejší, P. Tichý, On efficient numerical solution of linear algebraic systems arising in goal-oriented error estimates, Journal of Scientific Computing 83 (5) (2020). Task 1.2
- O. Bartoš, V. Dolejší, F. Roskovec, Goal-oriented mesh adaptation method for nonlinear problems including algebraic errors, Comput. Math. Appl. (submitted). Task 1.2
- V. Dolejší, F. Roskovec, Goal-oriented anisotropic *hp*-adaptive discontinuous Galerkin method for the Euler equations, Commun. Appl. Math. Comput. (accepted). Tasks 1.3, 4.1
- V. Dolejší, M. Svärd, Numerical study of two models for viscous compressible fluid flows, Journal of Computational Physics (under review). Task 1.3(??)
- V. Dolejší, F. Roskovec, M. Vlasák, A posteriori error estimates for parabolic problems, SIAM J. Numer. Anal., (under review). Task 1.4

Papers to be reported

- V. Dolejší, P. Tichý, On efficient numerical solution of linear algebraic systems arising in goal-oriented error estimates, Journal of Scientific Computing 83 (5) (2020). Task 1.2
- O. Bartoš, V. Dolejší, F. Roskovec, Goal-oriented mesh adaptation method for nonlinear problems including algebraic errors, Comput. Math. Appl. (submitted). Task 1.2
- V. Dolejší, F. Roskovec, Goal-oriented anisotropic *hp*-adaptive discontinuous Galerkin method for the Euler equations, Commun. Appl. Math. Comput. (accepted). Tasks 1.3, 4.1
- V. Dolejší, M. Svärd, Numerical study of two models for viscous compressible fluid flows, Journal of Computational Physics (under review). Task 1.3(??)
- V. Dolejší, F. Roskovec, M. Vlasák, A posteriori error estimates for parabolic problems, SIAM J. Numer. Anal., (under review). Task 1.4

Setting

- linear PDE $a(u,\phi) = \ell(\phi)\ \forall \psi \in V$
- goal to estimate $J(u) - J(u_h^k)$
- adjoint problem $a(\psi, z) = J(\psi)\ \forall \psi \in V$
- algebraic systems $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, $\mathbb{A}^{\mathsf{T}} z = \boldsymbol{c}$,
- $J(u - u_h^k) \approx r_h(u_h^k)(z_h^+ - z_h^k) + r_h(u_h^k)(z_h^k) =: \eta_{\mathrm{S},k} + \eta_{\mathrm{A},k}$

Results

- solved by BiCG method which solves both problems simultaneously
- cheap estimate of $\eta_{\mathrm{A},k}$
- efficient adaptive iterative solver

Setting

- linear PDE $a(u, \phi) = \ell(\phi)\ \forall \psi \in V$
- goal to estimate $J(u) - J(u_h^k)$
- adjoint problem $a(\psi, z) = J(\psi)\ \forall \psi \in V$
- algebraic systems $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, $\mathbb{A}^{\mathsf{T}}\mathrm{z} = \boldsymbol{c}$,
- $J(u - u_h^k) \approx r_h(u_h^k)(z_h^+ - z_h^k) + r_h(u_h^k)(z_h^k) =: \eta_{\mathrm{S},k} + \eta_{\mathrm{A},k}$

Results

- solved by BiCG method which solves both problems simultaneously
- cheap estimate of $\eta_{\mathrm{A},k}$
- efficient adaptive iterative solver

Setting

- linear PDE $a(u, \phi) = \ell(\phi) \ \forall \psi \in V$
- goal to estimate $J(u) - J(u_h^k)$
- adjoint problem $a(\psi, z) = J(\psi) \ \forall \psi \in V$
- algebraic systems $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, $\mathbb{A}^{\mathsf{T}}\mathrm{z} = \boldsymbol{c}$,
- $J(u - u_h^k) \approx r_h(u_h^k)(z_h^+ - z_h^k) + r_h(u_h^k)(z_h^k) =: \eta_{\mathrm{S},k} + \eta_{\mathrm{A},k}$

Results

- solved by BiCG method which solves both problems simultaneously
- cheap estimate of $\eta_{\mathrm{A},k}$
- efficient adaptive iterative solver

