Technische Universität München

Fachgebiet Hydromechanik

Compact fourth-order scheme for numerical simulations of Navier-Stokes Equations

ARPIRUK HOKPUNNA

Vollständiger Abdruck der von der Fakultät für Bauingenieur- und Vermessungswesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Die Dissertation wurde am 23.09.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingenieur- und Vermessungswesen am 10.12.2009 angenommen.

Acknowledgements

I would like to thank colleagues at Fachgebiet Hydromechanik of Technische Universität München and express my special appreciation to Prof. M. Manhart for supervising my research. His insightful suggestions have always been a guided light for my work. All of my work would not be made possible with out his constant helps and supports. I would like to express my sincere gratitude to Dr.-Ing. M. Haselbauer, Dr.-Ing. F. Schwertfirm, Dipl.-Ing. N. Peller. and Dipl.-Tech. math. C. Gobert. Discussion with them during coffee breaks have always been enjoyable moments and most of the time, come along some new interesting ideas. Many thanks to Dr.-Ing. C. Raap who always gives excellent suggestion regarding regulations and administrative requirement of the Faculty.

There are no word that could be enough to express an appreciation to my parents, Prof. Wech and Aj. Wilaiwan Hokpunna. Their love, their encouragement and the thoughtful plan in paving my basic education have shaped and equipped me with necessary quality to carried out this work. My last words of gratitude, and by no mean—least, I would like to express my indebted appreciation to my wife, Dr. med. Warangkana A. Hokpunna. Her companionship brings out the best in me.

Contents

Acknowledgements **I**

Contents **V**

List of tables **VII**

List of figures **IX**

Nomenclature **XIII**

Zusammenfassung **XVII**

Abstract **XIX**

1 Introduction **1**

1.1 Motivation 1

1.2 Contribution of this work 3

1.3 Outline 5

2 Finite Volume Discretisation of Navier-stokes on Staggered Grids **7**

2.1 Navier-Stokes equation 7

2.2 Staggered grid system 7

2.3 Discrete form of Navier-Stokes equations 9

2.3.1 Conservation of mass 9

2.3.2 Conservation of momentum 9

2.4 Numerical Approximations 10

2.4.1 Cell-centered deconvolution for the computation of mass fluxes 11

2.4.2 Cell-centered deconvolution for the pressure 11

2.4.3 Intercell deconvolution for the computation of momentum fluxes . . . 12

2.4.4 Intercell differentiation for diffusive fluxes 13

2.4.5 Convective velocities 13

2.4.6 Divergence-free convective velocities 15

2.4.7 Nonlinear correction 18

2.5 Projection method . . . 19
2.5.1 Discretisation of Poisson equation . . . 22
2.6 Boundary closures . . . 22
2.6.1 Closures for collocated variables . . . 23
2.6.2 Closures for staggered variables . . . 23
2.7 Analysis . . . 24
2.7.1 Comparative Fourier analysis of linear term . . . 24
2.7.2 Fourier analysis of nonlinear terms . . . 26
2.8 A priori testing . . . 28
2.9 Conclusion . . . 28

3 Validations 31
3.1 Taylor-Green vortex flow . . . 31
3.2 Doubly-Periodic shear layer . . . 32
3.3 Instability of plane channel flow . . . 36
3.4 Turbulent channel flow . . . 39
3.4.1 Choice of the approximation of the convective velocities . . . 40
3.4.2 Grid dependency study . . . 41
3.4.3 Comparison with the second-order scheme . . . 43
3.4.4 Effects of second-order solution of pressure and nonlinear correction . 44
3.5 Conclusion . . . 45

4 Parallel Solution to Tridiagonal Systems 51
4.1 Introduction . . . 51
4.2 Parallel algorithm solving tridiagonal systems . . . 52
4.2.1 Pre and post-processing parallel algorithms . . . 52
4.2.2 The algorithm . . . 54
4.3 Accuracy Analysis . . . 56
4.4 Complexity and performance . . . 59
4.5 Results . . . 61
4.6 Conclusion . . . 64

5 Approximate Projection Method 67
5.1 Background . . . 68
5.2 Consistent Projection methods . . . 70
5.3 Analysis of consistent Projection methods . . . 72
5.3.1 Effect of the divergence approximation . . . 72
5.3.2 Effect of pressure gradient approximation . . . 73
5.4 Approximate projection method . . . 76

5.5 Analysis of the approximate projection method 77
5.5.1 Local truncation errors 77
5.5.2 Conservation of mass and vorticity 82
5.6 Results 84
5.6.1 Doubly periodic shear layer 84
5.6.2 Turbulent channel flow 87
5.7 Conclusion 89

6 Application to Massive-scaled Simulations 91
6.1 Configuration of numerical simulations 92
6.2 Accuracy and performance of parallel compact fourth-order 92
6.2.1 Performance of Approximate projection method with strongly implicit procedure 93
6.2.2 Accuracy of the parallel fourth-order compact scheme 94
6.3 DNS of turbulent channel flow on fine grids 98
6.3.1 Comparison with spectral scheme on the same computational domain 99
6.3.2 Comparison with spectral scheme using a smaller computational domain 101
6.4 DNS of turbulent channel flow on coarse grids 107
6.4.1 DNS of turbulent channel flow on very coarse grids 108
6.5 Efficiency and scalability 109
6.6 Conclusion 111

7 Conclusion and outlook 115

Literature 121

List of Tables

2.1 Square root of L_2-norm of errors of the correction term ($\|N_{exact} - N_i\|_2$) over $kh \in [0, \pi/2]$. 27

3.1 Maximum error of the streamwise velocity and its convergence rate of T4 and DF4 convective velocities using fourth-order solution of pressure at $t = 1.2$, in doubly periodic shear layers flow. 34

3.2 Error and the convergence rate of the streamwise velocity at $t = 1.2$ using $DF4$ convective velocities and second-order solution of pressure ($\mathbf{D}_2\mathbf{G}_2$). . . . 34

3.3 Convergence study in y-direction of the instability of plane channel flow on *uniform* grid with $N_x = 64$. The table shows growth-rates of perturbations (G_p), their errors (ε_p) and the corresponding convergence rates (C_a). 37

3.4 Convergence study in y-direction of the instability of plane channel flow on *nonuniform* grid with $N_x = 64$. 38

3.5 Error of the perturbations growth rate (ε_p) relative to the growth rate on the finest grid (4.428E-03) using second-order approximation of pressure. . . . 39

3.6 Specification of numerical grids used in the grid dependency study. 40

3.7 Mean flow variables 42

3.8 CPU-seconds per time step spent in momentum equation(M) and the enforcement of continuity(P) on an Opteron 8216. 44

4.1 Computation and communication costs of the interface-splitting algorithm for solving a linear system of order n on p processors with single and γ right-hand sides. 61

4.2 Normalised error of the interface-splitting algorithm (column 2) subjected to different cut-off threshold (ε_c). The error of the differentiation (column 3) is shown for a comparison. 62

4.3 Normalised error of the interface-splitting algorithm applied to system with non-constant coefficients using different bandwidth. 62

5.1 Maximum error and convergence rate of the streamwise velocity at $t = 1.2$ subjected to different projection methods. Extra digits are given to P44 and P42. 84

5.2 Global flow parameters of the turbulent channel flow under different projection methods. 89

6.1 Numerical grid used in this study and the grid spacings at the wall and the center of the channel in term of channel half-width(H). 92
6.2 CPU-seconds per time integration on AMD opteron 8216 of the sequential fourth-order scheme with direct and iterative solvers for the pressure. 94
6.3 Recommended grid resolutions for DNS and LES from [WHS07]. 99
6.4 Summay of turbulent channel flow simulations on fine grids.. 100
6.5 Summay of turbulent channel flow simulations on coarse grids.. 108
6.6 Scalability of parallel compact scheme of T180-G1 case on commodity workstation.. 110
6.7 CPU-time per time step used in T180-G1 and the scalability of the fourth-order scheme on ALTIX 4700. 110
6.8 Comparison of CPU-time per time integration and the scalability of the fourth- and the second-order schemes in T390-G2 on ALTIX 4700. 111

List of Figures

2.1 Arrangement of variables on a nonuniform staggered grids consisting of pressure cells (solid), u-momentum (dash) and w-momentum (dotted) cells. . . 8

2.2 The stencil of compact fourth-order intercell deconvolution in Eq.(2.13). . . 13

2.3 Approximation stencil of $[\mathsf{w}]^{xy}_{is,j,k+\frac{1}{2}}$ in Eq.(2.15). The solid box is the pressure cell, the dashed box is the u-momentum cell and dotted boxes are the w-momentum cells. 14

2.4 Arrangement of the closure stencils near the Dirichlet boundary condition: (a) compact differentiation of collocated variables and (b) compact deconvolution and differentiation of staggered variables. Solid rectangles are the collocated cells(u) and dashed rectangles are staggered cells (w). Known values are shown by the circles and the arrows represent the position of the approximated surface-averaged values. 23

2.5 Comparison of the transfer function of standard second-order and higher-order schemes: second-order inter-cell deconvolution (T_{I2}), fourth-order compact inter-cell deconvolution (T_{I4}), fourth-order compact differentiation (T_{D4}), second-order cell-centered deconvolution (T_{C2}) and fourth-order cell-centered deconvolution (T_{C4}) 26

2.6 Convergence of numerical approximations applied to a Gaussian function (only even number of grid cells are plotted). T_{I2}:standard second-order intercell interpolation, T_{I4}:fourth-order compact deconvolution, T_{I6}:sixth-order compact deconvolution, T_{D4}:fourth-order compact interpolation. T_{C2}:second-order cell-centered deconvolution (mid-point rule), T_{C4}:fourth-order cell-centered deconvolution. 29

3.1 (a) Maximum errors of the velocity at $t = 10$ of the fourth-order and the second-order scheme applied to inviscid TGV with $c_1 = 0$ and $c_2 = 0$. (b) Maximum error of the fourth-order scheme applied to classical TGV(A) and convective TGV(B). 32

3.2 Contour plot of vorticity from -36 to 36. Left: second-order, Right: fourth-order (DF4). The resolution are 64^2, 128^2 and 256^2 from top to bottom. . . . 35

3.3 Time evolution of the perturbation energy on: (a) two uniform grids and (b) non-uniform grids with different stretching factor (S). All grids are computed with $N_x = 64$. On uniform grid, the resolution in the wall-normal direction of the two grids are $N_y = 64$ and 128 and the result with nonlinear correction are denoted by 64+NC and 128+NC. The result without the nonlinear correction are denoted by 64 and 128. 38

3.4 Mean profiles of streamwise velocity along wall-normal direction on grid A (a) and grid B (b). Averaged mass imbalance per unit volume on u-momentum cells of T4 (c) and DF4 (d). 41

3.5 (a) Convergence of mean streamwise velocity. (b) Mean streamwise velocity on grid M2 compared with other two spectral solutions from [KMM87] (dashed line) and [MKM99] (solid line). (c) Square root of Reynolds normal stresses on grid M2. 47

3.6 Skewness factor (a) and Flatness factor (b). 48

3.7 One-dimensional energy spectra in spanwise (a) and streamwise (b) directions of the streamwise velocity at $z^+ = 178$. 48

3.8 Comparison of the mean streamwise velocity profiles of second-order and fourth-order scheme: (a) mean streamwise velocity, (b) one-dimensional spectra of streamwise velocity in x-direction on grid F. 49

3.9 Mean streamwise velocity on grid C(a), M1(b) and M2(c) and the Reynolds shear stress on grid M2 (d). The thin solid line is the full fourth-order scheme (4M-4P-wNON). The open circle (4M-2P-wNON) is the solution using the fourth-order scheme for convective and diffusive terms but second-order approximation of divergence and pressure gradient. The plus symbol (4M-4P-woNON)is the full fourth-order schemes without the nonlinear correction. . 50

4.1 Matrix $\mathbf{Q}$ of the interface splitting algorithm and its N^k subdiagonal where $c = (k-1)m$. 54

4.2 Absolute speedup ($\mathcal{S}_\gamma$) of multiple right-hand side problem on Linux cluster. 63

4.3 Running time (a) and the scaled-efficiency (b) of the interface-splitting algorithm on Altix4700 compared to ScaLAPACK: 64

5.1 Contour plot of the divergence introduced by numerical approximations $\|T_y(ky)-T_x(kx)\|$ from Eq.(5.12): (a) second-order on collocated grids, (b) second-order on staggered grids, (c) fourth-order on collocated grids and (d) fourth-order on staggered grids. Horizontal and vertical axis are the components of the Fourier mode in x and y respectively. See (a) for numerical values of the contours. 75

5.2 Convergence of the approximate projection method compared to the solution interpolated from pseudo spectral method. 85
5.3 Error of the streamwise velocity (a) and pressure (b) of the approximate projection method (P42a) compared to the exact projection method (P44). 86
5.4 (a) Convergence of flow variables under a spatial grid refinement with fixed CFL of mixed-order projection method (Alg.2) and approximate projection method (Al.g3) towards the exact projection method (Alg.1). (b) Convergence of the pressure under a time-step refinement on $N = 64^2$ grid. 87
5.5 Cross-correlation of streamwise velocity near the wall (a) and near the center of the channel (b). 88
5.6 Time-averaged streamwise velocity profile (a) and r.m.s. of velocity (b) with the approximate projection and mixed-order projection methods. Results from the second-order projection method (P22) from chapter 3 are shown for comparison. 90

6.1 Comparison of statistics from the solutions of parallel and sequential versions of the fourth-order scheme on 128^3 grid cells. 95
6.2 Effect of the sampling procedures to the Reynolds shear stress in turbulent channel flow. 96
6.3 Effect of the nonlinear correction to the Reynolds normal stress in turbulent channel flow: solid-line:cell-averaged values and dash-line:surface-averaged values with nonlinear correction. The top pair is $< u'u' >$, the middle pair is $< v'v' >$ and the bottom pair is $< w'w' >$. 97
6.4 Comparison of probability density functions of the streamwise velocity between parallel and sequential versions of the fourth-order scheme at two positions: (a) close the wall at $z^+ = 5$ and (b) at the center of the channel. . . 102
6.5 Mean streamwise velocity profile (left) and r.m.s. of velocity fluctuations (right) of turbulent channel flow on the domain and the grids comparable with those used by spectral code[MKM99]. Top:$Re_\tau = 180$; middle:$Re_\tau = 395$; bottom:$Re_\tau = 590$. Plus symbol: u-component ; square symbol: v-component; triangle symbol: w-component. 103
6.6 Skewness factor (left) and flatness factor (right) of turbulent channel flow. Top:$Re_\tau = 180$. Middle:$Re_\tau = 395$. Bottom:$Re_\tau = 590$. 104
6.7 One-dimensional spectra of the streamwise velocity at $z^+ = 5$ (left) and at the center of the channel (right) of turbulent channel flow. Top: $Re_\tau = 180$. Middle: $Re_\tau = 395$. Bottom: $Re_\tau = 590$. 105
6.8 Mean streamwise velocity (a) and r.m.s. of velocities fluctuations(b) of T950-G4 (every two grid cells are shown). 106

6.9 One-dimensional energy spectra of the streamwise velocity of T950-G4. at $z^+ = 5$ (a) and at the center of the channel (b). 106

6.10 Mean streamwise velocity profile (left) and r.m.s. of velocity fluctuations (right) of turbulent channel flow on coarse grid. Top: $Re_\tau = 395$; middle: $Re_\tau = 590$; bottom: $Re_\tau = 950$. Plus symbol: u-component ; square: symbol: v-component; triangle symbol: w-component. 113

6.11 Mean streamwise velocity (left) and velocity fluctuations (right) normalised by the nominal parameters of $Re_\tau = 180$ (top) and 950 (bottom) on the grids twice larger than the recommended resolutions. 114

6.12 One-dimensional spectra E_{uu} of the streamwise velocity for $Re_\tau = 180$ (a) and 950 (b)on very coarse grids. 114

Nomenclature

Greek Symbols

$\alpha_1, \alpha_2, \beta_7, \beta_8$	Coefficiencts of the fourth-order inter-cell deconvolution
$\alpha_3, \alpha_4, \beta_9, \beta_{10}$	Coefficiencts of the fourth-order inter-cell differentiation
$\beta_1, \beta_2, \beta_3$	Coefficiencts of the cell-centered deconvolution for pressure cell
$\beta_4, \beta_5, \beta_6$	Coefficiencts of the cell-centered deconvolution for momentum cell
$\theta_{is,1} - \theta_{is,4}$	Coefficiencts of the divergence-free interpolation
ϵ	Machine accuracy
ε_c	Cut-off Threshold used in the interface-splitting algorithm
ρ	Density of the fluid
ρ_{12}	Cross correlation coefficient between u_1 and u_2
τ_{wall}	Shear stress at the wall
ν	Kinematic viscostiy
$\Theta(\mathbf{u}_1, \mathbf{u}_2)$	Difference of the convective and diffusive fluxes between two velocities
$\Omega_{i,j,k}$	Volume of pressure cell located at $[x, y, z] = [x(i),\ y(j),\ z(k)]$
$\Omega S_{is,j,k}$	Volume of momentum cell located at $[x, y, z] = [\frac{x(i)+x(i+1)}{2}),\ y(j),\ z(k)]$

Roman Symbols

$dt,\ \triangle t$	Time-step size
h_x, h_y, h_z	Grid spacing in x, y, z direction
$\mathbf{k}$	Vectorial wave number in the Fourier space
k	Wave number

k_1, k_2, k_3	Coefficiencts of the non-divergence-free interpolation
k_x, k_y, k_z	Components of the wave number in x, y, z direction
$\mathbf{u}$	Velocity vector
$\mathbf{u}^*$	Provisional velocity vector
u_b	Bulk flow velocity
u_τ	Friction velocity
$\widehat{u}_k$	Fourier component of u
u, v, w	Velocity in x, y, z direction
$\overline{u}, \overline{v}, \overline{w}$	Momentum in x, y, z direction
$\mathsf{u}, \mathsf{v}, \mathsf{w}$	Convective velocity in x, y, z direction
$\mathbf{x}$	Exact solution to $\mathbf{Ax} = \mathbf{b}$
$\widetilde{\mathbf{x}}$	Approximate solution to $\mathbf{Ax} = \mathbf{b}$
$\triangle x, \triangle y, \triangle z$	Grid spacing in x, y, z direction
x, y, z	Cartesian coordiantes
$\mathbf{z}^k$	Coefficient vector for computing the solution at k-th interface
$\mathbf{A}$	Tridiagonal matrix
$\mathcal{C}$	Sum of convective fluxes
$\mathbf{D}$	Discrete approximation operator of the discrete divergence $(\nabla \cdot ())$
$\mathbf{D}_2$	Second-order approximation of the discrete divergence
$\mathbf{D}_4$	Fourth-order approximation of the discrete divergence
$\mathcal{D}$	Sum of diffusive fluxes
D_x	Discrete one-dimenional approximation operator of $\frac{\partial}{\partial x}()$
$\mathbf{D}_2\mathbf{G}_2$	Discrete Laplacian of the second-order projection method
$\mathbf{D}_4\mathbf{G}_2$	Discrete Laplacian of the approximate projection method
$\mathbf{D}_4\mathbf{G}_4$	Discrete Laplacian of the fourth-order projection method

E_a	Absolute parallel efficiency
E_i	Incremental parallel efficiency
$\mathbf{G}$	Discrete approximation operator of the discrete gradient ($\nabla()$)
$\mathbf{G}_2$	Second-order approximation of the discrete gradient
$\mathbf{G}_4$	Fourth-order approximation of the discrete gradient
$\mathcal{H}(\mathbf{u})$	Sum of the convective and diffusive fluxes ($\mathbf{u}$)
$\mathbf{H}$	Discrete sum of the convective and diffusive fluxes
J	Banwidth of the truncated scalar product of $(\mathbf{z}^k)^T$
$\mathbf{N}$	Special block diagonal matrix used in interface-splitting algorithm
$\mathcal{P}$	Sum of pressure fluxes
$\mathbf{Q}$	Block diagonal matrix
Re	Reynolds number
Re_τ	Reynolds number based on friction velocity
$\mathcal{S}_1$	Absolute speedup of single r.h.s. problem
$\mathcal{S}_\gamma$	Absolute speedup of multiple r.h.s. problem
$\mathbf{T}$	Transformation matrix
$\mathbf{T}$	Strain rate tensor
T	Transfer function
$\mathsf{U}, \mathsf{V}, \mathsf{W}$	Primitive value of the convective velocity in x, y, z direction

Zusammenfassung

In dieser Dissertation wird ein kompaktes Finite-Volumen-Verfahren vierter Ordnung zur Lösung der Navier-Stokes Gleichung auf versetzten Gittern vorgestellt. Die vierte Ordnung wird durch eine Korrektur der nichtlinearen Terme und eine neuartige divergenzfreie Interpolation der Massenflüssen auf die Impulszellen erzielt. Für die Parallelisierung wird ein neuartiger *Interface-Splitting*-Algorithmus vorgestellt. Durch eine approximative Projektionsmethode wird die Lösung der Poissongleichung erheblich beschleunigt ohne an Genauigkeitsordnung zu verlieren.

Die Genauigkeit und Effizienz des parallelisierten, kompakten Schemas vierter Ordnung wird an Hand laminarer Testfälle und direkter numerischer Simulation turbulenter Kanalströmung mit Reynoldszahlen bis zu $Re_\tau = 950$ untersucht. Das benutzte System ist skalierbar und wesentlich genauer als vergleichbare Löser.

Abstract

In this dissertation, a compact fourth-order scheme for the solution of the Navier-Stokes equations on staggered grids is developed. The fourth order is ensured by a correction of the non-linear terms and a novel interpolation of the mass fluxes onto the momentum cells. The scheme is parallelised by a new interface splitting algorithm. An approximative projection method allows for an efficient solution of the pressure Poisson equation without loosing accuracy.

The accuracy and the efficiency of the parallel compact fourth-order scheme is evaluated in laminar test cases and a direct numerical simulation of turbulent channel flow up to $Re_\tau = 950$. The proposed scheme is highly scalable and can deliver accurate predictions of the first- and second-order statistics using the grid spacing twice coarser than the usual recommended values.

1 Introduction

1.1 Motivation

Over a half century, Computational Fluid Dynamics (CFD) plays a very important part helping engineers and scientists to understand the nature of turbulent flows. An accurate time-dependent numerical simulation of incompressible fluids can be obtained by Direct Numerical Simulations (DNS) which solve the discrete Navier-Stokes equations directly. DNS can be very accurate but it is extremely expensive. Physical requirement dictates that one must resolve the flow close to Kolmogorov length scale. This spatial resolution requirement is roughly rising with $O(Re^{9/4})$. Taking the number of time integration in to account, a complexity of $O(Re^3)$ is expected. This scaling restricts DNS to a low or moderate Reynolds number. We can, however, expect an accurate prediction when this grid resolution is used together with spectral scheme. When second-order scheme are used instead, but the same quality of the solution is desired, this high complexity must be multiplied by a factor of 16 which can transform a prohibitively expensive computation into an impossible computation. In order to reduce this factor, a higher-order scheme is required.

If one wish to learn only the large scale structure of the flow, a promising alternative approach to DNS is the Large Eddy Simulation (LES), in which the large-scale structures of the flow are resolved and the small-scale structures are modelled. In an essence of numerical simulations, these two approaches rely heavily on the accuracy of the spatial discretisations. A satisfactory simulation can not be obtained if the dynamics of the flow are not described in a sufficiently accurate way. Modeling effects of the small scales in an LES will not improve the overall accuracy of the solution when the numerical error was larger than the effects of the small scales [Gho96]. The numerical accuracy can be improved by increasing the numerical grid or increasing the accuracy order of the numerical approximations. The latter approach has become an active field of research recently because in three dimensional simulations, the cost of higher order is linearly proportional to second-order scheme but the number of grid point is reduced cubically. This scaling give a tremendous favour to higher-order methods over a brute force increasing the number of grid points.

Higher order approximations can be computed explicitly using Lagrange polynomial. The n-th order approximation of the m-order derivative requires at most $n + m$ abscissas. Alternatively, one can couple unknown values to the abscissas and solve a linear equation system. These implicit approximations have shorter stencils and have been called *compact scheme* by Lele and Sanjiva [Lel92]. They demonstrate the superiority of compact schemes over traditional explicit schemes and show that for intermediate wave numbers, the compact fourth-order scheme is even better than the explicit sixth-order scheme. They quantify the resolution characteristics of second and higher-order schemes and point out that for a relative error of 0.1%, the fourth-order compact differentiation requires 5 points per half-wave. Fourth-order explicit requires 8 grid points and the second-order requires 50. The explicit fourth-order requires 30% more grid points to achieve the same accuracy as the compact schemes. Therefore the compact fourth-order scheme is very attractive.

The finite-volume methods (FVM) hold a strong position in the CFD community because of their intrinsic conservation properties. Despite of the popularity of the second-order FVM, there are only a few papers addressing its developments towards higher-order. The complicated relationship of volume averaged values and surface fluxes made higher-order FVM more difficult than the finite difference (FD) counterpart. The first concept that tries to link compact schemes to FVM is presented by Gaitonde and Shang [GS97]. They present fourth- and sixth-order compact finite volume methods for linear wave phenomena. However, the so-called reconstruction procedure is needed to compute the primitive value and this costs significant computational time. A more economical approach is proposed by Kobayashi [Kob99]. He directly calculates the surface-averaged from the cell-averaged values. Explicit and implicit approximations based on the cell-averaged value up to twelfth-order are analysed. Pereira et al. [PKP01] presented a compact finite volume method for the two-dimensional Navier-Stokes equations on collocated grids. Piller and Stalio [PS04] presented a compact finite volume method on staggered grids in two dimensions. Lacor [CSM04] propose a finite volume method on arbitrary collocated structured grids and perform LES of turbulent channel flows at $Re_\tau = 180$. LES of the same flow with explicit filtering is performed in [EK05] using the spatial discretization proposed in [PKP01]. Fourth-order finite volume in cylindrical domain is developed in [SW07] and DNS of pipe flow of $Re_\tau = 360$ is performed.

Staggered grids have become a favorable arrangement over collocated grids because of the pressure decoupling problem. The pressure decoupling is not confined only in the lower order schemes. This problem is already reported in [PKP01] when using even number of cells with fourth-order scheme. This problem can be avoided by limiting ourselves to an odd number of cells which poses an undesirable limitation on grid design. Recently, staggered grids is shown to be more robust than collocated grids by Nagarajan [NLF03] in Large-eddy simulations. Thus compact finite volume on staggered grids deserves more attention.

1.2 Contribution of this work

Previous works on higher-order methods for finite-volume discretisation of Navier-Stokes equations have not yet clarified some fundamental questions of the fourth-order scheme described in the following paragraphs.

(i.) The approximation of convective velocities : On staggered arrangement, momentums and pressure are not located on the same control volumes. It is plausible that we can use the momentum flux to convect itself and the convective velocities in the other directions can be interpolated from other momentum cells. However, it is not straight forward to interpolate the momentum fluxes to the desired position without violating the mass-conservation. In second-order context, we do not have this problem because volume-averaged, surface-averaged and pointwise value are interchangeable via the midpoint rule. This is not the case in higher-order context. A simple solution to this problem is to use higher-order interpolation and accept divergence error at the level of truncation error. This approach would be acceptable in finite-difference where small conservation error is expected. In finite volume method, it is not acceptable since the conservation property is the most important reason for one to choose finite volume method over the finite difference. This issue is addressed by Verstappen and Veldman in [VV03] but their approach rely on a relatively large stencil which could result in a lower resolution characteristic. Thus it is necessary to find a proper higher-order divergence-free interpolation that is efficient and accurate.

(ii.) the treatment of non-linear terms: In finite-volume method, the approximation of the nonlinear convective fluxes $\overline{uu}$ using $\overline{u}\,\overline{u}$ is only second-order accurate unlike in finite-difference method. This problem was solved by Pereira et al. [PKP01]. Nevertheless, the reconstruction of the non-linear fluxes must be chosen wisely. It has been shown that this term is necessary to achieve higher-order accuracy in laminar flows. Its effects and importance in the simulation of the turbulent flow is never studied before. This term is very expensive. Therefore it is imperative that we must investigate it properly.

(iii.) the discretisation of pressure term: The role of the pressure discretization in higher-order methods is still a matter of controversy among researchers in this field. It has been shown in [ARM01] and [WD01] that the approximation of pressure term has to be the same order as the convective and diffusive ones. When the pressure is approximated using lower order, the overall accuracy is limited by this approximation. The time-dependent Stuart problems are used in their works. Piller and Stalio [PS04] argue that the local truncation error of the pressure is multiplied with the local truncation error of the time integration and deliver a fourth-order convergence rate, provided that both terms are at least second-order. They use the decaying Taylor-Green vortex flow to support this argument. This issue must be clarified because it is crucial to the cost of computations. The solution of the pressure can easily take more than half of the computation time. If a second-order

approximation of the Poisson equation for the pressure was sufficient, then the higher-order accuracy in the velocities can be achieved at a marginal cost. However, if a fourth-order treatment of the pressure is necessary, a 19-points stencil of the Laplacian operator must be used instead of the simple 7-point stencil.

(iv.) Efficient iterative solution for pressure: Stencils of the fourth-order Laplacians given by the projection method are three times wider than the second-order scheme. Such stencil requires a wider overlapping domains and imposes significant cost on the solution of pressure. The computation of the residual alone is already three times more expensive than the second-order projection. Accurate and efficient algorithm solving this Laplacian is needed to enable the fourth-order method to be useful in practice.

(v.) Parallelisation of tridiagonal systems: The implicitness that gives a higher resolving power to the compact schemes prevents a simple parallelisation. The compact fourth-order needs to solve a tridiagonal systems along the direction of approximation. This process is strictly two-ways sequential, one for the forward elimination and another one on the backward substitution. Therefore when the data along this line are distributed on different processors, a special implementation are needed. Classical algorithms solving tridiagonal matrices in parallel are either twice more expensive or requires frequent data transfer. Recent algorithm of Sun [Sun95] allows an efficient algorithm but when the solver are called, the caller have to had a full knowledge of grid connection which is against the cocept of domain decomposition. His algorithm requires two times of a uni-directional communication. Most of the computer architectures today are able to handle bi-directional communications and thus we still need more efficient algorithms.

The objective of this work is to answer the mentioned problems by a systematic study. The first phase of the work is dedicated to the fundamental development of the method where problem *(i.)-(iii.)* are solved. The developed algorithm is then evaluated by well known numerical benchmarks. Once the successful algorithm is developed, we proceed to the second phase where we are dealing with the problem *(iv)&(v.)*. The code is parallelised and efficient solver for higher-order discretisation of Poisson is developed.

The main achievement of this work is the highly-accurate fourth-order solver for the Navier-Stokes equations and the companion novel algorithms namely :

- Arbiatry order divergence-free interpolation
- Divergence-free approximate projection method
- Interface-splitting algorithm for parallel solution of diagonal dominant matrices

1.3 Outline

In chapter 2, the numerical approximations of each term in the Navier-Stokes equations are presented and analysed. This chapter is designed to present the development of a fourth-order method for finite volume discretisation of the Navier-Stokes Equations. A novel interpolation that preserves the discrete divergence-free property of the velocity fields on all discrete cells is presented. This method is generalised for arbitrary order of accuracy. Another fourth-order convective velocity that is not divergence-free is presented for comparison. Several choices of nonlinear corrections and the role of the pressure term are studied. The higher resolution properties of the cell-centered deconvolutions for divergence and gradient calculations are demonstrated.

In chapter 3 , the proposed scheme is evaluated. The necessity of fourth-order approximations of divergence and gradients are numerically verified. Fourier analysis shows that staggered grids can satisfy the incompressibility constraint better than collocated grids, thus retaining more accurate information in the flow field. The performance of the fourth-order scheme is carefully investigated. Despite the fact that higher-order schemes are shown to be vastly superior to second-order schemes in laminar flows by numerous authors, some recent papers report disappointing findings in the application of higher-order schemes to turbulent flows. Gullbrand[Gul00] applies fully-conservative explicit fourth-order scheme of Morinishi *et al.*[MLVM98] and Vasilyev[Vas00] to a DNS of turbulent channel flow. Knikker[Kni08] uses fully-conservative compact fourth-order scheme on the same flows. The grid resolutions used in their simulations are comparable to those used by the spectral code in [MKM99]. They both report that differences between second-order and the fourth-order schemes are negligible and significantly differ from the reference solution. Meinke *et al.*[MSKR02] comment that the sixth-order compact scheme is comparable with the second-order upwind scheme in large-eddy simulation of turbulent channel and jet flows. Shishkina and Wagner [SW07] also note a similar finding in their DNS of turbulent pipe flow but point out that the fourth-order scheme improve the third- and the fourth-order statistics. In this work we will show that in a turbulent channel flow, our fourth-order scheme can deliver a comparable result to the second-order scheme using only eight times less number of cells. The goal of this chapter is to verify the fourth-order convergence rate and carefully investigate whether those unfavourable findings will be observed in the proposed scheme.

The interface-splitting presented in Chapter 4 allows an efficient way of solving tridiagonal systems in parallel on distributed-memory machines. Factors determining the accuracy and efficiency of the algorithm are presented and the error bound is derived. The performance and scalability of the algorithm are evaluated on Gigabit cluster and ALTIX 4700.

Chapter 5 presents the divergence-free approximate projection which ensures the fourth-order accuracy of the spatial approximations without being excessively expensive. According to the projection method, we have to solve the 19-point Laplacian in order to preserve the fourth-order accuracy of the approximation used in the momentum equation. This would need three ghost cells. In second-order codes, two ghost cells are usually sufficient. Implementation of such Laplacian would need to engineer the whole code. The proposed approximate projection method presented in this chapter allows the fourth-order scheme to be added to second-order codes without modifying the number of ghost cells. The developed algorithm needs to solve only 13-point Laplacian without recomputing the divergence.

The presented numerical algorithms are combined and implemented in the MGLET code. This code has been developed over several decades and currently belong to Fachgebiet Hydromechanik, Technische Universität München. Detail information on the numerical approaches used in this code can be found in [Man04]. In chapter 6, the implementation of the parallel compact fourth-order scheme is evaluated. The parallel version is first compared to the sequential version using the turbulent channel flow $Re_\tau = 180$. The parallel version is then used to simulate the turbulent channel flow up to $Re_\tau = 950$. The grid resolutions necessary to achieve good predictions of the first- and second-order statistics are identified. The scalability of the parallelised fourth-order scheme is finally compared to the parallel version of the second-order scheme.

2 Finite Volume Discretisation of Navier-stokes on Staggered Grids

In this chapter we define the governing equations of incompressible fluids we intend to solve using a finite-volume method. We then describe our staggered grid system. Next, the numerical approximations of each term in the Navier-Stokes equations are discussed, follows by the projection method. Finally, we close the chapter by the Fourier analyses of the approximations of convolution, differentiation and the nonlinear terms.

2.1 Navier-Stokes equation

We solve the Navier-Stokes equations for incompressible flows of a Newtonian fluid in absence of external forces. The integral forms of the conservation laws of mass and momentum used for finite volume methods read as

$$\oint_A \mathbf{u} \cdot \mathbf{n} dA = 0 \tag{2.1}$$

$$\frac{\partial}{\partial t} \int_\Omega \mathbf{u} d\Omega + \oint_A (\mathbf{u}\mathbf{u}) \cdot \mathbf{n} dA = \nu \oint_A \mathbf{T} dA - \frac{1}{\rho} \oint_A p \cdot \mathbf{n} dA \tag{2.2}$$

Here, $\mathbf{u}$ defines the velocity vector, p the pressure, $\mathbf{T}$ the strain rate tensor, ρ the density and ν the kinematic viscosity of the fluid while $\mathbf{n}$ is the unit vector on dA pointing outside of the volume Ω.

2.2 Staggered grid system

On Cartesian grids, a system of staggered grids can be set up by putting collocated grid points along a real line x using a strictly increasing function $\xi(i)$, $x_i = \xi(i)$, $i = 0, ..., nx + 1$. Staggered grid points are defined by $xs_i = \frac{1}{2}(x_i + x_{i+1})$, $i = 0, ..., nx$. The control volume ΩS_i of the momentum u_{is} is defined on the closed interval $[x_i, x_{i+1}]$, likewise the control volume of the pressure cells is defined by $\Omega_i = [xs_{i-1}, xs_i]$. In this setting, half indices mark the position of the control surface of the corresponding control volumes (full indices) e.g. $xs_{i+1/2}$ is the x position of the East face of ΩS_i which is corresponding to x_{i+1}. This leads to the following mapping between two indices, $xs_{i+1/2} = x_{i+1}$ and $x_{i+1/2} = xs_i$. We explicitly call the vector of staggered grid point as xs to emphasize that $xs_i = \frac{1}{2}(\xi(i) + \xi(i+1)) \neq \xi(i + \frac{1}{2})$. This setting allows an accurate calculation of the divergence on the pressure cell because surfaces of pressure cells are placed exactly at the middle of the momentum cells.

The finite volume method describes the changes of a volume-averaged quantity by the net fluxes on the surface enclosing that control volume. These fluxes are surface-averaged quantity. In a second-order context, pointwise, cell-averaged and surface-averaged values are interchangeable because the second-order local truncation error is acceptable. In higher-order context, they are not and must be well identified. In this work, a cell-averaged value of f defined on a collocated control volume $\Omega_{i,j,k} = \triangle x_i \triangle y_j \triangle z_k$ is denoted by

$$[f]^{xyz}_{i,j,k} = \frac{1}{\Omega_{i,j,k}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{z_{k-1/2}}^{z_{k+1/2}} f(x,y,z) dx\ dy\ dz. \tag{2.3}$$

Accordingly, the cell-averaged value of an x-staggered variable g on $\Omega S_{is,j,k} = \triangle xs_i \triangle y_j \triangle z_k$ is represented by

$$[g]^{xyz}_{is,j,k} = \frac{1}{\Omega S_{is,j,k}} \int_{xs_{i-1/2}}^{xs_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{z_{k-1/2}}^{z_{k+1/2}} g(x,y,z) dx\ dy\ dz. \tag{2.4}$$

In order to make a distinction of averaged values defined on staggered grid points from the collocated ones, the s is appended to the indices. Here is stands for the i-th staggered grid point, $x = xs(i)$. For example $[u]^{xyz}_{is,j,k}$ is half-a-cell staggered from $[p]^{xyz}_{i,j,k}$ in the positive x-direction. Surface and line-averaged values can be defined in a similar way by reducing the dimension of integration to two and one respectively. For example $[p]^{yz}_{is,j,k}$ is surfaced-averaged value of p on the yz-plane located at xs_i.

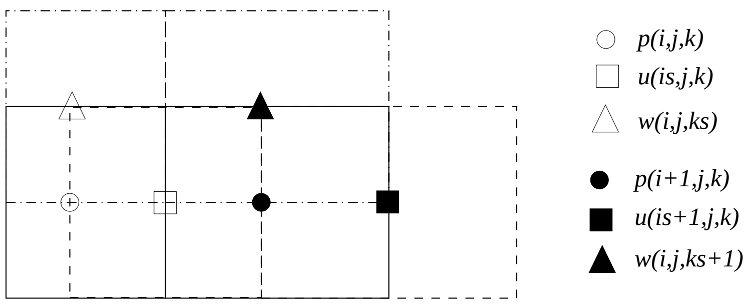


Figure 2.1: Arrangement of variables on a nonuniform staggered grids consisting of pressure cells (solid), u-momentum (dash) and w-momentum (dotted) cells.

2.3 Discrete form of Navier-Stokes equations

In this section, we introduce the finite volume discretisation of the Navier-Stokes equations and explain the necessity of the mass conservation on the momentum cells.

2.3.1 Conservation of mass

In incompressible flows, the conservation of the volume flux is equivalent to the conservation of the mass flux and they will be used interchangeably. On staggered grids, it is convenient to enforce the mass conservation on the control volumes of the pressure (Ω_i) where the divergence $[div]^{xyz}_{i,j,k}$ from these cells is given by

$$\begin{aligned}[div]^{xyz}_{i,j,k}\triangle x_i\triangle y_j\triangle z_k = &\left([u]^{yz}_{i+\frac{1}{2},j,k} - [u]^{yz}_{i-\frac{1}{2},j,k}\right)\triangle y_j\triangle z_k \\ &+\left([v]^{xz}_{i,j+\frac{1}{2},k} - [v]^{xz}_{i,j-\frac{1}{2},k}\right)\triangle x_i\triangle z_k \\ &+\left([w]^{xy}_{i,j,k+\frac{1}{2}} - [w]^{xy}_{i,j,k-\frac{1}{2}}\right)\triangle x_i\triangle y_j\end{aligned} \tag{2.5}$$

The mass conservation dictates that this divergence is zero.

