CaLES

Theory Manual

Version 1.1

November 14, 2024

Contents

1 Staggered grid 2

2 Poisson equation solver 2

3 Boundary condition 4

4 Low-storage Runge-Kutta scheme 6

5 Transform method 9

6 LES 10

7 Verification 14

List of Code Listings

This document introduces the numerics in CaLES, which solves the filtered incompressible NS equations using a projection method on staggered grids. This document is also available on Overleaf. If you want to contribute to the manual, please email the author at maochao.xiao@uniroma1.it to get an editable Overleaf link.

1 Staggered grid

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

$$\frac{u^{n+1}_{i+1/2,j}-u^{n}_{i+1/2,j}}{\Delta t}=H_{i+1/2,j}+L_{i+1/2,j} \tag{1}$$

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

$$H_{i+1/2,j}=-\frac{(uu)_{i+1/2,j}-(uu)_{i-1/2,j}}{\Delta x}-\frac{(uv)_{i+1/2,j+1/2}-(uv)_{i+1/2,j-1/2}}{\Delta y}-\frac{P_{i+1,j}-P_{i,j}}{\Delta x} \tag{2}$$

$$L_{i+1/2,j}=\nu\left(\frac{u_{i+3/2,j}-2u_{i+1/2,j}+u_{i-1/2,j}}{\Delta x^2}+\frac{u_{i+1/2,j+1}-2u_{i+1/2,j}+u_{i+1/2,j-1}}{\Delta y^2}\right) \tag{3}$$

The two terms in the v-momentum direction,

$$H_{i,j+1/2}=-\frac{(uv)_{i+1/2,j+1/2}-(uv)_{i-1/2,j+1/2}}{\Delta x}-\frac{(vv)_{i,j+1}-(vv)_{i,j}}{\Delta y}-\frac{P_{i,j+1}-P_{i,j}}{\Delta y} \tag{4}$$

$$L_{i,j+1/2}=\nu\left(\frac{v_{i+1,j+1/2}-2v_{i,j+1/2}+v_{i-1,j+1/2}}{\Delta x^2}+\frac{v_{i,j+3/2}-2v_{i,j+1/2}+v_{i,j-1/2}}{\Delta y^2}\right) \tag{5}$$

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

$$\partial_i u_i=\frac{u_{i+1/2,j}-u_{i-1/2,j}}{\Delta x}+\frac{v_{i,j+1/2}-v_{i,j-1/2}}{\Delta y} \tag{6}$$

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 separately.

2 Poisson equation solver

The elliptic Poisson equation is

$$\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2}+\frac{\partial^2\phi}{\partial z^2}=R \tag{7}$$

It can be solved via iterative methods, such as Jacobi, Gauss-Seidel (with successive overrelaxation). ADI (Alternating Direction Implicit) can also be used, but it can only obtain an

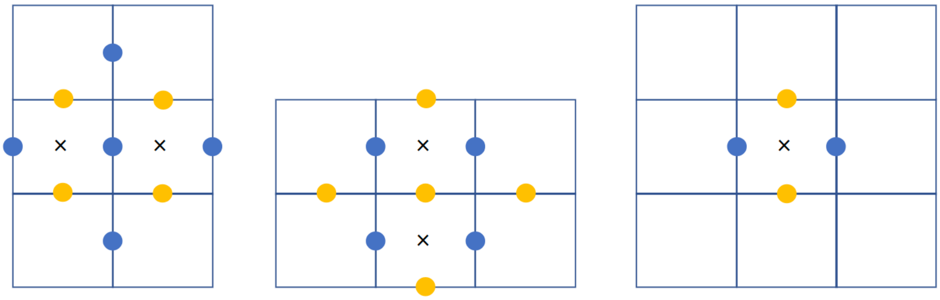

Figure 1 Stencil points for the equations of u, v and p

approximate solution of equation (7). 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 $\Delta x = L_x/N_x$ and $\Delta y = L_y/N_y$, where N_x and N_y are the numbers of cells in the x and y directions,

$$\frac{\phi_{i-1,j,k} - 2\phi_{i,j,k} + \phi_{i+1,j,k}}{\Delta x^2} + \frac{\phi_{i,j-1,k} - 2\phi_{i,j,k} + \phi_{i,j+1,k}}{\Delta y^2} + \frac{\phi_{i,j,k-1} - 2\phi_{i,j,k} + \phi_{i,j,k+1}}{\Delta z^2} = R_{i,j,k} \tag{8}$$

Fourier series in the x direction,

$$\phi_{i,j,k} = \frac{1}{N_x} \sum_{k_x=0}^{N_x-1} \hat{\phi}_{k_x,j,k} e^{\frac{\ell 2\pi k_x i}{N_x}} \tag{9}$$

Fourier series in the y direction,

$$\hat{\phi}_{k_x,j,k} = \frac{1}{N_y} \sum_{k_y=0}^{N_y-1} \hat{\hat{\phi}}_{k_x,k_y,k} e^{\frac{\ell 2\pi k_y j}{N_y}} \tag{10}$$

Hence,

$$\phi_{i,j,k} = \frac{1}{N_x N_y} \sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} \hat{\hat{\phi}}_{k_x,k_y,k} e^{\ell\left(\frac{2\pi k_x i}{N_x} + \frac{2\pi k_y j}{N_y}\right)} \tag{11}$$

A modified wavenumber is defined as

$$k_x'^2 = \frac{2\left(1 - \cos\left(\frac{2\pi k_x}{N_x}\right)\right)}{\Delta x^2} = \frac{4}{\Delta x^2} \sin^2\left(\frac{\pi k_x}{N_x}\right) = -\frac{\lambda_x}{\Delta x^2} \tag{12}$$

$$k_y'^2 = \frac{2\left(1 - \cos\left(\frac{2\pi k_y}{N_y}\right)\right)}{\Delta y^2} = \frac{4}{\Delta y^2} \sin^2\left(\frac{\pi k_y}{N_y}\right) = -\frac{\lambda_y}{\Delta y^2} \tag{13}$$