Setting

- linear PDE $a(u, \phi) = \ell(\phi)\ \forall\psi \in V$
- goal to estimate $J(u) - J(u_h^k)$
- adjoint problem $a(\psi, z) = J(\psi)\ \forall\psi \in V$
- algebraic systems $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, $\mathbb{A}^\mathsf{T}\mathrm{z} = \boldsymbol{c}$,
- $J(u - u_h^k) \approx r_h(u_h^k)(z_h^+ - z_h^k) + r_h(u_h^k)(z_h^k) =: \eta_{\mathrm{S},k} + \eta_{\mathrm{A},k}$

Results

- solved by BiCG method which solves both problems simultaneously
- cheap estimate of $\eta_{\mathrm{A},k}$
- efficient adaptive iterative solver

Setting

- linear PDE $a(u,\phi) = \ell(\phi)\ \forall \psi \in V$
- goal to estimate $J(u) - J(u_h^k)$
- adjoint problem $a(\psi, z) = J(\psi)\ \forall \psi \in V$
- algebraic systems $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, $\mathbb{A}^{\mathsf{T}}\mathrm{z} = \boldsymbol{c}$,
- $J(u - u_h^k) \approx r_h(u_h^k)(z_h^+ - z_h^k) + r_h(u_h^k)(z_h^k) =: \eta_{\mathrm{S},k} + \eta_{\mathrm{A},k}$

Results

- solved by BiCG method which solves both problems simultaneously
- cheap estimate of $\eta_{\mathrm{A},k}$
- efficient adaptive iterative solver

Setting

- linear PDE $a(u,\phi) = \ell(\phi)\ \forall\psi \in V$
- goal to estimate $J(u) - J(u_h^k)$
- adjoint problem $a(\psi, z) = J(\psi)\ \forall\psi \in V$
- algebraic systems $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, $\mathbb{A}^{\mathsf{T}}z = \boldsymbol{c}$,
- $J(u - u_h^k) \approx r_h(u_h^k)(z_h^+ - z_h^k) + r_h(u_h^k)(z_h^k) =: \eta_{\mathrm{S},k} + \eta_{\mathrm{A},k}$

Results

- solved by BiCG method which solves both problems simultaneously
- cheap estimate of $\eta_{\mathrm{A},k}$
- efficient adaptive iterative solver

Setting

- linear PDE $a(u, \phi) = \ell(\phi)\ \forall \psi \in V$
- goal to estimate $J(u) - J(u_h^k)$
- adjoint problem $a(\psi, z) = J(\psi)\ \forall \psi \in V$
- algebraic systems $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, $\mathbb{A}^{\mathsf{T}}\mathrm{z} = \boldsymbol{c}$,
- $J(u - u_h^k) \approx r_h(u_h^k)(z_h^+ - z_h^k) + r_h(u_h^k)(z_h^k) =: \eta_{\mathrm{S},k} + \eta_{\mathrm{A},k}$

Results

- solved by BiCG method which solves both problems simultaneously
- cheap estimate of $\eta_{\mathrm{A},k}$
- efficient adaptive iterative solver

Setting

- linear PDE $a(u, \phi) = \ell(\phi)\ \forall \psi \in V$
- goal to estimate $J(u) - J(u_h^k)$
- adjoint problem $a(\psi, z) = J(\psi)\ \forall \psi \in V$
- algebraic systems $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, $\mathbb{A}^{\mathsf{T}}\mathrm{z} = \boldsymbol{c}$,
- $J(u - u_h^k) \approx r_h(u_h^k)(z_h^+ - z_h^k) + r_h(u_h^k)(z_h^k) =: \eta_{\mathrm{S},k} + \eta_{\mathrm{A},k}$

Results

- solved by BiCG method which solves both problems simultaneously
- cheap estimate of $\eta_{\mathrm{A},k}$
- efficient adaptive iterative solver

Setting

- linear PDE $a(u, \phi) = \ell(\phi)\ \forall\psi \in V$
- goal to estimate $J(u) - J(u_h^k)$
- adjoint problem $a(\psi, z) = J(\psi)\ \forall\psi \in V$
- algebraic systems $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$, $\mathbb{A}^\mathsf{T}\mathrm{z} = \boldsymbol{c}$,
- $J(u - u_h^k) \approx r_h(u_h^k)(z_h^+ - z_h^k) + r_h(u_h^k)(z_h^k) =: \eta_{\mathrm{S},k} + \eta_{\mathrm{A},k}$

Results

- solved by BiCG method which solves both problems simultaneously
- cheap estimate of $\eta_{\mathrm{A},k}$
- efficient adaptive iterative solver