2.3.2 Conservation of momentum

Equation (2.2) describes the conservation of the momentum per unit mass. On a staggered grid, we conserve the three components: $[u]^{xyz}_{is,j,k}$, $[v]^{xyz}_{i,js,k}$ and $[w]^{xyz}_{i,j,ks}$ for the momentum in x, y and z respectively. Let us consider the discrete form of the momentum equation for the first component:

$$\Omega S_{is,j,k}\frac{\partial[u]^{xyz}_{is,j,k}}{\partial t} = -\mathcal{C}_{is,j,k} + \nu\mathcal{D}_{is,j,k} - \frac{1}{\rho}\mathcal{P}_{is,j,k} \tag{2.6}$$

The terms $\mathcal{C}_{is,j,k}$, $\mathcal{D}_{is,j,k}$ and $\mathcal{P}_{is,j,k}$ are shorthand notations of the net convective, diffusive and pressure fluxes, respectively. On a Cartesian grid they are defined as follows:

$$\begin{aligned}\mathcal{C}_{is,j,k} = &\left([\mathsf{u}u]^{yz}_{is+\frac{1}{2},j,k} - [\mathsf{u}u]^{yz}_{is-\frac{1}{2},j,k}\right)\triangle y_j\triangle z_k \\ &+\left([\mathsf{v}u]^{xz}_{is,j+\frac{1}{2},k} - [\mathsf{v}u]^{xz}_{is,j-\frac{1}{2},k}\right)\triangle xs_{is}\triangle z_k \\ &+\left([\mathsf{w}u]^{xy}_{is,j,k+\frac{1}{2}} - [\mathsf{w}u]^{xy}_{is,j,k-\frac{1}{2}}\right)\triangle xs_{is}\triangle y_j,\end{aligned} \tag{2.7}$$

$$
\begin{aligned}
\mathcal{D}_{is,j,k} = &\left(\left[\frac{\partial u}{\partial x}\right]^{yz}_{is+\frac{1}{2},j,k} - \left[\frac{\partial u}{\partial x}\right]^{yz}_{is-\frac{1}{2},j,k}\right)\triangle y_j \triangle z_k \\
&+ \left(\left[\frac{\partial u}{\partial y}\right]^{xz}_{is,j+\frac{1}{2},k} - \left[\frac{\partial u}{\partial y}\right]^{xz}_{is,j-\frac{1}{2},k}\right)\triangle xs_{is} \triangle z_k \\
&+ \left(\left[\frac{\partial u}{\partial z}\right]^{xy}_{is,j,k+\frac{1}{2}} - \left[\frac{\partial u}{\partial z}\right]^{xy}_{is,j,k-\frac{1}{2}}\right)\triangle xs_{is} \triangle y_j,
\end{aligned}
\tag{2.8}
$$

$$
\begin{aligned}
\mathcal{P}_{is,j,k} = &\left([p]^{yz}_{is+\frac{1}{2},j,k} - [p]^{yz}_{is-\frac{1}{2},j,k}\right)\triangle y_j \triangle z_k \\
&+ \left([p]^{xz}_{is,j+\frac{1}{2},k} - [p]^{xz}_{is,j-\frac{1}{2},k}\right)\triangle xs_{is} \triangle z_k \\
&+ \left([p]^{xy}_{is,j,k+\frac{1}{2}} - [p]^{xy}_{is,j,k-\frac{1}{2}}\right)\triangle xs_{is} \triangle y_j.
\end{aligned}
\tag{2.9}
$$

Here we introduce a distinction of the two velocities in the convective term into convective velocities and the convected velocities. The convective velocities are denoted by Sans-serif fonts u, v and w. The convected velocities (momentum per unit mass) are denoted by Roman font u, v and w. It is important to note that the convective velocities have to be conservative, i.e. the divergence over the momentum cells has to be zero. If the convective velocities were not mass-conservative, an additional source term,

$$
[s]^{xyz}_{is,j,k} \approx [div]^{xyz}_{is,j,k}[u]^{xyz}_{is,j,k} \tag{2.10}
$$

will be added to the r.h.s of momentum equation (Eq.(2.6)). Although this source term does not affect the global conservation of the momentum due to the telescoping property of FVM, the quality of the local solution is degraded and the Galilean invariant is violated as well.

All the discrete equations in this section are exact and no simplifications or approximations have been introduced so far. Approximation errors will be introduced when these fluxes are approximated from the volume-averaged values.

2.4 Numerical Approximations

In this section, we describe, for each term in the discrete Navier-Stokes equations, how it can be approximated by a fourth-order method. First the approximation for mass flux is presented, followed by the cell-centered approximation for the pressure. Then approximations of convective and diffusive fluxes for the momentum using the compact fourth-order approximations of Kobayashi[Kob99] are briefly described. A novel approximation of divergence-free convective velocities is presented. Finally the nonlinear corrections for staggered grids are discussed.

2.4.1 Cell-centered deconvolution for the computation of mass fluxes

The enforcement of the mass conservation in Eq.(2.5) requires the approximation of the surface-averaged values at the center of the momentum cells from the volume-averaged ones, which is called deconvolution. The second-order cell-centered deconvolution (the mid-point rule) can be improved to fourth-order by using this explicit formula:

$$[u]^{yz}_{is,j,k} = \beta_1 [u]^{xyz}_{is-1,j,k} + \beta_2 [u]^{xyz}_{is,j,k} + \beta_3 [u]^{xyz}_{is+1,j,k}. \tag{2.11}$$

The coefficients of this deconvolution can be found by Taylor expansion or the method of undetermined coefficients. On uniform grids they are $\beta_1 = \beta_3 = -\frac{1}{24}$ and $\beta_2 = \frac{13}{12}$. Let $h_{is-1} = \triangle xs_{is-1}$, $h_{is} = \triangle xs_{is}$ and $h_{is+1} = \triangle xs_{is+1}$ the coefficients on non-uniform grids are given by the following formulas:

$$\beta_1 = -\frac{h_{is}^2}{4(h_{is-1}+h_{is})(h_{is-1}+h_{is}+h_{is+1})}, \qquad \beta_2 = 1-(\beta_1+\beta_3),$$
$$\beta_3 = -\frac{h_{is}^2}{4(h_{is}+h_{is+1})(h_{is-1}+h_{is}+h_{is+1})}.$$

Here we do not explicitly compute β_2 from the grid spacing, but the consistency criterion is used instead. We call this approximation *cell-centered deconvolution* because it approximates the surface-averaged values at the center of the cells from the volume-averaged ones.

2.4.2 Cell-centered deconvolution for the pressure

The fourth-order deconvolution of the pressure on the surface of the momentum cells reads

$$[p]^{yz}_{i,j,k} = \beta_4 [p]^{xyz}_{i-1,j,k} + \beta_5 [p]^{xyz}_{i,j,k} + \beta_6 [p]^{xyz}_{i+1,j,k}. \tag{2.12}$$

Note that, even if the stencil is exactly the same as in the previously introduced cell-centered deconvolution, the coefficients can be different when the grid is not uniform. This is because the deconvolved values are not lying on the center of the cells and the following coefficients must be used :

$$\beta_4 = \frac{2h_{is}^2 + h_{is}h_{is+1} - h_{is-1}3h_{is} - h_{is-1}h_{is+1}}{(h_{is} + 2h_{is-1} + h_{is-2})K_1}, \qquad \beta_5 = 1 - (\beta_4 + \beta_6),$$

$$\beta_6 = -\frac{2h_{is-1}^2 + h_{is-1}h_{is-2} - h_{is-1}3h_{is} - h_{is}h_{is-2}}{(h_{is-1} + 2h_{is} + h_{is+1})K_1},$$

$$K_1 = h_{is-2} + 2h_{is-1} + 2h_{is} + h_{is+1}.$$

This cell-centered deconvolution is fourth-order on uniform grids and third-order on non-uniform grids. Nonetheless, it is more accurate than other approximations presented here, as will be demonstrated later in section 2.7.

2.4.3 Intercell deconvolution for the computation of momentum fluxes

According to Eq.(2.7), the convected velocities u on the surface enclosing the staggered control volumes are needed. In contrast to Eq.(2.11) where the deconvoluted quantity is located at *the center of the volume-averaged one*, here the desired surface-averaged fluxes are needed at the *interfaces between momentum cells* i.e. between the volume-averaged quantities. We call this an *intercell-deconvolution*. These surface-averages of the momentum are positioned at nonstaggered grid points e.g. x_i and they can be approximated by the following fourth-order compact deconvolution [Kob99]:

$$\alpha_1[u]^{yz}_{i-1,j,k} + [u]^{yz}_{i,j,k} + \alpha_2[u]^{yz}_{i+1,j,k} = \beta_7[u]^{xyz}_{is-1,j,k} + \beta_8[u]^{xyz}_{is,j,k}. \tag{2.13}$$

The stencil of this approximation is depicted in Fig.2.2. Note that $[u]^{yz}_{i,j,k}$ here is equivalent to $[u]^{yz}_{is-\frac{1}{2},j,k}$ in Eq.(2.7). On uniform grids, $\alpha_1 = \alpha_2 = \frac{1}{4}$ and $\beta_7 = \beta_8 = \frac{3}{4}$. The coefficients on non-uniform grids are:

$$\alpha_1 = \frac{h_{is}^2}{(h_{is} + h_{is-1})^2}, \qquad \alpha_2 = \frac{h_{is-1}^2}{(h_{is} + h_{is-1})^2},$$

$$\beta_7 = \frac{2h_{is}^2(h_{is} + 2h_{is-1})}{(h_{is} + h_{is-1})^3}, \qquad \beta_8 = \frac{2h_{is}^2(h_{is-1} + 2h_{is})}{(h_{is} + h_{is-1})^3}.$$

It is possible to tune these coefficients in the Fourier space and obtain a better resolution for high wave numbers [Lel92, KL96], however, at the expense of the asymptotic convergence rate. In this work, we aim to construct a genuine fourth-order numerical scheme for the Navier-Stokes equations, therefore only formal fourth-order schemes are studied here.

2.4.4 Intercell differentiation for diffusive fluxes

The compact fourth-order approximation of the first derivative, for example $[\partial u/\partial z]^{yz}_{is+1/2,j,k}$ in Eq.(2.8), is approximated based on the same stencil used earlier in Eq.(2.13) and the differentiation formula is given by

$$\alpha_3\left[\frac{\partial u}{\partial x}\right]^{yz}_{i-1,j,k}+\left[\frac{\partial u}{\partial x}\right]^{yz}_{i,j,k}+\alpha_4\left[\frac{\partial u}{\partial x}\right]^{yz}_{i+1,j,k}=\beta_9[u]^{xyz}_{is-1,j,k}+\beta_{10}[u]^{xyz}_{is,j,k}. \tag{2.14}$$

The coefficients on non-uniform grids are

$$\begin{aligned}
\alpha_3 &= \frac{h_{is}(h^2_{is-1}+h_{is-1}h_{is}-h^2_{is})}{K}, & \alpha_4 &= \frac{(h^2_{is}+h_{is-1}h_{is}-h^2_{is-1})h_{is-1}}{K},\\
\beta_9 &= \frac{-12h_{is-1}h_{is}}{K}, & \beta_{10} &= \frac{12h_{is-1}h_{is}}{K},\\
K &= (h_{is}+h_{is-1})(h^2_{is}+3h_{is-1}h_{is}+h^2_{is-1}).
\end{aligned}$$

The fourth-order intercell deconvolution given previously is fourth-order accurate as well on nonuniform grids but the compact differentiation here is third-order on nonuniform grids. However, the transfer function of Eq.(2.14) is superior to the one of Eq.(2.13) and therefore the fourth-order convergence is not compromised as will be shown later.

2.4.5 Convective velocities

The convective fluxes in Eq.(2.7) consist of the product of convective velocity (u, v and w) with the convected velocity u, v and w. Naturally without being concerned about the mass conservation over momentum cells, one could use the deconvoluted momentum (per unit mass) $[u]^{yz}_{i,j,k}$, in Eq.(2.13) as the convective velocity $[\mathsf{u}]^{yz}_{i,j,k}$. The approximation of the remaining convective velocities e.g. $[\mathsf{v}]^{xz}_{is,js,k}$ and $[\mathsf{w}]^{xy}_{is,j,ks}$ can be done in different ways using the known cell-averaged and face-averaged velocities. Piller and Stalio [PS04] propose to use compact interpolations by first approximating the velocities on the surfaces of the pressure cell e.g. $[\mathsf{w}]^{xy}_{i,j,ks}$ and then apply compact interpolation to shift these velocities to the faces of momentum cells e.g. $[\mathsf{w}]^{xy}_{is,j,ks}$ where they can be used to convect

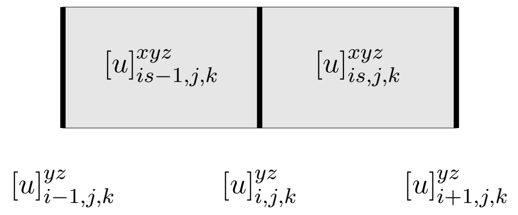


Figure 2.2: The stencil of compact fourth-order intercell deconvolution in Eq.(2.13).

the momentum fluxes. Here we present a more compact form of a fourth-order approximation. This approximation only utilizes the information on the two cells enclosing the interested surface (see Fig.2.3). The compact approximation for the convective velocity $[\mathsf{w}]^{xy}_{is,j,ks}$ reads

$$[\mathsf{w}]^{xy}_{is,j,ks} = k_1 - k_2 - k_3 + R(\triangle^4) \tag{2.15}$$

where,

$$\begin{aligned} k_1 &= \frac{6}{8}\left([w]^{xyz}_{is-\frac{1}{2},j,ks} + [w]^{xyz}_{is+\frac{1}{2},j,ks}\right) \\ k_2 &= \frac{1}{8}\left([w]^{yz}_{is+1,j,k} - 2[w]^{yz}_{is,j,k} + [w]^{yz}_{is-1,j,k}\right) \\ k_3 &= \frac{1}{8}\left([w]^{xy}_{is-\frac{1}{2},j,ks-\frac{1}{2}} + [w]^{xy}_{is+\frac{1}{2},j,ks-\frac{1}{2}} + [w]^{xy}_{is+\frac{1}{2},j,ks+\frac{1}{2}} + [w]^{xy}_{is-\frac{1}{2},j,ks+\frac{1}{2}}\right) \end{aligned}$$

The leading remainder terms are

$$R = -\frac{1}{384}\frac{\partial^4 w}{\partial x^4}\triangle x^4 - \frac{1}{192}\frac{\partial^4 w}{\partial x^2 z^2}\triangle x^2 \triangle z^2 - \frac{1}{1920}\frac{\partial^4 w}{\partial z^4}\triangle z^4.$$

This stencil is as small as the second order stencil thus it does not require extra boundary closures. The surface-averaged velocities obtained from the two deconvolutions in Eq.(2.13) and Eq.(2.15) can be used as convective velocities. These convective velocities are fourth-order accurate but not necessarily mass-conservative. We denote these convective velocities as $T4$.

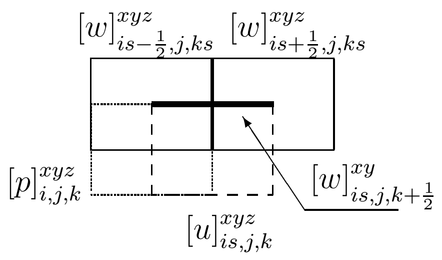

$[w]^{xyz}_{is-\frac{1}{2},j,ks}$
$[w]^{xyz}_{is+\frac{1}{2},j,ks}$
$[p]^{xyz}_{i,j,k}$
$[w]^{xy}_{is,j,k+\frac{1}{2}}$
$[u]^{xyz}_{is,j,k}$

Figure 2.3: Approximation stencil of $[\mathsf{w}]^{xy}_{is,j,k+\frac{1}{2}}$ in Eq.(2.15). The solid box is the pressure cell, the dashed box is the u-momentum cell and dotted boxes are the w-momentum cells.

2.4.6 Divergence-free convective velocities

The importance of the divergence-free property of the convective velocities has been addressed by Verstappen and Veldman [VV03]. They show that it is a necessary condition for the energy conservation. In their work, the fourth-order solution of the convective terms is not computed from a direct fourth-order approximation, but rather a Richardson extrapolation from second-order solutions of a small control volume and a larger one. A direct computation of fourth-order convective velocities is thus avoided. This approach ensures the mass and energy conservation on momentum cells but it is using a stencil width of 7-h in each direction for one momentum cell. This wide stencil can reduce the resolution properties of the scheme. In this section we propose a new divergence-free interpolation which only requires a stencil width of 4-h.

Concept

The simplest way of computing divergence-free convective velocities is to use a linear combination of the volume fluxes that are already divergence-free. This means we should compute the convective velocities from the volume fluxes over the surface of the pressure cells where the continuity is enforced. For example, the convective velocity of the u-momentum can be computed from the volume fluxes on neighbouring pressure cells sharing the same x-coordinate. The remaining difficulty is that we have to work with two directions of fluxes. In the first, the fluxes are aligned with the momentum e.g. the approximation of u for the u-momentum. In the second, the fluxes are normal to the momentum e.g. the approximation of w for the u-momentum. These fluxes are defined on different positions and when the grid was not uniform, they would require a different set of coefficients.

Now, we consider the discrete divergence written as a summation of matrix-vector multiplications:

$$D_x u + D_y v + D_z w = div. \tag{2.16}$$

Any linear transformations matrix T applied to this equation will not change the summation. This means if the same interpolation was used for all three velocities, the mass conservation will remain unchanged. Using constant coefficient is of course one of the possibilities, but this does not give a fourth-order convergence. In order to use the same interpolation, we have to convert one of the fluxes into a compatible form with the other one.

The interpolation of the fluxes aligned with the momentum can be done easily using Lagrange interpolations. Therefore we choose to convert the fluxes normal to the direction of the momentum. Inspired by the primitive value reconstruction of Gaitonde and Shang[GS97], we convert the fluxes normal to the direction of the momentum to line averaged ones such that the same interpolation can be used. To this end, we invoke the second fundamental theorem of Calculus:

$$\int_a^b f(x) = F(b) - F(a), \tag{2.17}$$

$$F = \int f(x). \tag{2.18}$$

In two dimensions, the surface-averaged value of the volume flux on the top of pressure cell and the associated line-averaged primitive are related by

$$[w]^{xy}_{i,j,ks} = \frac{1}{\triangle x_i}\left([W]^{y}_{is,j,ks} - [W]^{y}_{is-1,j,ks}\right). \tag{2.19}$$

The primitive values can be reconstructed at the top of the East and West faces of the pressure cell using this formula. We can now interpolate these values using the same method as was used for u. After that, the interpolated primitive values can be converted back to surface-averaged values using the same relationship. These two conversions are exact. However, a direct implementation of the above method is expensive. A total floating point operations of $8m$ is required for the $(2m)^{th}$-order interpolation, instead of just $4m-1$. The novelty of our approach is the elimination of these extra costs.

In what follows, we derive a method to approximate the second-order divergence-free convective velocities and generalize the method for arbitrary order of accuracy.

Second-order divergence-free convective velocities

To derive an expression for the convective velocities which is divergence-free and second-order accurate, we start from the mass conservation equation of the u-momentum cell on $\Omega S_{i,j,k}$:

$$\begin{aligned}[div]^{xyz}_{is,j,k}\triangle xs_i\triangle y_j\triangle z_k = {} & \left([\mathsf{u}]^{yz}_{i+1,j,k} - [\mathsf{u}]^{yz}_{i,j,k}\right)\triangle y_j\triangle z_k \\ & + \left([\mathsf{v}]^{xz}_{is,js,k} - [\mathsf{v}]^{xz}_{is,js-1,k}\right)\triangle xs_i\triangle z_k \\ & + \left([\mathsf{w}]^{xy}_{is,j,ks} - [\mathsf{w}]^{xy}_{is,j,ks-1}\right)\triangle xs_i\triangle y_j\end{aligned} \tag{2.20}$$

Applying the second fundamental theorem of Calculus to the above equation leads to

$$\begin{aligned}[div]^{xyz}_{is,j,k}\triangle xs_i\triangle y_j\triangle z_k = \left([\mathsf{u}]^{yz}_{i+1,j,k} - [\mathsf{u}]^{yz}_{i,j,k}\right)\triangle y_j\triangle z_k + \\ \left[\left([\mathsf{V}]^{z}_{i+1,js,k} - [\mathsf{V}]^{z}_{i,js,k}\right) - \left([\mathsf{V}]^{z}_{i+1,js-1,k} - [\mathsf{V}]^{z}_{i,js-1,k}\right)\right]\triangle z_k + \\ \left[\left([\mathsf{W}]^{y}_{i+1,j,ks} - [\mathsf{W}]^{y}_{i,j,ks}\right) - \left([\mathsf{W}]^{y}_{i+1,j,ks-1} - [\mathsf{W}]^{y}_{i,j,ks-1}\right)\right]\triangle y_j.\end{aligned} \tag{2.21}$$

The divergence of the convective velocities is expressed in terms of variables given at x_i and x_{i+1}. A desired variable f at x_i can be obtained by a second-order interpolation using the respective variables from xs_{i-1} and xs_i by the following formula

$$f_i = \gamma_{i,1} f_{is-1} + \gamma_{i,2} f_{is}, \tag{2.22}$$

where $\gamma_{i,1}$ and $\gamma_{i,2}$ are the respective interpolation coefficients. The reconstruction of the primitive values of w on the pressure cells can be started by assuming that $[\mathsf{W}]^{y}_{is-1,j,k}$ is known and the subsequent primitive values can be computed using Eq.(2.19) and Eq.(2.22). Together with Eq.(2.19) and Eq.(2.22) we obtain

$$\begin{aligned}[\mathsf{w}]^{xy}_{is,j,ks} \triangle x_{is} =& \gamma_{i+1,1}[\mathsf{W}]^{y}_{is,j,ks} + \gamma_{i+1,2}[\mathsf{W}]^{y}_{is+1,j,ks} \\ & - \gamma_{i,1}[\mathsf{W}]^{y}_{is-1,j,ks} - \gamma_{i,2}[\mathsf{W}]^{y}_{is,j,ks}.\end{aligned} \tag{2.23}$$

After regrouping of variables, the convective velocity on the top surface of $[u]^{xyz}_{is,j,k}$ is given by

$$[\mathsf{w}]^{xy}_{is,j,ks} = \theta_0 \, [\mathsf{W}]^{y}_{is-1,j,ks} + \theta_1 \, [w]^{xy}_{i,j,ks} + \theta_2 \, [w]^{xy}_{i+1,j,ks} \, , \tag{2.24}$$

with,

$$\begin{aligned}\theta_0 &= \frac{1}{\triangle x_{is}} (\gamma_{i+1,1} + \gamma_{i+1,2}) - (\gamma_{i,1} + \gamma_{i,2})), \\ \theta_1 &= \frac{\triangle x_i}{\triangle x_{is}} \left((\gamma_{i+1,1} + \gamma_{i+1,2}) - \gamma_{i,2} \right). \\ \theta_2 &= \frac{\gamma_{i+1,2} \triangle x_{i+1}}{\triangle x_{is}}.\end{aligned}$$

The coefficient of the unknown primitive value, $[\mathsf{W}]^{y}_{is-1,j,k}$ is reduced to a difference between the sum of two sets of the interpolation coefficients. The consistency dictates that the sum of any set of interpolation coefficients is equal to unity thus $\theta_0 = 0$ and $[\mathsf{W}]^{y}_{is-1,j,k}$ can be removed from the interpolation. This leads to a convenient way of computing convective velocities using the new set of interpolation coefficients, θ. In this formulation, the construction of primitive values and the back transformation are fully avoided.

The net volume flux leaving the control volume of $\Omega_{is,j,k}$ under the second-order divergence-free interpolation is

$$[div]^{xyz}_{is,j,k} = \gamma_{i,1}[div]^{xyz}_{i,j,k} + \gamma_{i+1,2}[div]^{xyz}_{i+1,j,k}. \tag{2.25}$$

This equation indicates that the imbalance of mass fluxes at the momentum cell is of the same order of magnitude as the one enforced at the pressure cells.

Fourth-order divergence-free convective velocities

A fourth-order Lagrange interpolation formula for the convective velocity $[\mathsf{u}]^{yz}_{i,j,k}$ at the cell face in x-direction of the u momentum can be obtained by

$$[\mathsf{u}]^{yz}_{i,j,k} = \gamma_{i,1}[\mathsf{u}]^{yz}_{is-2,j,k} + \gamma_{i,2}[\mathsf{u}]^{yz}_{is-1,j,k} + \gamma_{i,3}[\mathsf{u}]^{yz}_{is,j,k} + \gamma_{i,4}[\mathsf{u}]^{yz}_{is+1,j,k} \tag{2.26}$$

The coefficients of this interpolation are the same as the ones for a fourth-order Lagrange interpolation of pointwise values. We can proceed with the similar procedure as in the second-order divergence-free interpolation and arrive at

$$[\mathsf{w}]^{xy}_{is,j,ks} = \theta_{is,1}[\mathsf{w}]^{xy}_{i-1,j,ks} + \theta_{is,2}[\mathsf{w}]^{xy}_{i,j,ks} + \theta_{is,3}[\mathsf{w}]^{xy}_{i+1,j,ks} + \theta_{is,4}[\mathsf{w}]^{xy}_{i+2,j,ks}. \tag{2.27}$$

We call this convective velocity $DF4$. On uniform grids, the two sets of interpolating coefficients $\gamma_{i,1-4}$ and $\theta_{is,1-4}$ are $[\frac{-1}{16}, \frac{9}{16}, \frac{9}{16}, \frac{-1}{16}]$, in numerical order. The divergence-free interpolation of the convective velocities can be generalised for arbitrary order. Suppose that a $(2m)^{th}$-order Lagrange interpolation is used instead of Eq.(2.26), then the $(2m)^{th}$-order divergence-free interpolation of the convective velocity on the top surface of the u-momentum cell is given by

$$[\mathsf{w}]^{xy}_{is,j,ks} = \sum_{l=1}^{2m} \left(\theta_{is,l} \, [\mathsf{w}]^{xy}_{i-m+l,j,ks} \right) \tag{2.28}$$

with,

$$\theta_{is,l} = \left(\sum_{j=l}^{2m} \gamma_{i+1,j} - \sum_{j=l+1}^{2m} \gamma_{i,j} \right) \frac{\triangle x_{i-m+l}}{\triangle x_{is}} \tag{2.29}$$

Identical coefficients are used for v. This higher-order divergence-free interpolation can be applied for any position in the field.

2.4.7 Nonlinear correction

The convective term $[\mathsf{u}_j u_i]$ is the origin of the nonlinearity in the Navier-Stokes equations. This term is responsible for energy transfer between different scales and wave components smaller than the Nyquist limit can be created by it. In finite difference methods the product of u_j and u_i is the exact value of $\mathsf{u}_j u_i$. On the other hand, the approximation of the nonlinear term in finite volume methods using $[\mathsf{u}_j u_i] = [\mathsf{u}_j][u_i]$ is only second-order accurate. Additional operations are needed to achieve higher order accuracy in finite volume methods. In this work we use variants of the nonlinear correction approach proposed by Pereira *et al.*[PKP01].

A fourth-order accurate approximation of a nonlinear term can be obtained by adding some corrections to the second-order approximation:

$$[fg]^{yz} = [f]^{yz}[g]^{yz} + \frac{\triangle y^2}{12}\frac{\partial f}{\partial y}\frac{\partial g}{\partial y} + \frac{\triangle z^2}{12}\frac{\partial f}{\partial z}\frac{\partial g}{\partial z} + O\left(\triangle y^4, \triangle z^4\right). \tag{2.30}$$

The original formula of [PKP01] computes the correction term from the cell-averaged values. First the second-order interpolation is used to compute the face-averaged values then the computed values are used for the approximation of the first-derivative. Here, we use the surface-averaged values which are readily available from the approximation of the momentum fluxes (Eq.(2.13)). The use of the surface-averaged values allows a cheaper computation and better resolution characteristics. The fourth-order approximation for the convection term on the East face of u-momentum is given by:

$$\begin{aligned} [\mathsf{u}u]^{yz}_{i+1,j,k} = [\mathsf{u}]^{yz}_{i+1,j,k}[u]^{yz}_{i+1,j,k} &+ \frac{1}{48}\Big([u]^{yz}_{i+1,j+1,k} - [u]^{yz}_{i+1,j-1,k}\Big)^2 \\ &+ \frac{1}{48}\Big([u]^{yz}_{i+1,j,k+1} - [u]^{yz}_{i+1,j,k-1}\Big)^2. \end{aligned} \tag{2.31}$$

On the top face we use the following formula

$$\begin{aligned} [\mathsf{w}u]^{xy}_{is,j,ks} = {} & [\mathsf{w}]^{xy}_{is,j,ks}[u]^{xy}_{is,j,ks} \\ & + \frac{1}{24}\Big([u]^{xy}_{is+1,j,ks} - [u]^{xy}_{is-1,j,ks}\Big)\Big([w]^{xyz}_{i+1,j,ks} - [w]^{xyz}_{i,j,ks}\Big) \\ & + \frac{1}{24}\Big([u]^{xy}_{is,j+1,k} - [u]^{xy}_{is,j-1,k}\Big)\Big([w]^{xz}_{i,js,k} - [w]^{xz}_{i,js-1,k}\Big). \end{aligned} \tag{2.32}$$

The proposed forms above are one of many possibilities. In section 2.7.2 we consider some other possible forms of nonlinear corrections. However, the proposed forms here are the most accurate.

2.5 Projection method

When the momentum equation is integrated in time, the new velocity fields are not necessarily divergence-free and thus usually called provisional velocities. One of the most successful approaches ensuring the mass-conservation after the time integration is the fractional time step method (FTSM) which operates on these provisional velocities. Traditionally there are two classes of the FTSM namely *pressure-Poisson* and *projection methods*[BCM01]. The pressure-Poisson method does not truly solve for the pressure itself, instead, it solves for a pressure-like variable ϕ which is a Lagrange multiplier for a divergence-free velocity field that is closest to the provisional velocity field. On the other hand, projection methods solve for the divergence-free field which has the same vorticity as the provisional velocity.

Both approaches have to solve a Poisson equation, but with a different form of the discrete Laplacians.

Consider the explicit Euler time integration of u-momentum. Let $\mathbf{u}^n$ be the velocity and $\mathbf{H}^n$ be the contribution from convective and diffusive terms at time t_n. Let $\mathbf{u}^*$ be the provisional velocity evaluated without the pressure term and $\mathbf{u}^{n+1}$ be the divergence-free velocity field at the new time step when a suitable p is used. The equations for $\mathbf{u}^*$ and $\mathbf{u}^{n+1}$ are shown below.

$$\mathbf{u}^* = \mathbf{u}^n + dt\mathbf{H}^n \tag{2.33}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + dt\mathbf{H}^n - \frac{dt}{\rho}\nabla p \tag{2.34}$$

The divergence of the difference between (2.33) and (2.34) gives the Poisson equation for the pressure,

$$\nabla \cdot \nabla p = \frac{\rho}{dt}\nabla \cdot \mathbf{u}^*. \tag{2.35}$$

This equation is identical to the one obtained from taking the divergence of the momentum equation. Thus the pressure found in (2.35) is essentially the pressure at time t_n. Once the solution of pressure is obtained, the divergence-free velocity field can be recovered by

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{dt}{\rho}\nabla p. \tag{2.36}$$

The new velocity is divergence free and its vorticity is equal to that of the provisional velocity because,

$$\nabla \times \mathbf{u}^{n+1} = \nabla \times \mathbf{u}^* - \frac{dt}{\rho}\nabla \times \nabla p = \nabla \times \mathbf{u}^* \tag{2.37}$$

In this derivation, the projection method and the pressure-Poisson method are essentially the same. They will go separate ways when the approximations of gradient and divergence are introduced. Suppose discrete divergence operators $\mathbf{D}$ and $\mathbf{G}$ are used to approximate the divergence and gradient, respectively. Then the discrete form of equation (2.35) is

$$\mathbf{DG}p = \frac{\rho}{dt}\mathbf{Du}^*. \tag{2.38}$$

The projection method adheres to this derivation and the discrete Laplacian is given by $\mathbf{L} = \mathbf{DG}$. The Laplacian in a pressure-Poisson formulation represents the minimization which is not related to the Navier-Stokes equations and thus any discrete Laplacian will suffice. Solving Eq.(2.38) by a direct method and correcting the velocity using the respective discrete gradient will result in a machine accurate divergence. Using a Laplacian operator other than this one will leave a significant divergence in the velocity fields, even when solved with a direct method. When the Poisson equation is solved by iterative methods, pressure-Poisson formulations need to recalculate the divergence and then start the iteration again. On the other hand, projection methods only need to compute the divergence once. Therefore the projection method offers a clear computational advantage over the pressure-Poisson formulation when aiming at small mass-conservation errors. The projection operator deriving the method's name is defined as

$$\mathbf{P} = \mathbf{I} - \mathbf{G}(\mathbf{D}\mathbf{G})^{-1}\mathbf{D}.$$

The derivation shown here is equivalent to the Hemholtz-Hodge decomposition in [Den03] which states that any vector field $\mathbf{u}^*$ can be uniquely decomposed into two orthogonal fields: $\mathbf{u}$ a divergence-free field and ∇p a gradient of a potential field.

However, it should be noted that the projection method does not conserve the L_2-norm of the provisional velocity. Let $(\,\cdot\,,\,\cdot\,)$ be a scalar product of two vectors, the L_2-norm of the Helmholtz-Hodge decomposition is given by

$$(\mathbf{u}^*, \mathbf{u}^*) = (\mathbf{u}, \mathbf{u}) + (\nabla p, \nabla p) + 2(\nabla p, \mathbf{u}).$$

Because the two components are orthogonal, the third term on the r.h.s vanishes and we have

$$\begin{aligned}&(\mathbf{u}^*, \mathbf{u}^*) = (\mathbf{u}, \mathbf{u}) + (\nabla p, \nabla p),\\ &(\mathbf{u}^*, \mathbf{u}^*) \geq (\mathbf{u}, \mathbf{u}).\end{aligned}$$

If the divergence-free field $\mathbf{u}$ were to have the same L_2-norm as $\mathbf{u}^*$, the ∇p and $\mathbf{u}$ can not be orthogonal which is against the underlying concept of projection methods. This equation indicates that the energy is strictly decreasing, when the provisional velocity was not divergence-free. This fact is used by Chorin[Cho68] to show the stability of the projection method. Therefore the projection method is stable, but not energy-conserving. Even if the numerical scheme for the momentum equation was energy conserving, a reduction in L_2 norm of the momentum can be expected.

In fourth-order context, we have the freedom to use second-order or fourth-order approximations for $\mathbf{D}$ and $\mathbf{G}$. This leads to four possible choices of the Laplacian namely (i) $\mathbf{D}_2\mathbf{G}_2$, (ii) $\mathbf{D}_2\mathbf{G}_4$, (iii) $\mathbf{D}_4\mathbf{G}_2$ and (iv) $\mathbf{D}_4\mathbf{G}_4$. The first and the fourth Laplacian are formal second-order and fourth-order, respectively. The other two are non-formal. In an existing second-order code, the second-order projection method ($\mathbf{D}_2\mathbf{G}_2$) is usually implemented. On staggered grids this $\mathbf{D}_2\mathbf{G}_2$ is a well known 7-points Laplacian which can be solved in a very efficient way. The most important question here is whether $\mathbf{D}_2\mathbf{G}_2$ is sufficient to deliver a fourth-order accurate solution of the velocities. In this chapter we restrict the study to the two formal Laplacians. In-depth investigation of these four Laplacians will be presented in chapter 5.

2.5.1 Discretisation of Poisson equation

On uniform grids, the component in x direction of the fourth-order Laplace operator given by the projection method reads

$$\frac{\partial^2 [p]^{xyz}_{i,j,k}}{\partial x^2} = \frac{[p]^{xyz}_{i\pm3,j,k} - 54\,[p]^{xyz}_{i\pm2,j,k} + 783\,[p]^{xyz}_{i\pm1,j,k} - 1460\,[p]^{xyz}_{i,j,k}}{576}. \tag{2.39}$$

On nonuniform grids it is convenient to construct the Laplacian from the matrices D_{4x} and G_{4x} which are the approximations of the cell-averaged values of divergence and gradient, respectively. They are given by

$$D_{4x} = \frac{I_c(is) - I_c(is-1)}{xs(is) - xs(is-1)} \quad \text{and} \quad G_{4x} = \frac{I'_c(i+1) - I'_c(i-1)}{x(i+1) - x(i)},$$

where I_c and I'_c are the cell-centered deconvolutions defined in Eq.(2.11) and Eq.(2.12), respectively. The consistent Laplacian operator in x-direction is simply given by $L_{4x} = D_{4x}G_{4x}$ and the three-dimensional Laplacian is

$$\mathbf{L}_4 = D_{4x}G_{4x} + D_{4y}G_{4y} + D_{4z}G_{4z}. \tag{2.40}$$

In our code, we use fast Fourier transformations in the xy-plane and Gaussian elimination in z direction. After solving the Poisson equation, the divergence-free velocity is recovered by Eq.(2.36) with the respective discrete gradient.

2.6 Boundary closures

Near the boundary we cannot use all of the formula proposed earlier due to lack of information outside the domain. Either asymmetric stencils, a certain kind of extrapolation or lower-order stencils must be used. Carpenter et al.[GA93] pointed out that high-order closures might be unstable in a finite difference context. For example, the fourth-order scheme must be closed with third-order closure to ensure stability. In finite volume context, Kobayashi[Kob99] reported that the fourth-order closure is necessary for a fourth-order global accuracy. Neumann boundary conditions are intensively studied in his work. In this section we consider the treatment of the solid surface as a Dirichlet boundary condition depicted in Fig.2.4. Approximation stencils near solid surfaces should not extend too much into the inner domain because the strong differences in the velocity gradient near the wall and in the inner domain decrease the accuracy. The compact fourth-order schemes require only the two nearest cells at the boundary thus it is less sensitive to this problem compared to the explicit fourth-order scheme.

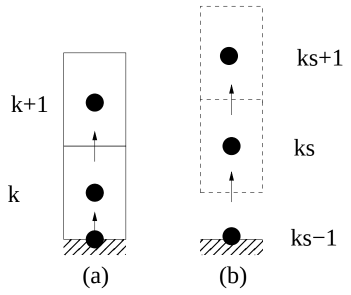


Figure 2.4: Arrangement of the closure stencils near the Dirichlet boundary condition: (a) compact differentiation of collocated variables and (b) compact deconvolution and differentiation of staggered variables. Solid rectangles are the collocated cells(u) and dashed rectangles are staggered cells (w). Known values are shown by the circles and the arrows represent the position of the approximated surface-averaged values.

2.6.1 Closures for collocated variables

For the deconvolution of collocated variables at the boundary (Fig.2.4a), the value given by the boundary can be used directly in the deconvolution formula of the inner domain (Eq.(2.13)). The boundary value can be moved to the right-hand-side and the resulting linear system can be solved by the Thomas algorithm. The only closure needed here is for the differentiation (Eq.(2.14)). In our work we use a third-order closure at the boundary:

$$\left[\frac{\partial u}{\partial z}\right]^{xy}_{is,j,0s} + 2\left[\frac{\partial u}{\partial z}\right]^{xy}_{is,j,1s} = \frac{3}{2h_z}\left([u]^{xyz}_{is,j,1} + [u]^{xyz}_{is,j,2}\right) - \frac{3}{h_z}[u]^{xy}_{wall} \tag{2.41}$$

This closure has the same convergence rate as the differentiation in the inner domain when the grid is not uniform. Thus using this third-order differentiation here does not degrade the global accuracy.

2.6.2 Closures for staggered variables

The first staggered cell in the domain is half-a-cell far from the boundary (Fig.2.4b) and the inter-cell deconvolution in Eq.(2.13) is closed by the following fourth-order approximation

$$[w]^{xy}_{i,j,1} + \frac{19}{21}[w]^{xy}_{i,j,2} = \frac{11}{7}[w]^{xyz}_{i,j,1s} + \frac{1}{7}[w]^{xyz}_{i,j,2s} + \frac{4}{21}[w]^{xy}_{wall}. \tag{2.42}$$

The third-order closure for the differentiation in Eq.(2.14) is

$$\left[\frac{\partial w}{\partial z}\right]^{xy}_{i,j,1} - \frac{11}{13}\left[\frac{\partial w}{\partial z}\right]^{xy}_{i,j,2} = \frac{36}{23h}[w]^{xyz}_{i,j,1s} - \frac{12}{23h}[w]^{xyz}_{i,j,2s} - \frac{24}{23h}[w]^{xy}_{i,j,wall}. \tag{2.43}$$

It is noteworthy that using deconvolved values for the differentiation is not recommended even though its asymptotic errors are fourth-order. The n^{th} order leading truncation term of the deconvolution transfers into $(n-1)^{th}$ order for the differentiation thus using them for the differentiation does not improve the accuracy.

In the cell-centered deconvolution and the approximation of convective velocities, we simply set the velocity to the wall value, for example zero in the case of a no-slip wall. The pressure cell within the wall is assumed to be equal to the pressure at the first cell in the domain. This ensures a second-order accurate enforcement of $\frac{\partial p}{\partial n} = 0$ at the wall. A similar extrapolation is also used by Verstaapen and Veldman in [VV03]. These treatments are sufficient for the fourth-order convergence which will be shown numerically in the next chapter.

2.7 Analysis

The accuracy of the NSE solver is determined by every single approximation step in the code. It is important to understand how large the errors are being generated in each term. Fourier analysis provides us with a quantitative error for each wave number. The convective and diffusive terms used in this work have been studied already in [Kob99] and [PS04]. In this section we perform a comparative study of numerical errors in the Fourier space. The fourth-order compact deconvolution, compact differentiation, and cell-centered deconvolution are compared.

2.7.1 Comparative Fourier analysis of linear term

In finite difference context, Fourier analysis of the discretisations of convective and diffusive terms lead to the study of the modified wave number k^* and the modified k^{2*}, respectively. In finite volume context, where the PDEs are integrated, the quantifications of the accuracy in Fourier space leads to the modified amplitude and modified wave number. These two quantities provide us a great deal of information. However, they can not be directly compared.

A better measurement for comparison is the transfer function where the approximated value is normalized by the exact one. In this section we use the concept of transfer function to compare each approximation in the momentum equation as well as the mass conservation equation.

In order to perform a Fourier analysis of a periodic function $u(x)$ over the domain $[0, L]$, the function $u(x)$ is decomposed into its respective Fourier components. We use a scaled wave number, $kh \in [0, \pi]$, similar to [Lel92] and each Fourier component is given by $\widehat{u}_k exp(ikh)$. The model equation we consider here is the one-dimensional transport equation in a periodic domain described by

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \Gamma \frac{\partial^2 u}{\partial x^2}. \tag{2.44}$$

The finite volume discretisation of the above equation is

$$\frac{1}{\Omega}\frac{\partial \overline{u}_i}{\partial t} + c\left[\widetilde{u}_{i+1/2} - \widetilde{u}_{i-1/2}\right] = \Gamma\left[\widetilde{\frac{\partial u}{\partial x}}|_{i+1/2} - \widetilde{\frac{\partial u}{\partial x}}|_{i-1/2}\right]. \tag{2.45}$$

In this one-dimensional problem we use the overbar to represent cell-averaged values and the approximations are represented by tilde symbol. Projecting the above equation in the Fourier space, we obtain the following equation for each wave-number k:

$$\frac{\partial \widehat{\overline{u}}_k}{\partial t} + ickT_I(k)\widehat{\overline{u}}_k = -\Gamma k^2 T_D(k)\widehat{\overline{u}}_k \tag{2.46}$$

The transfer function of an approximation is defined by the ratio of the approximated value over the exact value, for example $T_I(k) = \widehat{\widetilde{u}}_k/\widehat{u}_k$ defines the transfer function of a deconvolution. The transfer function of the differentiation is defined equivalently. The overall accuracy of a numerical solution of the transport equation is determined by T_I and T_D . For the purpose of a general analysis, let $c = \Gamma = 1$ such that only errors of the approximations are considered.