The eigenvalue λ_x has no units, but the wavenumber has unit $1/\mathrm{m}^2$. Hence,

$$\left(\frac{\lambda_x}{\Delta x^2} + \frac{\lambda_y}{\Delta y^2} + \delta_z^2\right) \hat{\hat{\phi}}_{k_x,k_y,k} = \hat{\hat{R}}_{k_x,k_y,k} \tag{14}$$

Given (k_x, k_y), equation (14) is a tridiagonal system of equations; there are in total $N_x \times N_y$ 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

$$k_x'^2 = \frac{2\left(1 - \cos\left(\frac{2\pi k_x}{N_x}\right)\right)}{\Delta x^2} = \frac{4}{\Delta x^2}\sin^2\left(\frac{\pi k_x}{N_x}\right) = -\frac{\lambda_x}{\Delta x^2} \tag{15}$$

$$k_y'^2 = \frac{2\left(1 - \cos\left(\frac{\pi k_y}{N_y}\right)\right)}{\Delta y^2} = \frac{4}{\Delta y^2}\sin^2\left(\frac{\pi k_y}{2N_y}\right) = -\frac{\lambda_y}{\Delta y^2} \tag{16}$$

The wavenumber k is indexed as $0, 1, \ldots, N-1$. The kind of transform is completely determined by the boundary conditions of ϕ.

3 Boundary condition

The Poisson equation on nonuniform grids is

$$\begin{aligned} &\frac{\phi_{i-1,j,k} - 2\phi_{i,j,k} + \phi_{i+1,j,k}}{\Delta x^2} + \frac{\phi_{i,j-1,k} - 2\phi_{i,j,k} + \phi_{i,j+1,k}}{\Delta y^2} \\ &+ \frac{1}{\Delta z_{f,k}}\left(\frac{\phi_{i,j,k+1} - \phi_{i,j,k}}{\Delta z_{c,k}} - \frac{\phi_{i,j,k} - \phi_{i,j,k-1}}{\Delta z_{c,k-1}}\right) = R_{i,j,k} \end{aligned} \tag{17}$$

where

$$R_{i,j,k} = \frac{\partial_i u_i^*}{\gamma^k \Delta t} \tag{18}$$

For boundary cells, the calculation of $R_{i,j,k}$ in equation (8) has to consider the boundary conditions in the implicit directions. We assume Dirichlet boundary conditions, i.e., $\phi_{i,j,1/2} = c$ and $\phi_{i,j,N_z+1/2} = c$. For $k = 1$,

$$\begin{aligned} &\frac{\phi_{i-1,j,1} - 2\phi_{i,j,1} + \phi_{i+1,j,1}}{\Delta x^2} + \frac{\phi_{i,j-1,1} - 2\phi_{i,j,1} + \phi_{i,j+1,1}}{\Delta y^2} \\ &- \frac{1}{\Delta z_{f,1}}\left(\frac{1}{\Delta z_{c,1}} + \frac{2}{\Delta z_{c,0}}\right)\phi_{i,j,1} + \frac{1}{\Delta z_{f,1}\Delta z_{c,1}}\phi_{i,j,2} = R_{i,j,1} \end{aligned} \tag{19}$$

$$R_{i,j,1} = \frac{\partial_i u_i^*}{\gamma^k \Delta t} - \frac{2c}{\Delta z_{f,1}\Delta z_{c,0}} \tag{20}$$

For $k = N_z$,

$$\begin{aligned} &\frac{\phi_{i-1,j,N_z} - 2\phi_{i,j,N_z} + \phi_{i+1,j,N_z}}{\Delta x^2} + \frac{\phi_{i,j-1,N_z} - 2\phi_{i,j,N_z} + \phi_{i,j+1,N_z}}{\Delta y^2} \\ &+ \frac{1}{\Delta z_{f,N_z}\Delta z_{c,N_z-1}}\phi_{i,j,N_z-1} - \frac{1}{\Delta z_{f,N_z}}\left(\frac{2}{\Delta z_{c,N_z}} + \frac{1}{\Delta z_{c,N_z-1}}\right)\phi_{i,j,N_z} = R_{i,j,N_z} \end{aligned} \tag{21}$$

$$R_{i,j,N_z} = \frac{\partial_i u_i^*}{\gamma^k \Delta t} - \frac{2c}{\Delta z_{f,N_z} \Delta z_{c,N_z}} \tag{22}$$

If homogeneous boundary conditions, the added term is zero at the right-hand side. In CaLES, 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 cells,

$$\frac{1}{\Delta z_{f,k}} \left(\frac{\hat{\hat{\phi}}_{i,j,k+1} - \hat{\hat{\phi}}_{i,j,k}}{\Delta z_{c,k}} - \frac{\hat{\hat{\phi}}_{i,j,k} - \hat{\hat{\phi}}_{i,j,k-1}}{\Delta z_{c,k-1}} \right) + \left(\frac{\lambda_x}{\Delta x^2} + \frac{\lambda_y}{\Delta y^2} \right) \hat{\hat{\phi}}_{i,j,k} = \hat{\hat{R}}_{i,j,k} \tag{23}$$

$$a\hat{\hat{\phi}}_{i,j,k-1} + b\hat{\hat{\phi}}_{i,j,k} + c\hat{\hat{\phi}}_{i,j,k+1} = \hat{\hat{R}}_{i,j,k} \tag{24}$$

$$a = \frac{1}{\Delta z_{f,k} \Delta z_{c,k-1}} \tag{25}$$

$$b = -(a+c) + \left(\frac{\lambda_x}{\Delta x^2} + \frac{\lambda_y}{\Delta y^2} \right) = -\frac{1}{\Delta z_{f,k}} \left(\frac{1}{\Delta z_{c,k-1}} + \frac{1}{\Delta z_{c,k}} \right) + \frac{\lambda_x}{\Delta x^2} + \frac{\lambda_y}{\Delta y^2} \tag{26}$$

$$c = \frac{1}{\Delta z_{f,k} \Delta z_{c,k}} \tag{27}$$

For $k = 1$,

$$\left(-\frac{1}{\Delta z_{f,1}} \left(\frac{2}{\Delta z_{c,0}} + \frac{1}{\Delta z_{c,1}} \right) + \frac{\lambda_x}{\Delta x^2} + \frac{\lambda_y}{\Delta y^2} \right) \hat{\hat{\phi}}_{i,j,1} + \frac{1}{\Delta z_{f,1} \Delta z_{c,1}} \hat{\hat{\phi}}_{i,j,2} = \hat{\hat{R}}_{i,j,1} \tag{28}$$