Setting

- extension of previous method to nonlinear problems PDE
- $\Rightarrow$ iterative solver for nonlinear algebraic systems
- each nonlinear iteration we solve linearized problem (=adjoint problem)

Results

- adjoint problem based on a linearization (not differentiation)
- proofs primal and adjoint consistencies
- stopping criteria for nonlinear and linear solvers
- adaptive algorithm using anisotropic *hp*-mesh adaptation

Setting

- extension of previous method to nonlinear problems PDE
- $\Rightarrow$ iterative solver for nonlinear algebraic systems
- each nonlinear iteration we solve linearized problem (=adjoint problem)

Results

- adjoint problem based on a linearization (not differentiation)
- proofs primal and adjoint consistencies
- stopping criteria for nonlinear and linear solvers
- adaptive algorithm using anisotropic *hp*-mesh adaptation

Setting

- extension of previous method to nonlinear problems PDE
- ⇒ iterative solver for nonlinear algebraic systems
- each nonlinear iteration we solve linearized problem (=adjoint problem)

Results

- adjoint problem based on a linearization (not differentiation)
- proofs primal and adjoint consistencies
- stopping criteria for nonlinear and linear solvers
- adaptive algorithm using anisotropic *hp*-mesh adaptation

Setting

- extension of previous method to nonlinear problems PDE
- ⇒ iterative solver for nonlinear algebraic systems
- each nonlinear iteration we solve linearized problem (=adjoint problem)

Results

- adjoint problem based on a linearization (not differentiation)
- proofs primal and adjoint consistencies
- stopping criteria for nonlinear and linear solvers
- adaptive algorithm using anisotropic *hp*-mesh adaptation

Setting

- extension of previous method to nonlinear problems PDE
- ⇒ iterative solver for nonlinear algebraic systems
- each nonlinear iteration we solve linearized problem (=adjoint problem)

Results

- adjoint problem based on a linearization (not differentiation)
- proofs primal and adjoint consistencies
- stopping criteria for nonlinear and linear solvers
- adaptive algorithm using anisotropic *hp*-mesh adaptation

Setting

- extension of previous method to nonlinear problems PDE
- ⇒ iterative solver for nonlinear algebraic systems
- each nonlinear iteration we solve linearized problem (=adjoint problem)

Results

- adjoint problem based on a linearization (not differentiation)
- proofs primal and adjoint consistencies
- stopping criteria for nonlinear and linear solvers
- adaptive algorithm using anisotropic *hp*-mesh adaptation

O. Bartoš, V. Dolejší, F. Roskovec, Goal-oriented mesh adaptation method for nonlinear problems including algebraic errors, Computers and Mathematics with Applications (submitted). Task 1.2

Setting

- extension of previous method to nonlinear problems PDE
- $\Rightarrow$ iterative solver for nonlinear algebraic systems
- each nonlinear iteration we solve linearized problem (=adjoint problem)

Results

- adjoint problem based on a linearization (not differentiation)
- proofs primal and adjoint consistencies
- stopping criteria for nonlinear and linear solvers
- adaptive algorithm using anisotropic *hp*-mesh adaptation

Setting

- extension of previous method to nonlinear problems PDE
- $\Rightarrow$ iterative solver for nonlinear algebraic systems
- each nonlinear iteration we solve linearized problem (=adjoint problem)

Results

- adjoint problem based on a linearization (not differentiation)
- proofs primal and adjoint consistencies
- stopping criteria for nonlinear and linear solvers
- adaptive algorithm using anisotropic *hp*-mesh adaptation

Setting

- extension of previous method to nonlinear problems PDE
- $\Rightarrow$ iterative solver for nonlinear algebraic systems
- each nonlinear iteration we solve linearized problem (=adjoint problem)

Results

- adjoint problem based on a linearization (not differentiation)
- proofs primal and adjoint consistencies
- stopping criteria for nonlinear and linear solvers
- adaptive algorithm using anisotropic *hp*-mesh adaptation

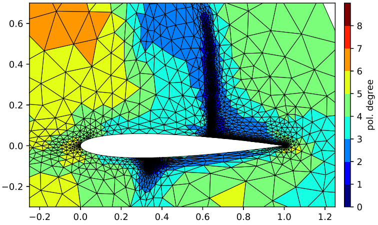
Subject

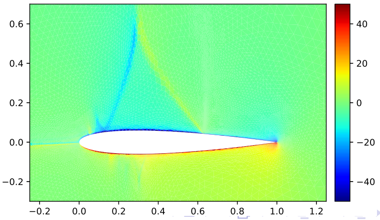
- extension to the Euler equations
- adjoint problem based on linearization