The transfer functions of the cell-centered deconvolution Eq.(2.11), inter-cell deconvolution (Eq.(2.13)) and the differentiation (Eq.(2.14)) are plotted in Fig.2.5. The significant improvement of the compact fourth-order deconvolution over the second-order is clearly shown. During the projection step, the mass fluxes are computed by the cell-centered deconvolution (Eq.(2.11)). According to Fig.2.5, the second-order cell-centered deconvolution (mid-point rule) is less accurate than the fourth-order inter-cell deconvolution (Eq.(2.13)), especially for $0.5 < kh < 2$. When the mid-point rule was used to enforce the mass conservation, the errors of the wave components in this range will remain in the velocity fields and thus degrade the level of accuracy that was achieved by the compact fourth-order. The fourth-order cell-centered deconvolution (Eq.(2.13)) is more accurate than the compact fourth-order inter-cell

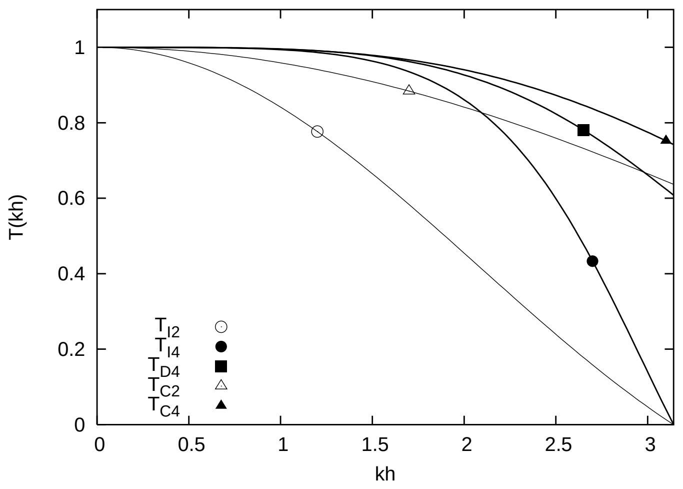


Figure 2.5: Comparison of the transfer function of standard second-order and higher-order schemes: second-order inter-cell deconvolution (T_{I2}), fourth-order compact inter-cell deconvolution (T_{I4}), fourth-order compact differentiation (T_{D4}), second-order cell-centered deconvolution (T_{C2}) and fourth-order cell-centered deconvolution (T_{C4})

deconvolution through out the Fourier space. Thus we do not need compact deconvolution for the approximations of mass and pressure fluxes. This finding has an important consequence because these two explicit stencils(Eq.(2.11) and Eq.(2.12)) lead to a narrow banded Laplacian operator which can be solved much easier than a full Laplacian operator which would arise from an implicit scheme for D or G.

The discussed cell-centered deconvolutions are used for the mass conservation and the pressure gradient in staggered grid arrangement. However, in a collocated grid arrangement, the inter-cell deconvolution has to be used for these two tasks. According to Fig.2.5, we could expect a more accurate mass conservation on staggered grids than the collocated ones. With this analysis, we can explain why the second-order solution of pressure on collocated grids strictly limits the accuracy to second-order as reported in [ARM01]. This limitation is, however, less severe on staggered grids. It will be shown later that, on staggered grids a convergence rate of approximately third-order can be achieved with the second-order solution of pressure.

2.7.2 Fourier analysis of nonlinear terms

Let us consider the fourth-order approximation of the nonlinear convective flux on the East surface of the u-momentum :

$$[\mathsf{u}u]^{yz}_{i+1,j,k} = C2 + N_i \tag{2.47}$$

where $C2 = [\mathsf{u}]^{yz}_{i+1,j,k}[u]^{yz}_{i+1,j,k}$, the second-order approximation of the nonlinear convective term. There are several straight forward methods which can be used to compute the nonlinear correction, N_i. In this study we consider three forms for the correction term $\frac{\triangle z^2}{12}\frac{\partial f}{\partial z}\frac{\partial g}{\partial z}$ in Eq.(2.30):

$$N_1 = \frac{1}{48}\left([u]^{yz}_{i+1,j,k+1} - [u]^{yz}_{i+1,j,k-1}\right)^2 \tag{2.48}$$

$$N_2 = \frac{1}{192}\left(\left([u]^{xyz}_{is,j,k+1} + [u]^{xyz}_{is+1,j,k+1}\right) - \left([u]^{xyz}_{is,j,k-1} + [u]^{xyz}_{is+1,j,k-1}\right)\right)^2 \tag{2.49}$$

$$N_3 = \frac{1}{192}\left(\left[\frac{\partial u}{\partial z}\right]^{xy}_{is,j,ks} + \left[\frac{\partial u}{\partial z}\right]^{xy}_{is+1,j,ks} + \left[\frac{\partial u}{\partial z}\right]^{xy}_{is,j,ks+1} + \left[\frac{\partial u}{\partial z}\right]^{xy}_{is+1,j,ks+1}\right)^2 \tag{2.50}$$

The first correction uses surface-averaged values, the second equation uses cell-averaged values and the third equation averages the first-derivatives provided by the compact differentiation. The nonlinear correction term is analysed using an ansatz function $u(x,z) = exp((i+0.4)kx + \alpha z)$. This function mimics an oscillating velocity under an exponential gradient in z-direction and its amplitude is varied in x-direction. The cell-averaged values here are treated as exact and surface-averaged values used for $C2$ are fourth-order accurate. The interpolated terms are computed by multiplying the analytical value with the modified amplitude of this ansatz function.

Instead of looking at the whole nonlinear convective term, we consider here just the transfer function of the correction term. The norm of the error, $L_2(|1 - T(N_i)|)$ over $kh \in [0, \pi/2]$ is shown in Tab.2.1. The first row is the norm of the analytical correction, and at the same time the error when we do not apply any correction. According to the table, nonlinear corrections improve the accuracy when the gradient is not too high and they are able to predict roughly two digits of the correction term. The correction using face-averaged values (N_1) is more accurate than the one using cell-average values (N_2). This is attributed to the inferior transfer functions of second-order approximations. The third form (N_3) is slightly more accurate than the second form(N_2) at the lowest gradient, but it performs poorly otherwise. Thus computing the nonlinear correction using face-averaged values is recommended.

Nonlinear correction form	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 1.5$
Without correction	2.7E-2	1.1E-1	2.7E-1
N_1	2.2E-3	4.4E-2	2.9E-1
N_2	3.9E-3	5.3E-2	3.2E-1
N_3	3.3E-3	8.4E-2	4.3E-1

Table 2.1: Square root of L_2-norm of errors of the correction term ($\|N_{exact} - N_i\|_2$) over $kh \in [0, \pi/2]$.

2.8 A priori testing

A simple Gaussian function $u(x) = 0.5exp(-ln(2)\frac{x^2}{9})$ as proposed in [HRT95] is used as our test function. The domain is set to $[-20, 20]$ such that errors at both ends do not affect the global accuracy. We initialize the cell-averaged values analytically and numerical approximations are computed from these cell-averaged values. In this test we consider fourth-order deconvolutions (Eq. (2.11) and (2.13)), the fourth-order differentiation (Eq.(2.14)) and the fourth-order nonlinear correction with Eq.(2.48). The errors of these approximations are investigated and plotted in Fig.2.6 in which second and compact sixth-order deconvolutions are presented for comparison. In this test case, the error graphs of every scheme are oscillating because they are strongly dependent on the sampling position. Therefore only the error on even number of cells are plotted for clarity. According to the figure, the error of the second-order inter-cell deconvolution ($I2$) is roughly four times larger than that of the cell-centered($C2$). The fourth-order inter-cell approximations ($I4, D4$) are comparable to the cell-centered deconvolution ($C4$) on large number of grid cells. The cell-centered deconvolution ($C4$) is the most accurate in general among the fourth-order approximations. The sixth-order compact scheme ($I6$) is better than the compact fourth-order scheme ($I4$). However, the compact sixth-order scheme($I6$) (from [Kob99]) is only significantly better than $C4$ only when $n > 20$. At this resolution, approximately 8 cells are used to represent the Gaussian bump ($x \in [-8, 8]$).

2.9 Conclusion

We have presented a fourth-order finite volume method using compact schemes for transported momentum and a divergence-free convective velocity. The accuracy of spatial approximations was studied by Fourier analysis and a priori testing. The deconvolution needed to approximate the momentum fluxes from the volume-averaged velocities is found to be the critical part of the whole scheme.

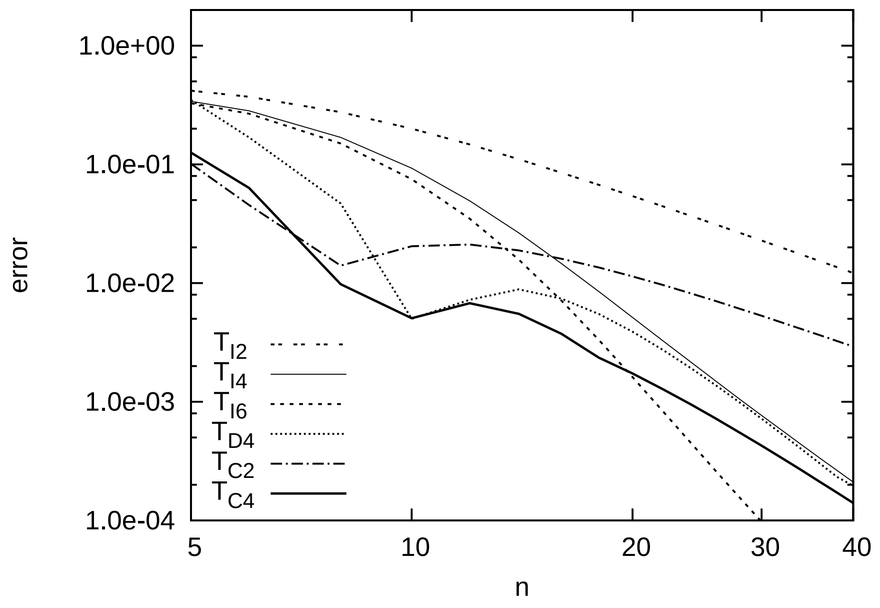


Figure 2.6: Convergence of numerical approximations applied to a Gaussian function (only even number of grid cells are plotted). T_{I2}:standard second-order intercell interpolation, T_{I4}:fourth-order compact deconvolution, T_{I6}:sixth-order compact deconvolution, T_{D4}:fourth-order compact interpolation. T_{C2}:second-order cell-centered deconvolution (mid-point rule), T_{C4}:fourth-order cell-centered deconvolution.

3 Validations

In this chapter, we evaluate the proposed scheme using well known benchmark test cases with increasing complexity. In all simulations, the time-integration is performed by the third-order Runge-Kutta scheme of [Wil80] and the continuity is enforced at every sub-step. This time integration is third-order accurate for both velocity and pressure. The nonlinear correction and fourth-order solution of the pressure are always used except stated otherwise.

3.1 Taylor-Green vortex flow

A family of Taylor-Green vortex flows (TGV) can be described by

$$u(x,y,t,Re) = c_1 - cos(x-c_1t)sin(y)e^{\frac{-2t}{Re}} \tag{3.1}$$

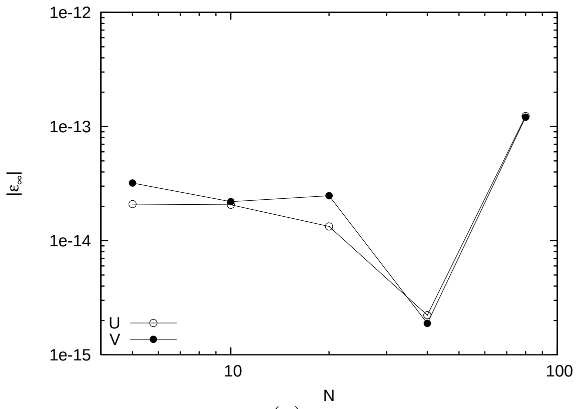
$$v(x,y,t,Re) = c_2 + sin(x-c_1t)cos(y)e^{\frac{-2t}{Re}} \tag{3.2}$$

$$p(x,y,t,Re) = -\frac{1}{4}(cos(2(x-c_1t)) + cos(2(y-c_2t)))e^{\frac{-2t}{Re}}. \tag{3.3}$$

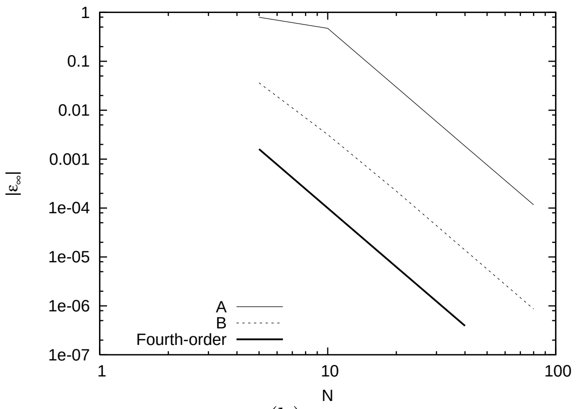
In this study, the domain is set to $(x,y) \in [0,2\pi]^2$. This test case has been widely used in literature. The classical TGV ($c_1 = 0, c_2 = 0, Re = 100$) is used in [PS04] and [CM04] to show that second-order solutions of pressure are sufficient to deliver fourth-order accurate velocities. We add a convection velocity (c_1, c_2) to the TGV because of the following reason. If there was no viscosity, the flow is essentially steady which is due to the equilibrium between convective and pressure terms. This equilibrium is obtained on uniform grids ($h_x = h_y$) for all central approximations, including the second-order. Therefore, a solution of the TGV with central approximations will reveal the convergence rate of the viscous term only.

The maximum errors of the velocities obtained by second and fourth-order schemes applied to the steady inviscid Taylor-Green vortex flow($c_1 = 0, c_2 = 0, \nu = 0$) are almost at machine accuracy (Fig.3.1(a)) which shows that the approximation of the convective term and the solution of the pressure does not affect the evolution of the momentum. Therefore the classical TGV can not be used to evaluate the momentum and the pressure terms. The nonlinear correction also does not affect the result of this flow.

Having ruled out the role of the pressure and continuity enforcement in this specific case,



(a)



(b)

Figure 3.1: (a) Maximum errors of the velocity at $t = 10$ of the fourth-order and the second-order scheme applied to inviscid TGV with $c_1 = 0$ and $c_2 = 0$. (b) Maximum error of the fourth-order scheme applied to classical TGV(A) and convective TGV(B).

we proceed to the validation of the deconvolution and the differentiation. The classical TGV ($c_1 = 0, c_2 = 0, Re = 100$) is used to test the differentiation , denoted by A in Fig3.1(b)). The convected inviscid TGV is used to test the deconvolution by setting $c_1 = 1, c_2 = 0$ and $\nu = 0$, denoted by B in Fig3.1(b). We keep the CFL number constant at 0.05 and march the solution to $t = 11\pi$ where the magnitude of the velocities is reduced to half of its initial value in case A. The maximum errors at the end of the simulations are shown in Fig.3.1(b) which clearly indicates that the convergence rates are fourth-order.

3.2 Doubly-Periodic shear layer

In order to expose the role of the pressure, we use a doubly periodic shear layer as the next test case. This simple 2D flow contains Kelvin-Helmholtz instabilities in which the shear layer is perturbed by a sinusoidal disturbance that leads to a roll-up of the vortex sheet into a cone-like shape. The domain $\Omega = [0, 1]^2$ is taken for this study and the initial velocities are given by

$$u = \begin{cases} tanh(\sigma(y - 0.25)) & \text{for y} \leq 0.5, \\ tanh(\sigma(0.75 - y)) & \text{for y} > 0.5. \end{cases} \tag{3.4}$$

$$v = \varepsilon sin(2\pi x). \tag{3.5}$$

This flow is governed by three parameters, the shear layer width parameter σ, the perturbation magnitude, ε and the Reynolds number. In this study the Reynolds number based on the maximum velocity and the length of the computational domain is set to $10,000$, the shear layer parameter and the perturbation magnitude are 30 and 0.05 respectively. This setting is similar to a *thick shear-layer problem* studied in [Bro95]. In order to show that the proposed scheme converges towards the correct solution, we generate a reference solution of this case on a 512^2 grid using a pseudo-spectral code. In this code, the computation is performed on the physical space and the derivatives are computed using FFT differentiations while the divergence form is used for the convective term. Authenticity of the solution is checked by successive refining the grid from 64^2 to 512^2. A smooth solution is already obtained at a resolution of 128^2 and the maximum difference between the solutions on 256^2 and 512^2 at the end of simulation is 10^{-9}. Further, the amplitudes of the wave numbers larger than $k \approx 300$ are clipped at machine accuracy. These assure the authenticity of the reference solution.

Comparing a finite volume solution with the one obtain from finite difference requires some interpolations or integrations in order to relocate and recast these two solutions in comparable forms. The solutions of finite volume methods can be deconvoluted to the pointwise values or we can integrate the solution from finite difference method. We chose the latter approach because interpolations and integrations on the fine grid of the reference solution is much more accurate than operating on the coarse grids used for the fourth-order scheme. The reference solution is first interpolated to the integration points using cubic splines and then the seventh-order integration is applied.

We provide a qualitative overview of the solutions using the contour plots of the vorticity in Fig.3.2. The improvement of the fourth-order over the second-order scheme is clearly visible. At the lowest resolution, the fourth-order solution preserves the correct shape of the vortex sheet while it is already distorted in second-order solution. The fourth-order has more wiggles but their magnitudes are smaller than those of the second-order. When the resolution is doubled, numerical wiggles are still disturbing the second-order solution. On the other hand, the solution of the fourth-order scheme is already smooth and the numerical wiggles at this resolution are comparable to those on the finest solution (256^2) of the second-order scheme. In this figure we see that even on the coarsest grid which is heavily under-resolved, the fourth-order scheme still delivers an appreciable solution unlike the second-order scheme.

The maximum errors of the fourth-order schemes and the convergence rates with respect to the reference solution are shown in Tab.3.1. Here we compare the two formulations of the convective velocities. On coarse grids, $T4$ is slightly more accurate than $DF4$. This is because the leading truncation term of $T4$ is smaller. The convergence rates of these two formulations approach fourth-order when the grid is sufficiently fine. The differences

N	$\lvert\varepsilon\rvert_\infty$ of u		Convergence Rate	
	T4	DF4	T4	DF4
64^2	2.5835E-02	2.7744E-02	—	—
96^2	1.0762E-02	1.1147E-02	2.16	2.25
128^2	4.4665E-03	5.0558E-03	3.06	2.75
196^2	8.9431E-04	1.2273E-03	3.97	3.49
256^2	2.6394E-04	2.5998E-04	4.24	5.39

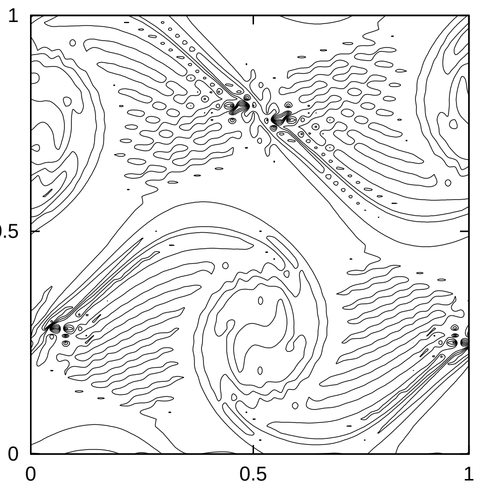
Table 3.1: Maximum error of the streamwise velocity and its convergence rate of T4 and DF4 convective velocities using fourth-order solution of pressure at $t = 1.2$, in doubly periodic shear layers flow. .

between the solutions using a different convective velocity are very small and the solutions are visually indistinguishable. For this reason, only the results from $DF4$ were shown in Fig.3.2.

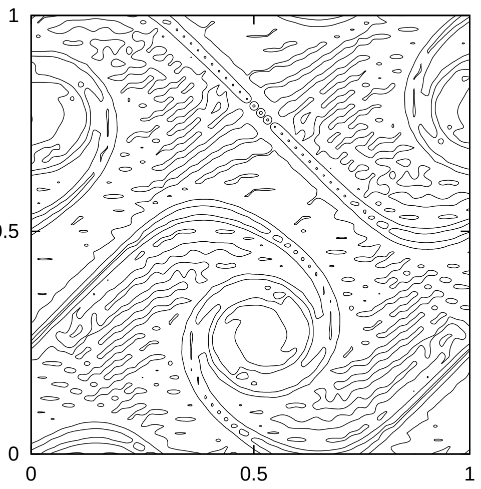
We repeat the simulation again, but with second-order solution of pressure, i.e. the divergence and the pressure gradient are computed with second-order schemes and the discrete Laplacian is $\mathbf{D}_2\mathbf{G}_2$. The result is given in Tab.3.2 for the $T4$ convective velocity. The effects of the second-order pressure can be observed at every grid size. The convergence rate of the L_∞-norm falls between second- and third-order. The convergence rate of the L_2-norm tends to third-order. At low resolution, the errors primarily stem from the wiggles inherit in the approximation of the convective term and therefore we see little differences between second-order and fourth-order pressure. Once the wiggles have disappeared ($N \geq 196^2$), the error of the solution using second-order pressure is significantly larger.

N	Error of u		Convergence Rate	
	$\lvert\varepsilon\rvert_\infty$	$\lvert\varepsilon\rvert_2$	L_∞	L_2
64^2	3.23E-02	1.36e-01	—	—
96^2	1.22E-02	1.03e-01	2.4	2.7
128^2	5.54E-03	8.32e-02	2.7	2.7
196^2	2.02E-03	5.93e-02	2.5	2.8
256^2	1.02E-03	4.57e-02	2.4	2.9

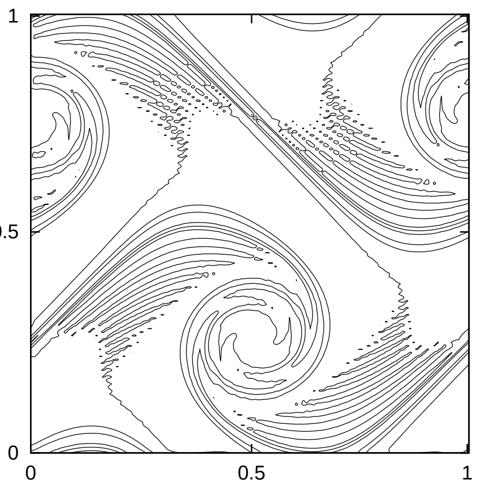
Table 3.2: Error and the convergence rate of the streamwise velocity at $t = 1.2$ using $DF4$ convective velocities and second-order solution of pressure ($\mathbf{D}_2\mathbf{G}_2$).



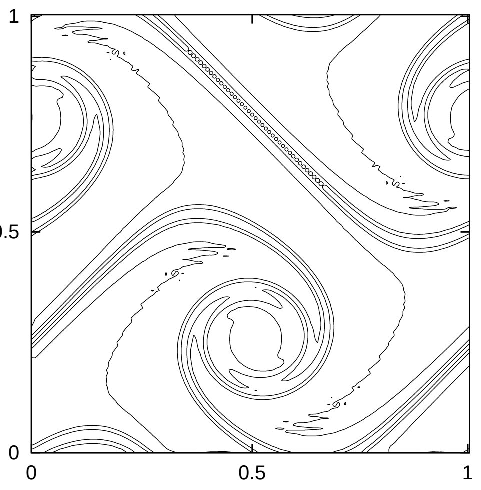
(a)



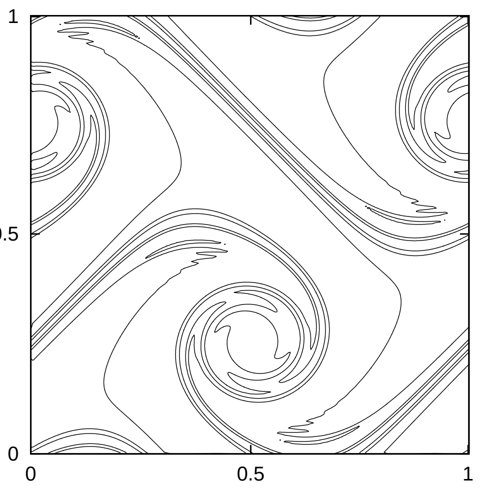
(b)



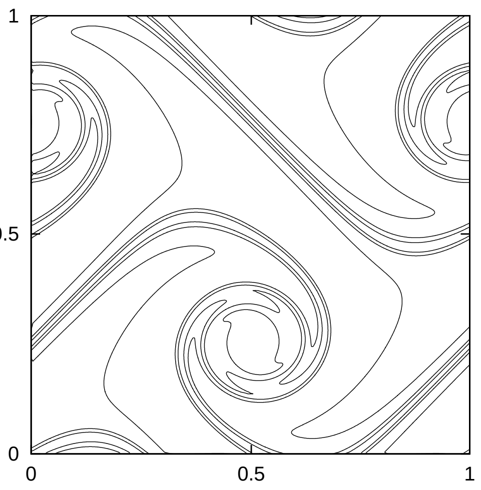
(c)



(d)



(e)



(f)

Figure 3.2: Contour plot of vorticity from -36 to 36. Left: second-order, Right: fourth-order (DF4). The resolution are 64^2, 128^2 and 256^2 from top to bottom.

3.3 Instability of plane channel flow

The instability of plane channel flow is a common test case used to validate higher-order accuracy of numerical schemes. In this test case, the parabolic profile of the channel flow is disturbed by the most unstable eigenfunction. The solution of the velocity fields can be described by the Orr-Sommerfeld equation:

$$u(x,y,t) = (1-y^2) + \varepsilon \mathcal{R}eal\left(\Psi(y)e^{i(\alpha x-\omega t)}\right) \tag{3.6}$$
$$v(x,y,t) = -\varepsilon \mathcal{R}eal\left(\alpha i \Psi(y)e^{i(\alpha x-\omega t)}\right) \tag{3.7}$$

The energy of the perturbation and its growth-rate is computed by :

$$E_d(t) = \int_\Omega \left((u(x,y,t)-u(x,y,0))^2 + (v(x,y,t)-v(x,y,0))^2\right) dxdy \tag{3.8}$$
$$G_p(t) = 2\omega_i = 2ln\left(\frac{\partial E_d(t)}{\partial t}\right) \tag{3.9}$$

The Orr-Sommerfeld eigenfunction $\Psi(y)$ here is a stream function and only the real part of the perturbation is taken into the velocities. This test-case is sensitive to the balance among the terms in the Navier-Stokes equations. The viscous term attenuates the perturbation while the convective term transfers energy from the main flow to the perturbation. If the approximation of the diffusion term is accurate and the convective term is under approximated, the growth-rate of the disturbance will be less than the analytical one. This is a common situation found in finite difference methods applied to this case [CM04, MZH85, RM91, GPPZ98, DM01]. On the other hand, the growth-rate will be larger than the analytical one when the situation is reverse. Higher-order convergence can only be achieved if every approximation is correctly treated. Therefore this is a formidable test case for numerical schemes and the boundary closures. The conditions of this test are set to the same conditions used in [MZH85] where $Re = 7500$, $\alpha = 1$, $\varepsilon = 0.0001$ and the only unstable mode is $\omega = 0.24989154 + 0.00223498i$. The expected analytical growth rate is $G_p(t)$ =4.46996E-03. The computational domain is $[L_x, L_y] = [2\pi H, 2H]$ based on the channel half-width H. The CFL is kept constant at 0.05 such that the errors are dominated by the spatial approximations. The simulations are calculated using double precision and the growth-rate of the perturbation is measured at $t = 50.29H/u_c$ where u_c is the velocity at the center of the channel.

Several grid systems in the wall-normal directions have been used to simulate this flow. Chebychev grids are used in [MZH85] and geometric grids are used in [RM91] and [CM04]. The geometric grids deliver better results at a much lower number of grid points. Further, the Chebychev grid does not offer flexibility in grid placement. Therefore we use the geometric grid in our code in which the grid spacing is increased or decreased with a constant factor.

N_y	$G_p(T4)$	$G_p(DF4)$	$\varepsilon_p(T4)$	$\varepsilon_p(DF4)$	$C_a(T4)$	$C_a(DF4)$
32	9.66E-03	1.09E-02	5.19E-03	6.44E-03	—	—
64	6.34E-03	6.37E-03	1.88E-03	6.44E-03	1.46	1.76
128	4.66E-03	4.65E-03	1.90E-04	1.78E-04	3.31	3.42
256	4.48E-03	4.47E-03	1.08E-05	1.41E-06	4.13	6.97
512	4.47E-03	4.47E-03	7.37E-07	3.19E-08	3.88	5.47

Table 3.3: Convergence study in y-direction of the instability of plane channel flow on *uniform* grid with $N_x = 64$. The table shows growth-rates of perturbations (G_p), their errors (ε_p) and the corresponding convergence rates (C_a).

Our grid dependency study shows that $N_x = 32$ is sufficient to conduct a convergence study in y-direction with the fourth-order scheme. In order to be on the safe side the results in Tab.3.3 and Tab.3.4 are computed using $N_x = 64$. Table 3.3 shows a clear convergence towards fourth-order for both formulations of convective velocities on uniform grids. A fourth-order convergence rate on nonuniform grids is shown in Tab.3.4. The smallest grid spacing on these grids is set to $1/N_y$ (half-size of the uniform grid). The predictions of the growth-rates on nonuniform grid are comparable to those on uniform grids with twice the number of grid cells. Both formulations of convective velocities deliver a comparable level of accuracy.

We further use this case to study the role of the nonlinear correction. To this end, we plot in Fig.3.3(a) the time evolution of the perturbation energy for two grid resolutions with and without nonlinear correction. The figure shows that the nonlinear correction facilitates additional energy transfer to the perturbation bringing the growth-rate closer to the analytical one. On the coarse grid, the nonlinear correction is helpful, but not as impressive as on the fine grid. It has to be noted that a growth-rate higher than the analytical one does not imply an instability of the numerical scheme. It means that the perturbation is gaining more energy from the mean field than what is lost by the dissipation. In fact, the total kinetic energy of the system is reducing in all cases shown there.

The spatial gradient of the perturbation in this flow varies greatly. The grid cells should be distributed in a way that the errors over the whole region are comparable. The result using the same number of grid cells can be significantly different when the grid cells were distributed differently. In Fig.3.3(b) we plot the perturbation growth rate on four different grids using the same number of grid cells ($N_x = 32$ and $N_y = 64$). The uniform grid ($S = 1.000$) and the nonuniform grid with $S = 1.046$ are the ones reported in Tab.3.3 and Tab.3.4, respectively. A stronger stretching ($S = 1.087$) leads to an excellent prediction of the growth rate while further stretching ($S = 1.124$) gives worse result. This is because the grid cells in the center of the channel are too large to capture the velocity variations accurately.

N_y	$G_p(T4)$	$G_p(DF4)$	$\varepsilon_p(T4)$	$\varepsilon_p(DF4)$	$C_a(T4)$	$C_a(DF4)$
32	5.919E-03	6.432E-03	1.45E-03	1.96E-03	—	—
64	4.628E-03	4.646E-03	1.58E-04	1.76E-04	3.20	3.48
128	4.475E-03	4.485E-03	4.55E-06	1.52E-05	5.12	3.53
256	4.470E-03	4.470E-03	1.91E-07	4.35E-07	4.57	5.06

Table 3.4: Convergence study in y-direction of the instability of plane channel flow on *nonuniform* grid with $N_x = 64$.

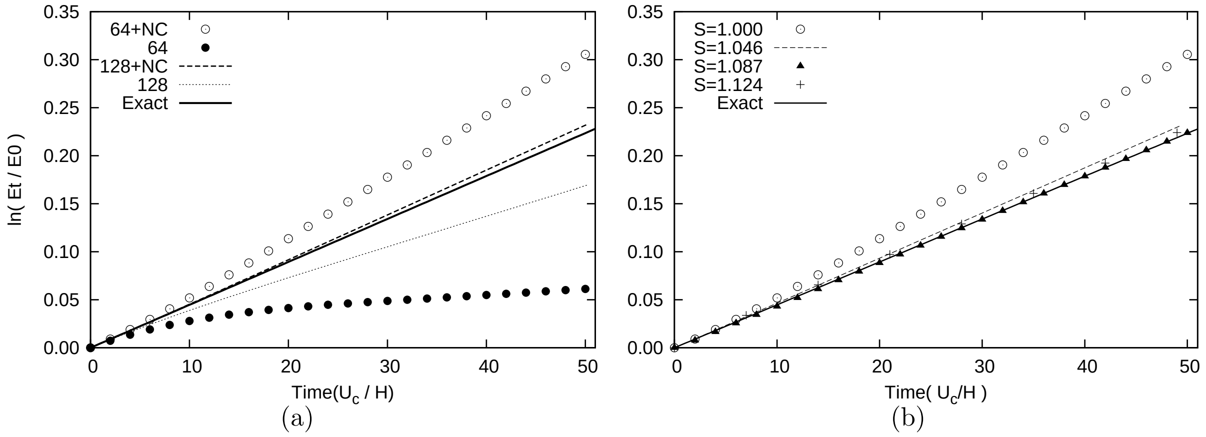


Figure 3.3: Time evolution of the perturbation energy on: (a) two uniform grids and (b) non-uniform grids with different stretching factor (S). All grids are computed with $N_x = 64$. On uniform grid, the resolution in the wall-normal direction of the two grids are $N_y = 64$ and 128 and the result with nonlinear correction are denoted by 64+NC and 128+NC. The result without the nonlinear correction are denoted by 64 and 128.

In the following, we investigate the effect of a second-order approximation of the pressure gradients by keeping all spatial approximations at fourth-order except the calculation of the pressure gradient, divergence and the discrete Laplacian, which are now computed using second-order schemes ($\mathbf{D}_2\mathbf{G}_2$). The accuracy limitation due to second-order pressure becomes evident in Tab.3.5. The growth rates of the perturbation are much less accurate here. Previously in Tab.3.3, the growth rate on the finest solution (64X-512Y) was accurately predicted up to the fourth digit and the error was 1.9E-7. Here on the same grid , the error of the solution using second-order pressure is 3.1E-4, three orders of magnitude larger. The saturation of the accuracy due to the approximation error in each direction is clearly seen in the first row and the first column. This poor accuracy is continued to the finest grid ($N_x = 256$ and $N_y = 512$) which predicts the perturbation growth at 4.428E-03. Nonetheless, the convergence the convergence rate towards this finest solution seen in the last row is approximately third-order.

In the previous three test cases, we have evaluated the proposed scheme including the

NY	NX				
	32	64	128	160	256
64	6.64e-04	3.18e-04	5.64e-04	5.93e-04	6.26e-04
128	1.43e-03	4.42e-04	1.94e-04	1.65e-04	1.33e-04
256	1.35e-03	3.62e-04	1.15e-04	8.50e-04	5.30e-05
512	1.16e-03	3.10e-04	6.20e-05	3.20e-05	—

Table 3.5: Error of the perturbations growth rate (ε_p) relative to the growth rate on the finest grid (4.428E-03) using second-order approximation of pressure.

boundary closures. A fourth-order convergence rate is obtained in all cases with the fourth-order approximation of pressure. The overall accuracy falls to third-order if the pressure was only treated by a second-order scheme. The nonlinear correction plays crucial role in energy transfer in the instability of plane channel flow. In the next step we investigate the application of the fourth-order scheme in a fully turbulent flow.

3.4 Turbulent channel flow

In this section we investigate the accuracy and the performance of the newly developed scheme in turbulent channel flow at $Re_\tau = 180$, based on the friction velocity u_τ. This test case has been employed to study turbulence and its modelling for the past two decades. Results of numerical simulations using spectral codes from several authors [MKM99, KMM87, Hu06] are publicly available for comparison. Numerical simulations using other classes of approximation should converge to these data.

The parameters of the numerical grids and the computational domain used in this study are listed in Tab.3.6 along with the three spectral simulations which are used for comparison. In our simulations, the streamwise, spanwise and wall-normal directions are set to x, y and z accordingly. The computational box is $[L_x, L_y, L_z] = [12.57H, 4.20H, H]$ which is the same domain used in [MKM99]. Homogeneity is assumed in streamwise and spanwise directions and thus periodic boundary conditions are applied. The top and the bottom walls are treated by no-slip boundary conditions. All spatial dimensions in this section are normalized by channel half-width if not stated otherwise. The flow is driven by a constant pressure gradient which is added as a source term in the momentum equation of the streamwise velocity. There is no control of mass flow within the simulations. The driving pressure gradient is balanced only by the shear force on the top and bottom walls. The initial velocity field is obtained from imposing a random perturbation on the logarithmic velocity profile. The criterion of a statistically steady state is assumed when the bulk flow velocity is not changed more than $0.5\%u_b$ over a period of $10Lx/u_b$. After the statistically steady state is reached, the flow is further advanced for $10Lx/u_b$. The sampling is then performed for $100Lx/u_b$ together with the averaging in the homogeneous directions.

This study is organized into four parts. In the first part, we rule out the T4 convective velocity. The accuracy of the proposed scheme with the DF4 convective velocity is demonstrated in the second part, the grid dependency study. In the third part, we evaluate the efficiency of the fourth-order scheme. Its computational cost and accuracy are compared against the second-order scheme. Having identified accuracy and performance of the full scheme, we discuss the necessity of the fourth-order solution of pressure and the influence of the nonlinear correction in the fourth part.

Grid	NX	NY	NZ	$\triangle x^+$	$\triangle y^+$	$\triangle z^+_{min}$	$\triangle z^+_{max}$
A	48	42	64	47.2	18.0	5.6	5.6
B	96	64	128	23.6	11.8	2.8	2.8
C	32	32	32	70.7	23.6	5.4	18.6
M1	64	64	64	35.4	11.8	2.7	9.7
M2	96	80	96	23.6	9.4	1.1	5.8
F	128	128	128	17.7	5.9	0.72	4.4
MKM1999,[MKM99]	128	129	128	17.7	5.9	0.054	4.4
KMM1987,[KMM87]	192	129	160	11.9	7.1	0.054	4.4
H2006,[Hu06]	256	256	121	16.9	8.4	0.062	4.7

Table 3.6: Specification of numerical grids used in the grid dependency study.

3.4.1 Choice of the approximation of the convective velocities

In laminar flows, the flow fields are usually smooth which was also the case for the flows presented earlier. In all previous simulations, the cell-averaged divergence of the convective velocities was always smaller than $10^{-6}u_{ref}/L$ for the respective characteristic length (L) and velocity (u_{ref}) in those cases. Consequently, the differences between the two alternatives for the convective velocities were negligible. In this section we investigate the behaviour of the two alternatives for the convective velocity, $T4$ and $DF4$ in a turbulent channel flow. In order to exclude other effects, we use uniform grids in every direction and turn off the nonlinear correction. Here, grid A and grid B listed in Tab.3.6 are considered. The convective velocities $T4$ and $DF4$ perform significantly different on the coarse grid but they are comparable on the fine grid (Fig.3.4a&b). According to this graph, the T4 convective velocity should not be used in LES and is probably equivalent to DF4 when the grid is DNS-like. However, the absolute values of the cell-averaged divergence on the u-momentum cell shown in Fig.3.4 shifts our favour entirely to DF4. The divergence-free convective velocity conserves mass close to machine accuracy on the momentum cell while the $T4$ formulation creates the maximum cell-averaged divergences of approximately $0.045u_b/H$. This means that an artificial momentum of the comparable amount are being removed or added per unit-momentum (see Eq.(2.10)). Therefore we choose DF4 to approximate the convective velocities for the remaining tests.

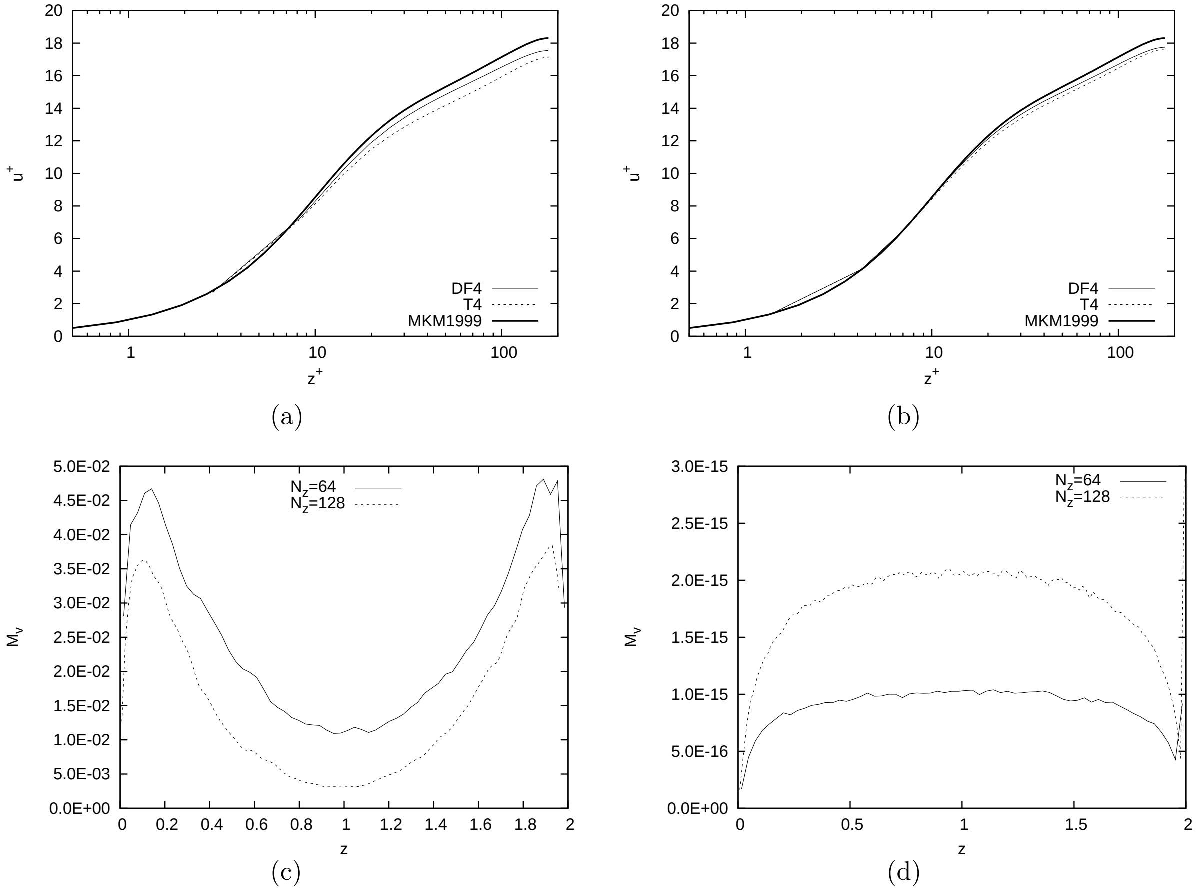


Figure 3.4: Mean profiles of streamwise velocity along wall-normal direction on grid A (a) and grid B (b). Averaged mass imbalance per unit volume on u-momentum cells of T4 (c) and DF4 (d).