$$a_1 = 0 \tag{29}$$

$$b_1 = b - a = -\frac{1}{\Delta z_{f,1}} \left(\frac{2}{\Delta z_{c,0}} + \frac{1}{\Delta z_{c,1}} \right) + \frac{\lambda_x}{\Delta x^2} + \frac{\lambda_y}{\Delta y^2} \tag{30}$$

$$c_1 = \frac{1}{\Delta z_{f,1} \Delta z_{c,1}} \tag{31}$$

For $k = N_z$,

$$\frac{1}{\Delta z_{f,N_z}} \frac{1}{\Delta z_{c,N_z-1}} \hat{\hat{\phi}}_{i,j,N_z-1} + \left(\frac{\lambda_x}{\Delta x^2} + \frac{\lambda_y}{\Delta y^2} - \frac{1}{\Delta z_{f,N_z}} \left(\frac{2}{\Delta z_{c,N_z}} + \frac{1}{\Delta z_{c,N_z-1}} \right) \right) \hat{\hat{\phi}}_{i,j,N_z} = \hat{\hat{R}}_{i,j,N_z} \tag{32}$$

$$a_1 = \frac{1}{\Delta z_{f,N_z} \Delta z_{c,N_z-1}} \tag{33}$$

$$b_1 = b - c = -\frac{1}{\Delta z_{f,N_z}} \left(\frac{1}{\Delta z_{c,N_z-1}} + \frac{2}{\Delta z_{c,N_z}} \right) + \frac{\lambda_x}{\Delta x^2} + \frac{\lambda_y}{\Delta y^2} \tag{34}$$

$$c_1 = 0 \tag{35}$$

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 $R_{i,j,k}$ for various boundary conditions.

For Dirichlet boundary conditions,

$$R_{i,j,1} = \frac{\partial_i u_i^*}{\gamma^k \Delta t} - \frac{2c}{\Delta z_{f,1} \Delta z_{c,0}} \tag{36}$$

$$R_{i,j,N_z} = \frac{\partial_i u_i^*}{\gamma^k \Delta t} - \frac{2c}{\Delta z_{f,N_z} \Delta z_{c,N_z}} \tag{37}$$

$$\begin{bmatrix} b-a & c & & \\ a & b & c & \\ & a & b & c \\ & & a & b-c \end{bmatrix} \begin{bmatrix} \hat{\hat{\phi}}_{i,j,1} \\ \hat{\hat{\phi}}_{i,j,2} \\ \hat{\hat{\phi}}_{i,j,3} \\ \hat{\hat{\phi}}_{i,j,4} \end{bmatrix} = \begin{bmatrix} \hat{\hat{R}}_{i,j,1} \\ \hat{\hat{R}}_{i,j,2} \\ \hat{\hat{R}}_{i,j,3} \\ \hat{\hat{R}}_{i,j,4} \end{bmatrix} \tag{38}$$

For Neumann boundary conditions,

$$R_{i,j,1} = \frac{\partial_i u_i^*}{\gamma^k \Delta t} + \frac{c \Delta z_{c,0}}{\Delta z_{f,1} \Delta z_{c,0}} \tag{39}$$

$$R_{i,j,N_z} = \frac{\partial_i u_i^*}{\gamma^k \Delta t} - \frac{c \Delta z_{c,N_z}}{\Delta z_{f,N_z} \Delta z_{c,N_z}} \tag{40}$$

$$\begin{bmatrix} b+a & c & & \\ a & b & c & \\ & a & b & c \\ & & a & b+c \end{bmatrix} \begin{bmatrix} \hat{\hat{\phi}}_{i,j,1} \\ \hat{\hat{\phi}}_{i,j,2} \\ \hat{\hat{\phi}}_{i,j,3} \\ \hat{\hat{\phi}}_{i,j,4} \end{bmatrix} = \begin{bmatrix} \hat{\hat{R}}_{i,j,1} \\ \hat{\hat{R}}_{i,j,2} \\ \hat{\hat{R}}_{i,j,3} \\ \hat{\hat{R}}_{i,j,4} \end{bmatrix} \tag{41}$$

For periodic boundary conditions,

$$R_{i,j,1} = \frac{\partial_i u_i^*}{\gamma^k \Delta t} \tag{42}$$

$$R_{i,j,N_z} = \frac{\partial_i u_i^*}{\gamma^k \Delta t} \tag{43}$$

$$\begin{bmatrix} b & c & a & \\ a & b & c & \\ & a & b & c \\ & c & a & b \end{bmatrix} \begin{bmatrix} \hat{\hat{\phi}}_{i,j,1} \\ \hat{\hat{\phi}}_{i,j,2} \\ \hat{\hat{\phi}}_{i,j,3} \\ \hat{\hat{\phi}}_{i,j,4} \end{bmatrix} = \begin{bmatrix} \hat{\hat{R}}_{i,j,1} \\ \hat{\hat{R}}_{i,j,2} \\ \hat{\hat{R}}_{i,j,3} \\ \hat{\hat{R}}_{i,j,4} \end{bmatrix} \tag{44}$$

Equation (44) is efficiently solved using the Thomas algorithm for periodic tridiagonal systems. Note that patched boundaries are not relevant here, since the tridiagonal system contains all cells along the z direction after transposing pencils to the z direction. Hence, we should obtain the same results from using different numbers of threads.

