**Theory Manual for CaNS 2.0**

This document explains the numeric in CaNS 2.0. The solver uses the projection method on staggered grids.

**1. Staggered grid**

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



The following scheme can be derived via finite difference/volume. Here, we regard it as a finite volume scheme. Taking 2D problems as an example, the two terms in the direction,





The two terms in the direction.





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





Figure 1 shows the stencils for solving , and .



Figure 1 Stencil points for solving , and

We assume a discontinuity between the wall and the first off-wall location. Across the discontinuity,  is large, while ,  and  are not large. Consequently, we should avoid the interpolation (reconstruction) or the differencing operator , where  is 1, or . Instead, one-sided operation biased towards the domain should be adopted. Table 1 lists the relevant terms in Eq. -. The conclusion is that no special treatments are needed near the wall for the projection method on a staggered grid,

Table 1 Terms that contain  or  in Eq. -

|  |  |  |
| --- | --- | --- |
| Eq. | Related term | Explanation |
|  |  | due to  on the wall |
|  |  | , computed from wall model |
| - | No |  |

A hand holding a piece of paper

Description automatically generated

I think there are basically two mistakes in the solution.

(1) Both velocity and its gradient are used at the wall, which makes the problem ill-posed (overdetermined). We should use only the velocity gradient as the wall boundary condition in WMLES; the velocity at the wall is non-zero.

(2) We should avoid differencing operator  across the layer between the wall and the first off-wall point. As Larsson reminded, that layer is under-resolved (discontinuous) in WMLES, so the wall-normal derivative computed using the wall point and the first off-wall point is incorrect. However, this solution does an interpolation using the velocity values at positions 1 and 2 and the velocity gradient at the wall, which implies that it assumes a smooth profile of velocity from the wall to position 1 to position 2. The assumption does not hold true, so this treatment can cause large errors.

In contrast, I still believe my current implementation is correct. It guarantees that the term  in Eq. . As for using ghost cells, it is just for convenience to guarantee , and for consistency with CaNS using ghost points for all kinds of boundary conditions. The size of the ghost cells is not important, as long as  can be exactly yielded. In CaNS, the ghost cells have the same size as the first off-wall cells , and . This exactly yields . As for , it is just normally computed as , using only the velocity information at positions 1 and 2.

Figure 2 shows the wall gradients provided by a wall model. In essence, only four locations need to be assigned values by the wall model, which uses, as input, the information at four interior locations.



Figure 2 Gradients provided by a wall model

Figure 3 shows a splitting of the domain into two subdomains. The calculation of wall gradients by the wall model only uses, as input, the information at interior points. Hence, it is confirmed that wall model is allowed to be called before updating ghost cell information.



Figure 3 Splitting of domain

**2. Poisson equation solver**

The elliptic Poisson equation can be solved via iterative methods, like Jacobi, Gauss-Seidel (with successive overrelaxation). ADI can be used, but it also needs iterations for a second-order accuracy in unsteady flows. This is different from AF method, whereby no iteration is involved if accurate  and  in  (assuming  is a constant matrix). However, there is no factorization of Poission equation. Fourier transform can be exploited for obtaining tridiagonal matrices. For a 3D problem with two periodic (homogeneous) directions, namely and , and one non-homogeneous direction, , it requires two consecutive FFTs in the homogeneous directions. In the following,  and ;  and  denote the number of cells,



Fourier series in the direction,



Fourier series in the direction,



Hence,



Inserting Eq. into Eq. ,



A modified wavenumber is defined as





Strictly speaking, the definition of eigenvalue  includes , but it is left out here for consistency with Pedro’s article. Inserting Eq. into Eq. ,



For a given set of , Eq. is a tridiagonal system of equations; there are in total systems of equations. If uniform grid spacing in the direction,



This is not as efficient as solving a tridiagonal system of equations in the direction. Hence, Eq. is the set of equations solved in channel flow. For square ducts with the Neumann boundaries in the direction, IDCT-II is performed in that direction. Please refer to Numerical Recipes for the relationship between DFT and DCT. On a staggered grid where is stored at the cell centers from 0 to , Eq. and Eq. provide the formulations of DCT-II and IDCT-II. Performing cosine series expansion of in the direction after Fourier series in the direction (Eq. ),



Inserting Eq. into Eq. ,



Modified wavenumber is defined as





Hence,



It has the same form as Eq. , which includes equations. Note that the index of input data points is always 0,1,…,, and is the number of those input points.

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

We use symbols in Orlandi’s book. In CaNS, the low-storage Runge-Kutta scheme is combined with the projection method (Harlow and Welch, 1965; Chorin, 1968; Kim and Moin, 1985)



where,  and . The variable  is not needed due to . The operator  denotes the convection term, and  is second-order derivative. The coefficients



The stability condition is



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



In a computer code, ,



The right-hand side of Eq. is similar to that of Eq. . Let ,



That is,



If  is discretized via second-order central differencing, the left-hand side of Eq. is a heptadiagonal matrix. In 2D problems, it is a pentadiagonal matrix (see Apeendix A). The modified Helmholtz equation Eq. can be solved via approximate factorization (AF) or transform method. In CaNS, the transform method is used.

