**CaNS 2.0 Theory Manual**

This document introduces the numerics in CaNS 2.0, which solves the incompressible NS equations using a projection method on staggered grids. LES capabilities have been fully supported in the extended version.

**1. Staggered grid**

We show the calculation of convective  and viscous  terms on a staggered grid,



The following scheme can be obtained using finite difference/volume scheme. We take a 2D problem as an example, the two terms in the *u*-momentum equation,





The two terms in the *v*-momentum direction,





In the projection method, we also need the divergence of velocity



Figure 1 shows the stencil points for solving *u*, *v* and *p*. The same stencil points are used even if the viscous cross-terms are included. The stencil points for calculating eddy viscosity are considered separatedly.

A diagram of a game

Description automatically generated

Figure 1 Stencil points for the equations of *u* (a), *v* (b) and *p* (c,d)

**2. Poisson equation solver**

The elliptic Poisson equation is



It can be solved via iterative methods, such as Jacobi, Gauss-Seidel (with successive overrelaxation). ADI (AF) can also be used, but it can only obtain an approximate solution of Eq. . For 3D problems with two periodic (homogeneous) directions, namely *x* and *y*, and one non-homogeneous direction, *z*, it requires two consecutive FFTs in the homogeneous directions. We denote  and , where  and  are the numbers of cells in the *x* and *y* directions,



Fourier series in the direction,



Fourier series in the direction,



Hence,



A modified wavenumber is defined as





The eigenvalue  has no units, but the wavenumber has unit . Hence,



Given , Eq. is a tridiagonal system of equations; there are in total  systems of equations. For square ducts, the same form can be obtained by performing inverse DCT-II in the *y* direction after DFT in the *x* direction. The modified wavenumber is defined as





The wavenumber  is indexed as 0, 1, …, *N*-1. The kind of transform is completely determined by the boundary conditions of .

**3. Boundary condition**

The Poisson equation on nonuniform grids is





For boundary cells, the calculation of  in Eq. has to consider the boundary conditions in the implicit directions. We assume Dirichlet boundary conditions, i.e.,  and . For ,





For ,





If homogenous boundary conditions, the added term is zero at the right-hand side. In CaNS, both *x* and *y* directions must be homogeneous boundary conditions when the two directions are treated implicitly. Consequently, only the *z* direction is essentially considered in the calculation of the right-hand side at the boundary. We perform transforms in the *x* and *y* directions. For internal cell,











For ,









For ,









We highlight that all the extra treatments of the boundary conditions are due to the implicit *z* direction. In the following, we list the calculation of *b* and  for various boundary conditions. If Dirichlet boundary conditions,







If Neumann boundary conditions,







If periodic boundary conditions,







Eq. is efficiently solved using Thomas algorithm for periodic tridiagonal systems.

**4. Low-storage Runge-Kutta scheme**

In CaNS, the low-storage Runge-Kutta scheme is combined with the projection method (Chorin 1968; Harlow & Welch 1965; Kim & Moin 1985)









where,  and . The variable  is not needed due to . The operator  denotes nonlinear terms, including convective and modelled stress terms,



The operator  denotes second-order derivative,



The coefficients



The stability condition is



Its implementation is







The eddy viscosity part is typically less than the laminar viscous part, so it can be neglected. As the Reynolds number increases, the inviscid part dominates the stability condition. If the viscous term is treated implicitly only in the direction,



Let ,



Let ,



Expand it.



If the viscous term is treated implicitly in all the three directions,



Let ,



Let ,



Expand it,



If  is discretized via second-order central differencing, the left-hand side of Eq. is a heptadiagonal matrix. The modified Helmholtz equation Eq. can be solved via approximate factorization (AF) or transform method. Hence, the right-hand sides of Eq. , and







In computer codes, the explicitly treated terms are lumped together, and the implicit terms are lumped together. In CaNS, the implementation of Eq. and is



The solved variable is  directly. Then, the Poisson equation can be solved to obtain  for the interior cells, and the implementation of the boundary conditions assigns values to the ghost cells. These values are used to correct the velocity field. Note that the correction essentially does not correct the ghost cell wall-parallel velocities, although the wall-normal velocity at a wall can be corrected. If there is a homogeneous pressure boundary condition, the wall-normal velocity at the wall is not corrected either.