4 Low-storage Runge-Kutta scheme

In CaLES, the low-storage Runge-Kutta scheme is combined with the projection method (Chorin, 1968; Harlow et al., 1965; Kim and Moin, 1985):

$$u_i^* = u_i^k + \Delta t \left(\alpha_{k+1} \left(H_i^k + \nu L_{jj} u_i^k \right) + \beta_{k+1} \left(H_i^{k-1} + \nu L_{jj} u_i^{k-1} \right) - \gamma_{k+1} \partial_i p^{k-1/2} \right) \tag{45}$$

$$L_{jj}\phi = \frac{\partial_i u_i^*}{\gamma_{k+1}\Delta t} \tag{46}$$

$$u_i^{k+1} = u_i^* - \gamma_{k+1}\Delta t\, \partial_i \phi \tag{47}$$

$$p^{k+1/2} = p^{k-1/2} + \phi \tag{48}$$

where $k = 0, 1, 2$, $u_i^0 = u_i^n$ and $u_i^3 = u_i^{n+1}$. The variable u_i^{-1} is not needed due to $\beta_1 = 0$. The operator H_i denotes nonlinear terms, including convective and modeled stress terms,

$$H_i = \frac{\partial u_i u_j}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{49}$$

The operator L_{ij} denotes the second-order derivative,

$$L_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} \tag{50}$$

The coefficients

$$\alpha_1 = \frac{8}{15}, \alpha_2 = \frac{5}{12}, \alpha_3 = \frac{3}{4}, \beta_1 = 0, \beta_2 = -\frac{17}{60}, \beta_3 = -\frac{5}{12}, \gamma_1 = \frac{8}{15}, \gamma_2 = \frac{2}{15}, \gamma_3 = \frac{1}{3} \tag{51}$$

The stability condition is

$$\Delta t \leq \min\left[\frac{1.65}{4\left((\nu_T + \nu)\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right)\right)}, \frac{\sqrt{3}}{\left(\frac{u_x}{\Delta x} + \frac{u_y}{\Delta y} + \frac{u_z}{\Delta z}\right)}\right] \tag{52}$$

Its implementation is

$$\Delta t_{exp} \leq \min\left[\frac{1.65}{4\left(\nu_T\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right) + \nu\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right)\right)}, \frac{\sqrt{3}}{\left(\frac{u_x}{\Delta x} + \frac{u_y}{\Delta y} + \frac{u_z}{\Delta z}\right)}\right] \tag{53}$$

$$\Delta t_{impl1d} \leq \min\left[\frac{1.65}{4\left(\nu_T\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right) + \nu\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)\right)}, \frac{\sqrt{3}}{\left(\frac{u_x}{\Delta x} + \frac{u_y}{\Delta y} + \frac{u_z}{\Delta z}\right)}\right] \tag{54}$$

$$\Delta t_{imp} \leq \min\left[\frac{1.65}{4\left(\nu_T\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right)\right)}, \frac{\sqrt{3}}{\left(\frac{u_x}{\Delta x} + \frac{u_y}{\Delta y} + \frac{u_z}{\Delta z}\right)}\right] \tag{55}$$

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 in all the three directions,

$$u_i^* = u_i^k + \Delta t\left(\alpha_{k+1} H_i^k + \beta_{k+1} H_i^{k-1} + \gamma_{k+1}\frac{\nu}{2} L_{jj}\left(u_i^* + u_i^k\right) - \gamma_{k+1}\partial_i p^{k-1/2}\right) \tag{56}$$

Let $\Delta u_i = u_i^* - u_i^k$,

$$\left(1 - \gamma_{k+1}\frac{\nu \Delta t}{2} L_{jj}\right)\Delta u_i = \Delta t\left(\alpha_{k+1} H_i^k + \beta_{k+1} H_i^{k-1} + \gamma_{k+1}\left(\nu L_{jj} u_i^k - \partial_i p^{k-1/2}\right)\right) \tag{57}$$

If L_{jj} is discretized via second-order central differencing, the left-hand side of equation (57) is a heptadiagonal matrix. The modified Helmholtz equation can be solved via approximate factorization (AF) or transform method. If the viscous term is treated implicitly only in the z direction,

$$\begin{aligned} u_i^* = u_i^k + \Delta t \Big(\alpha_{k+1} \left(H_i^{k+1} + \nu (L_{11} + L_{22}) u_i^k \right) + \beta_{k+1} \left(H_i^{k-1} + \nu (L_{11} + L_{22}) u_i^{k-1} \right) \\ + \gamma_{k+1} \frac{\nu}{2} L_{33} (u_i^* + u_i^k) - \gamma_{k+1} \partial_i p^{k-1/2} \Big) \end{aligned} \tag{58}$$

Let $\Delta u_i^* = u_i^* - u_i^k$,

$$\begin{aligned} \left(1 - \gamma_{k+1} \frac{\nu \Delta t}{2} L_{33}\right) \Delta u_i^* = \Delta t \Big(\alpha_{k+1} \left(H_i^k + \nu \left(L_{11} + L_{22}\right) u_i^k \right) + \beta_{k+1} \left(H_i^{k-1} + \nu \left(L_{11} + L_{22}\right) u_i^{k-1} \right) \\ + \gamma_{k+1} \left(\nu L_{33} u_i^k - \partial_i p^{k-1/2} \right) \Big) \end{aligned} \tag{59}$$

Hence, the right-hand sides of equations (45), (57) and (59) are

$$R_i^{exp} = \Delta t \left(\alpha_{k+1} \left(H_i^k + \nu L_{jj} u_i^k \right) + \beta_{k+1} \left(H_i^{k-1} + \nu L_{jj} u_i^{k-1} \right) - \gamma_{k+1} \partial_i p^{k-1/2} \right) \tag{60}$$

$$R_i^{imp} = \Delta t \left(\alpha_{k+1} H_i^k + \beta_{k+1} H_i^{k-1} + \gamma_{k+1} \left(\nu L_{jj} u_i^k - \partial_i p^{k-1/2} \right) \right) \tag{61}$$

$$\begin{aligned} R_i^{imp1d} = \Delta t \Big(\alpha_{k+1} \left(H_i^k + \nu \left(L_{11} + L_{22}\right) u_i^k \right) + \beta_{k+1} \left(H_i^{k-1} + \nu \left(L_{11} + L_{22}\right) u_i^{k-1} \right) \\ + \gamma_{k+1} \left(\nu L_{33} u_i^k - \partial_i p^{k-1/2} \right) \Big) \end{aligned} \tag{62}$$