Results

- proofs primal and adjoint consistencies
- adaptive algorithm using anisotropic *hp*-mesh adaptation

inviscid transonic flow around NACA0012, $J =$ lift coefficient





V. Dolejší, F. Roskovec, Goal-oriented anisotropic *hp*-adaptive discontinuous Galerkin method for the Euler equations, Commun. Appl. Math. Comput. (accepted) Task 1.3, 4.1

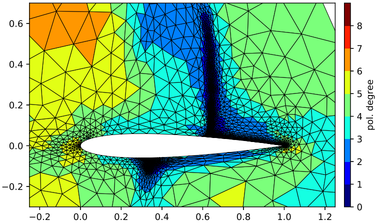
Subject

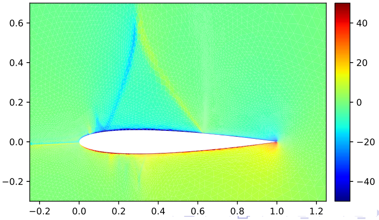
- extension to the Euler equations
- adjoint problem based on linearization

Results

- proofs primal and adjoint consistencies
- adaptive algorithm using anisotropic *hp*-mesh adaptation

inviscid transonic flow around NACA0012, J = lift coefficient





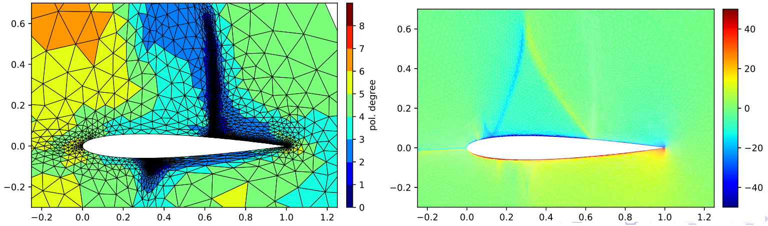
Subject

- extension to the Euler equations
- adjoint problem based on linearization

Results

- proofs primal and adjoint consistencies
- adaptive algorithm using anisotropic *hp*-mesh adaptation

inviscid transonic flow around NACA0012, J = lift coefficient



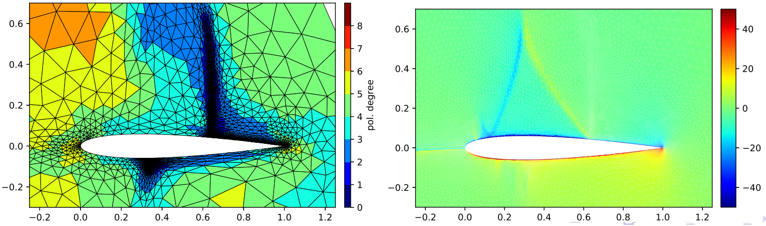
Subject

- extension to the Euler equations
- adjoint problem based on linearization

Results

- proofs primal and adjoint consistencies
- adaptive algorithm using anisotropic *hp*-mesh adaptation

inviscid transonic flow around NACA0012, J = lift coefficient



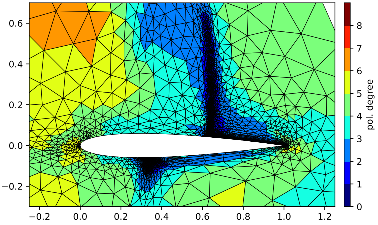
Subject

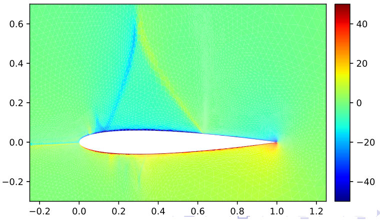
- extension to the Euler equations
- adjoint problem based on linearization

Results

- proofs primal and adjoint consistencies
- adaptive algorithm using anisotropic *hp*-mesh adaptation

inviscid transonic flow around NACA0012, $J =$ lift coefficient





V. Dolejší, M. Svärd, Numerical study of two models for viscous compressible fluid flows, Journal of Computational Physics (under review). Task 1.3(??)

