Study guide: Numerical solution of the Navier-Stokes equations

Hans Petter Langtangen[1,2]

Center for Biomedical Computing, Simula Research Laboratory[1]

Department of Informatics, University of Oslo[2]

Nov 11, 2014

The physical and mathematical problem

http://www.youtube.com/embed/P8VcZzgdfSc

http://www.youtube.com/embed/sI2uCHH3qIM

Lots of physical applications involve fluid flow

- Weather (flow in the atmosphere)
- Ocean currents
- Flight
- Drag on cars
- Blood circulation
- Breathing

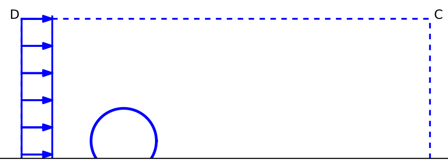
The physical assumptions behind the Navier-Stokes equations

Assumptions:

- Incompressible flow (velocity $< 1/3$ of the speed of sound)
- Laminar flow
- Simple fluids (constant viscosity ν)

Primary unknowns:

- velocity $\boldsymbol{u}(\boldsymbol{x}, t)$
- pressure $p(\boldsymbol{x}, t)$



The Navier-Stokes equations

Momentum balance (Newton's 2nd law):

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\frac{1}{\varrho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}$$

Mass balance (eq. of continuity):

$$\nabla\cdot\boldsymbol{u} = 0$$

Boundary conditions

- Dirichlet conditions: components of $\boldsymbol{u}$ are known
- Neumann conditions:
  - Stress condition: components of the stress vector $\boldsymbol{\sigma}\cdot\boldsymbol{n}$ are prescribed
  - Outflow or symmetry condition: $\partial\boldsymbol{u}/\partial n = 0$ (or components of this vector are zero)
- Pressure known at *a single point*

The classical splitting method

Idea: split the N-S equations into simpler problems (*operator splitting*).

A simple, naive approach

The equation for $\boldsymbol{u}$ looks like a diffusion equation...why not a Forward Euler scheme?

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\frac{1}{\varrho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}$$

$$\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^n}{\Delta t} + (\boldsymbol{u}^n\cdot\nabla)\boldsymbol{u}^n = -\frac{1}{\varrho}\nabla p^n + \nu\nabla^2\boldsymbol{u}^n + \boldsymbol{f}^n$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^n\cdot\nabla)\boldsymbol{u}^n - \frac{\Delta t}{\varrho}\nabla p^n + \Delta t\,\nu\nabla^2\boldsymbol{u}^n + \Delta t\boldsymbol{f}^n$$

Two fundamental problems:

1. $\nabla\cdot\boldsymbol{u}^{n+1}\neq 0$ (that equation is not used!)
2. no computation of p^{n+1}

A working scheme

Idea: Forward Euler in time, but evaluate ∇p at t_{n+1} and enforce $\nabla\cdot\boldsymbol{u}^{n+1}=0$.

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^n\cdot\nabla)\boldsymbol{u}^n - \frac{\Delta t}{\varrho}\nabla p^{n+1} + \Delta t\,\nu\nabla^2\boldsymbol{u}^n + \Delta t\boldsymbol{f}^n,$$

$$\nabla\cdot\boldsymbol{u}^{n+1} = 0$$

Note: *implicit system* for $\boldsymbol{u}^{n+1}$ and p^{n+1}

We solve the implicit system by a splitting technique

- Use old $\beta\nabla p^n$ for ∇p^{n+1} and advance to intermediate velocity $\boldsymbol{u}^*$
- Correct the $\boldsymbol{u}^*$ velocity by $\nabla\cdot\boldsymbol{u}^{n+1}=0$

Intermediate velocity (Forward Euler):

$$\boldsymbol{u}^* = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^n\cdot\nabla)\boldsymbol{u}^n - \beta\frac{\Delta t}{\varrho}\nabla p^n + \Delta t\,\nu\nabla^2\boldsymbol{u}^n + \Delta t\boldsymbol{f}^n$$

Seek correction $\delta\boldsymbol{u}$ such that

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* + \delta\boldsymbol{u}$$

fulfills

$$\nabla\cdot\boldsymbol{u}^{n+1} = 0$$

A Poisson equation must be solved to ensure $\nabla\cdot\boldsymbol{u}=0$

Subtract $\boldsymbol{u}^*$ equation from original $\boldsymbol{u}^{n+1}$ equation to find $\delta\boldsymbol{u}$:

$$\delta\boldsymbol{u} = \boldsymbol{u}^{n+1} - \boldsymbol{u}^* = -\frac{\Delta t}{\varrho}\nabla\Phi$$

where

$$\Phi = p^{n+1} - \beta p^n$$

The oldest methods had $\beta=0$, but $\beta\neq 0$ gives in general better speed and accuracy.

$\nabla\cdot\boldsymbol{u}^{n+1}=0$ implies

$$\nabla\cdot\delta\boldsymbol{u} = -\nabla\cdot\boldsymbol{u}^*$$

which gives

$$\nabla^2\Phi = \frac{\varrho}{\Delta t}\nabla\cdot\boldsymbol{u}^*$$

Summary

1. Compute the intermediate velocity $\boldsymbol{u}^*$
2. Solve the Poisson equation for Φ
3. Update the velocity: $\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{\Delta t}{\varrho}\nabla\Phi$
4. Update the pressure: $p^{n+1} = \Phi + \beta p^n$

Basically, we have $u=f$ approximation problems (1, 3, 4) and a Poisson equation to solve.