In computer codes, the explicitly treated terms are lumped together, and the implicit terms are lumped together. Equation (57) can be written as

$$\begin{aligned} \left(1 - \gamma_{k+1} \frac{\nu \Delta t}{2} L_{jj}\right) u_i^* = \Delta t \Big(\alpha_{k+1} H_i^{k+1} + \beta_{k+1} H_i^{k-1} \\ + \gamma_{k+1} \left(\nu L_{jj} u_i^k - \partial_i p^{k-1/2} \right) \Big) + \left(1 - \gamma_{k+1} \frac{\nu \Delta t}{2} L_{jj}\right) u_i^k \end{aligned} \tag{63}$$

In CaLES, it is split into

$$u_i^{**} = u_i^k + \Delta t \left(\alpha_{k+1} H_i^k + \beta_{k+1} H_i^{k-1} + \gamma_{k+1} \left(\nu L_{jj} u_i^k - \partial_i p^{k-1/2} \right) \right) \tag{64}$$

$$u_i^* - \gamma_{k+1} \frac{\nu \Delta t}{2} L_{jj} u_i^* = u_i^{**} - \gamma_{k+1} \frac{\nu \Delta t}{2} L_{jj} u_i^k \tag{65}$$

$$p^{k+1/2} = p^{k-1/2} + \phi - \gamma_{k+1}\frac{\nu\Delta t}{2}L_{jj}\phi \tag{66}$$

Equation (59) can be written as

$$\begin{aligned}\left(1-\gamma_{k+1}\frac{\nu\Delta t}{2}L_{33}\right)u_i^* = \Delta t\left(\alpha_{k+1}\left(H_i^k+\nu\left(L_{11}+L_{22}\right)u_i^k\right)+\beta_{k+1}\left(H_i^{k-1}+\nu\left(L_{11}+L_{22}\right)u_i^{k-1}\right)\right.\\ \left. +\,\gamma_{k+1}\left(\nu L_{33}u_i^k - \partial_i p^{k-1/2}\right)\right) + \left(1-\gamma_{k+1}\frac{\nu\Delta t}{2}L_{33}\right)u_i^k\end{aligned} \tag{67}$$

In CaLES, it is split into

$$\begin{aligned}u_i^{**} = u_i^k + \Delta t\left(\alpha_{k+1}\left(H_i^k + \nu(L_{11}+L_{22})u_i^k\right) + \beta_{k+1}\left(H_i^{k-1} + \nu(L_{11}+L_{22})u_i^{k-1}\right)\right.\\ \left. +\,\gamma_{k+1}\left(\nu L_{33}u_i^k - \partial_i p^{k-1/2}\right)\right)\end{aligned} \tag{68}$$

$$\left(1-\gamma_{k+1}\frac{\nu\Delta t}{2}L_{33}\right)u_i^* = u_i^{**} - \gamma_{k+1}\frac{\nu\Delta t}{2}L_{33}u_i^k \tag{69}$$

$$p^{k+1/2} = p^{k-1/2} + \phi - \gamma_{k+1}\frac{\nu\Delta t}{2}L_{33}\phi \tag{70}$$

The variable u_i^{**} is a better approximation of the final velocity than the sum of the terms in the right-hand side of equation (65). Then, the Poisson equation can be solved to obtain ϕ 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 $L_{jj}(u_i)^*$, 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 Error Reference source not found. with 3D implicit viscous diffusion can be solved via the transform method,

$$\left(1-\beta\left(\frac{\lambda_x}{\Delta x^2}+\frac{\lambda_y}{\Delta y^2}+\delta_z^2\right)\right)\Delta\hat{\hat{u}}_{k_x,k_y,k} = \hat{\hat{R}}_{k_x,k_y,k} \tag{71}$$

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

Table 1 Boundary conditions

	Explicit RK	Hybrid RK/CN	Hybrid RK/CN (1D)
Mesh-x	Uniform	Uniform	Uniform
Mesh-y	Uniform	Uniform	Uniform
Mesh-z	Non-uniform	Non-uniform	Non-uniform
cbcvel-x	PP, DD, NN, DN	PP, DD0, N0N0, D0N0	PP, DD, NN, DN
cbcvel-y	PP, DD, NN, DN	PP, DD0, N0N0, D0N0	PP, DD, NN, DN
cbcvel-z	PP, DD, NN, DN	PP, DD, NN, DN	PP, DD, NN, DN

Table 2 Boundary conditions in CaLES

	Explicit RK	Hybrid RK/CN	Hybrid RK/CN (1D)
Mesh-x	Uniform	Uniform	Uniform
Mesh-y	Uniform	Uniform	Uniform
Mesh-z	Non-uniform	Non-uniform	Non-uniform
bcvel-x	PP, DD, NN, DN	PP, DD0	PP, DD, NN, DN
bcvel-y	PP, DD, NN, DN	PP, DD0	PP, DD, NN, DN
bcvel-z	PP, DD, NN, DN	PP, DD, NN, DN	PP, DD, NN, DN

6 LES

The filtered NS equations obtained using a uniform filter are

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0 \tag{72}$$

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u_i u_j}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j} \tag{73}$$

It is written as

$$\frac{\partial \overline{u}_i}{\partial t} + \overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{74}$$

where

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j \tag{75}$$

We need a subgrid model to compute τ_{ij}, since $\overline{u_i u_j}$ 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 a central scheme is applied, we only have numerical dispersion errors and SGS dissipation errors.

The subgrid stress is modeled as

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = 2\nu_t \overline{S}_{ij} \tag{76}$$

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

$$\begin{aligned}\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} = {} & \nu \left(\frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2} + \frac{\partial^2 \bar{u}}{\partial z^2} \right) + \frac{\partial}{\partial x}\left(\nu_t \left(\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{u}}{\partial x} \right) \right) \\ & + \frac{\partial}{\partial y}\left(\nu_t \left(\frac{\partial \bar{u}}{\partial y} + \frac{\partial \bar{v}}{\partial x} \right) \right) + \frac{\partial}{\partial z}\left(\nu_t \left(\frac{\partial \bar{u}}{\partial z} + \frac{\partial \bar{w}}{\partial x} \right) \right)\end{aligned} \tag{77}$$

