**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 is



It 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 Costa’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 also required a uniform grid spacing in the z direction, which might be the reason that Eq. is the set of equations solved in channel flow. In addition, it is more efficient to solve a tridiagonal system of equations than a third Fourier transform. For a square duct with homogeneous 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 wavenumber *kx* is always indexed as 0,1,…,, with the number of those input points, even if the input data points are indexed as 1,2,…,*N*. For example, the input data points are indexed as *i*=1,2,…,*Nx* in CaNS, then *kx* is *i*-1.

The kind of transform is determined by the boundary conditions of , nothing to do with *R*, because the ghost cell values of *R* are not used in either the physical space or the spectral space, whereas the ghost cell values of are used in both spaces. In contrast, the ghost cell values of is used. In the following, we denote the form of Eq. and as



Hence, we need to solve



For nonuniform grid spacings in the z directions,



That is,









Now, we consider the boundary conditions. In this regard, special care should be given to the calculation of  in the right hand side and *b* in the left hand side.

For Poisson equation, the right hand side of Eq. in internal cells is



For boundary cells, the calculation of  has to consider the boundary conditions in the implicit directions. For Poisson equation, three directions need to be considered, and for momentum equations, three directions or only z directions need to be considered. Here, we take z direction as an example to illustrate its calculation. The Poisson equation at pressure cell (*i*, *j* ,*k*=1) is



If Dirichlet boundary condition,



Here, the boundary condition can be homogeneous (*c*=0) or nonhomogeneous (c≠0) in this step. If we assume periodic boundary conditions in the *x* and *y* directions,



That is,







In the following, we list the calculation of *b* and  for various boundary conditions,

If Dirichlet boundary conditions,







If Neumann boundary condition,







If periodic boundary conditions,







Eq. can be solved using Thomas algorithm for periodic tridiagonal systems. We would like to highlight that the calculation of  for boundary cells do not require homogeneous boundary conditions, whereas the Fourier transform requires homogeneous boundary conditions.

For a modified Poisson equation,



where is a known function of the coordinates,



**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 nonlinear terms, including convective term and modelled stress term,



where is modelled stress. The operator  is second-order derivative. The coefficients



The stability condition is



Its implementation is



The eddy viscosity part is typically much less than the 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 in all the three directions,



Let ,



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



Expand it,



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.

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



Let ,



Let ,



Expand it,



The left-hand side of Eq. is a tridiagonal matrix. Compare the RHS of Eq. , and ,



In a computer code, the explicitly treated terms are lumped/stored together, and the implicit terms are stored together. In CaNS, the implementation of Eq. and is



The solved variable is  directly, rather than .

Then, it comes to the correction/projection procedure. The solution of the Poisson equation yields  for the interior cells, then the boundary condition implementation assigns values for the ghost cells. These values are used to correct the velocity field from the prediction step. 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 homogeneous pressure boundary condition, the wall-normal velocity at a wall is not corrected either, i.e. it keeps zero from the prediction step.

When implicit 1D/3D are used, the ghost cells are not included at the left hand side. The added extra term to the right hand side for the first off-wall layer of cells is determined by the given boundary condition, i.e., gradient or value on the wall. Since the left-hand side terms are essentially caused by the viscous term , the wall-normal gradient (wall stress) should be used to computed extra term add 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. Consequently, the correct implementation of wall model is to completely change the boundary condition for both the left and right hand sides, rather than just modifying the right hand side. My current implementation is the right method.

In LES, eddy viscosity is introduced. Consequently, 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, so transform method cannot be applied. If we simplify the viscous term as did in Bae’s code, Eq. becomes





**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. For a general velocity profile,





We have









Note that







For brevity, we do not include viscosity. At the wall, , and , so















In equilibrium wall models, we assume that the velocity direction does not vary with *y* within the wall model layer, i.e., . Consequently,















Eq. - are the same as Eq. -, but note that  is different from  in 3D TBLs. Hence, one possible strategy of wall modeling for 3D TBL is to predict the magnitude of wall shear stress and its direction, respectively. In the current log-law wall model,  is assumed, that is, the direction of velocity does not vary in the wall model layer, while its magnitude varies as a classical near-wall law.