3.4.2 Grid dependency study

In this part, the accuracy of the fourth-order compact scheme is evaluated using Grid C to F listed in Tab.3.6. These four grids mimic different situations one may encounter in numerical simulations of turbulent flows. Grid C is on the extreme end of the resolution where one can perform a reasonable LES. $M1$ is too coarse for DNS simulations but an excellent resolution for LES. $M2$ is a reasonable and F is a good resolution for DNS. The grid spacings of grid F match the ones used in [MKM99] in the center of the channel.

The mean flow variables shown in Tab.3.7 demonstrate a convergence towards the spectral solution. The mean bulk and center line velocities clearly converge towards the spectral

Grid	u_c/u_τ	u_b/u_τ	C_f
C	16.68	14.55	9.35E-3
M1	17.62	15.20	8.57E-3
M2	18.19	15.63	8.20E-3
F	18.34	15.77	8.16E-3
KMM1987,[KMM87]	18.20	15.63	8.18E-3
MKM1999,[MKM99]	18.30	15.52	8.18E-3

Table 3.7: Mean flow variables

solution. On the grid(C), the bulk flow is underestimated by 6.4% and the error is reduced to 2.3% on M1. Results of grid M2 and grid F are very close and comparable to the reference solutions.

Qualitative convergence of the scheme is shown in Fig.3.5(a). Amiri and Hanami[EK05] incorporated LES into Pereira's fourth-order scheme and found that the result without explicit filtering is poor on the 64^3 grid which is equivalent to M1. Here we can obtain a satisfactory result on the same resolution without any filtering or modelling. Increasing resolution from grid M1 to grid M2 places the profile almost on top of the spectral solution. When this profile is plotted together with two spectral solutions in Fig.3.5(b), we see that it is lying between these two. The mean streamwise velocity on grid F is not plotted here because it is lying between the two spectral solutions as well

In Fig.3.5(c) the square-roots of the surfaced-averaged values of the Reynolds normal stresses e.g. $[uu]^{yz}_{i,j,k}$ extracted from the momentum equation are plotted against the two reference spectral solutions. It confirms that the grid M2 is sufficient to capture all scales of engineering interest. The Reynolds normal stresses of the fourth-order scheme do not differ from the reference solutions more than the difference between those two. This level of difference is much smaller compared to uncertainties occurring in physical experiments.

Next we consider the higher-order statistics, the skewness and the flatness factors. The Skewness factor shown in Fig.3.6(a) confirms the consistency and convergence of the scheme. The profiles of grid M2 are satisfactory close to the spectral solution. The profile of $S(u)$ is on top of the reference solution almost everywhere and the profile of $S(w)$ is satisfactory. The value of 0.06 is obtained near the wall compared to -1.3 when the second-order scheme was used (not shown). To the best of our knowledge, all second-order codes predict negative $S(w)$ near the wall when the grid resolution is not finer than those used in [MKM99] and [KMM87]. Increasing the solution to grid F clearly improves the accuracy of the solutions and brings $S(w)$ on top of the spectral solution. The flatness factors, are plotted in Fig.3.6(b). The profiles on both grids are highly satisfactory for the streamwise velocity but notable shortcomings are observed in $F(w)$. The deviations we see here in the skewness and flatness factors can be attributed to small scale structures at the far end of the spectrum. In order to reveal how much the fourth-order scheme is behind the spectral scheme, one-dimensional spectra of the cell-averaged values are investigated and plotted in Fig.3.7(a) and 3.7(b). In these figures, the Fourier spectra are normalised by the value of the first mode. The energy spectrum E_{uu} in the y-direction on grid C is far from the spectral solution, but the spectra on the other grids follow the one of the spectral code nicely. In the streamwise direction, where the convection is much stronger, the energy spectra on all grids follow the spectral solution up to about 60% of the Nyquist limit and then start to fall sharply. This is consistent with the sharp drop of the transfer function of the fourth-order compact deconvolution used for the convective fluxes shown in Fig.2.5.

3.4.3 Comparison with the second-order scheme

The improvement of the compact fourth-order over the second-order scheme is illustrated in Fig.3.8(a). The profile of the mean streamwise velocity of the fourth-order scheme is closer to the spectral solution at the same number of grid points. The second-order scheme requires twice number of grid cells in each direction to have a result comparable with the fourth-order scheme. It was shown earlier that the fourth-order scheme only requires $0.7M$ grid cells to match the solution of spectral codes up to the second-order statistics. The second-order scheme cannot match the first-order statistics even at $2.1M$ i.e. three-times more cells than the fourth-order. The energy spectra E_{uu} in x-direction show an interesting result. The second-order drops sharply at $k_x \approx 25$. This means the second-order scheme can accurately capture only 40% of the whole energy spectrum while the fourth-order can capture up to 60%.

In order to provide a clear picture of the efficiency of the fourth-order scheme, we document the CPU seconds used to advance the momentum equation (Eq.(2.6)) per time step in Tab.3.8. The runtimes are broken down into the computation of the momentum equation (M) and the solution of pressure (P) which includes the calculation of divergence, solution of Poisson equation and velocity correction. The computations are performed using double precision on an AMD Opteron 8216. The compiler is Intel FORTRAN and the optimization is set to *-O3*. Due to the logarithmic complexity of the FFT, the CPU-time of the pressure solver increases fastest. It occupies 50% of the total time on the finest grid in the fourth-order scheme. At the same number of grid cells, the fourth-order is 2.4-times more expensive in average. If one considers that the fourth-order scheme requires only half of the grid cells in each direction, this means the compact fourth-order scheme can deliver a result 5-times faster than the second-order scheme *per time integration*. A larger grid size allows a larger time step, therefore the compact fourth-order scheme can be effectively

10-times more efficient than the second-order scheme. This luxury efficiency can be spent to get a more accurate solution in a shorter time.

Grid	Fourth-order			Second-order			Ratio
	M	P	sum	M	P	sum	
C-32^3	0.11	0.05	0.16	0.03	0.032	0.06	2.73
M1-64^3	1.08	0.73	1.89	0.19	0.621	0.81	2.34
M2-$96^2 \times 80$	4.22	3.55	7.77	0.58	2.343	2.92	2.66
F–128^3	9.72	9.80	19.52	1.56	8.716	10.28	1.90

Table 3.8: CPU-seconds per time step spent in momentum equation(M) and the enforcement of continuity(P) on an Opteron 8216.

3.4.4 Effects of second-order solution of pressure and nonlinear correction

In what follows the effects of the second-order approximation of the pressure and the nonlinear correction for convective term are investigated. The first effect is studied by keeping the approximation of the momentum and the diffusive terms at fourth-order but compute the pressure gradient and the divergence by second-order approximations. The second effect is investigated by using the fourth-order scheme but turn off the nonlinear correction.

Numerical simulations are performed again on grid C, M1 and M2. In Fig.3.9(a)-(c), the mean streamwise velocity profiles are compared. The second-order solution of pressure leads to a strange behaviour. On the coarsest grid, it overestimates the velocity profile in the center of the channel. However on the M1 grid, the profile is underestimated and, by chance, collapses on the full fourth-order solution. However, later on grid M2 it departs from the full fourth-order and again overpredicts the velocity. The fourth-order solution of pressure is therefore essential to obtain accurate solutions and a convergent scheme. The nonlinear correction, on the other hand, does not strongly affect the solution. Turning it on and off does not significantly change the streamwise velocity profiles on the coarse grids (C and M1). The nonlinear correction slightly improves the mean streamwise velocity profile on grid M2. The nonlinear correction takes care of the small variations on the surfaces of the momentum cells. It will not improve the accuracy of the convective term on coarse grids when the error in the approximations of momentum and convective velocities are relatively large. This effect is similar to the relationship between the truncation errors and the subgrid-scale model mentioned by Ghosal[Gho96]. The nonlinear correction costs roughly 30% of the time used in the computation of the momentum equation. Thus when we knew a priori that the grid spacing is not DNS-like, the nonlinear correction can be turned off for a better cost/performance ratio. The remaining question is whether the nonlinear correction pays off on very fine grids. This is answered in Fig.3.9(c)&(d). The thin solid line of the fourth-order with nonlinear correction (4M-4P-wNON) is absorbed into the thick solid line of the reference solution while the plus symbols of the fourth-order without nonlinear correction (4M-4P-woNON) are slightly lower. Also, the solution without the nonlinear correction predicts the wrong value of the Reynolds shear stress (Fig.3.9(d)). Therefore if we do not use the nonlinear correction, the turbulence interactions may not be captured accurately. In summary, the nonlinear correction should be *turned off* when performing LES and let the subgrid-scale handle small scale interactions. In DNS, the nonlinear correction would be necessary when highly accurate solutions (comparable to spectral code) are sought and the grid was sufficiently fine.

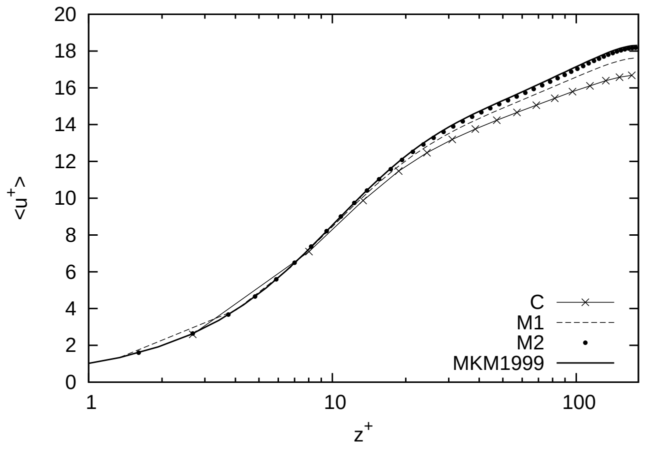
3.5 Conclusion

We studied two formulations for the fourth-order convective velocities, a conservative (DF4) and non-conservative (T4). A difference between them can hardly be observed in laminar flows because of the smoothness of the field. On staggered grids the second-order solution of the pressure limits the accuracy of the solver but not as strong as observed in [WD01] for collocated grids. A convergence rate of third-order can be achieved on staggered grids when the pressure is only treated with second-order approximations. This finding is supported by a comparative error analysis in Fourier space in the chapter.2. The cell-centered deconvolution used to enforce the continuity on staggered grids has higher resolving power than the inter-cell deconvolution on collocated grids. Thus more information is preserved in the velocity field. A third-order convergence rate can be obtained whilst it is capped at second-order on collocated grids. Nevertheless, a fourth-order solution of pressure is required to reach overall fourth-order convergence.

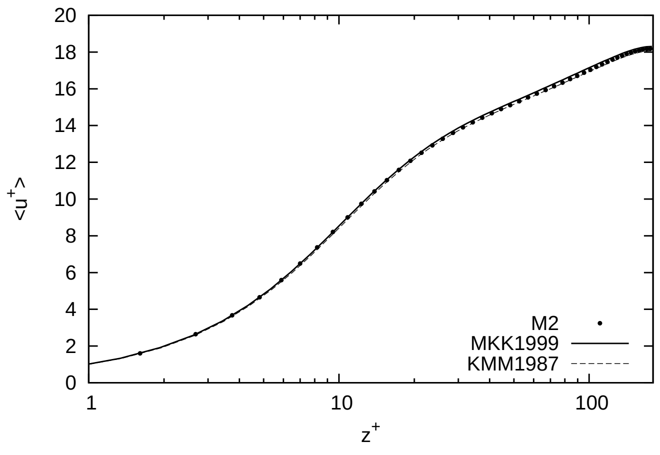
In turbulent flows the two convective velocities give significantly different results on coarse grids. Therefore divergence-free formulations should be used for the convective velocity to maintain the underlying conservation properties of the Navier-Stokes equations.

The high resolution property and the efficiency of the proposed scheme is demonstrated using a turbulent channel flow in which the convergence towards the spectral solution is demonstrated. The proposed scheme is robust and can give a reasonable solution using only 32^3 grid cells. Actually, our code can deliver a stable DNS solution even on 20^3 cells. The solution using $0.7M$ grid points is in excellent agreement with spectral solutions up to second-order statistics although it is only one-third and one-sixth of the grid cells used by spectral solutions in [MKM99] and [KMM87], respectively. Small deviations near the wall are observed in third and fourth-order statistics. Increasing the grid resolution improves these higher-order statistics. We have quantified the efficiency of the proposed scheme. It only requires half of the resolution per coordinate direction to have a comparable result with the second-order scheme. Effectively, the fourth-order scheme can be ten-times faster than the second-order scheme in advancing the momentum per one dimensionless time unit, at a comparable accuracy.

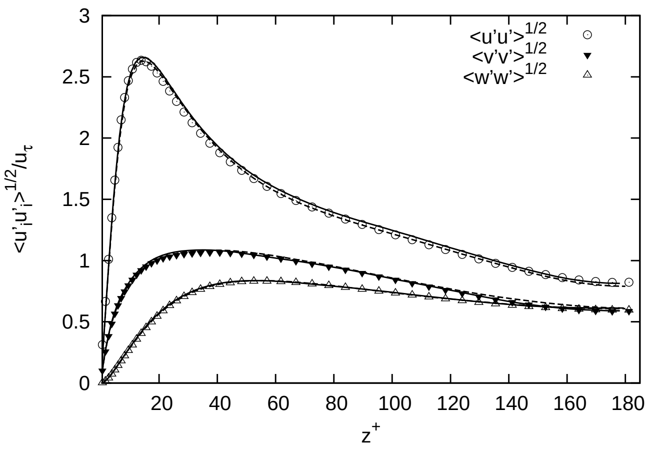
The fourth-order solution of pressure is essential, physically and numerically. The second-order pressure will only give, at best, a third-order convergence rate for the velocities and it delivers poor solutions in turbulent channel flows. The nonlinear correction is found to be useful on a very fine grid and unimportant otherwise. It could be turned off when performing LES for a better cost/performance ratio.



(a)



(b)



(c)

Figure 3.5: (a) Convergence of mean streamwise velocity. (b) Mean streamwise velocity on grid M2 compared with other two spectral solutions from [KMM87] (dashed line) and [MKM99] (solid line). (c) Square root of Reynolds normal stresses on grid M2.

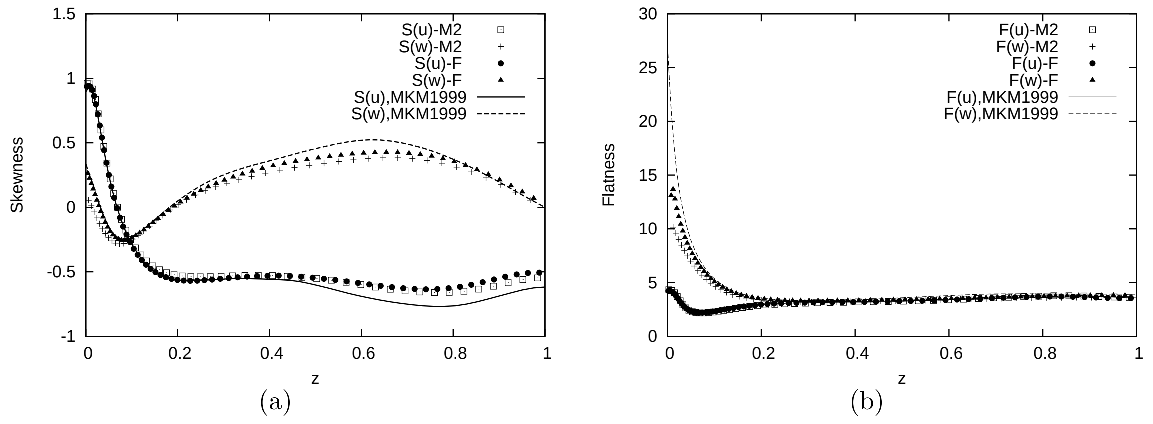


Figure 3.6: Skewness factor (a) and Flatness factor (b).

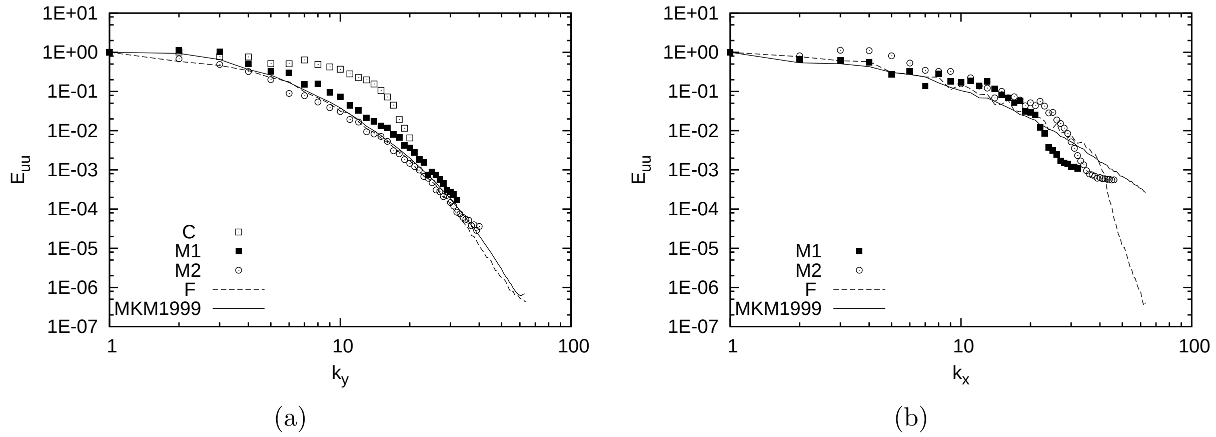
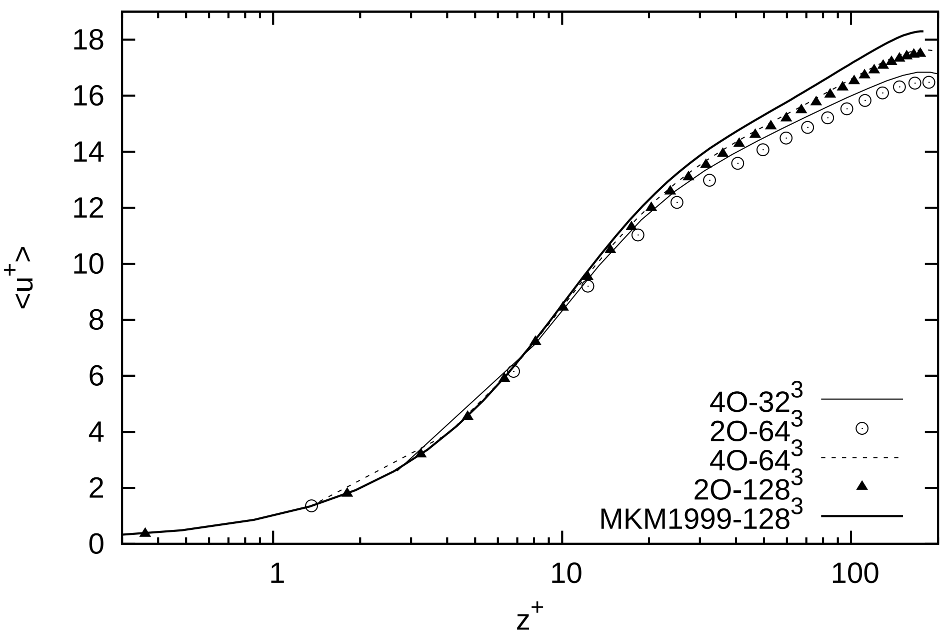
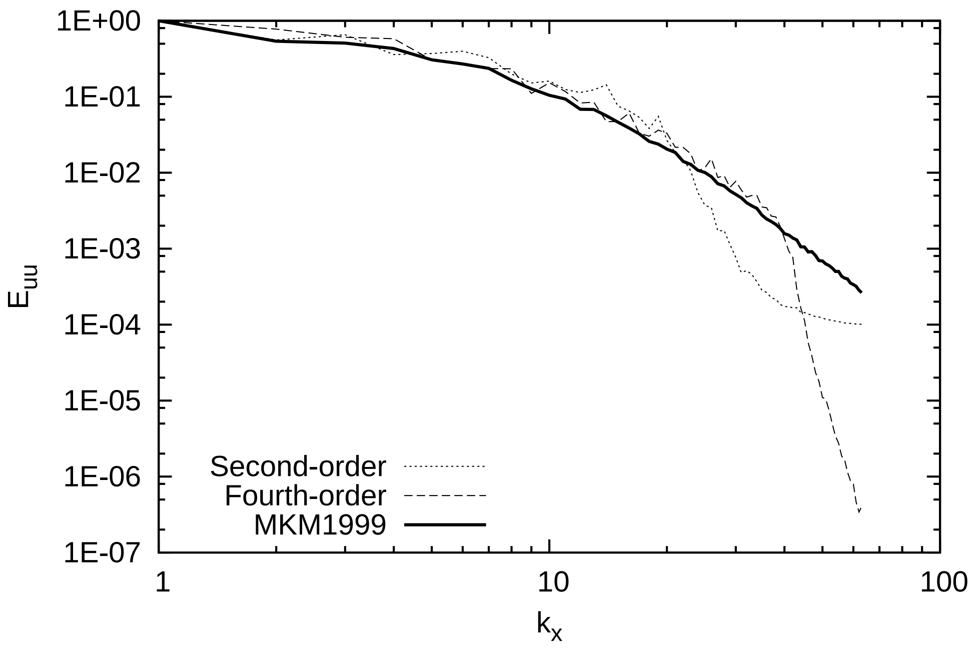


Figure 3.7: One-dimensional energy spectra in spanwise (a) and streamwise (b) directions of the streamwise velocity at $z^+ = 178$.



(a)



(b)

Figure 3.8: Comparison of the mean streamwise velocity profiles of second-order and fourth-order scheme: (a) mean streamwise velocity, (b) one-dimensional spectra of streamwise velocity in x-direction on grid F.

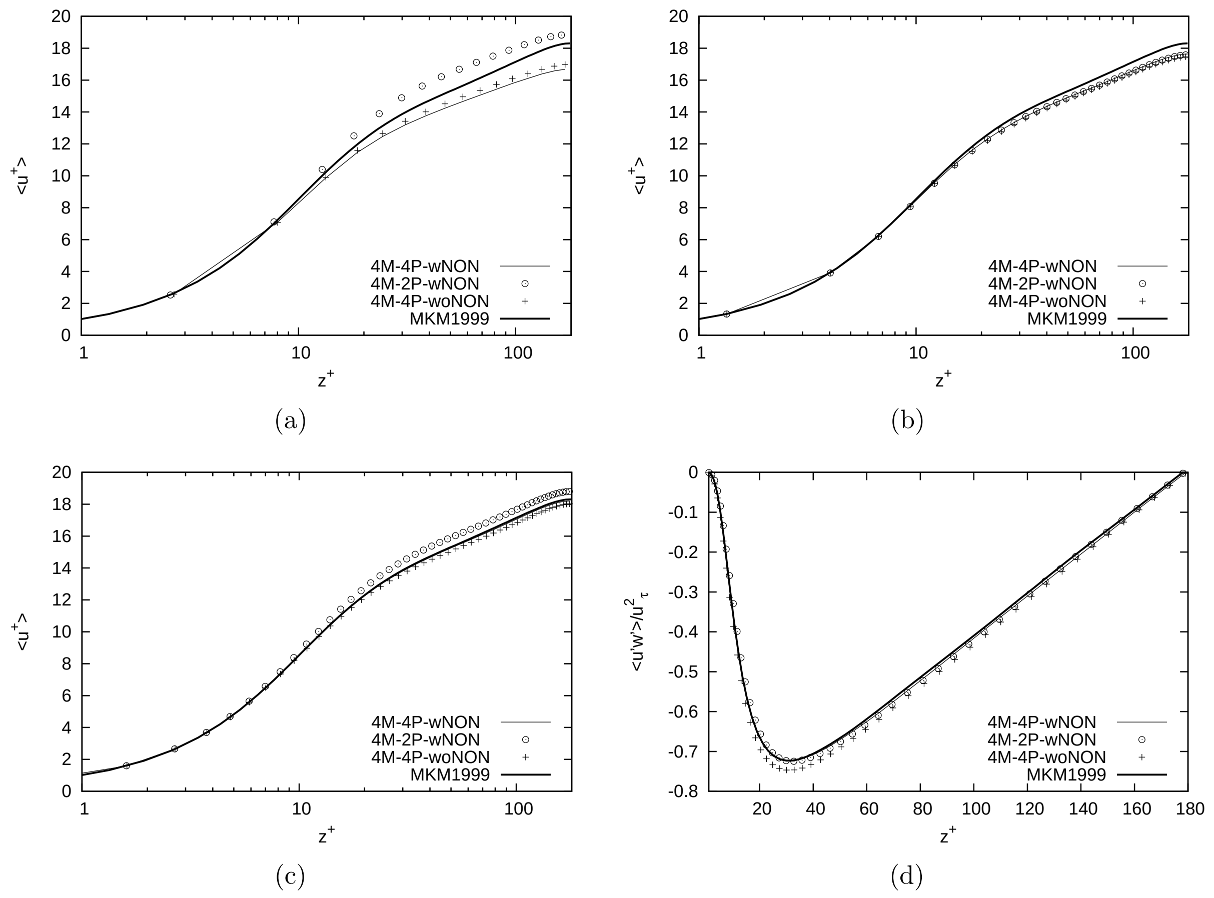


Figure 3.9: Mean streamwise velocity on grid C(a), M1(b) and M2(c) and the Reynolds shear stress on grid M2 (d). The thin solid line is the full fourth-order scheme (4M-4P-wNON). The open circle (4M-2P-wNON) is the solution using the fourth-order scheme for convective and diffusive terms but second-order approximation of divergence and pressure gradient. The plus symbol (4M-4P-woNON)is the full fourth-order schemes without the nonlinear correction.

4 Parallel Solution to Tridiagonal Systems

4.1 Introduction

In order to parallelised the compact scheme, we need an algorithm solving tridiagonal systems in parallel. Solving tridiagonal systems is one of the important kernels of scientific computing. These systems appear in many approximation problems such as spline interpolations, wavelets or numerical solutions to differential equations. These systems are usually solved repeatedly for an enormous number of right-hand sides. In three-dimensional problems, this number can easily reach 2^{32} which may not fit on a memory of a single processor machine. Efficient parallelisation of these systems is thus critical for scientific computing and our compact fourth-order scheme.

Modern parallel computers are mostly categorised into massive parallel processing (MPP) type which consists of a large number of powerful nodes. The right-hand-side of the system can be distributed among the processor, and thus a processor holding a certain subsystem may not have explicit access to the subsystems own by other processors. Despite the availability of NUMA architecture which allows a processor to access the memory of the other processors, this nonlocal access is much more expensive than the local one. This problem is more pronounce when the number of processor does not fit on a single partition where the computational nodes are connected by special interconnection networks. Therefore an efficient and scalable algorithm solving tridiagonal matrices in parallel should minimise the number of communications and synchronisations. Divide and conquer algorithms [Wan81], [Bon91], [Sun95], [AG96] and [Heg96] are well suited for the current trend in supercomputer architecture. Bondeli [Bon91] and Sun [Sun95] independently present the algorithm specialised for diagonal dominant tridiagonal matrix. The *reduced parallel diagonal dominant algorithm* in [Sun95] can solve a tridiagonal system of n equations for γ right-hand sides using $(5n/p + 4J + 1)\gamma$ operations on p processors, for some small number of J which will be described later. His algorithm is among the most efficient algorithms for this problem. This algorithm requires two times a unidirectional communication at two stages of the computation. The bidirectional communication links of modern computers are thus left unused. This shortcoming and problem of load-balancing among the processors can double the cost of the communication.

In the present work, we develop a novel interface-splitting algorithm with a complexity of $(5n/p + 4J - 4)\gamma$. This algorithm is designed for diagonal dominant tridiagonal matrices. The idea is to decrease the communication and reduce the data dependency. This is achieved by computing the solution at the interface between two processors before computing the solution on the inner points. This algorithm exploits an exponential decay of the inverse of diagonal dominant matrices demonstrated in [Nab99]. The proposed scheme is competitive. It has less complexity than the algorithm presented in [Sun95] and requires one less synchronisation phase. Therefore the proposed algorithm is less sensitive to load balancing and network congestion problems. This scheme is applicable for non-Toeplitz as well as periodic systems.

This chapter is organised as follows. First the interface-splitting algorithm is derived. Then, similarities and differences with existing divide-and-conquer algorithms are discussed and the complexity of the proposed algorithm is presented. Finally, accuracy and performance on workstations, an Ethernet cluster and a supercomputer of the proposed scheme are presented compared to ScaLAPACK.

4.2 Parallel algorithm solving tridiagonal systems

In this section we classify the parallel algorithm solving tridiagonal systems by two ways namely, (i) pre-processing and (ii) post-processing algorithms and point out that for a specific algorithm of each type, it is possible to formulate an equivalent algorithm belong to the other one.

4.2.1 Pre and post-processing parallel algorithms

We consider tridiagonal systems of size n:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{4.1}$$

where $\mathbf{A}$ is a strictly diagonal dominant matrix: $\mathbf{A} = [l_i, d_i, r_i]$, $|d_i| > |l_i| + |r_i|$ and $l_1 = r_n = 0$. In order to solve this system in parallel, one can assume that there is a simpler matrix $\mathbf{Q}$ and a perturbed right-hand side $\mathbf{v}$ with the accompanied transformation matrix $\mathbf{T}$ such that

$$\mathbf{Tv} = \mathbf{b}, \tag{4.2a}$$

$$\mathbf{Qx} = \mathbf{v}. \tag{4.2b}$$

This means that the original tridiagonal matrix is decomposed in the form $\mathbf{A} = \mathbf{TQ}$. The structure of the algorithm depends on the choice of the matrix $\mathbf{Q}$. Wang's partitioning algorithm [Wan81] and parallel line solver [ABM04] belongs to this type of factorisation. This factorisation can be considered as pre-processing schemes where the r.h.s is perturbed such that the solution of the simpler block matrix gives the desired solution.

On the other hand, one can assume a decomposition in the form $\mathbf{A} = \mathbf{Q'S}$ and solve

$$\mathbf{Q'w} = \mathbf{b}, \tag{4.3a}$$

$$\mathbf{Sx} = \mathbf{w}. \tag{4.3b}$$

The earliest algorithm of this type applied to tridiagonal matrices is the SPIKE algorithm [SK78], [LPJN93]. The subsequent algorithms are the parallel partition algorithm (PPT) of Sun [SSN89] and the divide and conquer algorithm (DAC) of Bondeli [Bon91]. Even though all of these algorithms are derived differently, their implementations can be identical. In practice, these three algorithms solve Eq.(4.3b) by two substeps, and the solution $\mathbf{x}$ is obtained by

$$\mathbf{x} = \mathbf{w} - \triangle\mathbf{w}, \tag{4.4}$$

using the correction vector $\triangle\mathbf{w}$. Sun further exploits the decay of the correction vector and derives the Reduced PDD in [Sun95]. These algorithms can be considered as post-processing schemes where the first solution $\mathbf{w}$ is corrected by solving Eq.(4.3b). All of these algorithms use the block subdiagonal matrix of $\mathbf{A}$ as $\mathbf{Q'}$.

A parallel algorithm solving the tridiagonal system will be efficient if the preprocessing step (Eq.(4.2a)) and the post-processing step (Eq.(4.3b)) are easy to solve in parallel. The post-processing step of the algorithms in the previous paragraph solves a block tridiagonal system of p subsystems where each block is 4×4 matrix. In case of a strictly diagonal dominant system, and if the subsystem size is sufficiently large, the matrix $\mathbf{S}$ can be reduced to a p-block diagonal matrix and solved by nearest neighbour communications. For the preprocessing scheme, an equivalent algorithm with the same advantages can be formulated for the solution of Eq.(4.2a). The preprocessing matrix is given by $\mathbf{T} = \mathbf{S}^T$ and it is straight forward to show that for any algorithm of the post-processing type, there is an equivalent variant in the pre-processing type, using the same block matrix, $\mathbf{Q} = \mathbf{Q}'$. There will be small differences between these two variants of the equivalent algorithm due to floating point operations, but it can make significant differences in the programming of the application using these algorithms. For example, if the post-processing algorithm is called from a certain subroutine, the calling routine must have knowledge of the global topology or at least its nearest neighbours. On the contrary when the pre-processing algorithm is used, knowledge of the topology is not necessary and the calling routine can proceed as if it was working alone because solving $\mathbf{Q}$ does not require information of the neighbours unlike solving $\mathbf{S}$. The pre-processing operations can be done in a higher level subroutine which has the knowledge of the topology. Therefore the pre-processing algorithm is more suitable for numerical programs adopting the domain decomposition concept.

4.2.2 The algorithm

The interface-splitting algorithm proposed here belongs to the pre-processing type. For simplicity, we assume that $\mathbf{A}$ is a tridiagonal matrix of size $n = pm$ where p is the number of processors and m is some integer which will be the size of our subsystems. The k-th processor is holding the r.h.s. b_l, $(k-1)m + 1 \leq l \leq km$. Let $\mathbf{D}^k$ be the k-th block subdiagonal matrix of $\mathbf{A}$, i.e. $\mathbf{D}^k = \{\, a_{ij} \mid (k-1)m + 1 \leq i, j \leq km\}$ and $\mathbf{N}^k$ be the matrix $\mathbf{D}^k$ except the last row being replaced by that of the identity matrix. Instead of using the block diagonal matrix of $\mathbf{D}^k$ as the independent subproblem like previous algorithms, the interface splitting algorithm uses $\mathbf{N}^k$. The matrix $\mathbf{Q}$ and the matrix $\mathbf{N}^k$ are illustrated in Fig.4.1 below.

$$
\mathbf{Q} = \begin{pmatrix} \mathbf{N}^1 & & & \\ & \mathbf{N}^2 & & \\ & & \ddots & \\ & & & \mathbf{N}^p \end{pmatrix}
\qquad
\mathbf{N}^k = \left(\begin{array}{cccc|c} d_{c+1} & r_{c+1} & & & \\ l_{c+2} & d_{c+2} & r_{c+2} & & \\ & \ddots & \ddots & \ddots & \\ & & l_{c+m-1} & d_{c+m-1} & r_{c+m-1} \\ \hline & & & & 1 \end{array}\right)
$$

Figure 4.1: Matrix $\mathbf{Q}$ of the interface splitting algorithm and its N^k subdiagonal where $c = (k-1)m$.

The matrix $\mathbf{Q}$ is accompanied by the transformation matrix $\mathbf{T}$ which is just the identity matrix whose km-th rows are replaced by a vector $(\mathbf{y}^k)^T$. $\mathbf{T}$ can be inverted easily thus allowing to solve (4.2a) by

$$\mathbf{v} = \mathbf{T}^{-1}\mathbf{b}. \tag{4.5}$$

It can be shown that the matrix $\mathbf{T}^{-1}$ has exactly the same structure as $\mathbf{T}$ but the km-th row are changed to $(\mathbf{z}^k)^T$ and $\left(z_j^k\right)^T = c_{km,j}$, $1 \leq j \leq n$ with $\mathbf{C} = \mathbf{A}^{-1}$. This is equivalent to computing the solution x_{km} explicitly using the row inverse of $\mathbf{A}$. The solution vector $\mathbf{v}$ can be obtained by manipulating b only at the interface i.e. $\mathbf{v} = \mathbf{b} - \mathbf{f}$ using a sparse vector f whose component is non-zero only in the neighbourhood of the interface. The application of this approach for a general matrix is of course expensive and prone to numerical instability. However when A is strictly diagonal dominant, the calculations of $(\mathbf{z}^k)^T$ and $(\mathbf{z}^k)^T \cdot \mathbf{b}$ are stable and accurate. Nabben has shown in [Nab99] that the components of matrix $\mathbf{C}$ decay exponentially away from the diagonal. The key to the efficiency of the proposed algorithm is based on the truncation of the scalar product $(\mathbf{z}^k)^T \cdot \mathbf{b}$ to a certain bandwidth $2J$ which introduces an approximation error $\mathbf{e}^k$ for the r.h.s. of (4.5) (see later).

Thus the interface-splitting algorithm partitions the whole system of Eq.(4.1) into p smaller independent subsystems. Each subsystem, $\mathbf{x}^k$, $x_j^k = x_{(k-1)m+j}$, $1 \leq j \leq m$, is separated from the others by the interface x_m^k. This solution at the interface is explicitly computed by a truncated scalar product $\widetilde{x}_m^k = (\mathbf{z}^k)^T \cdot \mathbf{b}$. The dependencies between the subsystems are replaced by these precomputed solutions. Then the k-th subsystem of $\mathbf{A}$ takes the following form

$$\mathbf{N}^k \mathbf{x}^k = \mathbf{b}^k - \mathbf{f}^k - \mathbf{e}^k. \tag{4.6}$$

The components of the vector $\mathbf{f}^k$ are zero except the first and the last component which are given by $f_1^k = -l_m^{k-1} \widetilde{x}_m^{k-1}$ and $f_m^k = -\widetilde{x}_m^k - b_m^k$. The vector $\mathbf{e}^k$ contains the error of the approximation which is a result of the approximations introduced to the top and the bottom interfaces. Note that $\mathbf{e}^k$ are non-zero only in the first and the last components. The above equation is exact, however due the error term $\mathbf{e}^k$ will be omitted thus we solve for the following approximate solution

$$\mathbf{N}^k \widetilde{\mathbf{x}}^k = \widetilde{\mathbf{v}} = \mathbf{b} - \mathbf{f}. \tag{4.7}$$

The interface-splitting algorithm consists of the following four steps:

1. Setting up the subsystems: The k-th processor takes $\mathbf{D}^k$ from $\mathbf{A}$ and the $\mathbf{x}^k$ is distributed to its respective owner. The last row of $\mathbf{D}^k$ is replaced by that of the identity matrix to obtain $\mathbf{N}^k$

2. Computation of $\mathbf{z}^k$ *:* Each processor obtains $\mathbf{z}^k$ by solving $\mathbf{A}^T \mathbf{z}^k = \mathbf{e}_{km}$ where $\mathbf{e}_{km}$ is the vector whose km-th component is one and zero otherwise. Assign the coefficient vector $s_j^k = z_j^k$, $(km - J + 1) \leq j \leq km$ and send z_j^k, $(km + 1) \leq j \leq (km + J)$ to processor p_{i+1} where it will be stored as $\mathbf{t}^k$. These two new vectors will be used to compute the solution at the bottom interfaces.

3. Computation of the solution at the interfaces: Compute $a_0^k = \mathbf{t}^T \mathsf{v}_l$ and $a_1^k = \mathbf{s}^T \mathsf{v}_r$ where $\mathbf{v}_l$ and $\mathbf{v}_r$ are the first and the last J component of $\widetilde{\mathbf{v}}^k$, respectively. The impact of J on the accuracy of the algorithm will be explained later. For even processors, send a_0^k to p_{k-1} and receive a_1^{k-1} from p_{k-1}. For odd processors, send a_1^k to p^{k+1} and receive a_0^{k+1} from it. The solutions $\widetilde{x}_m^k$ at the lower interfaces is given by $\widetilde{x}_m^k = a_1^k + a_0^{k+1}$. The right-hand side v_1^k of the upper interface is thus $v_1^k = b_1^k - (a_0^k - a_1^{k-1})l_1^k$.

4. Parallel solution of $\mathbf{x}^k$*:* Each processor can solve its own independent subsystem: $\mathbf{N}^k\mathbf{x}^k = \mathbf{b}^k$. Note that the solution at the last line can be skipped as it is already known.

Step 1 and 2 are only performed once for each matrix $\mathbf{A}$, therefore when we are solving multiple right-hand sides the cost in these two step can be neglected. However, computing $\mathbf{z}$ this way requires the knowledge of the global topology. This can be problematic for a complex code where many grid blocks of different sizes can be connected together. Because the components of $\mathbf{z}^k$ are exponentially decays , it is implied that the components of $\mathbf{A}$ far away from the k-th interface only have small influences on $\mathbf{z}^k$ and thus can be neglected, for example $a_{ij} \in \{a_{ij} \mid |i - km| > J + \theta,\ 1 \leq j \leq n,\ \exists\theta < J\}$. Therefore it is sufficient to solve for a portion of $\mathbf{z}^k$ from a smaller matrix $\mathcal{A}^k$, a diagonal block submatrix enclosing the desired k-th interface. This simplification of step 2 significantly reduces the complexity of a coding and makes the interface-splitting algorithm much more efficient in the case of single right-hand side problem. The selection of safety boundary width θ will be described later.