$$\begin{aligned}\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} = {} & \nu \left(\frac{\partial^2 \bar{v}}{\partial x^2} + \frac{\partial^2 \bar{v}}{\partial y^2} + \frac{\partial^2 \bar{v}}{\partial z^2} \right) + \frac{\partial}{\partial x}\left(\nu_t \left(\frac{\partial \bar{v}}{\partial x} + \frac{\partial \bar{u}}{\partial y} \right) \right) \\ & + \frac{\partial}{\partial y}\left(\nu_t \left(\frac{\partial \bar{v}}{\partial y} + \frac{\partial \bar{v}}{\partial y} \right) \right) + \frac{\partial}{\partial z}\left(\nu_t \left(\frac{\partial \bar{v}}{\partial z} + \frac{\partial \bar{w}}{\partial y} \right) \right)\end{aligned} \tag{78}$$

$$\begin{aligned}\frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} = {} & \nu \left(\frac{\partial^2 \bar{w}}{\partial x^2} + \frac{\partial^2 \bar{w}}{\partial y^2} + \frac{\partial^2 \bar{w}}{\partial z^2} \right) + \frac{\partial}{\partial x}\left(\nu_t \left(\frac{\partial \bar{w}}{\partial x} + \frac{\partial \bar{u}}{\partial z} \right) \right) \\ & + \frac{\partial}{\partial y}\left(\nu_t \left(\frac{\partial \bar{w}}{\partial y} + \frac{\partial \bar{v}}{\partial z} \right) \right) + \frac{\partial}{\partial z}\left(\nu_t \left(\frac{\partial \bar{w}}{\partial z} + \frac{\partial \bar{w}}{\partial z} \right) \right)\end{aligned} \tag{79}$$

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

$$\nu_t = (C_s \Delta D(y))^2 \overline{S} \tag{80}$$

where $\overline{S} = \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}$, and van Driest damping function,

$$D(y) = 1 - \exp\left(-\frac{y^+}{25} \right) \tag{81}$$

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 (Germano et al., 1991; Pope, 2001)

$$\nu_t = c_s \Delta^2 \overline{S} \tag{82}$$

$$c_s = C_s^2 = \frac{M_{ij}L_{ij}}{M_{ij}M_{ij}} \tag{83}$$

$$L_{ij} = \widetilde{\overline{u}_i\overline{u}_j} - \widetilde{\overline{u}}_i\widetilde{\overline{u}}_j \tag{84}$$

$$M_{ij} = 2\Delta^2\overline{S}\,\overline{S}_{ij} - 2\left(\alpha\Delta\right)^2\widetilde{\overline{S}}\,\widetilde{\overline{S}}_{ij} \tag{85}$$

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 equation (74)

$$\frac{\partial\widetilde{\overline{u}}_i}{\partial t} + \widetilde{\overline{u}}_j\frac{\partial\widetilde{\overline{u}}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial\widetilde{\overline{p}}}{\partial x_i} + \nu\frac{\partial^2\widetilde{\overline{u}}}{\partial x_j\partial x_j} - \frac{\partial T_{ij}}{\partial x_j} \tag{86}$$

$$T_{ij} = \overline{u_iu_j} - \widetilde{\overline{u}}_i\widetilde{\overline{u}}_j = \widetilde{\tau}_{ij} + \overline{u}_i\overline{u}_j - \widetilde{\overline{u}}_i\widetilde{\overline{u}}_j \tag{87}$$

Consequently,

$$L_{ij} = T_{ij} - \widetilde{\tau}_{ij} = \overline{u}_i\overline{u}_j - \widetilde{\overline{u}}_i\widetilde{\overline{u}}_j \tag{88}$$

The derivation of Germano's identity assumes uniform grids. It is used to determine the coefficient of SGS models. In the following, we assume $c_s(x,t)$ 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

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2c_s\overline{\Delta}^2\overline{S}\,\overline{S}_{ij} \tag{89}$$

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

$$T_{ij} = \overline{u_iu_j} - \widetilde{\overline{u}}_i\widetilde{\overline{u}}_j = u_iu_j - \widetilde{u}_i\widetilde{u}_j \tag{90}$$

$$T_{ij} - \frac{1}{3}T_{kk}\delta_{ij} = -2c_s\widetilde{\overline{\Delta}}^2\widetilde{\overline{S}}\,\widetilde{\overline{S}}_{ij} = -2c_s\widetilde{\Delta}^2\widetilde{S}\widetilde{S}_{ij} \tag{91}$$

Substituting equations (89) and (91) into equation (88).

$$L_{ij} = -2c_s\widetilde{\overline{\Delta}}^2\widetilde{\overline{S}}\,\widetilde{\overline{S}}_{ij} + \frac{1}{3}T_{kk}\delta_{ij} + 2c_s\overline{\Delta}^2\overline{S}\,\overline{S}_{ij} - \frac{1}{3}\widetilde{\tau}_{kk}\delta_{ij} \tag{92}$$

It is further written as

$$L_{ij} - \frac{1}{3}\delta_{ij}\left(T_{kk} - \widetilde{\tau}_{kk}\right) = 2c_s\overline{\Delta}^2\overline{S}\,\overline{S}_{ij} - 2c_s\widetilde{\overline{\Delta}}^2\widetilde{\overline{S}}\,\widetilde{\overline{S}}_{ij} \tag{93}$$

Since

$$L_{kk} = T_{kk} - \widetilde{\tau}_{kk} \tag{94}$$

Equation (93) becomes

$$L_{ij} - \frac{1}{3}L_{kk}\delta_{ij} = -2c_s\widetilde{\overline{\Delta}}^2\widetilde{\overline{S}}\,\widetilde{\overline{S}}_{ij} + 2c_s\overline{\Delta}^2\overline{S}\,\overline{S}_{ij} \tag{95}$$

Define

$$M_{ij} = 2\overline{\Delta}^2\overline{S}\,\overline{S}_{ij} - 2\widetilde{\overline{\Delta}}^2\widetilde{\overline{S}}\,\widetilde{\overline{S}}_{ij} \tag{96}$$

and squared error

$$E = \left(L_{ij} - \frac{1}{3} L_{kk} \delta_{ij} - c_s M_{ij} \right)^2 \tag{97}$$

We have

$$\frac{\partial E}{\partial c_s} = -2M_{ij} \left(L_{ij} - \frac{1}{3} L_{kk} \delta_{ij} - c_s M_{ij} \right) = -2 \left(L_{ij} M_{ij} - c_s M_{ij} M_{ij} \right) \tag{98}$$

