

centered.h Documentation

Haochen (Langford) Huang

November 26, 2025

Version: 1.0 Updated: 2025-01-01

Section 1

Introduction

This solver addresses the incompressible Navier-Stokes equations[4, 3] on collocated grids i.e. velocity \mathbf{u} as well as pressure p are stored at center of the cell:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = \frac{1}{\rho} (-\nabla p + \nabla \cdot (2\mu \mathbf{D})) + \mathbf{a} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

where the deformation tensor is defined as $\mathbf{D} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$.

For multiphase flows, the acceleration term \mathbf{a} includes the effects of interfacial forces, such as surface tension. With the incompressibility constraint in equation 2, the velocity \mathbf{u} and pressure gradient ∇p at the next time step can be determined using approximate projection methods. The discrete form is:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u})^{n+1/2} = \frac{1}{\rho^{n+1/2}} (-\nabla p^{n+1} + \nabla \cdot (2\mu^{n+1/2} \mathbf{D}^{**})) + \mathbf{a}^{n+1/2} \quad (3)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (4)$$

Here, the viscosity term is computed implicitly using the intermediate velocity \mathbf{u}^{**} .

The following subsections will provide a theoretical overview of each step, with practical program analysis presented in subsequent sections. Notably, in the following subsections, face-centered variables will be denoted by the subscript f , such as μ_f , to distinguish them from cell-centered variables. Readers are also referred to the excellent introduction written by Edoardo Cipriano [2], which has been a great inspiration for this documentation.

1.1 Algorithm Overview

To summarize, solving the Navier-Stokes equations involves four main steps: **Property Update**, **Advection**, **Viscosity**, and **Projection**, corresponding to distinct stages of the solution process. The first three steps involve solving the following equations:

$$\frac{c^{n+1/2} - c^{n-1/2}}{\Delta t} + \mathbf{F}(c, \mathbf{u}_f^n) = 0 \quad (5)$$

$$\rho^{n+1/2} \left[\frac{\mathbf{u}^{**} - \mathbf{u}^n}{\Delta t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u})^{n+1/2} - \mathbf{a}^{n-1/2} + \frac{\nabla p^n}{\rho^{n-1/2}} \right] = \nabla \cdot (2\mu_f^{n+1/2} \mathbf{D}^{**}) \quad (6)$$

$$\mathbf{u}^{***} = \mathbf{u}^{**} - \Delta t \left(\mathbf{a}^{n-1/2} - \frac{\nabla p^n}{\rho^{n-1/2}} \right) \quad (7)$$

Once \mathbf{u}^{***} is computed, we obtain \mathbf{u}^{n+1} with:

$$\mathbf{u}^{n+1} = \mathbf{u}^{***} + \Delta t \left(\mathbf{a}^{n+1/2} - \frac{\nabla p^{n+1}}{\rho^{n+1/2}} \right) \quad (8)$$

The pressure gradient ∇p^{n+1} in the previous equation is obtained by applying the incompressibility constraint:

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (9)$$

to solve a Poisson equation in the **Projection** step.

1.1.1 Property Update

Equation 5 is the discrete form of advection equation for markers

$$\frac{\partial c}{\partial t} + \nabla \cdot (c \mathbf{u}_f^n) = 0 \quad (10)$$

the scheme of advection term $\mathbf{F}(c, \mathbf{u}_f^n)$ varies depending on the practical condition, the \mathbf{u}_f^n here is the strict non-divergence face velocity obtained in the last final step (see section 1.4.1). If c denotes the volume of fluids and the header file `two-phase.h` is included, the geometric advection scheme in `vof.h` is activated. Otherwise, if c represents tracers, the BCG scheme is applied to solve the equation. For further details, refer to the `tracers.h` and `vof.h` documentation.

The density ρ and viscosity μ_f at the $n + 1/2$ time step are determined based on the distribution of c , this step is implemented in `two-phase-generic.h`. For those condition with large density ratio, there is option to smear the $\alpha_f = 1/\rho$ in the same header file, as it is considered in poisson equation solver. All these steps are achieved by event succession, note although the body force such as surface tension is associated with distribution of c , it is actually implemented in the **Projection** step also through succession of event ‘acceleration’.

1.2 Advection

In this step, we merge the first two terms on the L.H.S. of Equation 6, solving:

$$\rho^{n+1/2} \left[\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u})^{n+1/2} \right] = 0 \quad (11)$$

To construct the discrete form of the nonlinear advection term $\nabla \cdot (\mathbf{u} \otimes \mathbf{u})^{n+1/2}$ while preserving conservation, a divergence-free face velocity \mathbf{u}_f is required. Instead of using the classical Rhie–Chow (RC) interpolation [5] commonly employed in finite-volume solvers, **Basilisk** adopts the BCG scheme, which incorporates both the advective fluxes and the pressure-gradient contributions when computing face velocities. A projection operator is then applied to enforce the divergence-free condition.

Given the face velocity $\mathbf{u}_f^{n+1/2}$, equation (11) is advanced using functions provided in the header file `bcg.h`, which implements the full BCG discretization. For additional details on the BCG scheme, please refer to the `bcg.h` documentation.

1.3 Diffusion

This step aims to compute \mathbf{u}^{**} . Starting from Equation 11, Equation 6 becomes:

$$\rho^{n+1/2} \left[\frac{\mathbf{u}^{**} - \mathbf{u}^*}{\Delta t} - \mathbf{a}^{n-1/2} + \frac{\nabla p^n}{\rho^{n-1/2}} \right] = \nabla \cdot \left(2\mu_f^{n+1/2} \mathbf{D}^{**} \right) \quad (12)$$

Here, \mathbf{u}^{**} is computed implicitly, and \mathbf{u}^{***} is subsequently determined by equation 7.