The interface-splitting algorithm will be equivalent to a direct method up to a small factor of machine accuracy (ϵ) if $\mathbf{z}^k$ decays below ϵ within the subsystem i.e. $z_i^k < \epsilon$ for $|i - km| > J$ and $J \leq m$. The theory for the exponential decay of the component of $\mathbf{z}_k$, for non-symmetric banded matrices is well established in [Nab99]. The Accuracy of the interface splitting algorithm depends on how accurate the x_m^k. The factors determining the accuracy of x_m^k are (i) the row diagonal dominance factor $\sigma_i = d_i/(|l_i| + |r_i|)$ and (ii) the approximation bandwidth J used for the computation of x_m^k . The relationship of these two factors to the accuracy of the algorithm will be described later in section 4.4. In the next section we discuss the accuracy of the interface-splitting algorithm and an estimation for the bandwidth J necessary for a desired truncation threshold ε_c.

4.3 Accuracy Analysis

Solving Eq.(4.6) for $\mathbf{x}^k$ is equivalent to solving the following $(m+1)\times(m+1)$ system

$$\mathbf{F}^k\mathbf{u}^k = \mathbf{z} \tag{4.8}$$

for $\mathbf{u} = [x_m^{k-1} \quad \mathbf{x}^{k^T} \quad x_m^k]^T$ with

$$\mathbf{F} = \left(\begin{array}{c|c} 1 & \mathbf{0}^T \\ \hline \mathbf{w} & \mathbf{N}^k \end{array}\right), \quad \mathbf{z} = [x_m^{k-1} \quad b_1 \quad b_2 \quad \cdots \quad b_{n-1} \quad x_m^k]^T \text{ and } \mathbf{w} = [l_1^k \quad 0 \quad \cdots \quad 0]^T. \tag{4.9}$$

Due to the structure of $\mathbf{N}^k$, it follows that

$$\mathbf{F}^k = \left(\begin{array}{c|c|c} 1 & \mathbf{0}^T & 0 \\ \hline \mathbf{w} & \mathbf{G}^k & \mathbf{y} \\ \hline 0 & \mathbf{0}^T & 1 \end{array}\right). \tag{4.10}$$

The interface splitting algorithm introduces an approximation to x_m^{k-1} and x_m^k and the system $\mathbf{F}\widetilde{\mathbf{u}}^k = \widetilde{\mathbf{z}}^k$ is solved instead using $\mathbf{z} = [\widetilde{x}_m^{k-1} \quad b_1 \quad b_2 \quad \cdots \quad b_{n-1} \quad \widetilde{x}_m^k]^T$. These approximations create errors which can be written as follows

$$\mathbf{h}^k = \widetilde{\mathbf{u}}^k - \mathbf{u}^k \tag{4.11}$$

$$\mathbf{F}\mathbf{h}^k = \widetilde{\mathbf{z}}^k - \mathbf{z}^k \tag{4.12}$$

$$\mathbf{F}\mathbf{h}^k = \begin{pmatrix} e_m^{k-1} \\ \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \\ e_m^k \end{pmatrix} \tag{4.13}$$

The error at the inner indices (h_i^k, $1 < i < m+1$) is thus a sum of the error propagated from both interfaces. The position of the maximum error is given by the following proposition.

Proposition 1. *Errors of the interface-splitting algorithm for diagonal-dominant tridiagonal matrix are maximal at the interfaces.*

Proof. The errors of the interface-splitting algorithm h_i^k satisfy

$$h_i^k = -l_i h_{i-1}^k - r_{i+1} h_{i+1}^k.$$

The error of the inner indices i.e. h_{i-1}^k, $1 < i < m$ is not larger than the maximum error introduced at the interface because

$$\begin{aligned} \left|h_i^k\right| &\le \left|l_i h_{i-1}^k\right| + \left|r_i h_{i+1}^k\right| \\ \left|h_i^k\right| &\le (|l_i| + |r_i|) \max(h_{i-1}^k, h_{i+1}^k) \\ \left|h_i^k\right| &< \max(h_{i-1}^k, h_{i+1}^k) < \max(e_m^{k-1}, e_m^k). \end{aligned}$$

□

In what follows we assumes that the threshold ε_c was used to truncate the vector $\mathbf{z}^k$ and J is the minimum $j > 0$ satisfying $z^k_{i\pm j} > \varepsilon_c$. The maximum error of the interface-splitting algorithm thus consists of the truncated terms and the round-off error in the calculation of the dot product in step 3. These errors are however bounded by a small factor of the cut-off threshold.

Theorem 1. *Let* $\mathbf{A}$ *be the matrix of size* $n = pm$ *mentioned in Eq.*(4.1) *and* $\left(\mathbf{z}^k\right)^T$ *be the* km*-th row of the inverse of* A*, which can be used to compute the* m*-th solution of the* k*-th subsystem. Then the maximum error of the interface-splitting algorithm is bounded by*

$$e_{max} = \left(2L + \frac{1}{2}\right)\epsilon + \left(L - \frac{1}{2}\right)\varepsilon_c|\mathbf{b}|_\infty, \tag{4.14}$$

where ε_c *is the threshold for the cut-off coefficient,* ϵ *is the machine accuracy and* L *is the bandwidth in which the magnitude of the coefficients is reduced by one significant digit. Note that* $|b|_\infty$ *is the maximum norm of* $\mathbf{b}$.

Proof. Assuming that the right-hand-side vector can be represented exactly by machine number and let μ^k be the error of representing the exact $\mathbf{z}^k$ by $\widehat{\mathbf{z}}^k$, i.e. $\widehat{\mathbf{m}u}^k = \widehat{\mathbf{z}}^k - \mathbf{z}^k$. The hat here represents the exact numerical value that only differ from the exact one due to machine operations. Let $fl(x)$ be the floating point operation on x, the numerical computation of $\widehat{x}^k_m$ by $\mathbf{b}^T\widehat{\mathbf{z}}^k$ is given by

$$\widehat{x}^k_m = fl\left(\sum_{j=1}^{n}\left(z^k_j + \mu^k_j\right)b^k_j\right) \tag{4.15}$$

$$\widehat{x}^k_m = fl\left(\sum_{j=1}^{km}\left(z^k_j + \mu^k_j\right)b^k_j\right) + fl\left(\sum_{j=km+1}^{n}\left(z^k_j + \mu^k_j\right)b^k_j\right) + \delta, \tag{4.16}$$

with $|\delta| < 2\epsilon$. Because $\mathbf{z}^k$ decays exponentially, there is a smallest number L such that $z^k_{j-L} < \frac{1}{10}z^k_j$ for $j < km$ and $z^k_{j+L} < \frac{1}{10}z^k_j$ for $j > km$. This means that if the machine accuracy ϵ is in $(10^{-\eta+1}, 10^{-\eta})$ then Eq.(4.16) is equivalent to

$$\widehat{x}^k_m = fl\left(\sum_{j=km-\eta L}^{km}\left(z^k_j + \mu^k_j\right)b^k_j\right) + fl\left(\sum_{j=km+1}^{km+\eta L+1}\left(z^k_j + \mu^k_j\right)b^k_j\right) + \delta. \tag{4.17}$$

Let $\widehat{a}^k_0$ and $\widehat{a}^k_1$ be the first and the second sum in Eq.(4.17). Again, due to the decaying nature of $\mathbf{z}^k$, the round-off errors only affect the scalar product on the first L largest terms, thus the first sum is reduced to

$$\widehat{a}_0^k = fl\left(\sum_{j=km-\eta L}^{km} \left(z_j^k + \mu_j^k\right) b_j^k\right) \tag{4.18}$$

$$\widehat{a}_0^k = fl\left(\sum_{j=km-\eta L}^{km} z_j^k b_j^k\right) + fl\left(\sum_{j=km-L}^{km} \mu_j^k b_j^k\right) \tag{4.19}$$

and because the exponential decay is bounded by a linear decay, we arrive at the following error bound

$$\widehat{h}_m^k = \widehat{x}_m^k - \left(\widehat{a}_0^k + \widehat{a}_1^k\right) \tag{4.20}$$

$$< fl\left(\sum_{j=km-\eta L}^{km} z_j^k b_j^k\right) + fl\left(\sum_{j=km+1}^{km+\eta L+1} z_j^k b_j^k\right) + \left(L - \frac{1}{2}\right)\epsilon + \delta \tag{4.21}$$

$$< \left(2L + \frac{1}{2}\right)\epsilon. \tag{4.22}$$

Now, we have a bound in case of using the full $\mathbf{b}^T\mathbf{z}^k$. It is now straight forward that error of the truncated scalar product is bounded by

$$h_m^k = \widetilde{x}_m^k - x_m^k \tag{4.23}$$

$$= \widehat{h}_m^k - fl\left(\sum_{j=1}^{km-J} z_j^k b_j^k\right) - fl\left(\sum_{j=km+J+2}^{n} z_j^k b_j^k\right) \tag{4.24}$$

$$< \left(2L + \frac{1}{2}\right)\epsilon + \left(L - \frac{1}{2}\right)\varepsilon_c |b|_\infty. \tag{4.25}$$

□

If one aims to apply the interface-splitting algorithm in simulation-based applications, the cut-off threshold ε_c can be kept at certain digits below the truncation error. These truncated terms are usually far larger than the machine accuracy thus ε_c can be much larger than the machine accuracy. In such a case, the maximum error of the algorithm is bounded by $L\varepsilon_c$.

4.4 Complexity and performance

The complexity of interface-splitting algorithm depends on the magnitude of the cut-off threshold which reflects upon two parameters, the 1-digit decays bandwidth L and the necessary bandwidth J achieving the cut-off threshold. The numbers J and L only depend on two factors, the cut-off threshold and the degree of diagonal dominance of the system

($|d_i|/(|l_i|+|r_i|)$). For Toeplitz system $\mathbf{T} = [1,\ \lambda,\ 1]$, the entry of the matrices of the LU-decomposition of this system converges to certain values and the bandwidth J can be deduced from these values. The necessary bandwidth J achieving the cut-off threshold ε_c can be approximated for this special system by :

$$J = -\frac{ln\ \varepsilon_c}{ln\ \left(\frac{1}{2}\left(|\lambda| + \sqrt{(\lambda^2-4)}\right)\right)}. \tag{4.26}$$

For instance, J equal to 7 and 27 for 10^{-4} and 10^{-15} when $\lambda = 4$. These two numbers are much smaller than the usual size of the subsystem used in scientific computing. The inter-face splitting algorithm is an approximate method but it can be made equivalent to other direct methods by setting ε_c to machine accuracy.

In step 2, we have to solve a linear system for the km-th row of the $\mathbf{C}$. This system has to be larger than $2J$ such that the coefficients of $(\mathbf{z}^k)^T$ are sufficiently accurate. In this work we solve a system of size $2J + 4L$ which gives a 2-digits accurate representation of the smallest coefficient of the truncated $(\mathbf{z}^k)^T$. It would be rare that one would satisfy with the error larger than 10^{-4}. It is thus safe to assume that $L = J/4$. This value of L is substituted in to the operation counts of the interface-splitting algorithm and the results are shown in Tab.4.1. Note that the error threshold smaller than this will lead to a smaller L for the same level of the accuracy of the coefficient vector and consequently a smaller number of operations relative to J. In this table we list also the communication time for $\mathcal{N}$ which can be expressed by a simple model as $\tau_{com} = \alpha + \beta Q$ where α is the fixed latency and β is the transmission time per datum. For the single r.h.s systems, we assume that the system is so large (otherwise we would not need a parallelisation) such that each processor only knows their own r.h.s and the submatrix $\mathbf{A}^k$. This leads to a higher communication compared to [Bon91] and [Sun95] in which the global matrix is known to all processors. If the same assumption is taken here, the communication will be reduced to $\alpha + 2\beta$. For multiple right-hand side system, we do not count the computation in the first two steps. This leads to the absolute speedup of single r.h.s. problem ($\mathcal{S}_1$) and multiple r.h.s. problem ($\mathcal{S}_\gamma$) given in Eq.(4.27) and Eq.(4.28) when n is much greater than p and the communication cost is small.

$$\mathcal{S}_1 = \frac{p}{1 + 3\frac{J}{m}} \tag{4.27}$$

$$\mathcal{S}_\gamma = \frac{p}{1 + 0.8\frac{J}{m}} \tag{4.28}$$

The key number determining the performance of the interface-splitting algorithm is the ratio J/m. When this ratio is small, an excellent speedup can be expected otherwise the algorithm suffers a penalty. For example, the efficiency will drop from 92% to 56% when J

is increased from $0.1m$ to m on multiple right-hand side system. According to Tab.4.1, the interface-splitting algorithm is slightly faster than the reduced-PDD algorithm [Sun95] for multiple right-hand side problems. Our algorithm is also faster for a single right-hand side system, when $J < 0.375m$. Therefore the interface splitting algorithm is competitive both as approximate and direct methods (close to machine accuracy).

It is a common practice that the subsystem size should be reasonably large such that the speedup from load distribution justifies the communications and other overhead. This algorithm assumed that $J \leq m$, if this does not hold the program adopting this algorithm should issue a warning or an error to the user.

System	Matrix	Sequential	Interface-Splitting Algorithm	
			Computation	Communication (τ_{com})
Single r.h.s	Nonperiodic	$8n-7$	$8\frac{n}{p}+24J-17$	$\alpha+9\beta J/2+2$
	Periodic	$14n-16$	$8\frac{n}{p}+24J-17$	$\alpha+9\beta J/2+2$
Multiple r.h.s	Nonperiodic	$(5n-3)\gamma$	$\left(5\frac{n}{p}+4J-4\right)\gamma$	$\alpha+2\gamma\beta$
	Periodic	$(7n-1)\gamma$	$\left(5\frac{n}{p}+4J-4\right)\gamma$	$\alpha+2\gamma\beta$

Table 4.1: Computation and communication costs of the interface-splitting algorithm for solving a linear system of order n on p processors with single and γ right-hand sides.

4.5 Results

In this section, we present results of the interface-splitting algorithm applied to single and multiple r.h.s. on two parallel computers. First the accuracy of the interface-splitting algorithm is studied for a simple matrix in an approximation problem. In the second step, we evaluate its performance on a Linux cluster. Finally, performance and the scalability of the interface-splitting algorithm is presented and compared with ScaLAPACK package.

Table 4.2 shows the error of the inter-face splitting algorithm applied to a matrix $[1,4,1]$ encountered in approximation problem such as spline interpolation, compact differentiation [Lel92] and compact deconvolution [Kob99]. In this table we consider the case of differentiation of $f = sin(20\pi x)$ on $x = [0,1]$. The unknown are placed at $x_i = ih,\ 0 \leq i \leq 251$. This problem is solved using three partitions. The table shows that the bound given in Eq.(4.14) is not so far from the actual error as seen in the first and the last row. Sometimes the error can be much smaller than the bound due to the cancellation of the neglected terms. The error of the algorithm using the smallest $J = 7$ is already much smaller than the error of the local truncation error of the differentiation. If one consider the fact that there are 25 grid points per wavelength in this problem which is already very fine, we expect that the smallest J here should be adequate for most simulation-based applications. If higher accuracy is required, the band can be expanded as necessary. Note that the bandwidth J here is dependent only on the diagonal-dominance of the matrix surrounding the interface and the choice of cut-off threshold ε_c. The number of unknowns has no influence on this bandwidth.

In what follows, we demonstrate the accuracy of the proposed algorithm when applied to a matrix with non-constant coefficient. In order to have a reproducible test, we solve the test matrix $\mathbf{A} = [sin(i), 2(|sin(i)| + |cos(i)|), cos(i)]$ of order 1000 with $b_i = 1$ on 4 processors. The off diagonal coefficients of this matrix vary relatively fast. Their signs change approximately once every three rows. The numerical error of the algorithm shown in Tab.4.3 indicates the non-constant coefficients of the matrix do not have adverse effects on the algorithm.

J	ε_c	$\frac{1}{\lvert b\rvert_\infty}\lvert\tilde{f}'_{seq} - \tilde{f}'_{par}\rvert_\infty$	$\lvert f'_{exact} - \tilde{f}'_{seq}\rvert_\infty$
7	9.9167e-05	7.1325e-06	
15	2.6350e-09	7.2559e-11	1.7394e-03
27	3.6092e-16	3.8095e-17	

Table 4.2: Normalised error of the interface-splitting algorithm (column 2) subjected to different cut-off threshold (ε_c). The error of the differentiation (column 3) is shown for a comparison.

J	$\frac{1}{\lvert b\rvert_\infty}\lvert x_{seq} - x_{par}\rvert_\infty$
7	1.41E-005
15	2.06E-011
18	4.66E-014
20	4.44E-016
27	4.44E-016

Table 4.3: Normalised error of the interface-splitting algorithm applied to system with non-constant coefficients using different bandwidth.

Fixed size speedup of the algorithm is studied by solving a multiple r.h.s. system of order 25600 and 10^4 right-hand sides. In all following tests, the interface bandwidth J is set to 9. The absolute fixed size speedup of the interface-splitting algorithm on a Linux cluster in Fig.4.2 showing an almost ideal speedup. Here the absolute speedup is defined by the solution time on a single processor using the fastest sequential algorithm (Gaussian elimination) divided by the time used by the interface splitting algorithm. Because the algorithm only communicates to the nearest neighbour, it is therefore highly scalable. When the subsystem size is sufficiently large i.e. $J/m \ll 1$, the overhead will be small and the ideal absolute speed up can be achieved. At $p = 64$ the speedup is even slightly better than the ideal one, which could be attributed to a better caching.

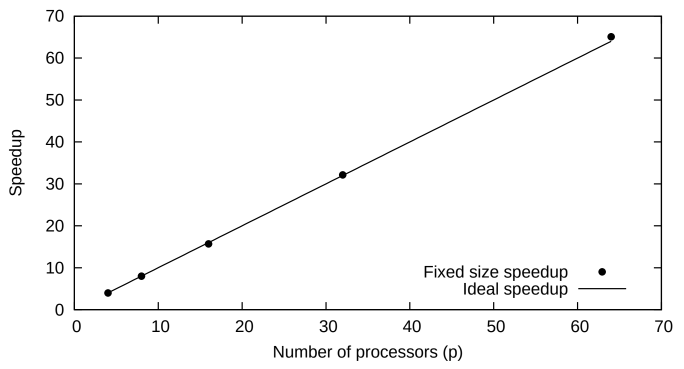
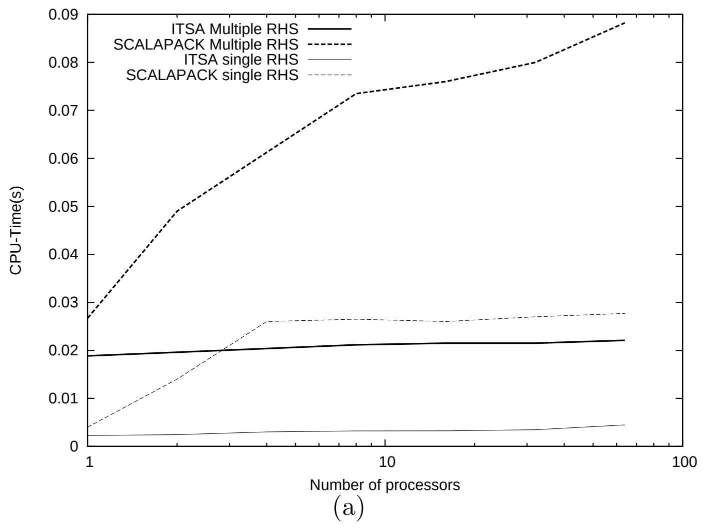


Figure 4.2: Absolute speedup ($\mathcal{S}_\gamma$) of multiple right-hand side problem on Linux cluster.

The scalability of the algorithm is studied by measuring runtimes on Altix4700 for two types of problem, a single r.h.s and a multiple r.h.s.. Here we consider a scaled problem where the total system size grows linearly with the number of processors i.e. $n_{tot} = pn_{sub}$. The subsystem size is set to 10^6 in a single r.h.s problem. For multiple right-hand side problems, it is set to 100 with 10^4 right-hand-sides. The results of the test are shown in Fig.4.3(a). The runtime of the interface-splitting algorithm grows approximately linear with the logarithm of the number of processors in both types of problem. This can be attributed to the fat-tree topology of the interconnection of the machine. Note that the number of unknowns in both problems are equal i.e. 10^6. The difference in CPU-time seen here is solely the communication time. Here we can not achieved an ideal speedup unlike on the cluster. This is of course a limitation imposed by the hardware. It is obvious that, the ScaLAPACK is not scaled well as the our algorithm. The CPU-time is approximately doubled when increased the number of processor from one to two. This is in accordance to the increase in complexity of the parallel algorithm used by ScaLAPACK. Interestingly the differences in the CPU-time of the multiple right-hand side problems are increasing sharply when the number of processors is increased. This indicates that the ScaLAPACK is very sensitive to the characteristics of the interconnection network unlike the proposed algorithm.

In Fig.4.3(b) efficiencies of the inter-face splitting algorithm and the ScaLAPACK are presented. On a single r.h.s., the efficiency of the proposed algorithm falls to 50% at $p = 64$ because the increase in communication time that occupies 50% of the CPU-time there.



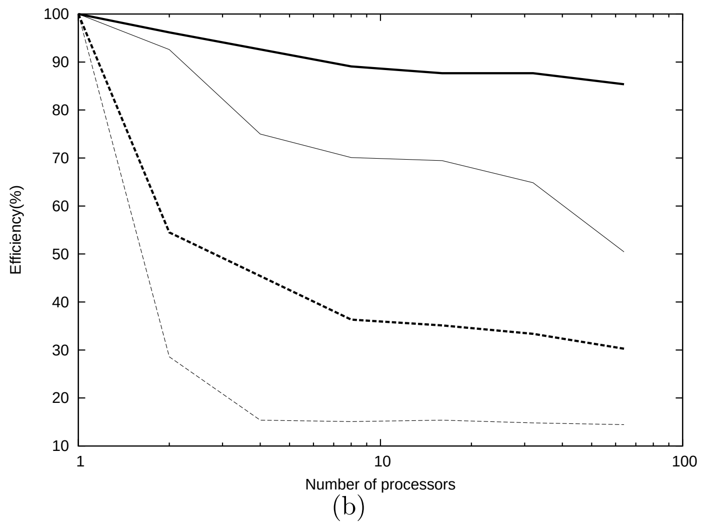


Figure 4.3: Running time (a) and the scaled-efficiency (b) of the interface-splitting algorithm on Altix4700 compared to ScaLAPACK: .

Interestingly, the increased communication time does not affect the solving time of the SCALAPCK (thin dash line) much, especially the single right-hand side problem. This is because the increase in the cost of computation is dominated. In the multiple r.h.s problem, both algorithms enjoy better efficiencies. At $p = 64$, the interface-splitting algorithm deliver an 85% efficiency compared to 30% of the ScaLAPACK. On both problems, the interface-splitting is at least four-times faster than the ScaLAPACK.

4.6 Conclusion

We have presented the inter-face splitting algorithm solving diagonal dominant tridiagonal systems. The accuracy of the algorithm depends on the diagonal dominance of the matrix and not the order of the matrix. This algorithm is an approximate algorithm but the user have full control over the accuracy. It can be used equivalently to the direct method if desired, provided that the subsystem size is sufficiently large. This algorithm is highly efficient and scalable, especially for multiple right-hand side problems. Excellent efficiencies are obtained on two parallel computers. The algorithm is at least four-times faster than the ScaLAPACK which used a direct algorithm. Unlike the existing algorithms, this algorithm fully utilise the bidirectional link which allows these communications to overlap and thus a low communication overhead can be expected from this algorithm.

5 Approximate Projection Method

In chapter 3, we have numerically verified that the fourth-order solution of pressure is essential for a fourth-order convergence rate of the velocities. Despite the fact that the fourth-order projection method used earlier can deliver excellent results and ultimately leads to a reduction of computational time. This was partially due to the cost of the direct solver of the fourth-order projection was not fully revealed. Because, the FFT were used in the xy-plane, the cost of the fourth-order and the second-order projection are approximately the same in this plane. We only had to solve the seven-banded system in the wall-normal direction for each wave number. The major cost in this process was the FFT and thus the cost of the fourth-order projection came at marginal cost (see Tab.3.8). In a general situations, the flow may not have any homogeneous directions and thus the FFT cannot be used. In these situations, we have to deal directly with the 19-banded matrix given by the fourth-order projection.

At first glance, it would seem that this is a simple task. One can just construct a sparse matrix out of this discrete Laplacian and use a certain type of preconditioned Krylov subspace algorithm to solve it. This approach has been tried and convergence of the iterative solution is obtained. However, the performance was not satisfactory due to two reasons, the matrix is singular and the precondition step is too expensive. Since the matrix is singular, we have to use BiCGSTAB or GMRES with restart which are quite expensive. Due to the fact that turbulent flows consist of a wide range of length scale, preconditioning is necessary. However, the matrix of the fourth-order projection method is new and thus there are no specialized preconditioner for it. The incomplete LU decomposition have been tried for the preconditioner, but a truncating threshold that can give a reasonable convergence rate leads to a large number of non-zero entry. Consequently, the memory requirement is too much to be used for large scale computing.

In the fourth-order projection method, one has to solve the special 19-point stencils. Comparing the size of this stencil to that of the standard fourth-order approximation of the Laplacian which is a 13. This 19-point stencil is expensive. The number of operation count is increased by 46% just in the computation of the residual alone ($div - \mathbf{L}p$), compared to the 13-points stencil. Even though this wide stencil has certain advantages due to the property of the projection method, this wide stencil spans over 3 cells in each direction which leads to a need of 3 ghost cells. This can be a big problem in the implementation, especially when one wish to upgrade an existing second-order code to a fourth-order scheme. In second-order codes, 1 or 2 ghost cells are usually used. These second-order codes, especially those with capability in solving different problems, such as particulate flow, scalar transport, chemical reactions and so forth, can be continuously developed over several decades and the code can be an assemble of a thousand of subroutines. This ghost cells parameter can be explicitly hard-wired anywhere in the code. Re-engineering them to allow other number of ghost cells is very difficult, time consuming and prone to errors. Therefore, it is preferable to deal with the 13-points stencil.

In this chapter we present the divergence-free approximate projection method which solves a 13-points stencil without compromising the fourth-order convergence rate of the spatial approximations. .

5.1 Background

A unique feature of the projection method is that the residual given by the corresponding discrete Laplacian of the projection method is the mass imbalance that will reside in the velocity fields after the velocity correction. This means when the discrete Poisson equation is solved by a direct method, the divergence of the velocity field after the correction is close to machine accuracy. During the solution process, only the pressure and the residual are needed. In pressure-Poisson formulation, this is not the case. The residual in the Poisson equation reflects the discrete mass-imbalance only up to the local truncation error. If one wish to achieve a certain value of mass conservation, it is necessary to correct the velocity and then recompute the divergence. If the result is unsatisfactory, then the Poisson equation must be solved again. In general cases, discrete Poisson equations are solved by iterative methods. This leads to a dual iteration in which the inner loop solves for the pressure and the outer loop check for the discrete mass imbalance. This inner-outer iteration must be repeated until the desired divergence is obtained. It is obvious that the pressure-Poisson formulations are at disadvantage because six variables (pressure, divergence, residual and the three velocities) must be continuously accessed instead of just two variables in the projection method. Memory bandwidth and cache misses can increase the computational time significantly.

On the other hand, projection methods only allow a unique form of the discrete Laplacian given by $\mathbf{L} = \mathbf{DG}$ where $\mathbf{D}$ and $\mathbf{G}$ are the discrete divergence and gradient, respectively (see section 2.5 for the derivation). On collocated grids, this restriction give rise to discrete Laplacians that are wider than usual and thereby increases the costs of computation. In second-order projection method, the number of the stencil point of the discrete Laplacian in 3D is 13 instead of 7. This is however a performance problem. In fact, a more serious problem of the projection method on collocated grids is the local-decoupling problem of the pressure. Here, the local-decoupling here means that the solution on odd cells are independent from the solution on even cells. This decoupling allows an unphysical oscillation pattern in the pressure field. Approximate projection method [ABS96] [ABC00] have been proposed to overcome these two specific problems. However the approximate projection methods proposed so far do not satisfy the discrete divergence exactly but only up to the local truncation errors. Loosing the perfect mass-conservation property, these approximate projection methods become similar to the pressure-Poisson formulations and do not offer any significant advantages.

On staggered grids, the situation is simpler. The Laplacian given by the projection method has a minimum number of stencil point and it does not suffer from the local-decoupling problem. Therefore, approximate projection for staggered grids does not exist in the second-order context. The situation is however changed when we move to fourth-order scheme. As mentioned earlier that the fourth-order projection method needs to solve the 19-points Laplacian while the 13-points stencil is sufficient to reach a formal fourth-order convergence rate. Therefore, if we solved the 13-points stencil and correct the velocity by fourth-order gradient, we can expect a fourth-order convergence rate in the velocity and an improvement in the performance can be gained here. However, the corrected velocity will not be divergence-free. Nevertheless, this shed some light on the possibility that, there could be some 13-points stencil which lead to a divergence-free velocity after the correction.

Recent developments in higher-order scheme for Navier-Stokes Equation such as those in [ARM01] and [Kni08] including our investigations require that the divergence and gradient approximations are as accurate as those of the convective and diffusive terms, in order to achieve the full potential of higher-order schemes. This requirement widen the Laplacian stencil and in some cases, the discrete Laplacian becomes a full matrix. For example, when the implicit schemes were used to approximate the divergence or the pressure, the Laplacian will become a full matrix. However, on staggered grids, the explicit approximations of divergence and gradient on staggered grids is accurate enough and can accommodate compact schemes without compromising the accuracy. The full matrix of the discrete Laplacian such as the one used in [ARM01] is thus not needed. Nagarajan *et al.* [NLF03] comment that staggered grids have better conservation property and it is more robust than the collocated counterpart. Therefore efficient projection method for higher-order schemes on staggered grids deserves more attention.

This chapter is organised as follows. First we outline advantages and disadvantages of projection methods then the Helmholtz-Hodge decomposition is introduced follows by the definition of consistent projection method. Effects of the choices in the discretisations of the pressure and the divergence are analysed. After that, the approximate projection method is presented and its local truncation error analysis is performed. Convergence and accuracy of the proposed method is then evaluated using numerical simulations of laminar flow. Finally, the applicability to realistic flows is demonstrated by direct numerical simulations of a turbulent channel flow.

5.2 Consistent Projection methods

We solve the incompressible Navier-Stokes equations (NSE) for Newtonian fluid. For convenience, we rewrite the NSE below

$$\frac{\partial \mathbf{u}}{\partial t} = (\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \triangle \mathbf{u} - \frac{1}{\rho} \nabla p \tag{5.1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{5.2}$$

These equations are non-linearly coupled and difficult to solve. Projections methods allow a simpler numerical procedure to be used. Projection methods are based on the Helmholtz-Hodge decomposition theorem which can be defined as follows

Theorem 2. *(Helmholtz-Hodge decomposition) A vector field $\mathbf{u}^*$ on a simple connected domain Ω with smooth boundary $\partial\Omega$ can be uniquely decomposed in the following form [BCM01]*

$$\mathbf{u}^* = \mathbf{u} + \nabla \phi \tag{5.3}$$

with $\nabla \cdot \mathbf{u} = 0$ in Ω and $\oint_{\Omega} \mathbf{u} \cdot \mathbf{n} dS = 0$

This theorem states that any vector fields can be decomposed into two components, the divergence-free vector field $\mathbf{u}$ and the curl-free vector field $\nabla\phi$. When apply this theorem to NSE, one can integrate Eq.(5.1) without enforcing the mass conservation and then project these provisional velocities back to the divergence-free space.

The momentum equation can be integrated in time in different ways. In this work the low storage third-order Runge-Kutta (RK) time integration of Williamson [Wil80] is considered. Let $\mathcal{H}(\mathbf{u}^k)$ be the contribution of convective and diffusive terms at the k-th substep of the time $t^n = n\triangle t$. Each substep k, $1 \leq k \leq 3$ of the third-order Runge-Kutta time integration for the time step n is given by

$$\mathbf{q}^k = a_k \mathbf{q}^{k-1} + \left(\mathcal{H}(\mathbf{u}^{k-1}) - \frac{1}{\rho} \mathbf{G} p^{k-1} \right) \triangle t \tag{5.4}$$

$$\mathbf{u}^{k*} = \mathbf{u}^{(k-1)} + b_k \mathbf{q}^k \tag{5.5}$$

where the variables of the zeroth substep are those of the n-th time integration i.e. $p^0 = p^n$ except for q^0 which is set to zero. The usage of the numerical superscript in this chapter e.g. p^0 is referred to the corresponding RK-substep. There will not be a reference to other step of the time integration. Therefore, the notation p^1, p^n and p^k will be clear in the context.

The provisional velocity $\mathbf{u}^{k*}$ above is usually not divergence-free and the mass imbalance can be accumulated from each substep and degrade the quality of the final solution. This problem can be cured by applying the projection operator $\mathbf{P} = \mathbf{I} - \mathbf{G}(\mathbf{DG})^{-1}\mathbf{D}$ to $\mathbf{u}^{k*}$. This projection operator requires a solution of Poisson equation $\mathbf{L}p = \triangle t\mathbf{Du^{k*}}/\rho$ under the consistent Laplacian operator $\mathbf{L} = \mathbf{DG}$. The projection method appended after Eq.(5.5) is given below

$$\mathbf{DG}\phi = \frac{\rho}{\triangle t}\mathbf{Du}^{k*} \tag{5.6}$$

$$\mathbf{u}^k = \mathbf{u}^{k*} - \frac{\triangle t}{\rho}\mathbf{G}\phi \tag{5.7}$$

$$p^k = p^{k-1} + \phi \tag{5.8}$$

The resulting $\mathbf{u}^k$ will have the divergence exactly as the values of the residual $\frac{\rho}{\triangle t}\mathbf{Du}^{k*} - \mathbf{DG}\phi$ in Eq.(5.6), when solve analytically. The above projection method obeys the Helmholtz-Hodge decomposition exactly in a discrete sense i.e. $\mathbf{u}$ satisfies the discrete divergence operator close to to machine accuracy and essentially the same for the conservation of the vorticity. This projection method is thus called *exact projection method* by Almgren, Bell and Crutchfield [ABC00]. It must be emphasised that the term exact here does not mean that the projection leads to the exact solution of p and $\mathbf{u}$.

In the context of fourth-order, $\mathbf{D}$ and $\mathbf{G}$ in equations (5.4), (5.6) and (5.7) can be approximated by second-order or fourth-order schemes. This leads to four possible choices of the Laplacian namely (i) $\mathbf{D}_2\mathbf{G}_2$, (ii) $\mathbf{D}_2\mathbf{G}_4$, (iii) $\mathbf{D}_4\mathbf{G}_2$ and (iv) $\mathbf{D}_4\mathbf{G}_4$. They are called here *composite Laplacians* for an obvious reason. We call the projection method solving one of these Laplacians and correcting the velocity using the respective discrete gradient *consistent projection method.* The consistent projection methods solving (i) and (iv) are the second-order and fourth-order exact projection methods, because they obeys the Helmholtz-Hodge decomposition up to their corresponding order of accuracy. The consistent projection methods solving (ii) and (iii) are thus called *mixed-order projection method.*

In [ARM01], [Kni08] and earlier in chapter 3 the two exact projection methods(i&iv) are studied ($\mathbf{D}_2\mathbf{G}_2$ and $\mathbf{D}_4\mathbf{G}_4$). The conclusions from these works indicate that the second-order exact projection method is unable to deliver the fourth-order convergent rate in both velocity and pressure and the fourth-order projection method is necessary. We have not found the investigation of mixed-order consistent projection methods in the literature.

5.3 Analysis of consistent Projection methods

Projections methods consist of three steps: (i) approximation of divergence, (ii) solution of pressure and (iii) correction of pressure and velocities. In this section we analyse how the approximations of divergence and pressure gradient affect the accuracy of the solutions.

5.3.1 Effect of the divergence approximation

Let us consider a two dimensional case. Assuming that velocity field $\mathbf{u} = u(x,y)\mathbf{i} + v(x,y)\mathbf{j}$ can be decomposed in to a Fourier series. The u-velocity on the physical space can then be written as $u(x,y) = \sum_{k=0}^{k_{max}} \hat{u}_{(kx,ky)} exp(i(k_x x + k_y y))$ and equivalently for $v(x,y)$. Suppose the initial velocity is divergence free, that is the velocity field satisfies :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \tag{5.9}$$

Applying the Fourier transform to the above equation gives the following relation for each wave number pair:

$$ik_x \hat{u}_{(kx,ky)} + ik_y \hat{v}_{(kx,ky)} = 0. \tag{5.10}$$

When a polynomial-based differentiations was used to approximate the divergence in Eq.(5.9), the resulting divergence may not be equal to zero because of the truncation error of the approximation. Consider below, the Fourier transform of a polynomial-based approximation to Eq.(5.9) :

$$ik_x T_x(kx)\hat{u}_{(kx,ky)} + ik_y T_y(ky)\hat{v}_{(kx,ky)} = \hat{q}(k_x, k_y), \tag{5.11}$$

where $\hat{q}(k_x, k_y)$ is the Fourier components of the divergence with respect to $T_x(k_x)$ and $T_y(k_y)$, the transfer function of the differentiation in x-direction and y-direction. One can make use of continuity equation and arrive at

$$\left(T_y(ky) - T_x(kx)\right) ik_y \hat{v}_{(kx,ky)} = \hat{q}(k_x, k_y), \tag{5.12}$$

Suppose that the two transfer functions, $T_x(k_x)$ and $T_y(k_y)$, belong to scheme A having $\mathbf{D}_A$ and $\mathbf{G}_A$ as its discrete gradient and the divergence operators, respectively. Projection methods search for a suitable field $\phi(x,y)$ such that

$$\mathbf{D}_A \cdot (\mathbf{u} - \frac{1}{\rho}\mathbf{G}_A \phi) = 0 \tag{5.13}$$

Then the velocities are corrected to

$$\tilde{u}(x,y) = \sum_{k=0}^{k_{max}} \left(\hat{u}_{(kx,ky)} - ik_x T_A(kx)\hat{\phi}_{(kx,ky)} \right) exp(i(k_x x + k_y y)) \tag{5.14}$$

$$\tilde{v}(x,y) = \sum_{k=0}^{k_{max}} \left(\hat{v}_{(kx,ky)} - ik_y T_A(ky)\hat{\phi}_{(kx,ky)} \right) exp(i(k_x x + k_y y)) \tag{5.15}$$

When compute the divergence of this *corrected velocity* analytically, the divergence is of course equal to the negative of the source term, $div = -q(x,y) = -\sum \hat{q}_{(kx,ky)} exp(i(k_x x + k_y y)$. This function $q(x,y)$ converges towards zero at the same rate as the convergence rate of scheme A (see Eq.(5.12)). It is evident that when the m-th order scheme was used for the divergence approximation, the velocity field satisfy the mass conservation at m-th order of accuracy. Thus when a second-order scheme was used to approximate the divergence, the error introduced into the velocity is $O(\triangle x^2)$. When the solution is integrated in time from t_0 to t_1, these errors will be accumulated to $(t_1 - t_0)(O(\triangle x^2))$ *Therefore fourth-order approximation of the divergence is essential to achieve fourth-order convergence rate in the velocity.*

The contour plots of the relative divergence introduced to the velocity field ($\|T_y(ky) - T_x(kx)\|_\infty$ in Eq.(5.11)) by the second- and fourth-order schemes are shown in Fig.5.1. In all figures, the most unreliable regions are the top-left and the bottom-right corners. In these regions, the approximation in one direction is very accurate while that of the other direction is poorly determined. This unbalanced leads to a larger value of q. According to the figures, the accuracy of the second-order and fourth-order approximation of the divergence on staggered grids are impressive. For a relative error of 0.1, the second-order approximation on staggered grid is comparable to the fourth-order on collocated grids (Fig.5.1(b) and Fig.5.1(c)). They can represent approximately 20% of all the modes within the Nyquist limit. On this level of relative error, the fourth-order can capture at least 60% of the whole wave space. For more accurate result, at a relative error of 0.001, the fourth-order scheme on collocated grids is much better than the second-order scheme on staggered grids. However, the fourth-order scheme on staggered grids is the most accurate in all levels of relative error.

5.3.2 Effect of pressure gradient approximation

In what follows, we consider how the approximation of the pressure gradient affects the accuracy of the velocity using Fourier analysis in 3D.

The velocity field in the vector form on the Fourier series can be written as

$$\mathbf{u}(\mathbf{x},t) = \sum_{\mathbf{k}} e^{i\mathbf{k}\cdot\mathbf{x}}\hat{\mathbf{u}}(\mathbf{x},t). \tag{5.16}$$

The divergence of the velocity on the physical space translates to the following relationship on the Fourier space

$$\mathbf{k}\cdot\hat{\mathbf{u}} = 0. \tag{5.17}$$

This means the divergence-free condition of a vector field $\mathbf{u}$ on the physical space translate into the orthogonality between the Fourier mode ($\mathbf{k}$) and the Fourier component vector $\hat{\mathbf{u}}$.

In the momentum equation there are only two terms that can generate a divergence, namely (i) the convection and (ii) the pressure terms. Let the nonlinear term be $\mathbf{w}(\mathbf{x},t)$, one can apply the Helmholtz-Hodge decomposition to this term and arrive at the following relationship of nonlinear convective and the pressure terms

$$\mathbf{k}\cdot\widehat{\mathbf{w}} = \mathbf{k}\cdot\widehat{\mathbf{p}} \tag{5.18}$$

This is similar to the situation we saw earlier in Eq.(5.10). The only difference is that the pressure must cancel out the image of the nonlinear convection in the direction of $\mathbf{k}$. Therefore when we approximate the convective term with higher-order schemes but kept the approximation of the pressure gradient at second-order, parts of the nonlinear term parallel to $\mathbf{k}$ will be cancelled out at second-order rate. Consequently, the accuracy of **the convective term is effectively degraded to second-order**. When the momentum equation is integrated in time the second-order errors is multiplied with $\triangle t$ and thus the errors of $O(\triangle t \triangle x^2)$ will be added to the momentum in each time integration. This first-order in time will be accumulated over time and eventually leads to the global error of $O(\triangle x^2)$. It should be noted that, if the second-order scheme are used for the divergence approximation, the error of $O(\triangle x^2)$ will be added at each RK substep. The error introduced by the second-order approximation of the pressure term, in a time-dependent problem, is thus less severe than the effect of second-order approximation of the divergence.