The squared error is minimized when

$$c_s = \frac{L_{ij} M_{ij}}{M_{ij} M_{ij}} \tag{99}$$

Equation (99) is proposed by Lilly (1992). The dynamic procedure uses the following assumptions:

1. Uniform Δx and Δy;
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 CaLES, 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 et al. (1995); Cabot and Moin (2000). In the following, the bar denoting the first filter is left out for brevity. In CaLES, the implementation of the dynamic model is

$$\nu_t = c_s S \tag{100}$$

$$c_s = \frac{M_{ij} L_{ij}}{M_{ij} M_{ij}} = C_s^2 \Delta^2 \tag{101}$$

$$L_{ij} = u_i u_j - \widetilde{u}_i \widetilde{u}_j \tag{102}$$

$$M_{ij} = 2 \left(S S_{ij} - \alpha^2 \widetilde{S} \widetilde{S}_{ij} \right) \tag{103}$$

where

$$\widetilde{S}_{ij} = \frac{1}{2} \left(\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i} \right) \tag{104}$$

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

$$\widetilde{S}_{ij} \neq \frac{1}{2} \left(\widetilde{\frac{\partial u_i}{\partial x_j}} + \widetilde{\frac{\partial u_j}{\partial x_i}} \right) \tag{105}$$

though we assume uniform cell size locally within a test filter size. The calculation of $\widetilde{S}_{ij}$ requires the wall boundary conditions for $\widetilde{u}_i$. It can be shown that

$$\widetilde{u}_j = \frac{1}{2} \left(\widetilde{u}_{j-1/2} + \widetilde{u}_{j+1/2} \right) = \frac{1}{4} \left(u_{j-1} + 2u_j + u_{j+1} \right) \tag{106}$$

where

$$u_j = \frac{1}{2}\left(u_{j-1/2} + u_{j+1/2}\right) \tag{107}$$

We compute the two terms in equation (102) 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

$$\frac{U}{u_\tau} = \frac{1}{\kappa}\ln\left(\frac{y u_\tau}{\nu}\right) + B \tag{108}$$

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

$$f(u_\tau) = \frac{U}{u_\tau} - \frac{1}{\kappa}\ln\left(\frac{y u_\tau}{\nu}\right) - B \tag{109}$$

So

$$f'(u_\tau) = -\frac{1}{u_\tau}\left(\frac{U}{u_\tau} + \frac{1}{\kappa}\right) \tag{110}$$

The iteration formula is

$$u_{\tau,k+1} = u_{\tau,k} - \frac{f(u_{\tau,k})}{f'(u_{\tau,k})} \tag{111}$$

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 CaLES. 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 the wall model is correct.

Appendix A

The eigenvalues are given by $\lambda_k = -4\sin^2(\theta_k)$. Table 5 and Table 6 list the transforms and inverse transforms for different combinations of boundary conditions. N is the number of input points, $0, 1, \ldots, N-1$ 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.

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

Table 3 Selected verification cases

	Re_τ	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
Turbulent channel3D	$180-10^{10}$	Explicit
Turbulent channel3D	$180-10^{10}$	Implicit 1D

Table 4 Selected validation cases

	Re	Time advancement
Square duct		Explicit

Table 5 Transforms on a staggered grid (pressure)

BC	θ_k	Transform	Inverse transform
PP		DFT	IDFT
NN	$\frac{\pi k}{2N}$	DCT-II	$\frac{1}{2N}$DCT-III
DD	$\frac{\pi(k+1)}{2N}$	DST-II	$\frac{1}{2N}$DST-III
ND	$\frac{\pi(2k+1)}{4N}$	DCT-IV	$\frac{1}{2N}$DCT-IV
DN		DST-IV	$\frac{1}{2N}$DST-IV

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

BC	θ_q	Transform	Inverse transform
PP		DFT	IDFT
NN	$\frac{\pi k}{2(N-1)}$	DCT-I	$\frac{1}{2(N-1)}$DCT-I
DD	$\frac{\pi(k+1)}{2(N+1)}$	DST-I	$\frac{1}{2(N+1)}$DST-I
ND	$\frac{\pi(2k+1)}{4N}$	DCT-III	$\frac{1}{2N}$DCT-II
DN		DST-III	$\frac{1}{2N}$DST-II

1. Staggered NN

$$X_k = 2\sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}k\left(n+\frac{1}{2}\right)\right], \quad k = 0,\dots,N-1 \tag{112}$$

$$x_n = \frac{1}{2N}\left\{X_0 + 2\sum_{k=1}^{N-1} X_k \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right]\right\}, \quad n = 0,\ldots,N-1 \tag{113}$$

2. Staggered DD

$$X_k = 2\sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N}(k+1)\left(n+\frac{1}{2}\right)\right], \quad k = 0,\ldots,N-1 \tag{114}$$

$$x_n = \frac{1}{2N}\left\{(-1)^n X_{N-1} + 2\sum_{k=0}^{N-2} X_k \sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)(k+1)\right]\right\}, \quad n = 0,\ldots,N-1 \tag{115}$$

3. Staggered ND

$$X_k = 2\sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)\right], \quad k = 0,\ldots,N-1 \tag{116}$$

$$x_n = \frac{1}{2N}\left\{2\sum_{k=0}^{N-1} X_k \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right]\right\}, \quad n = 0,\ldots,N-1 \tag{117}$$

4. Staggered DN

$$X_k = 2\sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)\right], \quad k = 0,\ldots,N-1 \tag{118}$$

$$x_n = \frac{1}{2N}\left\{2\sum_{k=0}^{N-1} X_k \sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right]\right\}, \quad n = 0,\ldots,N-1 \tag{119}$$

5. Non-staggered NN

$$X_k = x_0 + (-1)^k x_{N-1} + 2\sum_{n=1}^{N-2} x_n \cos\left(\frac{\pi}{N-1}kn\right), \quad k = 0,\ldots,N-1 \tag{120}$$

$$x_n = \frac{1}{2(N-1)}\left\{X_0 + (-1)^n x_{N-1} + 2\sum_{k=1}^{N-2} X_k \cos\left(\frac{\pi}{N-1}nk\right)\right\}, \quad n = 0,\ldots,N-1 \tag{121}$$