Navier-Stokes equations

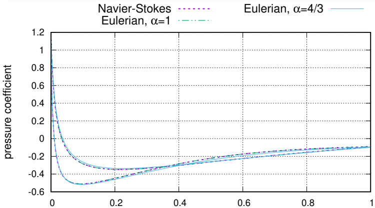
$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \boldsymbol{f}_i(\boldsymbol{w}) = \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w})$$

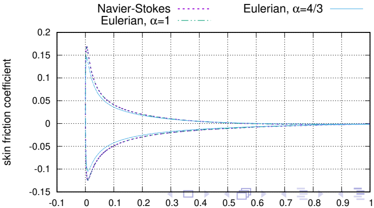
$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \left(0,\ \tau_{i1}^{\mathrm{V}},\ \tau_{i2}^{\mathrm{V}},\ \tau_{i1}^{\mathrm{V}} v_1 + \tau_{i2}^{\mathrm{V}} v_2 + k \frac{\partial \theta}{\partial x_i}\right)^{\mathrm{T}}, \quad i = 1, 2$$

Eulerian model proposed by Magnus Svärd

$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \nu(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_i}, \qquad \nu(\boldsymbol{w}) = \alpha \frac{\mu}{\rho}, \quad \alpha \in [1, 4/3],\ i = 1, 2$$

quantitative comparison: difference between models about 1%





Navier-Stokes equations

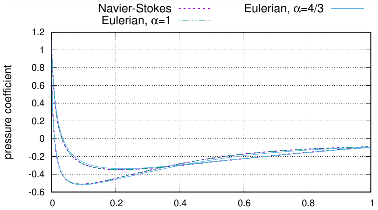
$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \boldsymbol{f}_i(\boldsymbol{w}) = \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w})$$

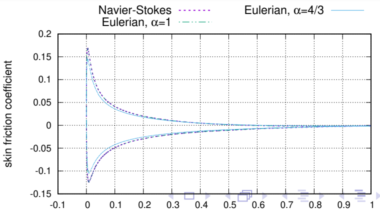
$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \left(0,\ \tau_{i1}^{\mathrm{V}},\ \tau_{i2}^{\mathrm{V}},\ \tau_{i1}^{\mathrm{V}} v_1 + \tau_{i2}^{\mathrm{V}} v_2 + k \frac{\partial \theta}{\partial x_i}\right)^{\mathrm{T}},\ i = 1, 2$$

Eulerian model proposed by Magnus Svärd

$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \nu(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_i}, \quad \nu(\boldsymbol{w}) = \alpha \frac{\mu}{\rho}, \quad \alpha \in [1, 4/3],\ i = 1, 2$$

quantitative comparison: difference between models about 1%





Navier-Stokes equations

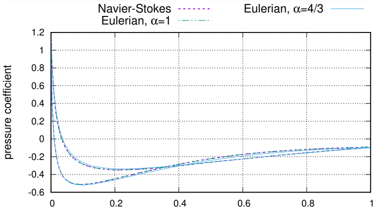
$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \boldsymbol{f}_i(\boldsymbol{w}) = \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w})$$

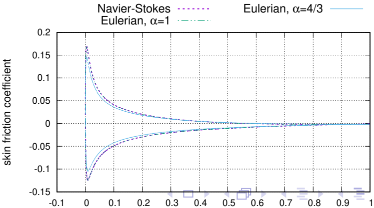
$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \left(0,\ \tau_{i1}^{\mathrm{V}},\ \tau_{i2}^{\mathrm{V}},\ \tau_{i1}^{\mathrm{V}} v_1 + \tau_{i2}^{\mathrm{V}} v_2 + k\frac{\partial \theta}{\partial x_i}\right)^{\mathrm{T}},\ i = 1, 2$$

Eulerian model proposed by Magnus Svärd

$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \nu(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_i}, \quad \nu(\boldsymbol{w}) = \alpha \frac{\mu}{\rho}, \quad \alpha \in [1, 4/3],\ i = 1, 2$$

quantitative comparison: difference between models about 1%





Navier-Stokes equations

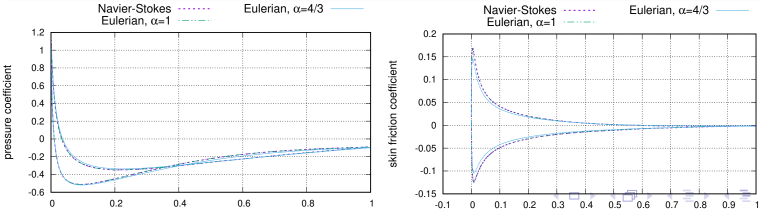
$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \boldsymbol{f}_i(\boldsymbol{w}) = \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w})$$