When implicit 1D/3D are used, the added term to the right-hand side at the boundary is determined by the boundary condition. The left-hand side terms are due to the viscous term , so the wall-normal velocity gradient (Neumann boundary condition) should be used to compute the extra terms added to the left-hand side, rather than no-slip boundary condition. Hence, the only correct approach for coupling a wall model with an implicit scheme is to modify the boundary condition from no-slip to Neumann boundary condition. Hence, the correct implementation of wall model is to completely change the boundary condition for both the left- and right-hand sides, rather than to just modify the right-hand side.

**5. Transform method**

Equation with 3D implicit viscous diffusion can be solved via transform method,



The transform needs both staggered and non-staggered forms. For example, *u* and *v* use staggered and non-staggered transforms in the direction, respectively. Table 1 lists the limitations of the transform method, and Table 2 lists the actual limitations of CaNS.

Table 1 Boundary conditions

|  |  |  |  |
| --- | --- | --- | --- |
|  | Explicit RK | Hybrid RK/CN | Hybrid RK/CN (1D) |
| Mesh- | Uniform | Uniform | Uniform |
| Mesh- | Uniform | Uniform | Uniform |
| Mesh- | Non-uniform | Non-uniform | Non-uniform |
| cbcvel- | PP, DD, NN, DN | PP, D0D0, N0N0, D0N0 | PP, DD, NN, DN |
| cbcvel- | PP, DD, NN, DN | PP, D0D0, N0N0, D0N0 | PP, DD, NN, DN |
| cbcvel- | PP, DD, NN, DN | PP, DD, NN, DN | PP, DD, NN, DN |

Table 2 Boundary conditions in CaNS

|  |  |  |  |
| --- | --- | --- | --- |
|  | Explicit RK | Hybrid RK/CN | Hybrid RK/CN (1D) |
| Mesh- | Uniform | Uniform | Uniform |
| Mesh- | Uniform | Uniform | Uniform |
| Mesh- | Non-uniform | Non-uniform | Non-uniform |
| bcvel- | PP, DD, NN, DN | PP, D0D0 | PP, DD, NN, DN |
| bcvel- | PP, DD, NN, DN | PP, D0D0 | PP, DD, NN, DN |
| bcvel- | PP, DD, NN, DN | PP, DD, NN, DN | PP, DD, NN, DN |

**6. LES**

The filtered NS equations obtained using a uniform filter,





It is written as





We need a subgrid model to compute , since  is unknown. In practical simulations, the filtering is implicitly achieved in the finite-volume discretization process. It can be understood as follows. We have a DNS turbulent flow field, and we initialize FVM by average flow quantities over each cell. Then, we can do the first time step (infinitesimal), using accurate subgrid stress and numerical flux. The quantities are cell-average values, equivalent to box-filtered quantities. In practical simulations, we must have spatial/temporal discretization errors and SGS errors. If central scheme is applied, we only have numerical dispersion errors and SGS dissipation errors. The subgrid stress is modelled as



The isotropic part is ignored in CaNS. Consequently, the viscous term becomes







The resolved viscous terms can be explicitly/implicitly treated, while the modelled terms are always explicitly treated. Next, we discuss the Smagorinsky models. The static Smagorinsky model reads as



where , and van Driest damping function,



The channel test case demonstrates that the performance of the model is sensitive to grid resolutions, grid anisotropy and model constant. The dynamic Smagorinsky model is (Pope 2020)









where the bar denotes the first filter, and the tilde denotes the second filter. In the following, we first discuss its derivation, then the implementation details. We do a second filtering operation on Eq. ,





Consequently,



The derivation of Germano’s identity assumes uniform grids. It is used to determine the coefficient of SGS models. In the following, we assume  is constant within a filter width and that it is the same for the two filtering operations. In practice, this is satisfied when averaging is applied in the homogeneous (or temporal) directions. The Smagorinsky model is



If the two filtering operations are low-pass spectral filters, we have





Substituting Eq. and into Eq. ,



It is further written that



Since



Eq. becomes



Define

 a

and squared error



We have



The squared error is minimized when



Eq. is proposed by Lilly in 1992 (Pope 2020). The dynamic procedure uses the following assumptions:

1. Uniform  and ;
2. Model coefficient is uniform at least within a filter width;
3. Model coefficient is the same for the two filtering operations;
4. The solution is obtained by minimizing the mean-square error.