6. Non-staggered DD

$$X_k = 2\sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N+1}(k+1)(n+1)\right], \quad k = 0,\ldots,N-1 \tag{122}$$

$$x_n = \frac{1}{2(N+1)}\left\{2\sum_{k=0}^{N-1} X_k \sin\left[\frac{\pi}{N+1}(n+1)(k+1)\right]\right\}, \quad n = 0,\ldots,N-1 \tag{123}$$

7. Non-staggered ND

$$X_k = x_0 + 2\sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(k+\frac{1}{2}\right)n\right], \quad k = 0,\ldots,N-1 \tag{124}$$

$$x_n = \frac{1}{2N}\left\{2\sum_{k=0}^{N-1} X_k \cos\left[\frac{\pi}{N}n\left(k+\frac{1}{2}\right)\right]\right\}, \quad n = 0,\ldots,N-1 \tag{125}$$

8. Non-staggered DN

$$X_k = (-1)^k x_{N-1} + 2\sum_{n=0}^{N-2} x_n \sin\left[\frac{\pi}{N}\left(k+\frac{1}{2}\right)(n+1)\right], \quad k = 0,\ldots,N-1 \tag{126}$$

$$x_n = \frac{1}{2N}\left\{2\sum_{k=0}^{N-1} X_k \sin\left[\frac{\pi}{N}(n+1)\left(k+\frac{1}{2}\right)\right]\right\}, \quad n = 0,\ldots,N-1 \tag{127}$$

Appendix B

In LES, eddy viscosity is introduced. The hybrid RK-CN scheme (56) becomes

$$u_i^* = u_i^k + \Delta t\left(\gamma_{k+1}H_i^k + \rho_{k+1}H_i^{k-1} + \frac{\alpha_{k+1}}{2Re}L\left(u_i^* + u_i^k\right) - \alpha_{k+1}\partial_i p^{k-1/2}\right) \tag{128}$$

where operator L denotes the viscous term,

$$L(u_i) = \frac{\partial}{\partial x_j}\left(2(\nu+\nu_T)S_{ij}\right) = \frac{\partial}{\partial x_j}\left((\nu+\nu_T)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) \tag{129}$$

Re-express it using $\Delta u_i^* = u_i^* - u_i^k$,

$$\Delta u_i^* - \frac{\Delta t \alpha^{k+1}}{2Re}L(\Delta u_i^*) = \Delta t\left(\gamma_{k+1}H_i^k + \rho_{k+1}H_i^{k-1} + \frac{\alpha_{k+1}}{Re}L(u_i^k) - \alpha_{k+1}\partial_i p^{k-1/2}\right) \tag{130}$$

Let $\beta_{k+1} = \frac{\Delta t \alpha_{k+1}}{2Re}$,

$$(1-\beta_{k+1}L)(\Delta u_i^*) = \Delta t\left(\gamma_{k+1}H_i^k + \rho_{k+1}H_i^{k-1} - \alpha_{k+1}\partial_i p^{k-1/2}\right) + 2\beta_{k+1}L\left(u_i^k\right) \tag{131}$$

Expand it,

$$(1-\beta_{k+1}(L_1+L_2+L_3))\Delta u_i^* = \Delta t\left(\gamma_{k+1}H_i^k + \rho_{k+1}H_i^{k-1} - \alpha_{k+1}\partial_i p^{k-1/2}\right) + 2\beta_{k+1}L\left(u_i^k\right) \tag{132}$$

$$L_1(\Delta u_i^*) = \frac{\partial}{\partial x_1}\left((\nu+\nu_T)\left(\frac{\partial \Delta u_i^*}{\partial x_1} + \frac{\partial \Delta u_1^*}{\partial x_i}\right)\right) \tag{133}$$

$$L_2(\Delta u_i^*) = \frac{\partial}{\partial x_2}\left((\nu+\nu_T)\left(\frac{\partial \Delta u_i^*}{\partial x_2} + \frac{\partial \Delta u_2^*}{\partial x_i}\right)\right) \tag{134}$$

$$L_3(\Delta u_i^*) = \frac{\partial}{\partial x_3}\left((\nu+\nu_T)\left(\frac{\partial \Delta u_i^*}{\partial x_3} + \frac{\partial \Delta u_3^*}{\partial x_i}\right)\right) \tag{135}$$

Equation (131) 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, equation (132) becomes

$$(1-\beta_{k+1}\left(L_{11}+L_{22}+L_{33}\right))\Delta u_i^* = \Delta t\left(\gamma_{k+1}H_i^k+\rho_{k+1}H_i^{k-1}-\alpha_{k+1}\partial_i p^{k-1/2}\right)+2\beta_{k+1}L_{jj}\left(u_i^k\right) \tag{136}$$

$$L_{jj}(\Delta u_i^*) = \frac{\partial}{\partial x_j}\left((\nu+\nu_T)\frac{\partial \Delta u_i^*}{\partial x_j}\right) \tag{137}$$

References

Balaras, E., C. Benocci, and U. Piomelli (1995). Finite-difference computations of high reynolds number flows using the dynamic subgrid-scale model. *Theoretical and computational fluid dynamics 7*(3), 207–216.

Cabot, W. and P. Moin (2000). Approximate wall boundary conditions in the large-eddy simulation of high reynolds number flow. *Flow, Turbulence and Combustion 63*, 269–291.

Chorin, A. J. (1968). Numerical solution of the navier-stokes equations. *Mathematics of computation 22*(104), 745–762.

Germano, M., U. Piomelli, P. Moin, and W. H. Cabot (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics 3*(7), 1760–1765.

Harlow, F. H., J. E. Welch, et al. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of fluids 8*(12), 2182.

Kim, J. and P. Moin (1985). Application of a fractional-step method to incompressible navier-stokes equations. *Journal of computational physics 59*(2), 308–323.

Lilly, D. (1992). A proposed modification of the germano sugrid-scale closure method. *Phys Fluids A 4*, 633–635.

Pope, S. B. (2001). Turbulent flows. *Measurement Science and Technology 12*(11), 2020–2021.