According to the above analysis, the approximations of divergence and pressure gradient must be fourth-order accurate, if one wish to achieve fourth-order convergence rate for the velocity. The result from the analysis in this section indicates that, within the framework of the consistent projection method, the fourth-order convergence rate can only be achieved through the consistent fourth-order projection method solving $\mathbf{D}_4\mathbf{G}_4$.

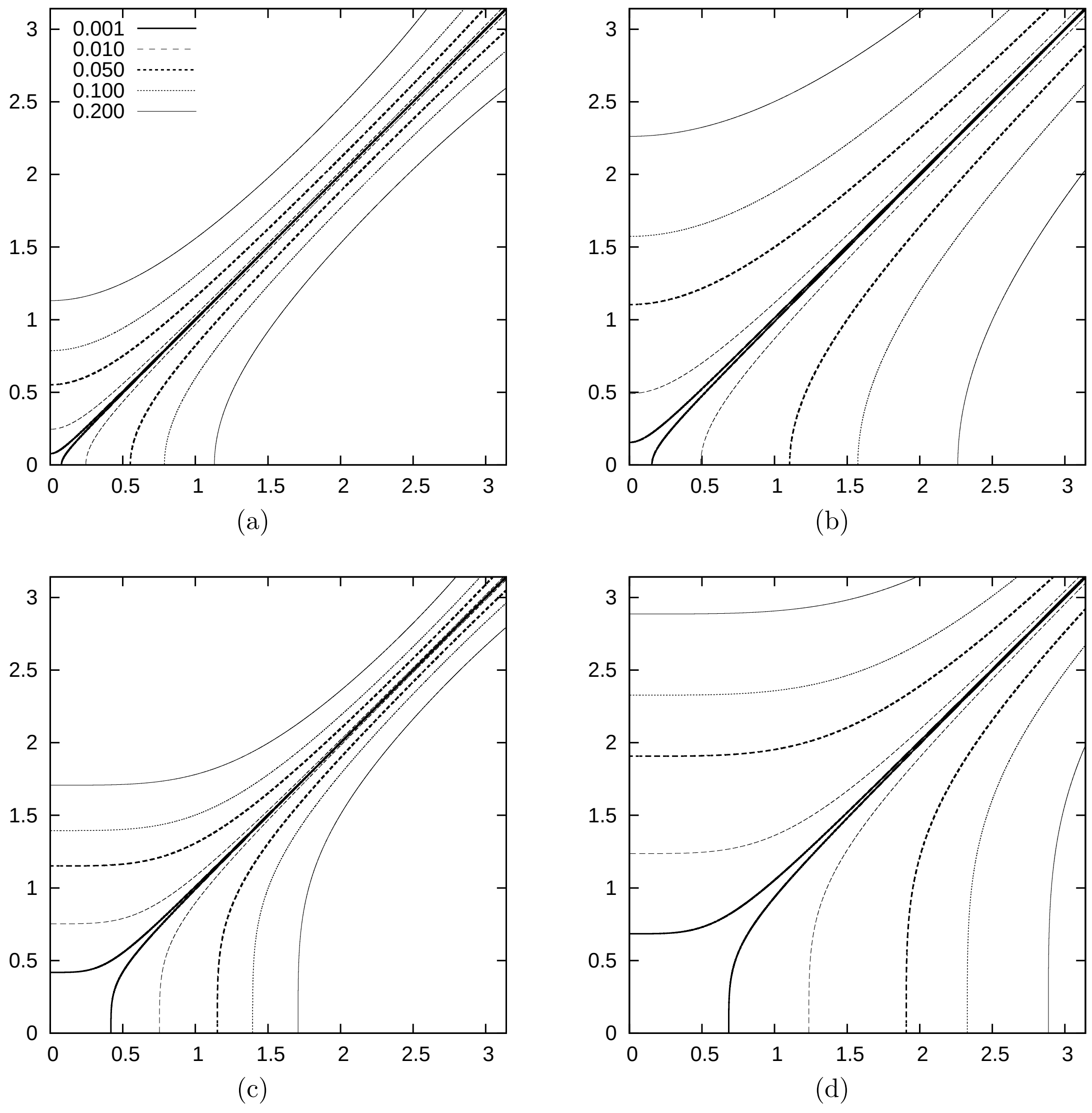


Figure 5.1: Contour plot of the divergence introduced by numerical approximations $\|T_y(ky) - T_x(kx)\|$ from Eq.(5.12): (a) second-order on collocated grids, (b) second-order on staggered grids, (c) fourth-order on collocated grids and (d) fourth-order on staggered grids. Horizontal and vertical axis are the components of the Fourier mode in x and y respectively. See (a) for numerical values of the contours.

5.4 Approximate projection method

The key concept of approximate projection methods is to solve another discrete Laplacian that is close to $\mathbf{D_4G_4}$ but simpler to solve. The two most obvious choices are the explicit second-order and fourth-order Laplacian which are the standard 7-point and 13-point stencils, respectively. The residuals display by these two Laplacians are not exactly the discrete mass imbalance and thus require dual-iteration mentioned previously are needed.

A close observation on Eq.(5.6) and Eq.(5.7) reveals that it is possible to fully satisfy the discrete mass conservation regardless of the discrete gradient used in Eq.(5.4). This can be achieved only when the discrete Laplacian is composite and the corresponding discrete gradient is used in Eq.(5.6) and Eq.(5.7). The divergence-free approximate projection method for the Runge-Kutta time-integration considered in this work is defined below:

$$\mathbf{Q}^k = a_k\mathbf{Q}^{k-1} + \left(\mathcal{H}(\mathbf{u}^{k-1}) - \frac{1}{\rho}\mathbf{G}_4 p^{k-1}\right)\triangle t \tag{5.19}$$

$$\mathbf{u}^{k*} = \mathbf{u}^{(k-1)} + b_k\mathbf{Q}^k \tag{5.20}$$

$$\mathbf{D}_4\mathbf{G}_2\phi = \frac{\rho}{\triangle t}\mathbf{D}_4\mathbf{u}^{k*} \tag{5.21}$$

$$\mathbf{u}^k = \mathbf{u}^{k*} - \frac{\triangle t}{\rho}\mathbf{G}_2\phi \tag{5.22}$$

$$p^k = p^{k-1} + \phi \tag{5.23}$$

This projection ensures that the velocity is divergence-free because Eq.(5.22) is consistent with Eq.(5.21). However, it is not consistent with Eq.(5.19) and thus called approximate projection method. On uniform grid, we have to solve $\mathbf{D}_4\mathbf{G}_2$ whose component in x-direction reads

$$D_{4x}G_{2x}p = \frac{1}{24\triangle x^2}(-p_{i\pm2,j,k} + 28p_{i\pm1,j,k} - 54p_{i,j,k}). \tag{5.24}$$

The component in y and z-direction can be found by switching the running index to j and k, respectively. On nonuniform grid, the method described in section 2.5.1 can be used to construct the discrete Laplacian.

It is clear that in steady state problems, this approximate projection method converges to that of the exact fourth-order projection method. This is because the velocities will cease to change only when either the momentum equation is in balance or the divergence of the provisional velocity is zero. If one these conditions is reached, the projection step will not change the velocity field. The remaining question is how will it perform in time-dependent problems which will be investigated in the subsequent sections.

5.5 Analysis of the approximate projection method

In this section, we analyse the local truncation error of the approximate projection method. First the local truncation error is analysed by following through each RK sub steps. The self-correction mechanism of the approximate projection method which improves the accuracy of the pressure term is presented. We then point out the form of the error in the velocity introduced by the approximate projection method.

5.5.1 Local truncation errors

In what follows the accuracy of the approximate projection method is analysed by comparing to the exact projection method ($\mathbf{D}_4\mathbf{G}_4$). Assuming that the initial velocity and the pressure are prescribed exactly. The k-th substep of the RK time integration in Eq.(5.7) gives out a provisional velocity u_1^{k*}. The consistent fourth-order projection and the approximate projection operate on this provisional velocity. The consistent fourth-order projection corrects the velocity and the pressure by the following steps:

$$\mathbf{D}_4\mathbf{G}_4\phi_1^k = \frac{\rho}{\triangle t}\mathbf{D}_4 u_1^{k*} \tag{5.25}$$

$$\mathbf{u}_1^k = \mathbf{u}_1^{k*} - \frac{\triangle t}{\rho}\mathbf{G}_4\phi_1^k \tag{5.26}$$

$$p_1^k = p^{k-1} + \phi_1^k \tag{5.27}$$

Similarly, the approximate projection performs:

$$\mathbf{D}_4\mathbf{G}_2\phi_2^k = \frac{\rho}{\triangle t}\mathbf{D}_4 u_1^{k*} \tag{5.28}$$

$$\mathbf{u}_2^k = \mathbf{u}_1^{k*} - \frac{\triangle t}{\rho}\mathbf{G}_2\phi_2^k \tag{5.29}$$

$$p_2^k = p^{k-1} + \phi_2^k \tag{5.30}$$

The subscript 1 and 2 denote the solutions of the exact projection and approximate projection methods, respectively. These two projection methods are called *incremental* projection method [BCM01] because the initial pressure is used in the momentum equation and it is then corrected by the solution of the Poisson equation. On the other hand, one can omit the pressure in the momentum equation and solve for the whole pressure itself. Such strategy is called *pressure-free* projection method [BCM01]. A work of Kim and Moin in [KM85] is an example of this type. In consistent projection methods, if the Poisson equation was solved by a direct solver, these two type of projection methods will be essentially the same. However, if an iterative solver was used instead, different results will be obtained because the magnitude of the solution is different. For example if the stopping criterion is set to $1E-3$, the incremental projection method will have ϕ up to 3 digits but the pressure-free projection method will have p^k accurate up to the same digit. Since to $\phi < p^k$ in general, the final solution of p^k will be more accurate under the incremental projection method.

Let us first consider the error in the first RK substep. The error between these two projection method in velocity field after the first substep is simply

$$\begin{aligned} e^1_{apx} = \|\mathbf{u}^1_1 - \mathbf{u}^1_2\|_\infty &= \frac{\triangle t}{\rho}\|\mathbf{G}_2\phi_2 - \mathbf{G}_4\phi_1\|_\infty \\ &= \|\left(\mathbf{G}_2\left(\mathbf{D}_4\mathbf{G}_2\right)^{-1} - \mathbf{G}_4\left(\mathbf{D}_4\mathbf{G}_4\right)^{-1}\right)\mathbf{D}_4 u^{1*}_1\|_\infty \\ &= \|\left(\mathbf{G}_2\left(\mathbf{D}_4\mathbf{G}_2\right)^{-1} - \mathbf{G}_4\left(\mathbf{D}_4\mathbf{G}_4\right)^{-1}\right)\|_\infty\|\mathbf{D}_4 u^{1*}_1\|_\infty \\ &= O(\triangle x^2)\|\mathbf{D}_4 u^{1*}_1\|_\infty \end{aligned}$$

The numerical superscript–"1" of $\mathbf{u}$ and ϕ in the above equation denote that they are belong to first RK substep and it should not be confused with the power. The inverse of the Laplacian is denoted by "-1". According to the above error, it would be straight forward to approximate the bound of e_{apx} if we can find a bound of the initial divergence $\|\mathbf{D}_4 u^{k*}_1\|_\infty$. Since we assume the exact initial conditions, $\|\mathbf{D}_4 u^{k*}_1\|_\infty$ naturally converges to zero at fourth-order rate. The error e^1_{apx} is thus in the order of $O(\triangle x^6)$ because we solved a second-order Laplacian. Likewise, the error in the pressure term is sixth-order i.e. $p^1_2 - p^2_1 = O(\triangle x^6)$

In the second substep, the velocity of both projections have now been changed and the Poisson equations in this substep for the exact projection becomes

$$\mathbf{D}_4\mathbf{G}_4\phi^2_1 = \frac{\rho}{\triangle t}\mathbf{D}_4 u^{2*}_1 = \mathbf{D}_4\left(b_2\left(\rho\mathcal{H}(\mathbf{u}^1_1) - \mathbf{G}_4 p^1_1\right) + \frac{a_2 b_2}{\triangle t}\mathbf{q}^1_1\right) \tag{5.31}$$

and that of the approximate projection is given by

$$\mathbf{D}_4\mathbf{G}_2\phi^2_2 = \frac{\rho}{\triangle t}\mathbf{D}_4 u^{2*}_2 = \mathbf{D}_4\left(b_2\left(\rho\mathcal{H}(\mathbf{u}^1_2) - \mathbf{G}_4 p^1_2\right) + \frac{a_2 b_2}{\triangle t}\mathbf{q}^1_2\right). \tag{5.32}$$

Apply the corresponding inverse of the discrete Laplacian to equations (5.32) and (5.31), then compute the difference gives the equation for the error in the incremental pressure at this step. Follows from the errors obtained in the previous paragraph, the right hand side of this error equation (source term), the keys determining the error of the incremental pressure in this substep are thus the difference in the order of accuracy and the magnitude of the source term. Consider Eq.(5.31), since the p^1_1 is responsible for cancelling the divergence part of $\mathcal{H}(\mathbf{u}^n)$, the only contributor to the divergence in the second substep is thus the *added velocity* given by $\mathbf{du}_1 = \mathbf{u}_1^2 - \mathbf{u}_1^1$. In order quantify how large is this added velocity, let us consider the divergence of the momentum equation.

Recall that, taking the divergence of the momentum equation leads to the following relationship

$$\triangle p = -\rho \frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i}, \tag{5.33}$$

where i and j here are the Einstein summation indices. Use this relation together with $\mathbf{u}_1^2 = \mathbf{u}_2^2 + O(\triangle x^6)$, Eq.(5.31) is equivalent to

$$\mathbf{D}_4\mathbf{G}_4\phi_1^2 = -\left(\frac{\partial du_j}{\partial x_i}\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\frac{\partial du_i}{\partial x_j} + \frac{\partial du_j}{\partial x_i}\frac{\partial du_i}{\partial x_j}\right) + O(\triangle x^4), \tag{5.34}$$

Note that in this equation, we try to keep the equation simple therefore the r.h.s of Eq.(5.33) is left at analytic form and the error of the approximation is put to the big-O in the above equation. This means the r.h.s of the above equation is in the order of $O(\triangle t)$. The Poisson equation for the approximate projection method takes the same r.h.s. Therefore, the error bound for the increment pressure term (ϕ) of the approximate projection method (relative to the exact projection method) at the second RK substep is given by

$$\|\phi_1^2 - \phi_2^2\|_\infty = \|\left[(\mathbf{D}_4\mathbf{G}_4)^{-1} - (\mathbf{D}_4\mathbf{G}_2)^{-1}\right]\mathbf{D}_4 u_1^{2*}\|_\infty = O(\triangle t \triangle x^2) \tag{5.35}$$

This incremental pressure is used to correct the velocity in Eq.(5.29) which leads to the local truncation error of $O\left(\triangle t^2 \triangle x^2\right)$ in the velocity.

At this point, it would seem that the approximate projection method deliver fourth-order accurate velocity but a third-order accurate pressure. When integrate the NSE for a long time, the time-error may be accumulated and leads to third-order and second-order global errors in the velocity and the pressure, respectively. Actually, this accumulation of error in the pressure does not occur due to the self-correction property which can be describe in the following paragraphs.

The provisional velocity at the third RK substep is given by

$$\mathbf{u}_2^{3*} = \mathbf{u}_2^2 + b_3\triangle t\left(\mathcal{H}(\mathbf{u}_2^2) - \frac{1}{\rho}\mathbf{G}_4 p_2^2\right) + a_3 b_3 \mathbf{q}_2^2 \tag{5.36}$$

$$= \mathbf{u}_2^2 + b_3\triangle t\left(\mathcal{H}(\mathbf{u}_2^1 + \mathbf{du}_2^2) - \frac{1}{\rho}\mathbf{G}_4\left(p_2^1 + \phi_2^2\right)\right) + a_3 b_3 \mathbf{q}_2^2 \tag{5.37}$$

Before we continue further, let us consider an extreme case where the time step size is close to the machine accuracy i.e. $\triangle t \approx \epsilon$. When we perform the time integration in this case, the velocity will stay constant and only the pressure will be changed. Since $\mathbf{q} \approx 0$, each i-th RK substep is having the same amount of convection and diffusion i.e. $\mathcal{H}(\mathbf{u}_2^1) = \mathcal{H}(\mathbf{u}_2^2) = \mathcal{H}(\mathbf{u}_2^3) = \mathcal{H}(\mathbf{u}^n)$. We simply perform an iteration for the pressure field $\check{p}$ satisfying $\mathbf{D}_4\mathbf{G}_4\check{p} = \frac{\rho}{\triangle t}\mathbf{D}_4\mathcal{H}\left(\mathbf{u}^n\right)$. Since $p^n = \check{p} + O(\triangle x^4)$ and we solve $\mathbf{D}_4\mathbf{G}_2$ which is a second-order approximation instead of $\mathbf{D}_4\mathbf{G}_4$, the pressure after the first RK substep is given by $p_2^1 = \check{p} + O(\triangle x^6)$. Substitute this pressure into Eq.(5.31), the source term obviously becomes $O(\triangle x^6)$. The solution p_2^2 is thus equal to $\check{p} + O(\triangle x^8)$. This convergence continues until $p_2 - \check{p} \approx \epsilon$.

Now, let us come back to a general case where $\triangle t$ is in the same order of magnitude as the grid size. We define Θ to be a function of the difference of the net diffusive and the convective fluxes of two velocities: $(\mathbf{w}_1 + \mathbf{w}_2)$ and $\mathbf{w}_1$, i.e.

$$\Theta(\mathbf{w}_1, \mathbf{w}_2) = \mathcal{H}(\mathbf{w}_1 + \mathbf{w}_2) - \mathcal{H}(\mathbf{w}_1). \tag{5.38}$$

Apply this relationship to Eq.(5.37), we have

$$\mathbf{u}_2^{3*} = \mathbf{u}_2^2 + b_3\triangle t\left(\mathcal{H}(\mathbf{u}_2^1) - \frac{1}{\rho}\mathbf{G}_4\left(p_2^1 + \phi_2^2\right)\right) + a_3b_3\mathbf{q}_2^2 + b_3\triangle t\Theta(\mathbf{u}_2^1, \mathbf{du}_2^2) \tag{5.39}$$

Applying the fourth-order divergence approximation to the above equation leads to

$$\mathbf{D}_4\mathbf{u}_2^{3*} = b_3\triangle t\mathbf{D}_4\left[\mathcal{H}(\mathbf{u}_2^1) - \frac{1}{\rho}\mathbf{G}_4\left(p_2^1 + \phi_2^2\right)\right] + \mathbf{D}_4\left[a_3b_3\mathbf{q}_2^2 + b_3\triangle t\Theta(\mathbf{u}_2^1, \mathbf{du}_2^2)\right]. \tag{5.40}$$

The first square bracket would be zero if we had solved $\mathbf{D}_4\mathbf{G}_4$ and the second square bracket would be our source term for ϕ_2^3. Since we actually solved $\mathbf{D}_4\mathbf{G}_2$, the first bracket leads to a divergence in the order of $O(\triangle t\triangle x^2)$. For clarity, we write the divergence of the provisional velocity at this substep for the exact projection below

$$\mathbf{D}_4\mathbf{u}_1^{3*} = \mathbf{D}_4\left[a_3b_3\mathbf{q}_1^2 + b_3\triangle t\Theta(\mathbf{u}_1^1, \mathbf{du}_1^2)\right]. \tag{5.41}$$

Apply the respective inverse of the Laplacian operator to Eq.(5.41) and Eq.(5.40), then correct the pressure leads to the following difference in pressure

$$\begin{aligned}
p_2^3 - p_1^3 &= \left(\mathbf{D}_4\mathbf{G}_2\right)^{-1}\mathbf{D}_4\left[\mathcal{H}(\mathbf{u}_2^1) - \frac{1}{\rho}\mathbf{G}_4\left(p_2^1 + \phi_2^2\right)\right] + O(\triangle t\triangle x^2)\\
(\phi_2^1 - \phi_1^1) + (\phi_2^2 - \phi_1^2) + (\phi_2^3 - \phi_1^3) &= \\
&\left(\mathbf{D}_4\mathbf{G}_2\right)^{-1}\mathbf{D}_4\left[\mathcal{H}(\mathbf{u}_2^1) - \frac{1}{\rho}\mathbf{G}_4\left(p_2^1 + \phi_2^2\right)\right] + O(\triangle t\triangle x^2)
\end{aligned}$$

Since we are not interested in the solution attributed to $\mathbf{D}_4\left[a_3b_3\mathbf{q}_2^2 + b_3\triangle t\Theta(\mathbf{u}_2^1, \mathbf{du}_2^2)\right]$ and

$\mathbf{D}_4 \left[a_3 b_3 \mathbf{q}_1^2 + b_3 \triangle t \Theta(\mathbf{u}_1^1, \mathbf{du}_1^2)\right]$, we simply bound the error due to these terms to third-order. The first bracket on the l.h.s. of the above equation can be neglected and only the following equation determines the leading truncation error

$$(\phi_2^2 - \phi_1^2) + (\phi_2^3 - \phi_1^3) = (\mathbf{D}_4\mathbf{G}_2)^{-1} \mathbf{D}_4 \left[\mathcal{H}(\mathbf{u}_2^1) - \frac{1}{\rho}\mathbf{G}_4\left(p_2^1 + \phi_2^2\right)\right] + O(\triangle t \triangle x^2) \quad (5.42)$$

It follows that,

$$\left(\phi_2^2 - \phi_1^2 - (\mathbf{D}_4\mathbf{G}_2)^{-1} \mathbf{D}_4 \left[\mathcal{H}(\mathbf{u}_2^1) - \frac{1}{\rho}\mathbf{G}_4\left(p_2^1 + \phi_2^2\right)\right]\right) + (\phi_2^3 - \phi_1^3) = O(\triangle t \triangle x^2) \quad (5.43)$$

Since $p_1^1 = p_2^1 + O(\triangle x^6)$ and $u_1^1 = u_2^1 + O(\triangle x^6)$, the above equation can be written as

$$\left(\phi_2^2 - \phi_1^2 - (\mathbf{D}_4\mathbf{G}_2)^{-1} \mathbf{D}_4 \left[\mathcal{H}(\mathbf{u}_1^1) - \frac{1}{\rho}\mathbf{G}_4\left(p_1^1 + \phi_2^2\right)\right]\right) + (\phi_2^3 - \phi_1^3) = O(\triangle t \triangle x^2) \quad (5.44)$$

The solution of the Poisson equation in the above equation is thus only a correction for ϕ_2^2, that is

$$(\mathbf{D}_4\mathbf{G}_2)^{-1} \mathbf{D}_4 \left[\mathcal{H}(\mathbf{u}_1^1) - \frac{1}{\rho}\mathbf{G}_4\left(p_1^1 + \phi_2^2\right)\right] = -\left(\phi_2^2 - \phi_1^2\right) + O(\triangle t \triangle x^4). \quad (5.45)$$

Therefore the error in computing ϕ_2^2 at the second RK substep is damped in the third substep. We call this error cancellation *the self-correction mechanism*. Similar to the extreme case discussed earlier, this self-correction will be saturated at the same magnitude of changes introduced to the velocity field. In our case, the limit is given by the error introduced to the velocity field at the second substep which is $O(\triangle t^2 \triangle x^2)$.

In summary, the local truncation error of the approximate projection method relative to the exact projection method is $O(\triangle t^2 \triangle x^2)$ for the velocity and $O(\triangle t \triangle x^2)$ for the pressure. Since $\triangle t \approx \triangle x$ in explicit time-integrations, the approximate projection does not compromise the accuracy of the third-order time integration or the fourth-order spatial approximation because $O(\triangle t^2 \triangle x^2) = O(\triangle t^4) = O(\triangle x^4)$ for explicit time integrations. This local truncation error leads to $O(\triangle t \triangle x^2)$ global error in the velocity. The local truncation error of the pressure term in the approximate projection method is $O(\triangle t \triangle x^2)$. This local truncation error **is not accumulated** due to the self-correction mechanism of the approximate projection method and therefore the global error is also $O(\triangle t \triangle x^2)$.

5.5.2 Conservation of mass and vorticity

The resulting velocities from both approaches are divergence-free with respect to the fourth-order divergence approximation ($\mathbf{D}_4$). This can be shown by applying $\mathbf{D}_4$ to Eq.(5.26) and Eq.(5.29). The resulting equation is exactly the corresponding Poisson equations. However, the difference in the vorticity of these two velocities is :

$$\mathbf{G}_4 \times \mathbf{u}_1^k - \mathbf{G}_4 \times \mathbf{u}_2^k = -\frac{\triangle t}{\rho}\left(\mathbf{G}_4 \times \mathbf{G}_4\phi_1^k - \mathbf{G}_4 \times \mathbf{G}_2\phi_2^k\right) \tag{5.46}$$

$$= -\frac{\phi_2^k \triangle t}{\rho}((G_{4y}G_{2z} - G_{4z}G_{2y})\mathbf{i} + (G_{4z}G_{2x} - G_{4x}G_{2z})\mathbf{j} + (G_{4x}G_{2y} - G_{4y}G_{2x})\mathbf{k}) \tag{5.47}$$

In analytical operations, $\nabla \times \nabla \cdot \phi = 0$ for all scalar field ϕ. This property is not automatically inherited to discrete operators. However, for a uniform grid with periodic domain, this property is inherited if the same operator was used for both computations. For example, $\mathbf{G}_4 \times \mathbf{G}_4\phi$ is equal to zero because there is no order in the application of the operators. Therefore, the first cross product is zero in these conditions.

The second cross product, on the other hand, is not equal to zero and the error is the commutative error between the second and the fourth-order operators. Consider the value of the **i**-component. Here the second-order and the fourth-order approximations for the first derivative are applied in a different order. Since these approximations are not exact and some information will be lost after applying the approximations. Two different approximations applied in different order will give different results. It can be shown by the Fourier analysis using a similar procedure done in section 5.3.1, that this cross-product will be equal to zero if $k_x = k_y$ and otherwise it convergences to zero at second-order rate. Therefore the error of the approximate projection method lies in the conservation of the vorticity.

Before we proceed with the result section, first let us summarize the projection methods discussed so far. The fourth-order exact projection (P44), the consistent mixed-order(P42), and the approximate projection method (P42a) are summarized in Alg.1, Alg.2 and Alg.3 respectively.

Algorithm 1 Exact projection method (P44)

```
Input: $\mathbf{u}^*$, $\mathbf{D}_4$, $\mathbf{G}_4$, $\mathbf{D}_4\mathbf{G}_4$, $p^n$, $\triangle t$, $\varepsilon_1$
Output: $\mathbf{u}^{n+1}$ and $p^{n+1}$ with $|\mathbf{D}_4\mathbf{u}^{n+1}|_\infty \leq \varepsilon_1$
Require: $\mathbf{G}_4$ was used for the pressure term in the momentum equation
 1: $div = \mathbf{D}_4\mathbf{u}$
 2: solve: $\mathbf{D}_4\mathbf{G}_4\phi = \frac{\rho}{\triangle t}div$ until $|\mathbf{D}_4\mathbf{G}_4\phi - div|_\infty \leq \varepsilon_1$
 3: $\mathbf{u} = \mathbf{u} - \frac{\triangle t}{\rho}\mathbf{G}_4\phi$
 4: $p = p + \phi$
```

Algorithm 2 Mixed-order projection method (P42)

```
Input: $\mathbf{u}^*$, $\mathbf{D}_4$, $\mathbf{G}_2$, $\mathbf{D}_4\mathbf{G}_2$, $p^n$, $\triangle t$, $\varepsilon_1$
Output: $\mathbf{u}^{n+1}$ and $p^{n+1}$ with $|\mathbf{D}_4\mathbf{u}^{n+1}|_\infty \leq \varepsilon_1$
Require: "$\mathbf{G}_2$" was used for the pressure term in the momentum equation
 1: $g = \mathbf{D}_4\mathbf{u}$
 2: solve: $\mathbf{D}_4\mathbf{G}_2\phi = \frac{\rho}{\triangle t}div$ until $|\mathbf{D}_4\mathbf{G}_2\phi - div|_\infty \leq \varepsilon_1$
 3: $\mathbf{u} = \mathbf{u} - \frac{\triangle t}{\rho}\mathbf{G}_2\phi$
 4: $p = p + \phi$
```

Algorithm 3 Divergence-free approximate projection method (P42a)

```
Input: $\mathbf{u}^*$, $\mathbf{D}_4$, $\mathbf{G}_2$, $\mathbf{G}_4$, $\mathbf{D}_2\mathbf{G}_4$, $p^n$,$\triangle t$, $\varepsilon_1$
Output: $\mathbf{u}^{n+1}$ and $p^{n+1}$ with $|\mathbf{D}_4\mathbf{u}^{n+1}|_\infty \leq \varepsilon_1$
Require: "$\mathbf{G}_4$" was used for the pressure term in the momentum equation
 1: $div = \mathbf{D}_4\mathbf{u}$
 2: solve: $\mathbf{D}_4\mathbf{G}_2\phi = \frac{\rho}{\triangle t}div$ until $|\mathbf{D}_4\mathbf{G}_2\phi - g|_\infty \leq \varepsilon_1$
 3: $\mathbf{u} = \mathbf{u} - \frac{\triangle t}{\rho}\mathbf{G}_2\phi$
 4: $p = p + \phi$
```

Algorithm 4 Second-order projection method (P22)

```
Input: $\mathbf{u}^*$, $\mathbf{D}_2$, $\mathbf{G}_4$, $\mathbf{D}_2\mathbf{G}_2$, $p^n$, $\triangle t$, $\varepsilon_1$
Output: $\mathbf{u}^{n+1}$ and $p^{n+1}$ with $|\mathbf{D}_2\mathbf{u}^{n+1}|_\infty \leq \varepsilon_1$
Require: $\mathbf{G}_2$ was used for the pressure term in the momentum equation
 1: $div = \mathbf{D}_2\mathbf{u}$
 2: solve: $\mathbf{D}_2\mathbf{G}_2\phi = \frac{\rho}{\triangle t}div$ until $|\mathbf{D}_2\mathbf{G}_2\phi - div|_\infty \leq \varepsilon_1$
 3: $\mathbf{u} = \mathbf{u} - \frac{\triangle t}{\rho}\mathbf{G}_2\phi$
 4: $p = p + \phi$
```

5.6 Results

In this section we evaluate the accuracy of the approximate projection in comparison with mixed order projection and the second-order projection methods using the Doubly periodic shear layer flow and the turbulent channel flows used earlier in chapter 3.

5.6.1 Doubly periodic shear layer

In this test, the approximate projection method (P42a) and the mixed-order projection method (P42) are applied to the doubly periodic shear layer. The approximations of the convective and the diffusive terms are fourth-order accurate. The pressure gradient and the discretisation of the Laplacian are used according to the respective algorithms. The Poisson equations are solved by direct method. The results are compared with the reference solution used earlier. The convergence rates of the streamwise velocity are shown in Tab.5.1 with surprising findings. The accuracy of P42 is not bad at all. At low resolutions, it is comparable to the exact projection (P44) and the convergence rate is definitely higher than three. Compared to the result of the second-order projection(P22), the errors of P42 are three times smaller on the two finest grids ($N = 192^2$ and 256^2). Nonetheless, the approximate projection method is much better. The numerical errors differ from the exact projection at the fourth digit! If we consider that P42a only requires three more floating point operations per direction compared to P42, the gain in the accuracy is extraordinary. The convergence rate is also plotted in in Fig.5.2 which shows clearly fourth-order convergence rate in every variables.

Grid	Maximum Error				Convergence rate			
	P44	P42	P42a	P22	P44	P42	P42a	P22
64	2.5835E-02	2.68E-02	2.5834E-02	3.23E-02	—	—	—	—
96	1.0762E-02	1.16E-02	1.0762E-02	1.22E-02	2.16	2.07	2.16	2.4
128	4.4665E-02	5.03E-03	4.4661E-02	5.54E-03	3.06	2.90	3.06	2.7
192	8.9431E-04	1.20E-03	8.9429E-04	2.02E-03	3.97	3.53	3.97	2.5
256	2.6394E-04	4.54E-04	2.6399E-04	1.02E-03	4.24	3.39	4.24	2.4

Table 5.1: Maximum error and convergence rate of the streamwise velocity at $t = 1.2$ subjected to different projection methods. Extra digits are given to P44 and P42.

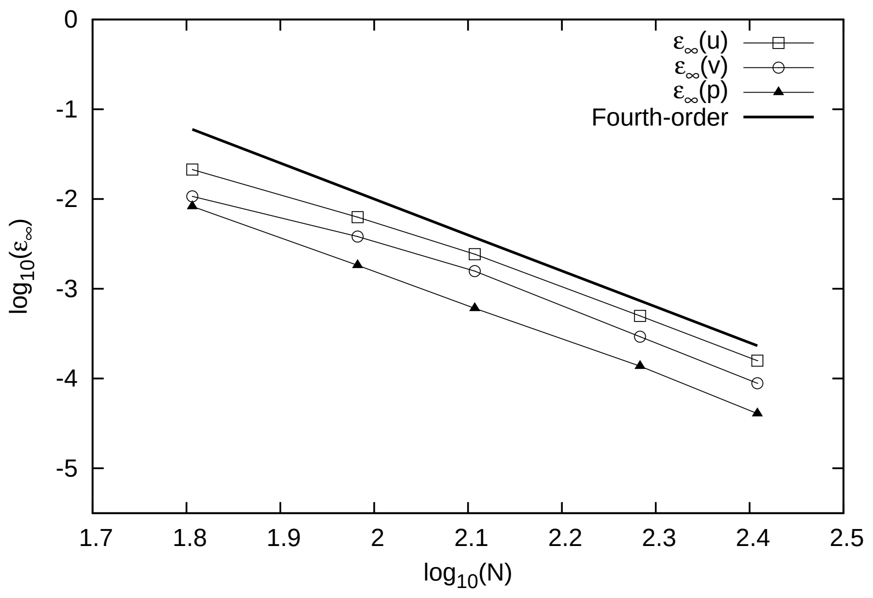


Figure 5.2: Convergence of the approximate projection method compared to the solution interpolated from pseudo spectral method.

The results in this test can can easily mislead us to believe that the convergence rate of the proposed projection method (P42a) is fourth-order. In the analysis section we found that the global convergence rate, compared to P44 is in fact $O(\triangle t \triangle x^2)$. It is possible that the errors of the projection method are so small and hidden under the local truncation error of other spatial approximations. In order to verify this, the errors in each time step are monitored. The solution from the full fourth-order scheme with the exact projection method at time t=0.8 is used as the initial condition and then the solution is integrated in time using fixed $CFL = 0.032$ such that the error of the time integration is negligible. The mean streamwise velocity and the pressure from P42 and P42a are compared with P44 and the maximum errors are plotted in Fig.5.3(a) and 5.3(b). The error in the streamwise velocity of the mixed-order projection (P42) is linearly accumulated in time while that of the approximate projection (P42a) is relatively constant. This can be explained by Fig.5.3(b) which shows that the level of pressure error is constant in P42. However, with the approximate projection, the error of the pressure is decreased by approximately a constant factor (one order of magnitude per time step in this case) for several time steps and then saturated at the level comparable to the error of the velocity. Therefore P42a does not lead to error accumulation in the velocity, unlike P42.

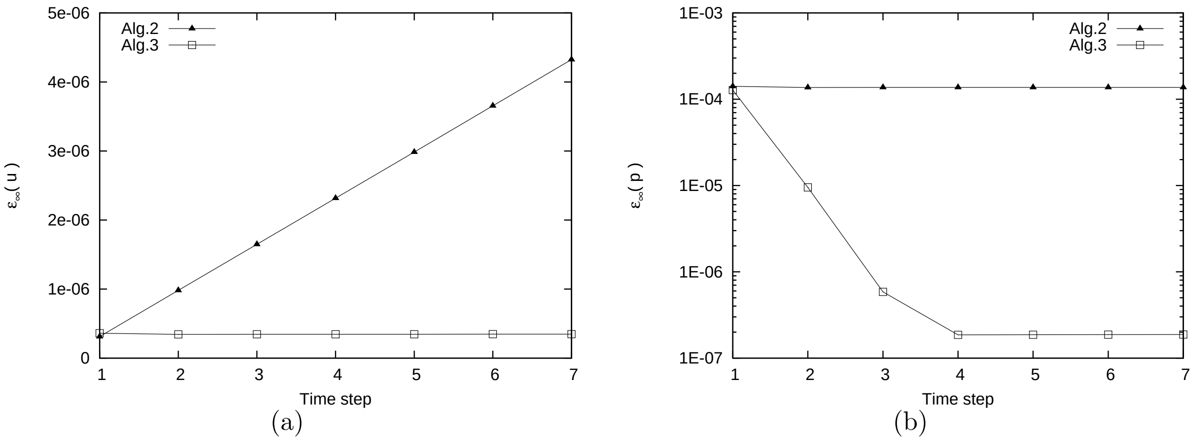
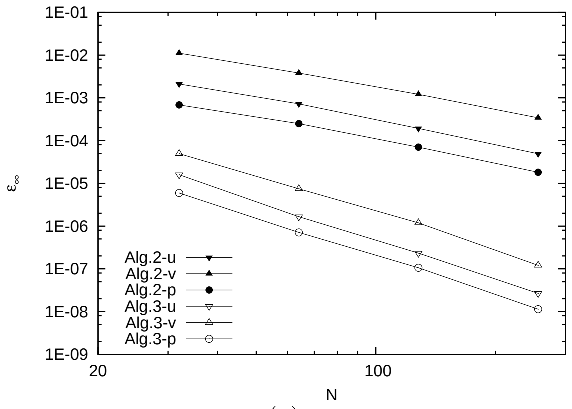
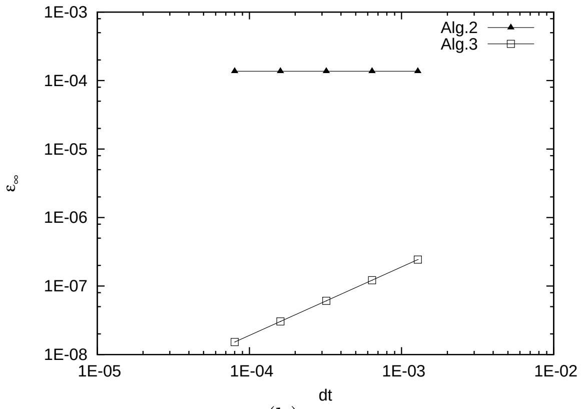


Figure 5.3: Error of the streamwise velocity (a) and pressure (b) of the approximate projection method (P42a) compared to the exact projection method (P44).

The convergence of the flow variables from P42 and P42a towards the approximate projection method are plotted in Fig5.4(a). It shows clearly that the convergence rate of P42 and P42a are second- and third-order, respectively. The extra convergence rate of the approximate projection method comes from the first-order convergence in time shown in Fig.5.4(b). These findings agree well with the analysis. The errors of both projection algorithms are smaller than that of the convective and diffusive terms and thus the comparable errors were observed in Tab.5.1 up to N=192. At the finest solution, error of P42 becomes larger than the error of the approximations of convective and diffusive terms. However, the error of the approximate projection method is well below those errors.



(a)



(b)

Figure 5.4: (a) Convergence of flow variables under a spatial grid refinement with fixed CFL of mixed-order projection method (Alg.2) and approximate projection method (Al.g3) towards the exact projection method (Alg.1). (b) Convergence of the pressure under a time-step refinement on $N = 64^2$ grid.

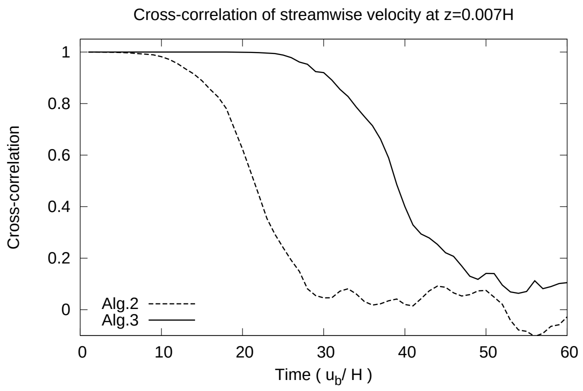
5.6.2 Turbulent channel flow

In this section we investigate the accuracy of the mixed-order projection (P42) and approximate projection methods (P42a) by investigating turbulent channel flow. Since it is not obvious how to choose the grid resolution to evaluate the projection method because the grid too coarse would not reveal any difference since the errors are mainly attributed to the approximations of convective and diffusive terms. The grid too fine would also not helpful because all both algorithms may be very close to the exact projection. It would be, however, too exhaustive to repeat all the grids used earlier in chapter 3. Therefore only Grid M1 is used in this test since it is not too coarse and not too fine. We use the solution obtained on grid M1 in chapter 3 as the initial condition. The solution is then integrated in time using fourth-order approximations for convective and diffusive terms but with the different approximation of the pressure gradient as required by the respective projection method.