If the viscous term is treated implicitly only in the direction,



Re-expressed using ,



Let ,



That is,



The left-hand side of Eq. is a tridiagonal matrix. Note that has a different form in Eq. from that in Eq. .

**4. Approximate factorization**

Equation with 3D implicit viscous diffusion can be solved via approximate factorization. Introducing matrix symbols (not tensors) ,  and ,



Note that each of ,  and  is a tridiagonal matrix. Using approximate factorization (AF),



Re-expressed as



The sweep procedure is described in Anderson’s book (P245). In the *i* direction,  is first solved, then , etc. In summary, hybrid Runge-Kutta/Crank-Nicolson finally solves tridiagonal systems of equations, and there are no constraints on the kinds of boundary conditions. The three-step explicit Runge-Kutta helps improve the accuracy, and the implicit Crank-Nicolson helps increase .

**5. Transform method**

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



The formulations of the eigenvalues are given in Table 6 and Table 7 in Appendix C. Note that both staggered and non-staggered transforms are used. In a square duct, and use staggered and non-staggered transforms in the direction, respectively.

In CaNS, transform method (non-staggered) is also used in the and direction when the diffusion term is treated implicitly in all the directions. Table 2 lists the theoretical limitations of CaNS, and Table 3 lists the actual limitations. The same constraints apply to both the implicit and implicit 1D scheme. Consequently, the wall modeling capability always applies in the direction, and only in explicit (or implicit 1D) form, applies to and directions. Also note that homogenous Neumann boundary condition always applies for walls with/without wall models, due to the no-penetration wall-normal boundary condition.

Table 2 Limitations of time-advancement schemes

|  |  |  |  |
| --- | --- | --- | --- |
|  | Explicit RK | Hybrid RK/CN | Hybrid RK/CN (1D, *z*) |
| 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 3 Limitations of CaNS

|  |  |  |  |
| --- | --- | --- | --- |
|  | Explicit RK | Hybrid RK/CN | Hybrid RK/CN (1D, z) |
| 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 (PP, D0D0) |
| bcvel- | PP, DD, NN, DN | PP, D0D0 | PP, DD, NN, DN (PP, D0D0) |
| bcvel- | PP, DD, NN, DN | PP, DD, NN, DN | PP, DD, NN, DN |

**6. LES**

The nondimensionalized log law reads as



We use Newton-Raphson to solve in Eq. . We construct





The Newton-Raphson iteration formula is



The current coupling is though modifying the ghost cell wall-parallel velocity values to match the wall stress yielded from a wall model. No-penetration boundary condition is applied on the walls, so zero convective flux is guaranteed; this implementation strategy ensures no convective flux on the walls. Of course, there is slip velocity on the wall.

The classical Smagorinsky model reads as



where , and van Driest damping function, . For varying eddy viscosity, the viscous term becomes







where the stress tensor , with the isotropic part ignored in the Boussinesq assumption. Due to the added eddy viscosity, the time step limitation has to be altered.

**7. Verification and validation**

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 4 Selected verification cases

|  |  |  |
| --- | --- | --- |
|  |  | Time advancement |
| Laminar channel2D |  | Explicit |
| Laminar channel2D |  | Implicit 1D |
| Laminar channel3D |  | Explicit |
| Laminar channel3D |  | Implicit 1D |
| Turbulent channel2D | 180-10^10 | Explicit |
| Turbulent channel2D | 180-10^10 | Implicit 1D |
| Turbulentchannel3D | 180-10^10 | Explicit |
| Turbulent channel3D | 180-10^10 | Implicit 1D |

Table 5 Selected validation cases

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

**Appendix A**

For an interior , and control volumes, we have





In the following, we derive the Poisson equation of pressure. We define



We have



where



We should have  for every cell at the end of the time cycle. Hence,





Eq. , , , are the set of equations used for solving the Navier-Stokes equations. When there is variable eddy viscosity, the viscous term is





**Appendix B**

Pentadiagonal matrix yielded from Eq. in 2D problems.



**Appendix C**

The eigenvalues are given by . Table 6 and Table 7 list the transforms and inverse transforms for different combinations of boundary conditions. is the number of input points, 0,1,…,, 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 Pedro’s paper and scipy.

Table 6 Transforms on a staggered grid

|  |  |  |  |
| --- | --- | --- | --- |
| 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 7 Transforms on a non-staggered grid

|  |  |  |  |
| --- | --- | --- | --- |
| 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 formulations 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 D**

We show that on rectangular meshes, FVM and FDM are equivalent. The momentum equation is





Volume integral,





Gaussian theorem,





Here, we assume the cell spacings are non-uniform in the direction. Due to second-order spatial accuracy, the volume average of is equal to the cell-center value. The FDM discretization multiplied by volume





This form can be interpreted as FVM if

1. and is interpreted as volume average (or volume-center value). Since the grid spacings are non-uniform in the direction, the location of is different from the control volume center. Hence, it is better to interpret it as control volume average.

**References**