1.4 Projection

The final step enforces the non-divergence constraint. A key challenge in this stage is the misalignment introduced by the collocated grid arrangement: velocities are stored at cell centers, whereas pressure gradients are naturally defined on cell faces.

1.4.1 From Cell-Centered to Face-Centered: Computing p^{n+1}

We compute the pressure p^{n+1} from

$$\frac{\mathbf{u}_f^{n+1} - \mathbf{u}_f^{***}}{\Delta t} = -\frac{\nabla p^{n+1}}{\rho^{n+1/2}} + \mathbf{a}_f^{n+1/2}, \quad (13)$$

$$\nabla \cdot \mathbf{u}_f^{n+1} = 0, \quad (14)$$

which leads to the following cell-centered pressure equation, obtained by applying the projection operator to the face velocity:

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1/2}} \right) = \nabla \cdot \left(\frac{\mathbf{u}_f^{***}}{\Delta t} \right). \quad (15)$$

The intermediate face velocity is given by

$$\mathbf{u}_f^{***} = \frac{\mathbf{u}^{***}[\] + \mathbf{u}^{***}[-1]}{2} + \Delta t \mathbf{a}_f^{n+1/2}. \quad (16)$$

Here we emphasize that, since \mathbf{u}^{***} already includes all contributions except pressure, the face velocity \mathbf{u}_f^{***} can be obtained simply by averaging adjacent cell-centered values and adding the acceleration term. No additional treatment—such as RC interpolation or the BCG construction described in section 1.2—is required.

Finally, the divergence-free face velocity is updated as

$$\mathbf{u}_f^{n+1} = \mathbf{u}_f^{***} - \frac{\Delta t}{\rho^{n+1/2}} \nabla p^{n+1}, \quad (17)$$

and will be used for tracer advection in the next time step.

1.4.2 From Face-Centered to Cell-Centered: Updating \mathbf{u}^{n+1}

With p^{n+1} computed, the cell-centered velocity \mathbf{u}^{n+1} is updated:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{***}}{\Delta t} = -\frac{\nabla p^{n+1}}{\rho^{n+1/2}} + \mathbf{a}^{n+1/2} \quad (18)$$

To transform face-centered values to cell-centered values, the face-centered pressure gradient is computed and combined with the face-centered acceleration $\mathbf{a}_f^{n+1/2}$ to yield:

$$\mathbf{g}_f^{n+1} = \mathbf{a}_f^{n+1/2} + \frac{\nabla p^{n+1}}{\rho_f^{n+1/2}} \quad (19)$$

The cell-centered \mathbf{g}^{n+1} is obtained by averaging:

$$\mathbf{g}^{n+1} = \frac{\mathbf{g}_f^{n+1}[1] + \mathbf{g}_f^{n+1}[\]}{2} \quad (20)$$

Finally, the velocity is updated:

$$\mathbf{u}^{n+1} = \mathbf{u}^{***} + \Delta t \mathbf{g}^{n+1} \quad (21)$$

1.5 Notes on the Projection Method and VOF-NS Equations

1.5.1 Approximate vs. Exact Projection

The approximate projection method, as initially described, pertains to an algorithm where the discrete Laplace operator does not precisely equal the discrete divergence of the discrete gradient, according to Almgren (2000)[1]. This situation arises exclusively on grids where only the cell-center is active, due to the misalignment caused by the face-centered vector. However, **Basilisk** circumvents this issue. Nonetheless, as previously discussed, while the face-centered velocity u_f maintains strict non-divergence, the cell-centered velocity u_c does not, as it is derived by subtracting the average of the updated pressure-acceleration at the face center. Consequently, this is why the projection method utilized by **Basilisk** is termed an approximate projection.

1.5.2 The Non-conserving Form of the VOF-NS Equations

When applying the VOF method, equation 1 cannot be directly obtained, since the density is no longer a constant but a time-dependent function, given by

$$\rho(\mathbf{x}, t) = \rho_1 c^n + \rho_2 (1 - c^n). \quad (22)$$

Consequently, the incompressible Navier–Stokes equations take the form

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot (2\mu \mathbf{D}) + \mathbf{a}, \quad (23)$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (24)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (25)$$

Here, the mass continuity equation 24 and the incompressibility constraint 25 are decoupled for the same reason. By substituting the incompressibility condition back into the continuity equation, we obtain

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0. \quad (26)$$

Combining equation 26 with the momentum equation 23, we finally recover the Navier–Stokes equations in the form of equation 1. This formulation allows us to avoid explicitly handling the highly discontinuous advection terms that arise under large density ratios, such as in the case of an air–water interface ($\rho_{\text{water}} : \rho_{\text{air}} \sim 1000 : 1$).

References

- [1] A. S. Almgren, J. B. Bell, and W. Y. Crutchfield. “Approximate projection methods: Part I. Inviscid analysis”. In: *SIAM J. Sci. Comput.* 22.4 (2000), pp. 1139–1159.
- [2] E. Cipriano. *Introduction for centered.h*. <http://basilisk.fr/sandbox/ecipriano/doc/centered>. 2024.
- [3] S. Popinet. “A quadtree-adaptive multigrid solver for the Serre–Green–Naghdi equations”. In: *J. Comput. Phys.* 302 (2015), pp. 336–358.
- [4] S. Popinet. “Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries”. In: *J. Comput. Phys.* 190.2 (2003), pp. 572–600.
- [5] C. M. Rhie and W. L. Chow. “Numerical study of the turbulent flow past an airfoil with trailing edge separation”. In: *AIAA J.* 21.11 (1983), pp. 1525–1532.