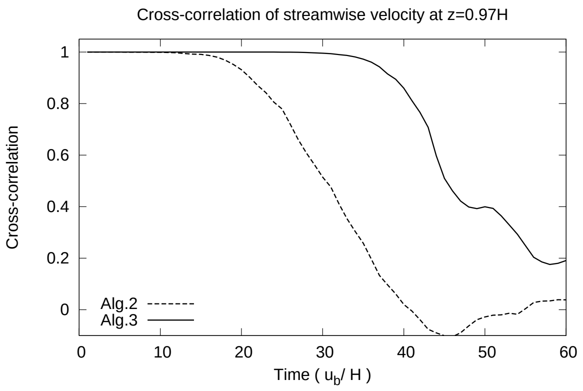
First the correlations of the streamwise velocity between non-exact projection methods and the exact projection method are investigated. The cross correlation between two momentums is computed by

$$\rho_{12}(u) = \sum \frac{(u_1 - \overline{u}_1)(u_2 - \overline{u}_2)}{\sigma_1 \sigma_2}, \tag{5.48}$$

where u_1 is the streamwise velocity from Alg.1(P44), and the overbar denote an averaging on the streamwise-spanwise plane. σ_1 and σ_2 are the standard deviation of u_1 and u_2 respectively. The simulations are performed for $60H/u_b$ with CFL=0.51. The results are shown in Fig.5.5(a) and Fig.5.5(b) for two positions. Near the wall, the cross-correlation of the streamwise velocity from the approximate projection method is approximately one up to $t = 20H/u_b$ while that of the mixed-order projection method drops already at $t \approx 10H/u_b$. The cross-correlation of P42 drops to approximately zero at $t = 30H/u_b$ while that of P42a lasts twice longer. Near the center of the channel, both algorithms have a better correlation with the exact projection method. This is because, at the center of the channel, large scale structures are dominant and thus it takes longer time before they start to be affected by the errors introduced in the projection step. These two figures show that the approximate projection is *closer* to the exact project since it can produce the streamwise velocity field that has higher, and longer correlation with the one provided by the exact projection method. The one-dimensional spectra were investigated at the end of the simulation (not shown here), the spectra from the exact projection method and the exact projection method are virtually identical. However, the ones from mixed-order projection method are significantly different.



(a)



(b)

Figure 5.5: Cross-correlation of streamwise velocity near the wall (a) and near the center of the channel (b).

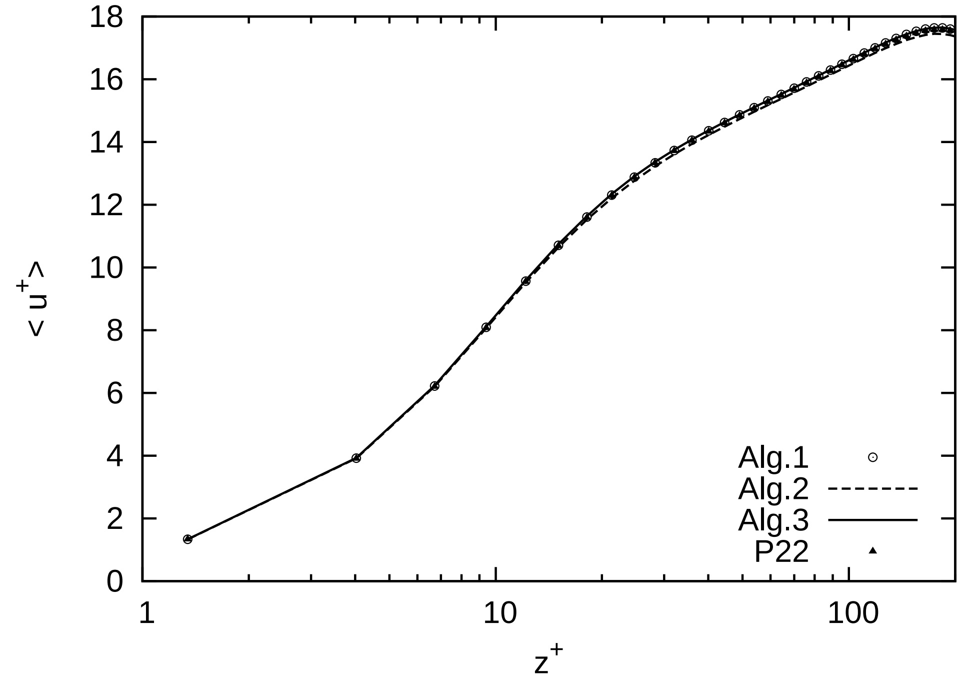
In Tab.5.2, the bulk flow parameters are listed. On the rightmost column, the imposed parameters are shown. Note that the flow conditions used here are slightly different from what used earlier. In chapter 3, the flow conditions were set to match the conditions in [MKM99]. Here, the conditions are set to the one used in [KMM87]. This leads to small differences in τ_{wall} and u_b. According to previous analysis, one would expect that P42 arrives at different time-averaged profile due to second-order approximation of pressure in the momentum equation. This is indeed true because the bulk velocity from P42 is slightly lower than the ones seen obtained from P44 and P42a. Similar to what have seen earlier with the doubly periodic shear layers flow, the global parameter of the solutions from P44 and P42a are identical in the first three digits. The first- and second-order statistics of the flow shown in Fig.5.6(a) and Fig.5.6(b). The profile of the mean streamwise velocity of the approximate projection (P42a) is passing through the circle symbols of the exact projection while the one of the mixed-order projection (P42) follows the bottom of the circle symbols from $z^+ = 30$ onwards. Similar situation is seen in the r.m.s of the streamwise velocity but the other two r.m.s profiles are collapsing. This level of differences was expected since the maximum differences seen in the global flow parameter was only 1% (in the bulk flow velocity). Similar study on grid 32^3 grid was also performed and came the same conclusion and thus are not reported here. When compare the second-order projection to the P42 and P42a, the bulk flow velocity normalized by the shear stress velocity is significantly higher. In the mean streamwise velocity profile, the differences are insignificant. However, the level of the velocity fluctuations of P22 are notably lower than the others. It should be emphasize that at this resolution P22 is accidentally collapse on the profile of P44 (see Fig.3.9(b)). Perhaps the difference between these projection methods can be seen better in the mean streamwise profile on some other grids.

	P44	P42	P42a	P22	Nominal
u_b	0.9786	0.9698	0.9790	1.0519	—
u_b/u_τ	15.296	15.142	15.302	16.443	15.631
Re_τ	180.005	180.192	180.000	179.992	179.996
τ_{wall}	11.516	11.540	11.516	11.515	11.515

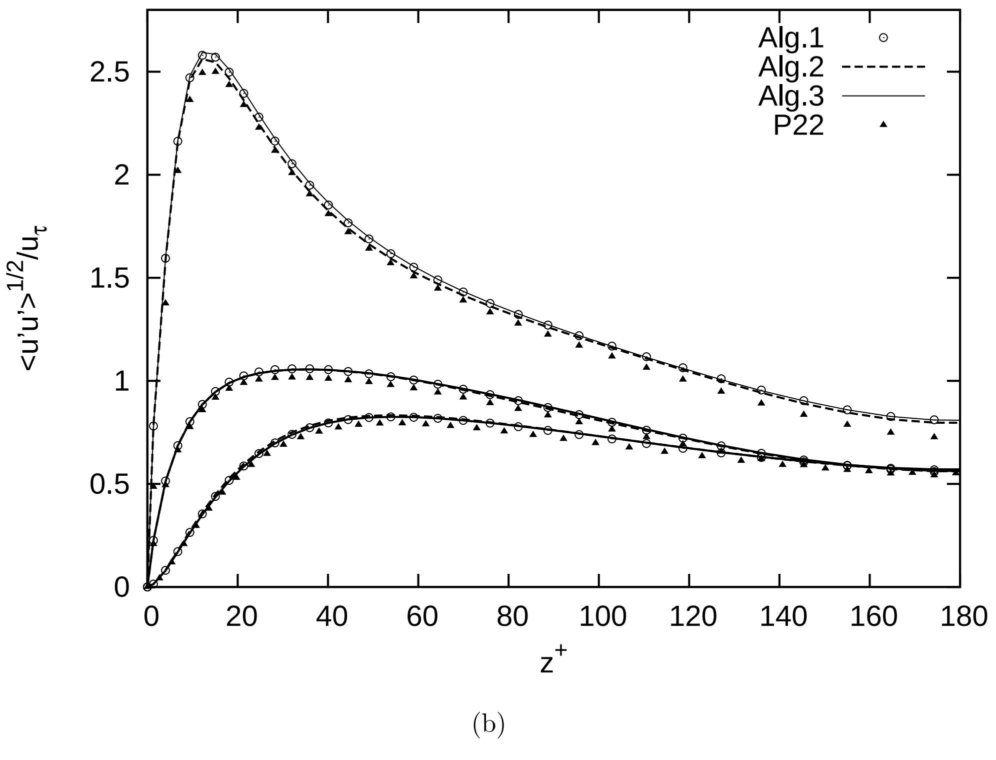
Table 5.2: Global flow parameters of the turbulent channel flow under different projection methods.

5.7 Conclusion

The mixed-order projection method (P42) is much better than using second-order projection (P22) which was shown to be unacceptable in chapter 3. Here, the solutions from P42 are hardly distinguished from the ones obtained from the exact projection method, when plotted together. Nevertheless, the approximate projection which solves the same discrete Laplacian gives even better result. Further, it was shown that the errors introduced by the approximate projection at each substep of Runge-Kutta time-integration are in the order of $O(\triangle x^4)$ when $\triangle x \approx \triangle t$. Therefore it does not degrade the overall accuracy of the spatial approximations nor the time integration. It was evident that the mixed-order projection method (P42) reaches the point where the projection error becomes dominant and its solution discorrelates from the exact projection method faster than the approximate projection method(P42a). Since the complexities of both projections are comparable, the approximate projection method is thus preferred.



(a)



(b)

Figure 5.6: Time-averaged streamwise velocity profile (a) and r.m.s. of velocity (b) with the approximate projection and mixed-order projection methods. Results from the second-order projection method (P22) from chapter 3 are shown for comparison.

6 Application to Massive-scaled Simulations

A Direct Numerical Simulation (DNS) is a useful tool in turbulent investigation. When the flow field is sufficiently resolved, DNS can be very accurate and it provides us with a great detail of dynamics and time history of the whole flow field. The quality of different simulations of the same flow can be slightly different due to numerical methods, grid resolutions, and boundary conditions used to simulate them. Among the wide range of DNS in the literature, turbulent channel flow is one of the most studied. This is because the simplicity of the geometry and the well-defined boundary conditions. The forcing of the flow is usually done by assuming a constant pressure gradient which is added as a body force in the streamwise momentum equation. These conditions allow the simulations to be repeated with a great precision. A large number of publications have shown that the statistics of turbulent channel flow $Re_\tau = 180$ converge the one reported in [KMM87]. This code is subsequently used to simulate the flow at higher Reynolds number up to $Re_\tau = 590$ in 1999 [MKM99]. Later in 2003, the simulation of turbulent channel flow at $Re_\tau = 2003$ [HJ08] is made public. This database is the highest Reynolds number in the literature at the time of this work. This simulation was performed on the MareNostrum at the Barcelona supercomputing center. According to amount of CPU hours and specification of the MareNosturm at that time, we can estimate that the simulation had occupied half of the MareNostrum for about six months. Disregarding the cost of the computer itself and only take the operation budget of the supercomputing center alone, this simulation costs roughly 1.5 million euro. If the performance of the code was 50% poorer, this run would take a year and the cost would have been doubled. This fact emphasizes that the code solving the Navier-Stokes equation must be parallelised with excellent scalability if one aims at attacking high Reynolds number flows.

It is relatively straight forward to parallelise the compact scheme on shared memory computer because the approximation problem is just a one-dimensional. For example, if the total number of grid cells is $N_x \times N_y \times N_z$, the deconvolution in x-direction will have $N_y \times N_z$ independent systems. Processors can shared this work load and the programming can be easily implemented by OpenMP directive. However, this shared memory paradigm is not as scalable as the distributed one. We addressed already how to solve tridiagonal systems on distributed memory machine in chapter 4. The proposed algorithm is expected to be very efficient when the subsystem size is chosen wisely. In this section, all the developed algorithms are combined in to the MGLET. The accuracy and the scalability of this parallelised version of the fourth-order compact scheme are evaluated First the accuracy and the performance of the parallelised version is compared to the sequential one using the turbulent channel flow at $Re_\tau = 180$. The accuracy of the code is further tested with higher Reynolds numbers up to $Re_\tau = 950$. We proposed new grid resolutions which allow direct numerical simulations using the compact fourth-order scheme to deliver the first and the second-order statistics accurately. Finally, the scalability of the proposed scheme is evaluated.

6.1 Configuration of numerical simulations

The computational box in this section is set to $[L_x, Ly, Lz] = [2\pi H, \pi, 2H]$, except for the grid G1 which is twice and one-third larger in the streamwise and the spanwise, respectively. The numerical grids used in this chapter are listed in Tab.6.1. The simulations are first started on a coarse grid (grid F in chapter 3) and the result are interpolated to other grids in the table. The u_b is monitored until it fluctuates within $0.1U_{bulk}$ for 2 through flow before the sampling is begun. The pressure gradient and the viscosity are changed according to Dean's correlation $Re_\tau \approx 0.09Re^{0.88}$ such that the mean bulk velocity remain close to unity. The simulations are run without further control of the mass flow.

Grid	N_x	N_y	N_z	$N_{total} \times 10^6$	$z_{wall}(H)$	$z_{center}(H)$
G1	128	128	128	2.1	0.00400	0.024685
G2	256	192	192	9.4	0.00220	0.016538
G3	384	384	256	37.7	0.00110	0.016113
G4	480	400	320	61.4	0.00082	0.010197

Table 6.1: Numerical grid used in this study and the grid spacings at the wall and the center of the channel in term of channel half-width(H).

6.2 Accuracy and performance of parallel compact fourth-order

The parallelisation of the compact fourth-order introduces two approximation procedures namely (i) the interface splitting algorithm and (ii) the approximate projection method. The first approximation opts for a performance by trading off some negligible errors. Otherwise the calculations of convective and diffusive terms could be four times more expensive if the direct method from the ScaLAPACK was used. The approximate projection method offers a convenient implementation to existing second-order codes. The use of three ghost cell required by the fourth-order projection method has been avoided. It also reduces the cost of the projection step significantly. In what follows, the parallelised version is compared to the sequential one in term of accuracy and performance.

6.2.1 Performance of Approximate projection method with strongly implicit procedure

The approximate projection method requires solutions of Poisson equation. In the previous chapters, we have avoided this problem by simply using the direct solver. In general computational domains which we may not have homogeneous directions, iterative solver is a better choice. We choose the strongly implicit procedure (SIP) [Sto73] as the iterative kernel.

The algorithm of the projection method has the same structure as Alg.3 in chapter 5 except that the Poisson equation is solved by iterative method using SIP as the smoother. SIP is claimed to be a second-order approximation [Sto73] of the second-order discrete Laplacian. Therefore it is a reasonable approximation to the discrete Laplacian $\mathbf{D}_4\mathbf{G}_2$ used in the approximate projection method. In order to solve the Poisson equation, the residual is first computed using $\mathbf{D}_4\mathbf{G}_2$ and the error in the pressure is smoothened out by the special incomplete **LU**-decomposition. Once the desired residual is reached, the velocities are then corrected by the second-order approximation of the pressure gradient. This iteration is repeated until the residual $(div - \mathbf{D}_4\mathbf{G}_4(p + \phi))$ reaches the desired value or the maximum number of iteration is exceeded.

The performance of SIP in solving $\mathbf{D}_4\mathbf{G}_2$ is compared with the direct solver in Tab.6.2. In this table, the number of iteration was set to 16 for the SIP and the threshold for stopping criteria is set very small such that the algorithm went through every iteration. The results were produced using INTEL Fortran compiler on AMD Opteron 8216. The direct solver is faster when the number of grid cells are small. At the highest number of cells, the SIP is slightly faster, but with a non-machine-accuracy divergence. The second-order SIP works well with the approximate projection method. In fact, it works as well with the exact projection method but this combination does not fit well with the parallelisation because of the incompatibility of the ghost cell. In summary, the iterative solver for the approximate projection method works correctly and efficiently. It should be noted that the CPU times is different from Tab.3.8 reported earlier because several factors such as storage of the coefficients, the calculation of the velocity correction and the version of the compiler.

However, the results reported in this table are performed under identical conditions which allow a direct comparison.

Grid	Time		Divergence	
	DIRECT	SIP	DIRECT	SIP
$32 \times 32 \times 32$	0.16	0.24	5.76E-15	9.93E-6
$64 \times 64 \times 64$	2.77	3.21	2.56E-14	8.37E-5
$96 \times 80 \times 96$	6.79	6.60	1.56E-14	2.17E-5
$128 \times 128 \times 128$	22.86	18.84	2.25E-14	4.47E-5

Table 6.2: CPU-seconds per time integration on AMD opteron 8216 of the sequential fourth-order scheme with direct and iterative solvers for the pressure.

6.2.2 Accuracy of the parallel fourth-order compact scheme

The most important requirement of parallel algorithms is that it gives a correct answer. Performance of the algorithm is then the next property to consider. To verify the accuracy of the parallelised version of the compact fourth-order scheme, the previous turbulent channel flow on grid F is used. The posteriori testing is performed and the statistics of the flow fields are compared. The computational domain is decomposed into four equal parts by two slicing, one in the streamwise and the other in the spanwise direction. These domains are distributed to four processors. The initial solution is taken from the result of the sequential version. The flow is then let to evolve for $600H/u_b$ which is equivalent to 48 through flow. During this time, the bulk velocity fluctuates within 0.4% which suggests that the flow does not go in to a different equilibrium. This means the transition phase can be skipped and the statistics sampled during this period can be used directly. The mean streamwise velocity and the r.m.s. of the velocity fluctuations from the second-half of the sampling differ from the first half not more than 0.7%.

The statistics of the cell-average values of this simulation are plotted in Fig.6.1 together with another simulation obtained from the sequential version, sampled in the same way. The profiles of the mean streamwise velocity and r.m.s. of the velocities fluctuations from both version are collapsing. Note that the cell-averaged of the r.m.s of the velocities fluctuations shown here are not the Reynolds normal stresses shown earlier in Fig3.5(c) due to a reason which will be described later. The skewness factors close to the wall are in excellent agreement. There are small deviations near the center of the channel which can be attributed to incomplete convergence of the statistics or minor differences in the distribution of the velocity.

The most notable deviation can be seen in the flatness factor $F(w)$ at the first cell next to the wall. Here, the sequential version predicts a smaller flatness than the one predicted from the spectral scheme in [MKM99] but the parallelised version predicts a higher value (≈ 47), a comparable deviation from the spectral scheme. This difference between the two versions is only seen at the first cell of the wall-normal velocity. This should not be an effect of the interface splitting algorithm. The error of the interface splitting algorithm is set to be five order of magnitude smaller than the maximum velocities (the approximate bandwidth J was set to 9) and the velocity there is much smaller than the maximum velocity. Therefore, the difference in the first cell here should come from the treatment of the pressure term. Recall that we use the third-order approximation for the pressure term in the momentum equation for both versions. The only difference here is the velocity correction and the discrete Laplacian. Due to the velocity close to the war has a steep gradient and the level of fluctuation changes rapidly, the difference between the exact projection and the approximate projection become notable in the fourth-order statistics, close to the wall.

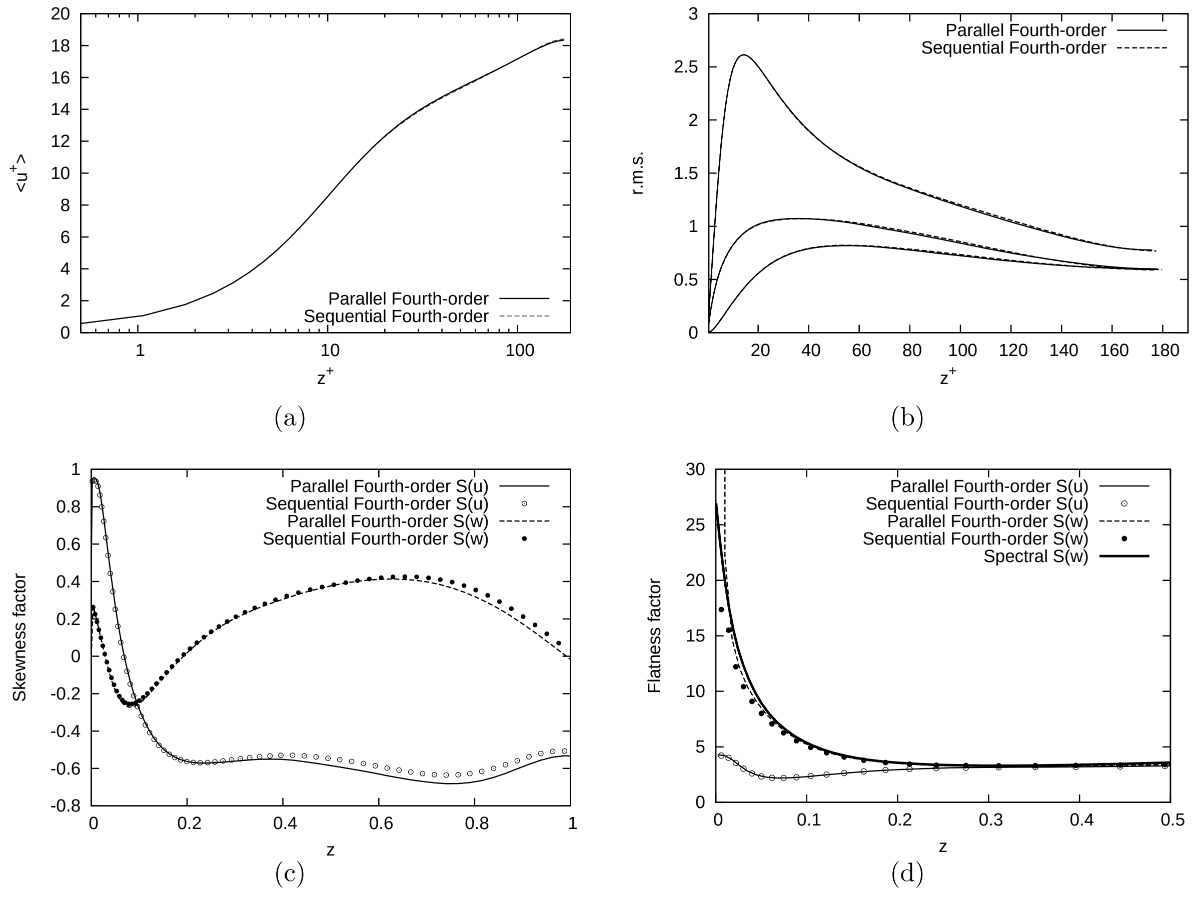


Figure 6.1: Comparison of statistics from the solutions of parallel and sequential versions of the fourth-order scheme on 128^3 grid cells.

Another factor that can affect accuracy of the interpretation of the statistics are the type of average values being sampled. When the grids are staggered, one can store the statistics on their original positions or averages them to the pressure cells. MGLET samples the second-order statistics at the pressure cells by simply averaging the two cells sharing the same pressure cell. The third- and the fourth-order statistics, however, are stored at their original positions. The averaging procedures, of course, introduce a filter to the velocity field and thus the information on small scale are less accurate. This would imply that we need even higher-order approximations to process the statistics.

The effect of using lower-order approximation statistics can be highlighted using the Reynolds shear stress profiles shown in Fig. 6.2. When the cell-averaged values are used to compute $< u'w' >$, u and w are averaged to the pressure cell where $< uu >$, $< ww >$ and $< uw >$ are computed and stored. The profile of the cell-averaged $< u'w' >$ close to the wall of both versions significantly deviate from the theoretical predictions of the Reynolds shear stress. Post-processing by de convolving the stored cell-averaged $< u'w' >$ (not shown) reduce the deviation by half. However, when $< u'w' >$ is taken directly from the surface-averaged at the top of the u-momentum cell, the profile follows the theoretical prediction perfectly. Another effect that one must take in to account when interpret the statistics

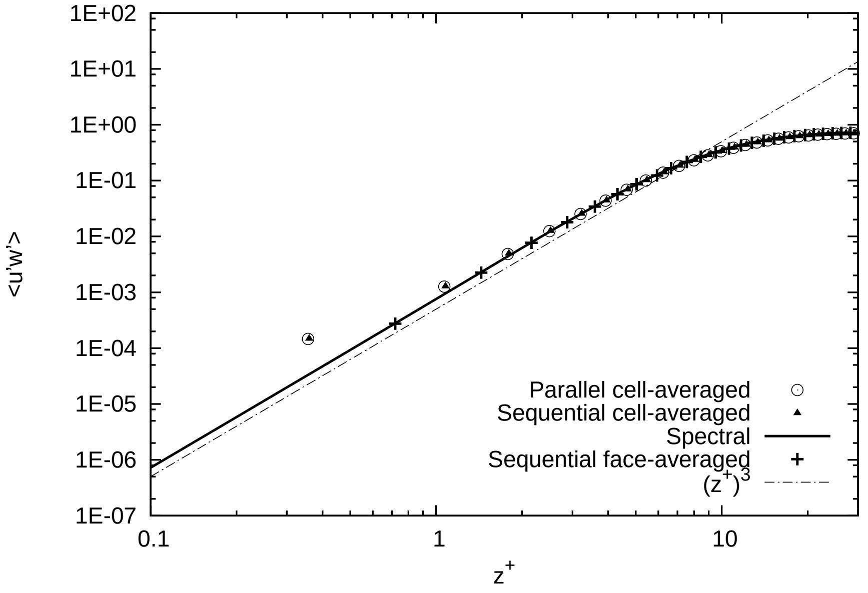


Figure 6.2: Effect of the sampling procedures to the Reynolds shear stress in turbulent channel flow.

of a solution of the finite volume discretisation is the nonlinear correction. Previously in chapter 3, we report the r.m.s of the velocities using fourth-order approximation of $< u'_i u'_i >$ extracted from the momentum equation using a sequential code. In Fig.6.1(b), we use the cell-averaged values because we do not yet have the infrastructure to gather these special statistics from different processors. In Fig.6.3, the cell-averaged and the surface-averaged values (with nonlinear correction) of the r.m.s. of the velocity fluctuations are compared. Actually, the contribution of the nonlinear-correction is small. When the solution is solved using the full fourth-order scheme, but sample the surface-averaged $< u'_i u'_i >$ without the nonlinear correction gives the r.m.s. profiles in between the two profiles plotted in Fig.6.3. This is not unexpected because the magnitude of the nonlinear correction is second-order. Since, we are dealing with $< u'_i u'_i >$ which is already a second-order quantity. We could only expect a small differences in the profile. Nevertheless, this correction is necessary for a fine comparison with the spectral scheme. If this correction is neglected, the level of the r.m.s will be notable lower than the spectral solution.

In practice, one can deconvolve the cell-average values to pointwise ones but it would involve a triple deconvolution. A very high order approximation would be needed to avoid a damping of high wave component. At this moment, there are no infrastructures to sample face-averaged values or pointwise values in the parallel version of the code and therefore the results in the rest of the chapter will be reported using the cell-average values. This should be noted for future improvement.

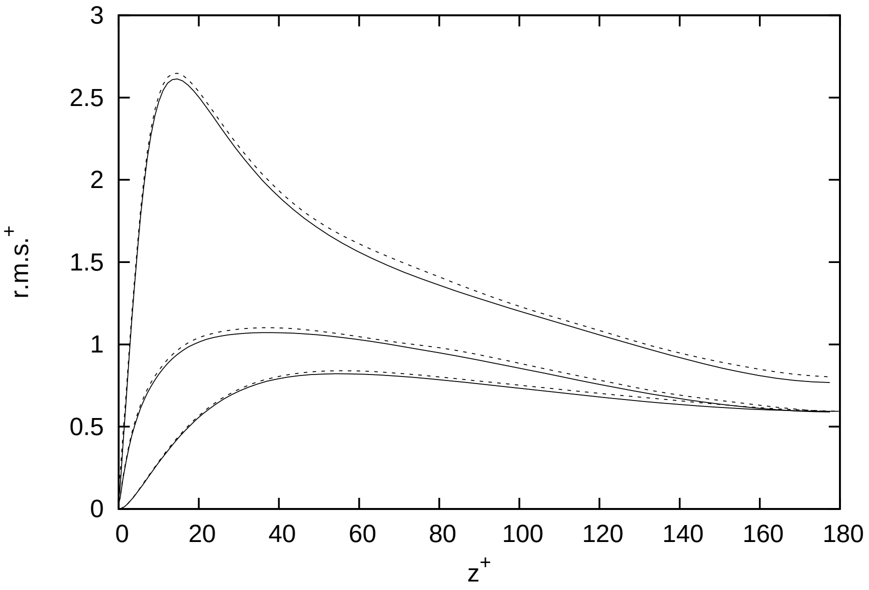


Figure 6.3: Effect of the nonlinear correction to the Reynolds normal stress in turbulent channel flow: solid-line:cell-averaged values and dash-line:surface-averaged values with nonlinear correction. The top pair is $< u'u' >$, the middle pair is $< v'v' >$ and the bottom pair is $< w'w' >$.

The probability density functions (PDF) are investigated in Fig.6.4. At first it was difficult to make a comparison of the fourth-order results with the reference [MKM99] . This is due to the normalisation, which can bring a different scaling when the cumulative density functions are computed with different intervals. This results in a different weighting and the

PDF curve can be scaled up or down. In Fig. 6.4(a) , PDF(u) at $z^+ = 5$ are shown. The number of interval and the range are set to 201 similar to the reference. The right figure shows a similar plot at the center of the channel. Here, the number of interval is set to 160. In both figures, all the distributions are comparable. The distribution of the fourth-order at the center of the channel is slightly broader than that of the spectral scheme. Nonetheless, there are no notable differences between the parallel and the sequential versions. According to these results, we can conclude that the parallelisation of the compact fourth-order is implemented correctly and the parallel algorithms used here do not have negative effects on the quality of the solution.

6.3 DNS of turbulent channel flow on fine grids

Ideally, numerical grids used in DNS should be able to represents all active structures of the flows. The Kolmogorov length scale, $\eta = (\nu^3/\epsilon)^4$, is commonly accepted as the smallest scale that is dynamically significant. This requirement is perhaps too severe to fulfil. Another argument on grid resolution is that the grid should be sufficient to accurately capture the viscous dissipation mechanism which is the sink of energy in turbulent flows. It is possible to use model spectrum of isotropic turbulent flow to show that the viscous dissipation is taken place in the range $0.1 \leq k\eta \leq 1$ which is corresponding to 6η to 60η [WHS07]. In the same reference, the recommended resolution for wall bounded flow is given in Tab.6.3. In the center of the channel, one can assume that the turbulence structures are similar to isotropic turbulent and use the corresponding grid resolution. In this section, the parallel compact fourth-order scheme is applied to turbulent channel flows using the grid resolution close to those recommended resolution. Note that this grid resolution is coarser than those given in [MM98] which are $[7.5^+, 4.4^+, 0.33^+]$ for the streamwise, spanwise and wall-normal directions, respectively.

Another criterion one should be careful when performing a numerical simulation of turbulence flow is the computational domain. It should be large enough to contain all significant structures of the flow. When there is a homogeneous direction and periodic boundary conditions are used, the domain length in that direction must be large enough such that the correlations drops to zero. Further, Kim and Adrian [KA99] have shown that in high Reynolds number flows, the small structures can interact in accumulated ways and become a very large structure. This phenomenon can be important in the numerical simulation of turbulent channel flow in which the homogeneity is assumed on the streamwise-spanwise plane. In their work, they explain that the large structures of the channel flow can be a result of interaction among the small hairpin structures. They uses premultiplied spectra to show that larger structures (low wave number) are created when the Reynolds number was increased. When the computational box is not large enough to allow these large-scaled structure to formed, statistics and turbulence mechanisms may not represent the correct physics. If they contain enough energy, the numerical simulations of turbulent channel flow on different domain (with the same grid resolutions) may deliver different results. However it is not yet known at what Reynolds number these large-scaled structures become significant and how much they affect the flow statistics. To get rid of this uncertainty, three DNS of turbulent channel flows up to $Re_\tau = 590$ are performed using the same computational box and comparable resolutions as those used in [MKM99] which will be used as the reference. Computational domain, grid resolutions, nominal flow conditions which are used to enforce the flows and the resulting Reynolds number are listed in Tab.6.4.

In the next step, we extend the study to a higher Reynolds number, the turbulent channel flow $Re_\tau = 950$. In order to avoid excessive use of resources, we scaled down the computational domain and perform a simulation using the recommended grid resolutions. It will be explain later that this domain is sufficiently large for this simulation. All of the simulations in this section are run on ALTIX 4700. The number of processors in each simulation spans from 8 to 128.

Flow direction	DNS	Wall-resolved LES
$\triangle x^+$ (streamwise)	10-15	50-100
$\triangle y^+$ (spanwise)	5	10-20
$\triangle z^+$ (wall-normal)	1	1
Number of points in $0 < z^+ < 10$	3	3

Table 6.3: Recommended grid resolutions for DNS and LES from [WHS07].

6.3.1 Comparison with spectral scheme on the same computational domain

Table 6.4 lists the Reynolds numbers and the resulting resolution of the respective grid used to simulate the flows. The nominal Reynolds number based on the friction velocity is given by the pressure gradient and the dynamic viscosity imposed to the flow. The corresponding effective Reynolds number is computed from the mean shear stress obtained from the simulations.

The mean streamwise velocity profiles of the fourth-order scheme in Fig.6.5 collapse on those of the spectral scheme. The r.m.s. of the cell-averaged fluctuations in the spanwise and wall-normal velocities are in excellent agreement with the spectral scheme. The peaks of u-r.m.s. are correctly predicted. Away from the wall, the predictions of r.m.s. of the

Case	Nominal Re_τ	Effective Re_τ	$\triangle x^+$	$\triangle y^+$	$\triangle z_w^+$	$\triangle z_c^+$	$\frac{tu_b}{L_x H}$
T180-G1	180.0	179.7	17.7	5.90	0.72	4.4	100
T395-G2	392.2	391.9	9.61	4.81	0.86	6.5	41.1
T590-G3	587.2	575.1	9.45	4.70	0.63	9.3	17.3
T950-G4	943.1	943.6	12.41	6.21	0.78	9.7	21.0

Table 6.4: Summay of turbulent channel flow simulations on fine grids..

streamwise velocity are slightly lower than what predicted by the spectral scheme. This difference is due to the averaging procedure mentioned earlier which explicitly filters the field and remove the small scale from the statistics. This shortcoming can be improved by deconvolve the field to pointwise value before the taking the sampling or deconvolve the field to the surface or adding nonlinear correction into it such that $\overline{u_i u_i}$ represent the true Reynolds normal stresses in the momentum equation like what we did earlier in chapter 3.

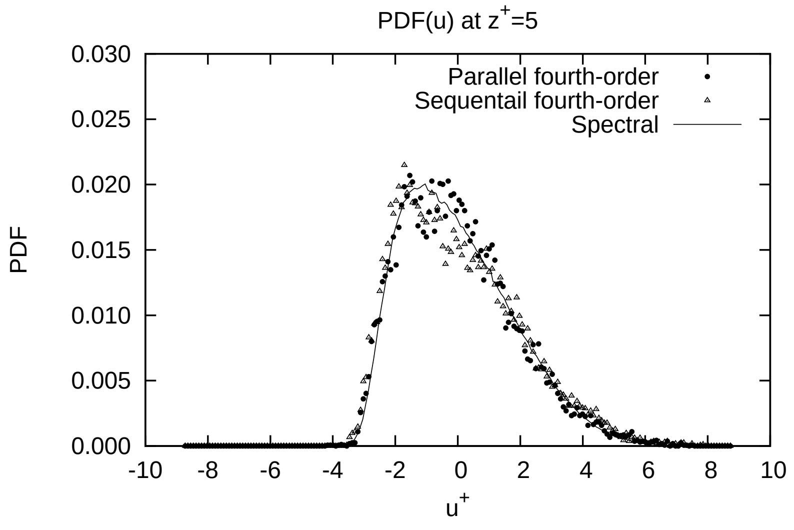
The profiles of the skewness factor of the wall-normal velocity at higher Reynolds numbers (Fig.6.6) agree with the spectral scheme better than what seen on the lowest Reynolds number ($Re_\tau = 180$). The flatness factors are also correctly predicted. At the center of the channel, the spectral code predicts $[F(u), F(v), F(w)] = [3.58, 3.73, 3.92]$ and the fourth-order predicts $[3.57, 3.40, 3.76]$.

One-dimensional energy spectra are investigated in Fig.6.7 which shows some interesting results. Near the wall, E_{uu} of the fourth-order scheme in T180-G1 drops near the end of the spectrum while that of the other Reynolds number follows the profile of the spectrum almost to the end of the Nyquist limit. On the other hand, the spectrum on T180-G1 follows that of the spectral scheme up to 75% of the resolvable frequency but the profiles on higher Reynolds number follows the predictions of spectral scheme up to only 60%. This opposite trend can be attributed to the grid spacing in the streamwise and the wall normal direction including the profile of the energy spectrum itself. The grid spacing in the streamwise direction of T180-G1 is approximately twice larger than those used in the other two cases. Therefore, near the wall it is not as accurate as the other cases. At the center of the channel, the grid spacing of T180-G1 in the wall-normal direction is two-third of the others and the slope of the reference spectrum is steeper which means the small scale structures is less significant than the other two cases. These two factors lead to a better prediction of energy cascade and in turn predicts an accurate energy spectrum there.

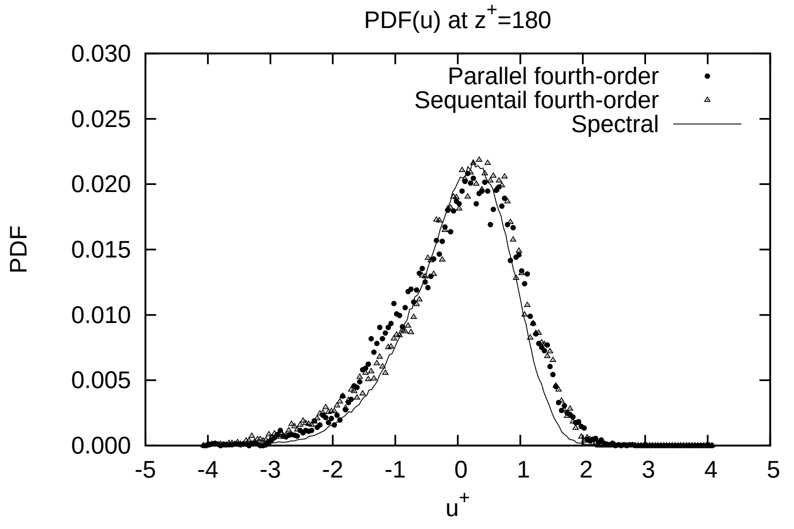
6.3.2 Comparison with spectral scheme using a smaller computational domain

In what follows, the turbulent channel flows at $Re_\tau = 950$ is investigated. The results in this section are compared with results obtained using spectral code in [HJ08]. Here the numerical simulations are performed on the same physical domain as used previously on T395-G2 and T590-G3. This domain is 12-times smaller than that used in the reference—a factor of four and three in the streamwise and spanwise directions respectively. It can be expected that if there were very large scale structures in the flow which exceed the computational domain used in this simulation, physics of the flow cannot be accurately represented. It is a common practice to justify the size of computational domain by taking the length that the autocorrelations of the velocity become zero. However, it can not be done prior to the simulation. Further, the autocorrelation goes in the opposite direction with the onset of the large scale structures. The higher the Reynolds number, the shorter length the velocities are correlated. Investigation of the autocorrelation reported in [Hu06] reveals that the autocorrelation R_{uu} drops to zero at approximately $l_x = 3.98$ and 2.76 for $Re_\tau = 360$ and 720, respectively. Linear interpolation shows that, for $Re_\tau = 590$, R_{uu} should reach zero at $l_x = 3.22$ which is larger than the domain used in [MKM99]. It is indeed result in a significant R_{uu} in their simulation. Our earlier simulations at this Reynolds number inherit this deficiency as well. According this findings, the domain used here for $Re_\tau = 950$ is large enough to be free from the error of periodic boundary conditions. However, it may not be large enough to capture large scale structures, if they are at all present. The purpose of this section is to investigate how the fourth-order will perform in an even higher Reynolds number using a comparable solution, assuming that the large-scale structures do not yet affect the flow.

The first- and second-order statistics in Fig.6.8 shows a similar agreement seen previously. Near the wall, the one-dimensional energy spectrum E_{uu} collapse on that of the reference solution (Fig6.9(a)). At the center of the channel, the profile of E_{uu} from the fourth-order is highly satisfactory over 60% of the whole spectrum (6.9(b)). According to the energy spectrum from the spectral code, 99% of the fluctuations energy is stored within the first 40 mode. Similarly, 99.9% and 99.99% of the fluctuations energy are stored in the first 76 and 120 modes, respectively. The fourth-order delivers the same number of mode for 99% but 74 and 106 modes for the other two predictions. The wavelength of the wave corresponding to 74-th mode is about $0.085H$ and it is resolved by approximately 6.5 cells per wavelength. This means over 99.9% of fluctuations energy are resolved by more than 6 cells. At this resolution, the error of the approximation of the 74-th wave number is less than 0.1%. This resolving power of the fourth-order scheme is thus highly satisfactory.