The dynamic procedure is a method for computing the model coefficient, rather than obtaining an optimal or theoretical coefficient. This is particularly relevant when considering errors arising from numerical discretization and the static form of the subgrid-scale (SGS) model. In CaNS, the default filter is a 3D filter, because it is consistent with the initial implicit 3D box filter and provides more accurate results than a 2D filter in the channel test case. The choice of filter is discussed by (Balaras 1995; Cabot & Moin 1999). In the following, the bar denoting the first filter is left out for brevity. In CaNS, the implementation of the dynamic model is









where



In CaNS, the grid in the *z* direction is stretched, so the derivative and filtering operations are not commutable,



though we assume uniform cell size locally within a test filter size. The calculation of  requires the wall boundary conditions for . It can be shown that



where



We compute the two terms in Eq. by averaging velocities from cell faces to cell centers, and then performing the second filtering operation using these cell-centered velocities, as implemented in Bae’s code.

Regarding the wall model, the nondimensionalized log law reads as



We use Newton-Raphson iteration to solve for . We construct



So



The iteration formula is



The wall stress yielded from the wall model is represented by modifying the values of wall-parallel velocities at the ghost cells. No-penetration boundary condition is set on the wall-normal velocity, which ensures zero convective flux on the wall.

**7. Verification**

Verification is to demonstrate that a wall model has been correctly implemented into CaNS. It is done via comparison to other codes or analytical solutions (if existing). Validation is to demonstrate whether a model is good or not, in terms of representing/modeling the true physics. It is done via comparing to experiments, or analytical solutions (if existing). Here, verification is needed to show if the implementation of wall model is correct.

Table 3 Selected verification cases

|  |  |  |
| --- | --- | --- |
|  |  | Time advancement |
| Laminar channel2D |  | Explicit |
| Laminar channel2D |  | Implicit 1D |
| Laminar channel3D |  | Explicit |
| Laminar channel3D |  | Implicit 1D |
| Turbulent channel2D | 180-1010 | Explicit |
| Turbulent channel2D | 180-1010 | Implicit 1D |
| Turbulentchannel3D | 180-1010 | Explicit |
| Turbulent channel3D | 180-1010 | Implicit 1D |

Table 4 Selected validation cases

|  |  |  |
| --- | --- | --- |
|  | Re | Time advancement |
| Square duct |  | Explicit |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Appendix A**

The eigenvalues are given by . Table 5 and Table 6 list the transforms and inverse transforms for different combinations of boundary conditions. *N* is the number of input points, 0,1,…, *N*-1 instead of the number of cells. Zero-value points are always left out. The current index of input points is consistent with Scipy and wiki. The factors in the inverse transforms are consistent with Costa’s paper and Scipy.

Table 5 Transforms on a staggered grid (pressure)

|  |  |  |  |
| --- | --- | --- | --- |
| BC |  | Transform | Inverse transform |
| PP |  | DFT | IDFT |
| NN |  | DCT-II | DCT-III |
| DD |  | DST-II | DST-III |
| ND |  | DCT-IV | DCT-IV |
| DN |  | DST-IV | DST-IV |

Table 6 Transforms on a non-staggered grid (velocity)

|  |  |  |  |
| --- | --- | --- | --- |
| BC |  | Transform | Inverse transform |
| PP |  | DFT | IDFT |
| NN |  | DCT-I | DCT-I |
| DD |  | DST-I | DST-I |
| ND |  | DCT-III | DCT-II |
| DN |  | DST-III | DST-II |

We list the transforms and inverse transforms for the above 8 types. The formulas below have been checked using Python.

1. staggered NN





1. staggered DD





1. staggered ND





1. staggered DN





1. Non-staggered NN





1. Non-staggered DD





1. Non-staggered ND





1. Non-staggered DN





**Appendix B**

In LES, eddy viscosity is introduced. The hybrid RK-CN scheme Eq. becomes



where operator  denotes the viscous term,



Re-express it using ,



Let ，



Expand it,









Eq. is no longer a modified Helmholtz equation, so transform method cannot be applied. If we simplify the viscous term as done by Bae’s code, Eq. becomes





**References**

Balaras E. 1995. Finite-Difference Computations of High Reynolds Number Flows Using the Dynamic Subgrid-Scale Model. *Theoret. Comput. Fluid Dynamics*

Cabot W, Moin P. 1999. Approximate Wall Boundary Conditions in the Large-Eddy Simulation ofHigh Reynolds Number Flow. *Flow, Turbulence and Combustion*

Chorin AJ. 1968. Numerical Solution of the Navier-Stokes Equations

Harlow FH, Welch JE. 1965. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids*. 8(12):2182–89

Kim J, Moin P. 1985. Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*. 59(2):308–23

Pope SB. 2020. Turbulent Flows