$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \left(0,\ \tau_{i1}^{\mathrm{V}},\ \tau_{i2}^{\mathrm{V}},\ \tau_{i1}^{\mathrm{V}} v_1 + \tau_{i2}^{\mathrm{V}} v_2 + k \frac{\partial \theta}{\partial x_i}\right)^{\mathrm{T}}, \quad i = 1, 2$$

Eulerian model proposed by Magnus Svärd

$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \nu(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_i}, \quad \nu(\boldsymbol{w}) = \alpha \frac{\mu}{\rho}, \quad \alpha \in [1, 4/3],\ i = 1, 2$$

quantitative comparison: difference between models about 1%



Navier-Stokes equations

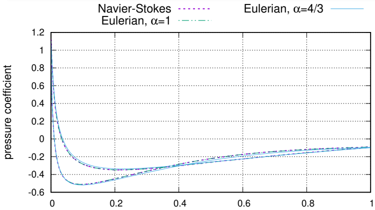
$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \boldsymbol{f}_i(\boldsymbol{w}) = \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w})$$

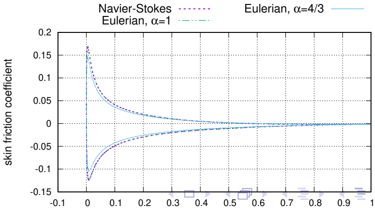
$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \left(0,\ \tau_{i1}^{\mathrm{V}},\ \tau_{i2}^{\mathrm{V}},\ \tau_{i1}^{\mathrm{V}} v_1 + \tau_{i2}^{\mathrm{V}} v_2 + k \frac{\partial \theta}{\partial x_i}\right)^{\mathrm{T}}, \quad i = 1, 2$$

Eulerian model proposed by Magnus Svärd

$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \nu(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_i}, \qquad \nu(\boldsymbol{w}) = \alpha \frac{\mu}{\rho}, \quad \alpha \in [1, 4/3],\ i = 1, 2$$

quantitative comparison: difference between models about 1%





Navier-Stokes equations

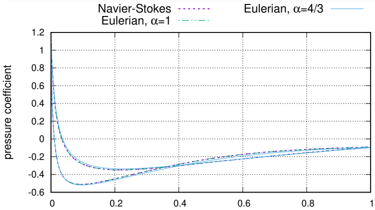
$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \boldsymbol{f}_i(\boldsymbol{w}) = \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w})$$

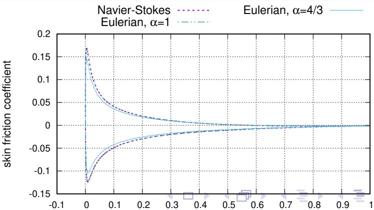
$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \left(0,\ \tau_{i1}^{\mathrm{V}},\ \tau_{i2}^{\mathrm{V}},\ \tau_{i1}^{\mathrm{V}} v_1 + \tau_{i2}^{\mathrm{V}} v_2 + k \frac{\partial \theta}{\partial x_i}\right)^{\mathrm{T}},\ i = 1, 2$$

Eulerian model proposed by Magnus Svärd

$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \nu(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_i}, \qquad \nu(\boldsymbol{w}) = \alpha \frac{\mu}{\rho}, \quad \alpha \in [1, 4/3],\ i = 1, 2$$

quantitative comparison: difference between models about 1%





Navier-Stokes equations

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \boldsymbol{f}_i(\boldsymbol{w}) = \sum_{i=1}^{2} \frac{\partial}{\partial x_i} \mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w})$$

$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \left(0,\ \tau_{i1}^{\mathrm{V}},\ \tau_{i2}^{\mathrm{V}},\ \tau_{i1}^{\mathrm{V}} v_1 + \tau_{i2}^{\mathrm{V}} v_2 + k \frac{\partial \theta}{\partial x_i}\right)^{\mathrm{T}},\quad i = 1, 2$$

Eulerian model proposed by Magnus Svärd

$$\mathbb{R}_i(\boldsymbol{w}, \nabla \boldsymbol{w}) = \nu(\boldsymbol{w}) \frac{\partial \boldsymbol{w}}{\partial x_i}, \qquad \nu(\boldsymbol{w}) = \alpha \frac{\mu}{\rho}, \quad \alpha \in [1, 4/3],\ i = 1, 2$$

quantitative comparison: difference between models about 1%