Boundary conditions

Problem: p condition at one point only in the original N-S equations. Now we need boundary conditions for Φ along the whole boundary (Poisson equation).

- Use conditions for $\boldsymbol{u}$ also for $\boldsymbol{u}^*$
- Known pressure: known Φ
- Known pressure gradient: known $\partial\Phi/\partial n$
- Otherwise $\partial\Phi/\partial n = 0$

Spatial discretization by the finite element method

- $\boldsymbol{u}^*, \boldsymbol{u}^{n+1} \in V^{(u)}$ (modulo nonzero Dirichlet cond.)
- $p^{n+1} \in V^{(\Phi)}$ (modulo nonzero Dirichlet cond.)
- Test function $\boldsymbol{v}^{(u)} \in V^{(u)}$ for vector equations (velocity)
- Test function $v^{(\Phi)} \in V^{(\Phi)}$ for scalar equations (pressure)
- Take inner product of vector equation and $\boldsymbol{v}^{(u)}$
- Integrate $\nabla^2\boldsymbol{u}\cdot\boldsymbol{v}^{(u)}$ by parts
- Integrate $\nabla p\cdot\boldsymbol{v}^{(u)}$ by parts (optional)
- Notation: $\boldsymbol{u}$ is $\boldsymbol{u}^{n+1}$, $\boldsymbol{u}_1$ is $\boldsymbol{u}^n$, p is p^{n+1}, p_1 is p^n (as in code)

$$\int_\Omega (\boldsymbol{u}^*\cdot\boldsymbol{v}^{(u)} + \Delta t((\boldsymbol{u}_1\cdot\nabla)\nabla\boldsymbol{u}_1)\cdot\boldsymbol{v}^{(u)} - \frac{\Delta t}{\varrho}p\nabla\cdot\boldsymbol{v}^{(u)} + \Delta t\,\nu\nabla\boldsymbol{u}_1\cdot\nabla\boldsymbol{v}^{(u)} - \Delta t f_1)\,\mathrm{d}x + \int_{\partial\Omega_{N,u}}\left(\nu\frac{\partial\boldsymbol{u}}{\partial n} - p\boldsymbol{n}\right)\cdot\boldsymbol{v}^{(u)}\,\mathrm{d}s \quad (1)$$

$\forall\boldsymbol{v}^{(u)} \in V^{(u)}$.

Natural boundary condition:

Increasing the implicitness

Stability (due to Forward Euler-style scheme):

$$\Delta t \le \frac{h^2}{2\nu + Uh}\,. \quad (5)$$

h: minimum element size, U: typical velocity.

Better stability by a Backward Euler scheme:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^{n+1}\cdot\nabla)\boldsymbol{u}^{n+1} - \frac{\Delta t}{\varrho}\nabla p^{n+1} + \Delta t\,\nu\nabla^2\boldsymbol{u}^{n+1} + \Delta t\boldsymbol{f}^{n} \quad (6)$$

$$\nabla\cdot\boldsymbol{u}^{n+1} = 0\,. \quad (7)$$

Intermediate velocity ($\nabla p^{n+1} \to \beta p^n$):

$$\boldsymbol{u}^* = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^*\cdot\nabla)\boldsymbol{u}^* - \beta\frac{\Delta t}{\varrho}p^{n+1} + \Delta t\,\nu\nabla^2\boldsymbol{u}^* + \Delta t\boldsymbol{f}^{n+1}$$

Methods based on slight compressibility

$\nabla\cdot\boldsymbol{u} = 0$ is problematic. Allow slight compressibility in the fluid:

$$p_t + c^2\nabla\cdot\boldsymbol{u} = 0\,.$$

c: speed of sound.

Now we have evolution equations for $\boldsymbol{u}$ and p:

$$\boldsymbol{u}_t = -(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \frac{1}{\varrho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}\,, \quad (9)$$

$$p_t = -c^2\nabla\cdot\boldsymbol{u}\,. \quad (10)$$

Forward Euler:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^n\cdot\nabla)\boldsymbol{u}^n - \frac{\Delta t}{\varrho}\nabla p^n + \Delta t\,\nu\nabla^2\boldsymbol{u}^n + \Delta t\boldsymbol{f}^n\,, \quad (11)$$

$$p^{n+1} = p^n - \Delta t c^2\nabla\cdot\boldsymbol{u}^n\,. \quad (12)$$

Applications

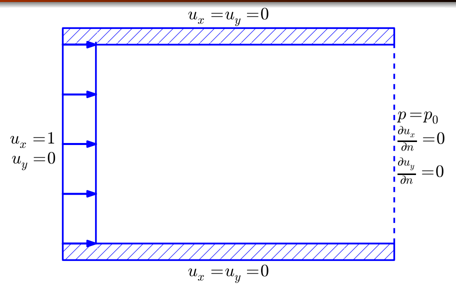


Figure : Flow in a channel.

$u_y = 0$, $\frac{\partial u_x}{\partial n} = 0$