For varying eddy viscosity, the viscous term becomes







It can also be expressed as







where the stress tensor . The isotropic part is ignored in CaNS. The resolved viscous terms can be treated either explicitly or implicitly, whereas the modelled terms, like the inviscid terms, are always treated explicitly. It is convenient to lump the modelled terms together with the inviscid terms. Consequently, we only need to add the modelled terms to the right hand side even if the viscous terms are treated implicitly. In contrast, if a wall model is used to change the boundary conditions and an implicit scheme is used, we have to further modify the right hand side at boundary cells to reflect the change of boundary condition values, assuming that a Neumann boundary condition is already set. When adding a wall model, we do not need to consider that homogeneous boundary conditions must be used in the FT direction, because the modification to the right hand side is irrelevant to FT. Before FT, all aspects have been considered in the physical space. We just need to be aware that the resulting wall model should not be used in the FT directions. Due to the added eddy viscosity, the time step criterion has to be altered. Specifically, variable  should be added to Eq. . However, this may not be important when the viscous term is treated implicitly, though it may be necessary when the viscous term is treated explicitly.

Now, we discuss various sources of error in LES. First, we discuss the errors in the analytical form. Then, we discuss the errors from numerical discretization. The incompressible NS equations are





We apply filtering,





Physical-space box filter is commonly used in practical simulations, so we use it as an example to discuss the properties of filtering. In box filtering,



We have







If we have



We would have



The spatial derivative operator and filtering operator are commutable if the integration region is not a function of spatial coordinates. In finite volume, the filtering volume  is identical to the control volume. In cartesian grids, this implies uniform  and , but it does not require . In CaNS, we have the commute errors since the grid spacing is nonuniform along the z direction. We assume uniform grid spacings, then Eq. and become





It can be written as





Then, we need a subgrid model to compute , since  is unknown. In practical simulations, the filtering is achieved in the finite-volume discretization process. This is called implicit filtering. It can be understood in this way. We have a continuous turbulent field, and we initialize FVM by average flow quantities over each cell. Then, assuming we use accurate SGS and numerical flux and infinitesimal time step, we can march the first time step without any errors. Then, we can march a second infinitesimal step without any errors, assuming numerical flux and SGS are accurate. The stored values are always cell-averaged values, which is essentially box-filtered quantities. Hence, the box filtering is achieved automatically in the discretization process of FVM. 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. Here, we use 1D viscous Burgers equation to demonstrate discretization errors and SGS dissipation. The filtered Burgers equation is



In the framework of FVM, the filtering (cell averaging) is implicit, and can be interpreted as a top hat filter in the physical space. Note that, there is no numerical approximation in Eq. . An approximation of Eq. by first-order upwind scheme would yield



A second-order central scheme yields



Since the two schemes are not higher than second order, they are equivalent to FDM with the variable stored at cell centers.





Then, Eq. becomes





The numerical viscosity from the first-order upwind discretization is



It is proportional to grid space  and velocity magnitude . Hence, the total viscosity is molecular viscosity, subgrid viscosity, and numerical viscosity. In CaNS, we use a second-order central scheme, so the numerical viscosity is zero, which helps us concentrate on SGS viscosity and numerical dispersion. Since numerical dispersion is unavoidable and depends on numerical schemes, an optimal SGS model must depend on the numerical errors (including mesh resolutions and numerical scheme), which is why the coefficient takes different values for second- and fourth-order numerical schemes.

Next, we discuss two SGS models. The classical Smagorinsky model read as



where  and , and van Driest damping function,



The test cases of a channel without a wall model demonstrate that the performance of the static Smagorinsky model is very sensitive to grid resolutions, grid anisotropy and model constant *Cs*. This drives us to implement its dynamic version. Here, the formulas in Pope’s book are









where



We first discuss the derivation of the dynamic Smagorinsky model, then we discuss the details of its implementation. Here, we use the symbols in Pope’s book. The bar denotes the first filtering operation, and the tilde denotes the second filtering operation. The second filter has a larger filer size. For a low-pass spectral filter, we have





Note that for other filters, we do not have Eq. and . We use the first filter to filter the NS equation,





We further do a filtering operation on Eq. ,





Consequently,



The significance of the identity is that  is known. Note that the assumptions in the derivation of Germano’s identity are uniform grid and Gaussian filter. It is easy to determine the coefficient of SGS models by Germano’s identity. In the following, we assume  is constant within a filter width and also that it is the same for the first and second filtering operations. In practice, this is satisfied when averaging is applied in the homogeneous (or temporal) directions.

Take Smagorinsky model as an example.



Assume the two filtering operations are low-pass spectral filters (), then we have



Substituting Eq. and into Eq. ,



It is further written that



Since



Eq. becomes



Define



Define error



We have



Hence,



Eq. is proposed by Lilly, 1992. To summarize, the dynamic procedure has the following simplifications:

1. Uniform  and ;
2. The two filtering operations are performed by low-pass spectral filter;
3. Model coefficient is uniform at least within filter width;
4. Model coefficient is the same for the two filtering operations;
5. The solution is obtain by minimizing the mean-square error.

Consequently, the dynamic procedure might be just a method to compute the model coefficient, instead of obtaining an optimal/theoretical coefficient. This is especially true if errors from numerical discretization and the static SGS model form are considered. In practice, 2D or 3D box filters can be applied. In CaNS, the default choice is 3D filter, because it is consistent with the first filter (implicit 3D box filter), as indicated by W. Cabot (1999), and that it yields more accurate results than the 2D filter from tests.

Next, we discuss the implementation of the dynamic Smagorinsky model in CaNS. The formulas are a bit different from above. The implementation is









where



since derivative and filtering operations are incommutable when the filtering size is nonuniform. In CaNS, the grid spacing in the z direction is stretched, so we must calculate using



though we assume uniform cell size locally within a test filter size. However, the calculation of  requires the wall boundary conditions of , which may be ambiguous when wall model is applied.

It can be shown that



where



Hence, we compute the two terms in Eq. by first averaging velocities to cell centers, then do the second filtering operation using the cell-centered velocities, as done in Bae’s code.

It has been found in our results that the predicted velocity profile between the first and second cell centers are very inaccurate, so the gradient between those two cell centers should not be used in the calculation of viscous flux and subgrid viscosity. Regarding the viscous terms,







Special treatments should be given to



Note that it is better to leave  as it, since it has no discontinuity (first principle). The method for computing the derivative is



Note that this treatment results in



Regarding the subgrid model, the velocity gradient for the first off-wall layer of cells should be calculated in a similar way. Note that this is equivalent to do a linear extrapolation to the first off-wall points. The values at the first off-wall points should be replaced by the interpolated values immediately after they are updated, simply because the computed values are incorrect. This surely also influences the inviscid terms. Note that the primary goal here is to get an accurate approximation of the first-order wall normal derivative, rather than an accurate value. Consequently, the second-order approximation is



For nonuniform grids,



where





Regarding the strain rate at the first off-wall layer of cells, we can use 1st or 2nd-order one-sided differencing scheme. Take the 2nd-order scheme as an example,



The test shows that a 1st -order scheme is enough for the computation of the strain rate at the first off-wall layer.

**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**

The seven-diagonal systems of equations from Eq. is



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 Costa’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.

**Appendix E**

In finite difference, the fourth-order central approximation is



In finite volume,







Note that that Eq. has the same form as Eq. . When the accuracy is not greater than second order, we also have , because reconstructed distribution has only one slope within a cell for FVM. We also note that the formula can not be obtained by using Lagrange interpolation. As stressed in Li’s PPT, reconstruction is essentially not interpolation for both FVM and FDM. The only relation is that reconstruction is equivalent to interpolation when the accuracy is not greater than second order.

**References**