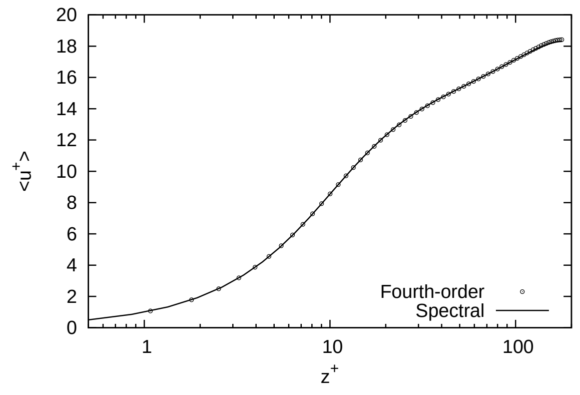


(a)

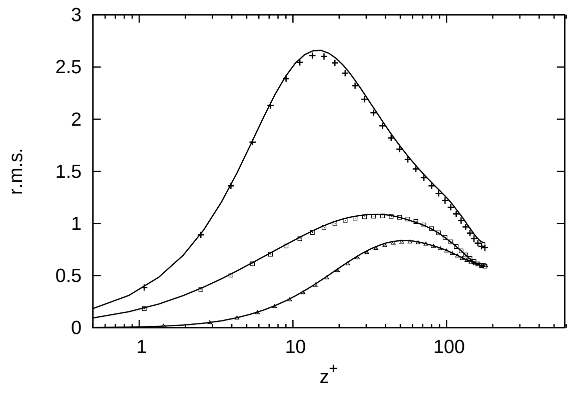


(b)

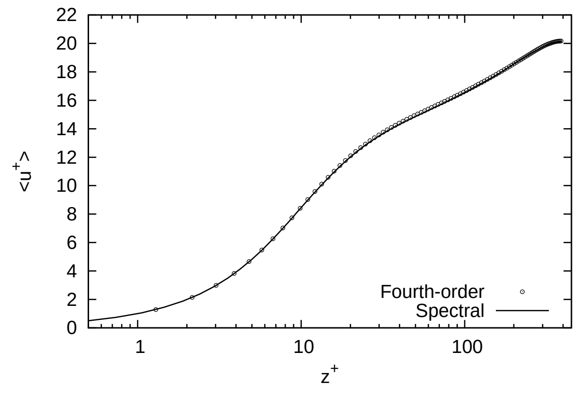
Figure 6.4: Comparison of probability density functions of the streamwise velocity between parallel and sequential versions of the fourth-order scheme at two positions: (a) close the wall at $z^+ = 5$ and (b) at the center of the channel.



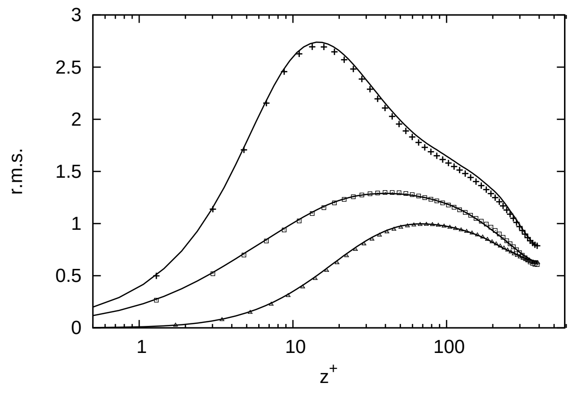
(a)



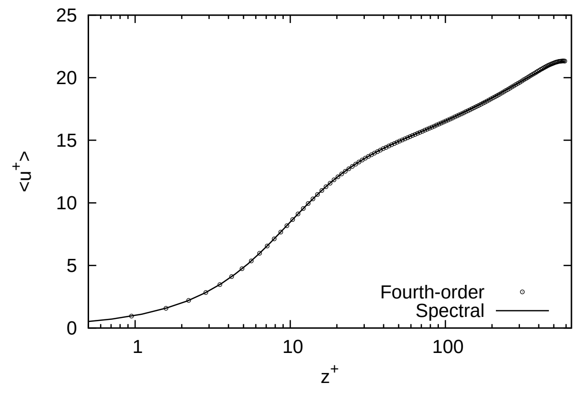
(b)



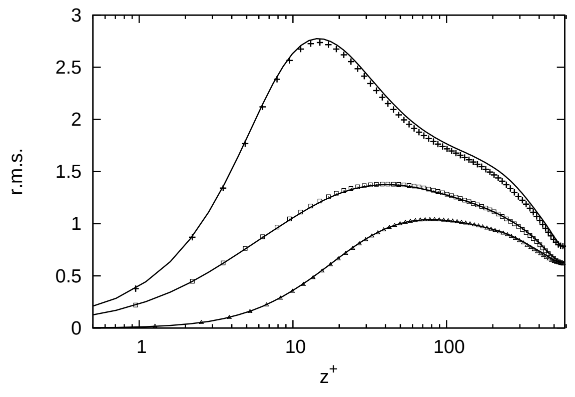
(c)



(d)

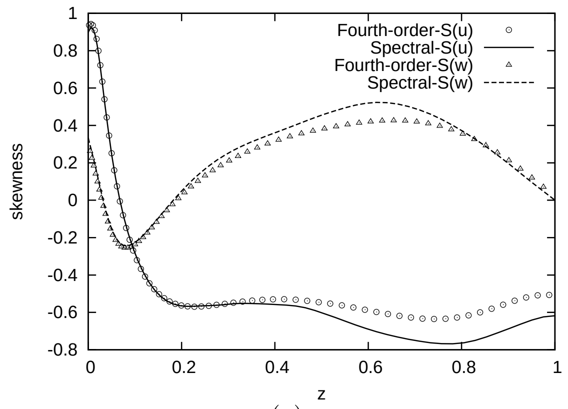


(e)

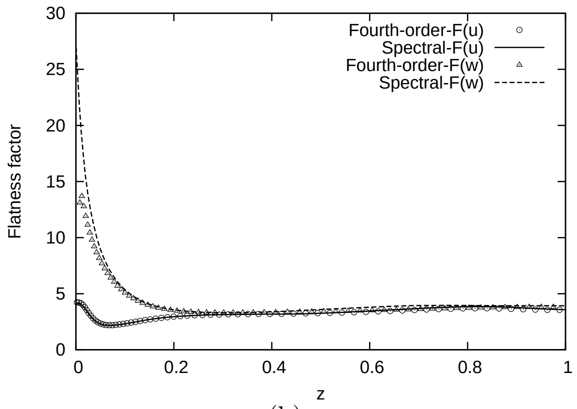


(f)

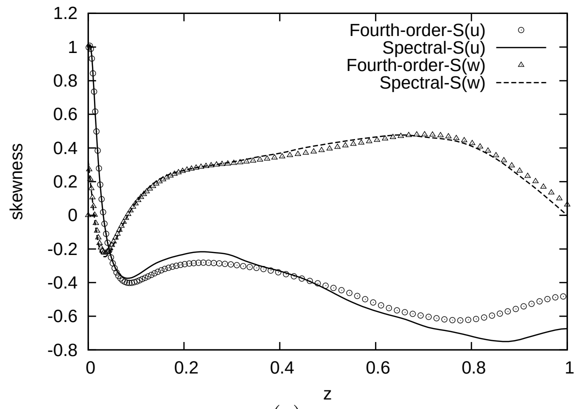
Figure 6.5: Mean streamwise velocity profile (left) and r.m.s. of velocity fluctuations (right) of turbulent channel flow on the domain and the grids comparable with those used by spectral code[MKM99]. Top:$Re_\tau = 180$; middle:$Re_\tau = 395$; bottom:$Re_\tau = 590$. Plus symbol: u-component ; square symbol: v-component; triangle symbol: w-component.



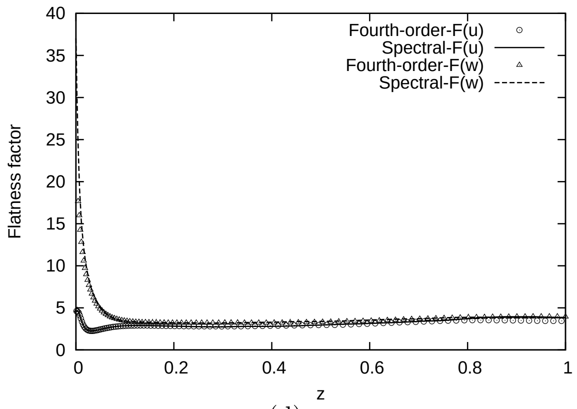
(a)



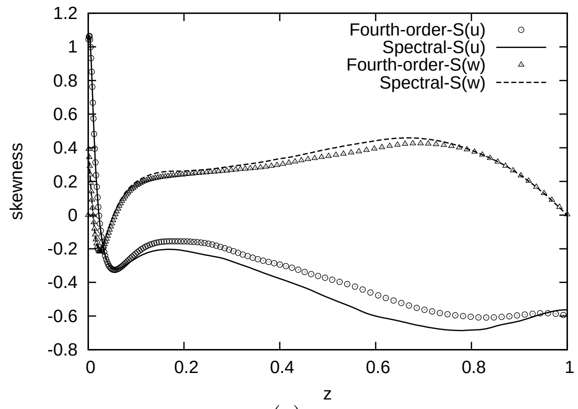
(b)



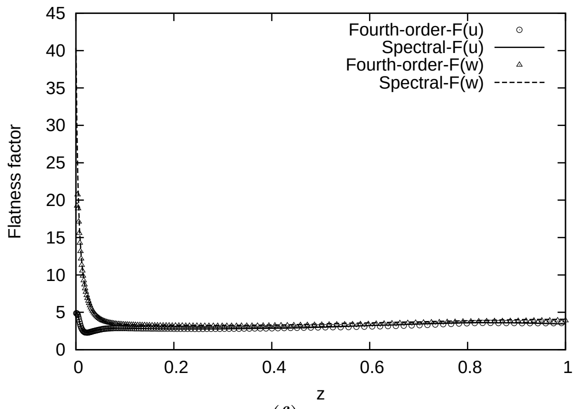
(c)



(d)

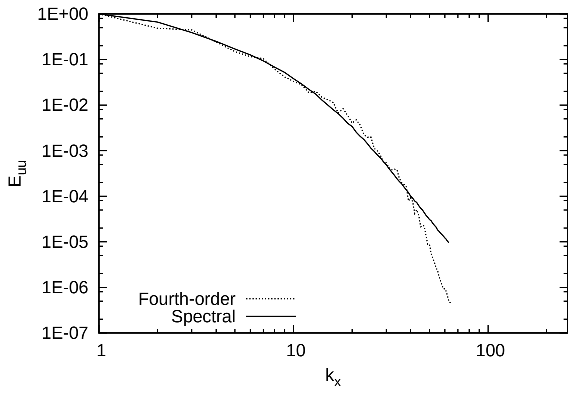


(e)

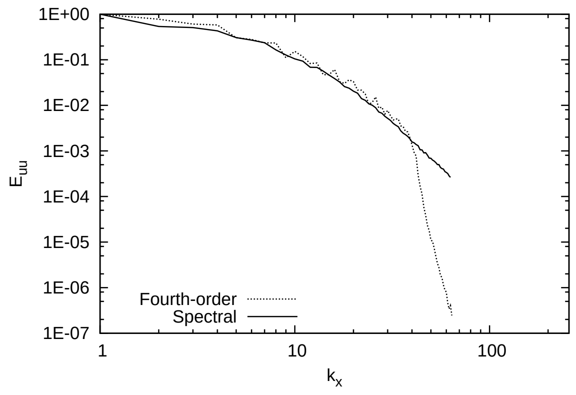


(f)

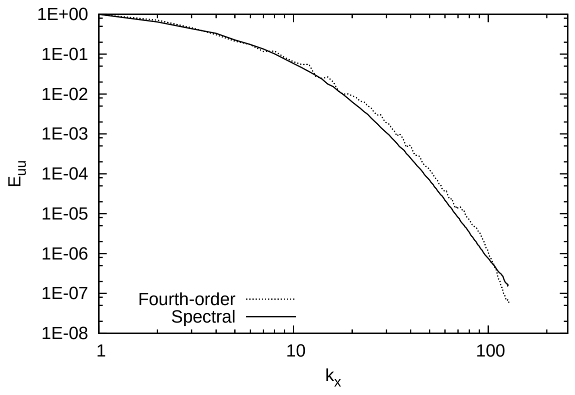
Figure 6.6: Skewness factor (left) and flatness factor (right) of turbulent channel flow. Top:$Re_\tau = 180$. Middle:$Re_\tau = 395$. Bottom:$Re_\tau = 590$.



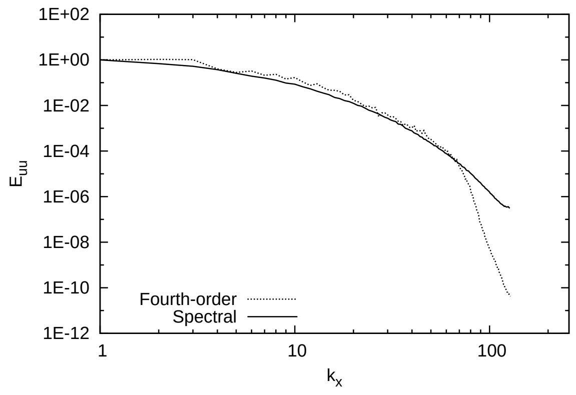
(a)



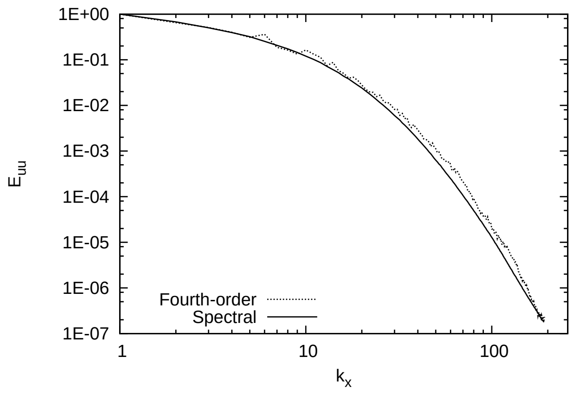
(b)



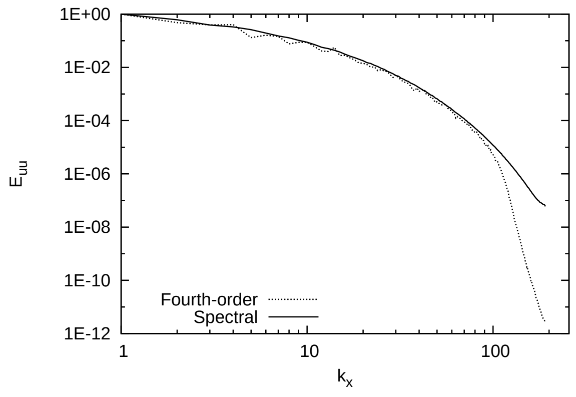
(c)



(d)

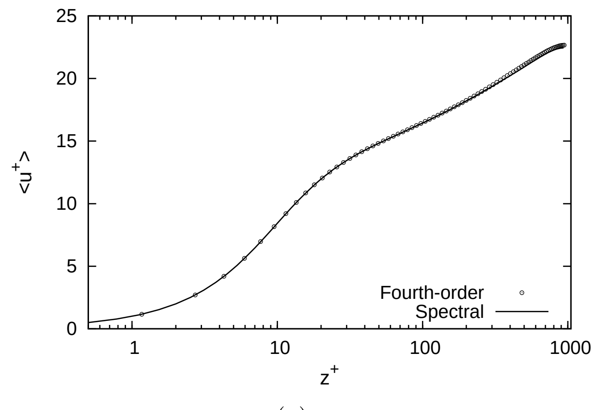


(e)

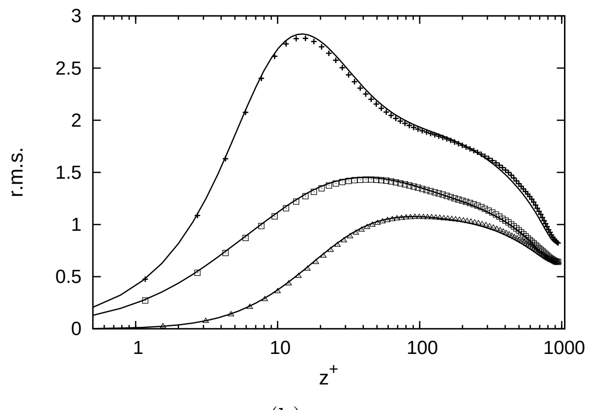


(f)

Figure 6.7: One-dimensional spectra of the streamwise velocity at $z^+ = 5$ (left) and at the center of the channel (right) of turbulent channel flow. Top: $Re_\tau = 180$. Middle: $Re_\tau = 395$. Bottom: $Re_\tau = 590$.

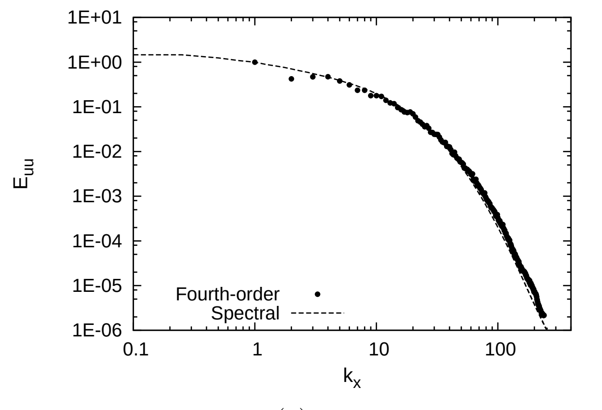


(a)

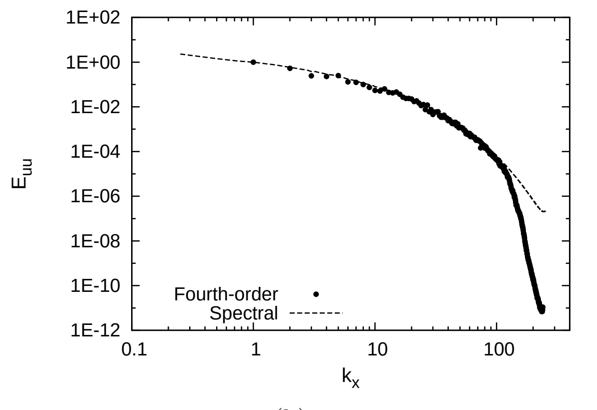


(b)

Figure 6.8: Mean streamwise velocity (a) and r.m.s. of velocities fluctuations(b) of T950-G4 (every two grid cells are shown).



(a)



(b)

Figure 6.9: One-dimensional energy spectra of the streamwise velocity of T950-G4. at $z^+ = 5$ (a) and at the center of the channel (b).

6.4 DNS of turbulent channel flow on coarse grids

In general, the main reason for using non-spectral schemes is to simulate fluid flows with complex geometry. In such cases, it may not be possible to represent the geometries and boundary conditions perfectly. The verification of the DNS solutions by physical experiments is therefore limited to certain accuracy. Taking the current computational resources into account, it is more practical for a DNS of non-spectral schemes to aim for correct predictions of the first- and second-order statistics. It is, perhaps, a philosophical question to defined the word *correct* statistics. Here it only means that the solution has reached *grid independent* solution for the first- and second-order statistics. We borrow the term *grid independent solution* from RANS which usually means that the first-order statistics are unchanged when the grid is significantly refined. Example of such solutions are the solutions on grid M2 and grid F seen in chapter 3. Here we extend it to the second-order statistics as well. For most engineering applications, these two statistics are the most important quantities. Therefore it is valuable to verify a necessary grid resolution that can deliver such solutions using the compact fourth-order scheme.

Earlier in chapter 3, grid M2 delivered the excellent result up to the second-order statistics. Third- and fourth-order statics were in good agreement. The number of cells on this grid was about one-third of the one used by the spectral scheme. This can not be interpreted as a superiority of the fourth-order scheme. It only means that the error in the approximation of the small scales *missed* by the fourth-order scheme are very small and do not appear in the first- and second-order statistics. It is seen from the previous sections that the energy spectra can be represented up to roughly 60% of the Nyquist limit before starting to experience a sharp drop. The energy spectrum depends on the balance of the terms in the momentum equation, more precisely, the balance between the productions that feeds the energy to small scales and the dissipation which drains the energy. This balance depends as well on the profile of the energy spectrum. These interactions are nonlinear and difficult to analyse. In this section, the turbulent channel flows are performed again but using the grid that was used earlier for the Reynolds number one step lower. Summary of the cases and the respective resolutions based on wall-unit are listed in Tab.6.5. Note that all the grid spacings here exceed the recommended values.

The mean streamwise velocity profile in Fig.6.10 shows that grid spacing twice coarser than the recommended values in some directions, does not affect the mean streamwise velocity. However, there are some noteworthy observations can be made here. The r.m.s. of the streamwise fluctuation near the peak is least accurate on the lowest Reynolds number. At this Reynolds number the grid spacing in the streamwise direction is also the coarsest ($38z^+$). The profile of the streamwise fluctuations approaches that of the spectral scheme again close

Case	Effective Re_τ	$\triangle x^+$	$\triangle y^+$	$\triangle z_w^+$	$\triangle z_c^+$	$\frac{tu_b}{L_x H}$
T395-G1	390.3	38.33	12.81	1.56	9.63	35
T590-G2	579.5	14.22	9.47	1.27	9.58	68
T950-G3	965.9	15.96	7.98	2.13	15.97	59

Table 6.5: Summay of turbulent channel flow simulations on coarse grids..

to the center of the channel. On the other hand, the profile of the r.m.s. of the velocity fluctuations of T950-G3 is very good close to the peak, but the fluctuations are slightly over predicted the center. This indicates that the grid spacing near the channel is relatively coarse. Here the grid spacing in the wall-normal direction is the coarsest among other cases. According to the grid resolutions used in this study, it could be possible that the first- and second-order statistics of the turbulent channel flows can be predicted accurately using grid spacings twice larger than the recommended values in Tab.6.3.

6.4.1 DNS of turbulent channel flow on very coarse grids

In order to verify whether the grid resolution resolutions twice coarser than the recommended values can be used to produce an excellent prediction of the first- and second-order statistics, the turbulent channel flows are investigated for two Reynolds numbers, 180 and 950 using $[\triangle x^+, \triangle y^+, z^+] = [30, 10, 2]$ at the wall. The maximum grid spacing in the wall-normal direction is set to 10^+ and 16^+ for $Re_\tau = 180$ and 950 , respectively. This leads to the total number of cells of 0.37M and 9.2M. A factor of 5.7 and 6.7 smaller than the grid conforming to the respective reference resolution.

The results of the mean streamwise velocity and the second-order statistics in Fig.6.11 are very encouraging. The mean streamwise velocity profiles are almost collapsing on the reference profiles. The level of fluctuations in $Re_\tau = 180$ is lower than the reference solution as expected. In case of $Re_\tau = 950$, the r.m.s. is higher than the reference solution starting from $z^+ = 100$ up to the center of the channel. The mean streamwise profile is also notably lower than the that of the spectral scheme. This can be attributed to a relatively coarse grid in the wall-normal direction, away from the wall. At the center of the channel we used $z^+_{max} = 16$, compared to 10 in $Re_\tau = 180$. One-dimensional energy spectra of the streamwise velocity also behave nicely, even though a slight overshoot in the energy spectrum on the lower Reynolds number is observed.

According to the result in this section, we can conclude that , for a turbulent channel flow, the grid resolutions $[\triangle x^+, \triangle y^+, \triangle z^+_{wall}] = [30, 10, 2]$ are sufficient to deliver a satisfactory result up to the second-order statistics.

6.5 Efficiency and scalability

In this section, the performance of the parallel compact scheme is evaluated on two types of parallel machine, a low cost workstation and a supercomputer. The workstation used in this test is a four-processors of dual core AMD 8216. The supercomputer where the benchmarking of large-scaled simulation is performed is ALTIX 4700 located at Leibniz-Rechenzentrum.

The result from the workstation is presented in Tab.6.6. This table documents the result of the parallelisation of T180-G1. The domain is sliced first in the streamwise direction, then the spanwise direction and repeated in this order for each time the number of processors is doubled. According to the table, there is a significant overhead going from a single processor to two processors. This overhead is as high as one-third of the total work. The overhead is attributed to three factors, the communications, domain decomposition of the SIP and the interface splitting algorithm. All simulations in this section use 16 iterations of SIP. The absolute efficiency which is defined by $E_a = \frac{100T_1}{pT_p}$, relative to the time used on a single processor (T_1), is not reflecting the overhead of the parallelsation. It is thus a pessimistic indicator for the scalability. A more practical indicator, named here *incremental efficiency*, measures the performance gain when the number of processor is increased. It is defined by $E_i = \frac{100T_{p_1}}{(p_2/p_1)T_{p_2}}$ for $p_2 > p_1$. This indicator says clearly how much the increased processors are being used, relatively to the simulation on p_1 processors.

In this table, E_i is measured between the two consecutive number of processors. On the workstation, roughly 75% of the added processors helps in speeding up the solution process. The performance on ALTIX 4700 in Tab.6.7 shows a similar finding. Performance on a single processor is slightly lower than what seen on the workstation. The values of absolute efficiency are also lower. For the incremental efficiency the ALTIX 4700 shows an interesting behaviour. At lower number of processors, the incremental efficiencies are lower than what seen on the workstation but on eighth and sixteen processors the value are higher. This mean that the overhead on ALTIX 4700 increased sharply up to four processors and then become almost saturated. Incremental efficiency of 92% is the evident of this saturation. There are three factors contribute to this behaviour. First, when we increase the number of processor from one to two, we introduce communication overheads in the streamwise direction. Increasing the number of processor from two to four, overheads are added again in the spanwise direction. After this, no extra direction is added. The increased overheads at eight and sixteen processors are therefore not as high. This contribute to a higher incremental efficiency. This behaviour also seen on the cluster as well. The second contributor is the architectural design. ALTIX 4700 is composed of building blocks which are blade systems consisting of two dual core Intel Itanium2 Montecito per blade. These four cores share the same 8.5 GB/s memory bus. This blade is then linked to the other blades via 6.4Gb/s NUMAlink4. The peak performance of the blade is 25.6 GFlops and therefore 204.8 GB/s must be fed to each processor to achieve their full potential. Unfortunately, the bus is only 4% of this value and thus results in a sharp drop of performance in the first four processors. This effect is clearly shown in the speedup value of 1.76 at four processors, compared to 2.21 on the workstation.

p	CPU-seconds	Speedup	Efficiency (%)	
			E_a	E_i
1	12.47	—	—	—
2	8.47	1.47	74	74
4	5.65	2.21	55	75
8	3.54	3.52	44	80

Table 6.6: Scalability of parallel compact scheme of T180-G1 case on commodity workstation..

p	CPU-seconds	Speedup	Efficiency (%)	
			E_a	E_i
1	13.61	—	—	
2	9.87	1.38	69	69
4	7.75	1.76	43	64
8	4.21	3.23	40	92
16	2.38	5.72	35	88

Table 6.7: CPU-time per time step used in T180-G1 and the scalability of the fourth-order scheme on ALTIX 4700.

The comparison of the performance between the second-order and the fourth-order on ALTIX 4700 using a turbulent channel flow on T395-G2 is shown in Tab.6.8. On four and eighth processors, the fourth-order is slower than the second-order by a factor of 2.7. This factor is reduced to 2.1 on thirty two processors. Both schemes achieve almost ideal scalability, relatively to performance on four processors. In these regions, the incremental efficiency is free from the memory bottle neck mentioned earlier. The amount of memory per variable on 4 and 8 are 17.9MB and 8.96MB, respectively. Thus on eighth processor a single variable fits perfectly on the processor's cache. This better data locality is compensated with the increased communications and results in excellent scalability in both schemes. The part of the code which is the most communication intensive is the solution of pressure where the SIP is used. There, the subroutine must access two variables, the pressure and the residual. On sixteen processors, both data can be stored entirely in the cache which gives a super-linear relative speedup, shown by 101% incremental efficiency.

Another reason that made the fourth-order scheme less sensitive to the communication is the pattern of communication in the SIP. The original second-order code, first solve the **L**-system then synchronise the residual. It then solves **U**-system and synchronises again. The modified version for the fourth-order scheme only synchronises once after both systems are solved. This halves the number of the communications. Even though, the second-order scheme needs to synchronise just one ghost cell compared to two ghost cells in the fourth-order scheme, the lower number of communication helps avoiding the idle time of the processor in which they must wait for the mismatched send-receive to be complete. According to the result form chapter 2, the second-order needs twice more grid point per direction to have the same accuracy as the fourth-order scheme. Taking this resolution requirement and the time integration in to account, we can expect that the fourth-order is $6-8$ times more expensive than the second-order code. Comparing to the factor of 10 found earlier in chapter 2, the advantages of the fourth-order scheme is reduced slightly in the parallel version. This can be attributed to the overhead in the interface-splitting algorithm and the SIP solver. Nevertheless, the parallel version of the fourth-order scheme is still much more efficient than the second-order scheme.

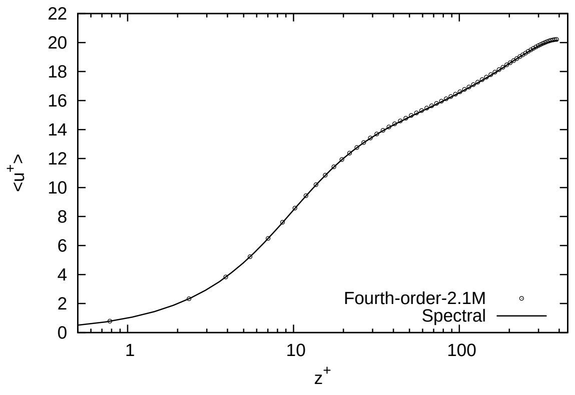
p	CPU-seconds		time-ratio	Incremental efficiency (%)	
	second-order	fourth-order		second-order	fourth-order
4	9.60	25.80	2.69	—	—
8	5.00	13.60	2.72	96	95
16	2.70	6.70	2.48	93	101
32	1.70	3.65	2.15	80	92

Table 6.8: Comparison of CPU-time per time integration and the scalability of the fourth- and the second-order schemes in T390-G2 on ALTIX 4700.

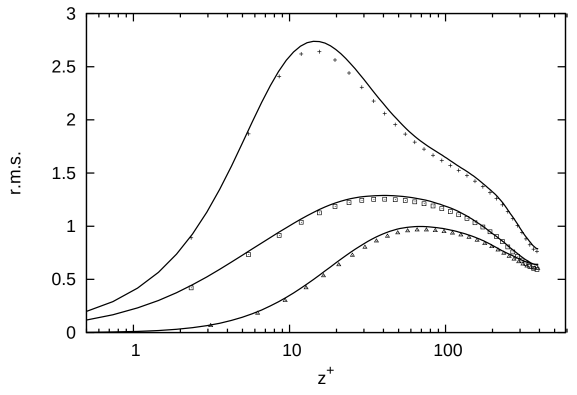
6.6 Conclusion

The results shown in this chapter highlights the desirable properties of the parallelised compact fourth-order scheme. It is highly accurate and can matched up with the spectral scheme with great accuracy. It can recreate the one-dimensional energy spectra up to 99.9% of the fluctuation energy using the same grid as the spectral scheme. If one satisfied with the grid independent solution of the first- and the second order statistics, a grid twice coarser than the usually recommended values can also be used. This is made possible by the fact that the small scales do not contribute much to the large scale structures.

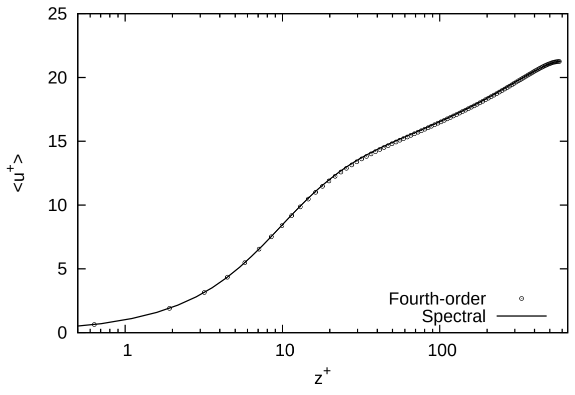
In general, it is relatively difficult to obtain a good scalability in fixed size speed up. Here we obtain approximately linear scalability which is a nice property thanks to the linear complexity of the algorithms we used. In every algorithms presented in this work, the complexity does not depends on the number of processor. Therefore the proposed scheme should be scalable for any number of processors, provided that the interconnection fabric of the parallel machine is scalable.



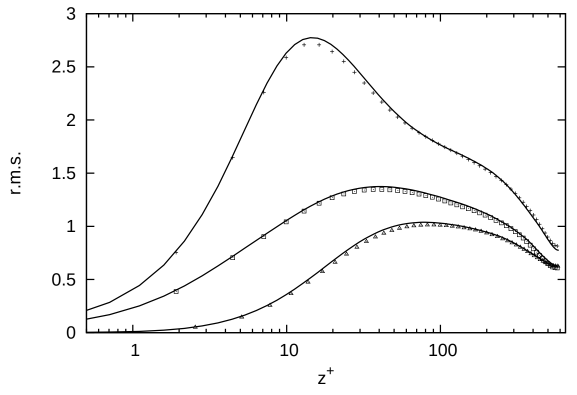
(a)



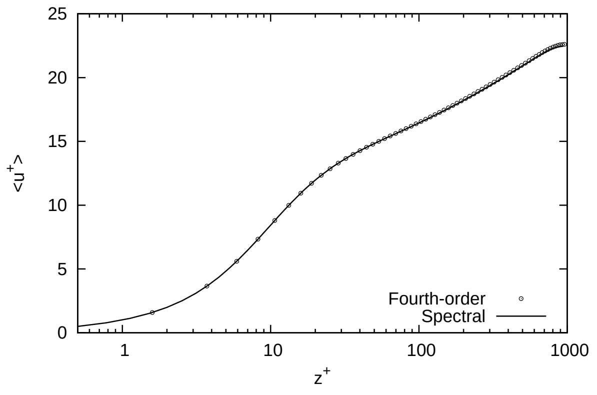
(b)



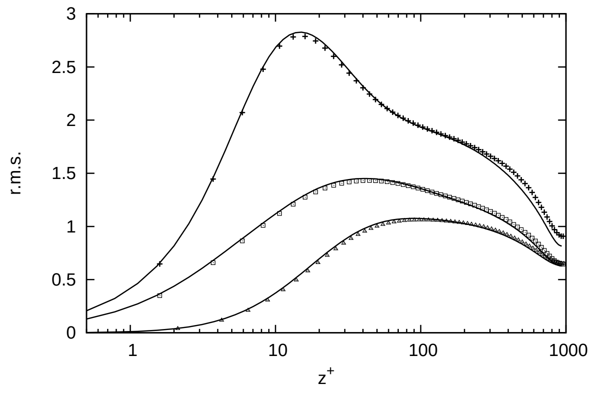
(c)



(d)

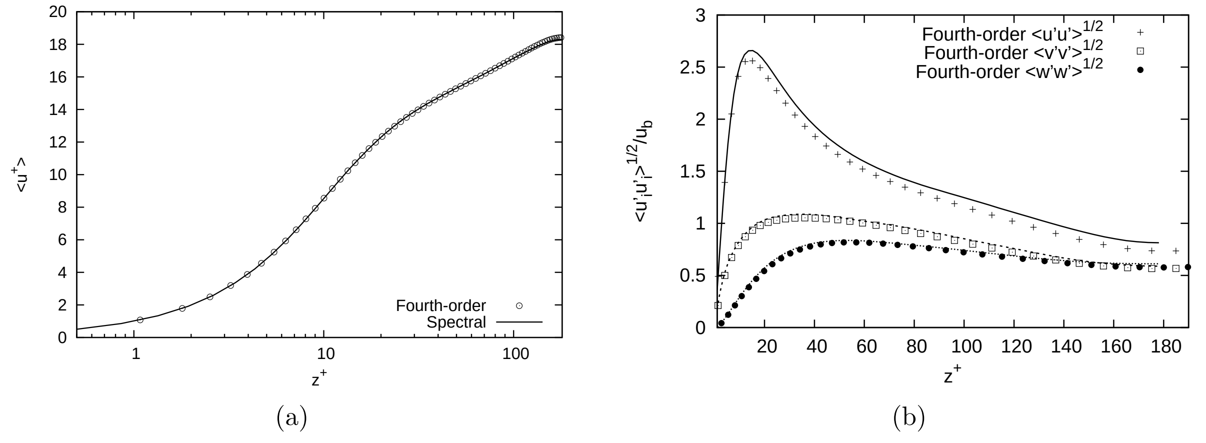


(e)



(f)

Figure 6.10: Mean streamwise velocity profile (left) and r.m.s. of velocity fluctuations (right) of turbulent channel flow on coarse grid. Top: $Re_\tau = 395$; middle: $Re_\tau = 590$; bottom: $Re_\tau = 950$. Plus symbol: u-component ; square: symbol: v-component; triangle symbol: w-component.



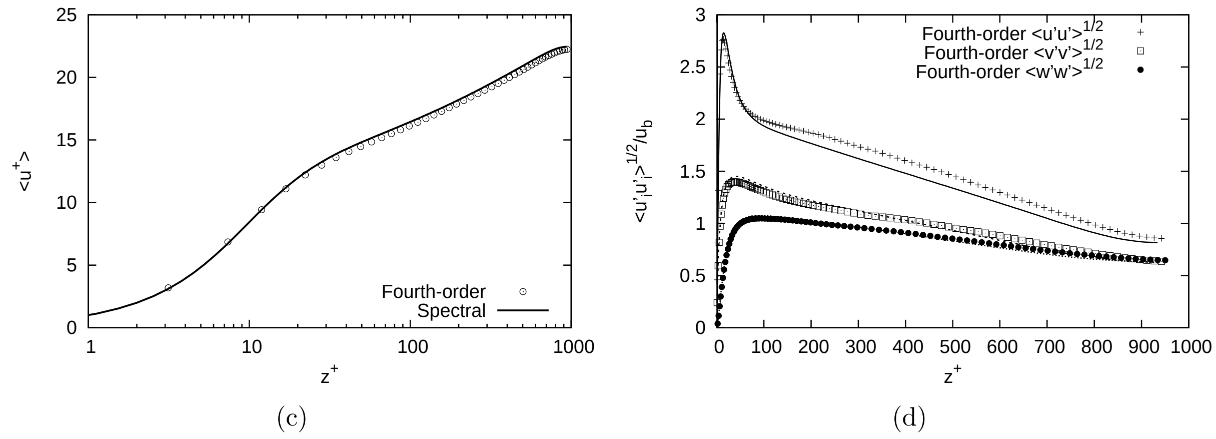


Figure 6.11: Mean streamwise velocity (left) and velocity fluctuations (right) normalised by the nominal parameters of $Re_\tau = 180$ (top) and 950 (bottom) on the grids twice larger than the recommended resolutions.

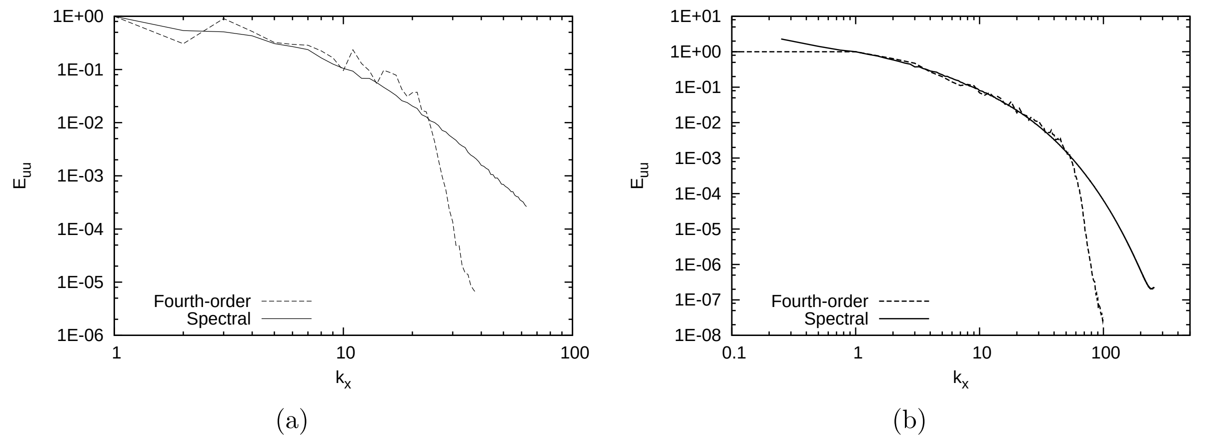


Figure 6.12: One-dimensional spectra E_{uu} of the streamwise velocity for $Re_\tau = 180$ (a) and 950 (b)on very coarse grids.

7 Conclusion and outlook

On the course of this work, the compact fourth-order scheme for numerical solutions of the Navier-Stokes equations (NSE) tailored specially for staggered grids have been developed. The scheme is efficiently parallelised and can be conveniently applied to any existing second-order code having two ghost cells. This development is a complete fourth-order scheme for finite volume discretisation on staggered grids which ensures a perfect mass conservation on every control volumes. Compared to earlier works on higher-order method for Navier-Stokes equations, we have considered many aspects of the numerical scheme such as accuracy, efficiency, scalability including compatibility for existing CFD code. The outcome of the work is a highly efficient algorithm solving Navier-Stokes equations which outperforms the standard second order schemes in both accuracy and performance. Careful evaluations have been carried out for laminar and turbulent flows. The approximations for the momentum term is the least accurate among the spatial approximations used in the NSE. The role and the importance of the pressure term have been investigated. Improving all other approximations to fourth-order but keeping that of the the pressure term at second-order will prevent the convergence rate to reach fourth-order. We also found an evident suggesting that the enforcement of continuity may be more accurate on staggered grids than on collocated ones. When the approximation of the pressure gradient and divergence is kept at second-order, the overall convergence rate of third-order can be achieved while it is limited to second-order on the collocated grids. This could be the reason why previous developments of higher-order scheme on collocated grids deliver disappointed results for turbulence flows, despite the fact that higher-order schemes were shown to be much more accurate than the second-order scheme on laminar flows.

Tridiagonal matrices of the compact scheme has been parallelised efficiently by the interface-splitting algorithm. This algorithm is an approximate method, but the accuracy of the approximation is predetermined by the cut-off threshold of the coefficients vector. The overhead of the algorithm is small when the size of the subsystem is properly chosen. The proposed algorithm has a minimum communication and the least number of floating point operations. The algorithm is designed in a way that the the additional computation and the bidirectional communication can be overlapped. This algorithm is shown to be at least four-times faster than the ScaLAPACK library. In optimal conditions, ideal absolute speedup can be obtained.

The developed NSE solver uses the novel divergence-free interpolation for the convective velocities. This interpolation ensures the divergence-free property of the convective velocities on all control volumes inside the computational domain. This divergence-free property of the convective velocity is a necessary condition for Galilean invariant of the numerical solution of the NSE and it can be very important in numerical simulations of turbulent flows on coarse grids. The divergence-free approximate projection method is developed to enforce the mass conservation using a narrow banded matrix. This method has a very good data locality and does not need to recompute the divergence during the solution process. The proposed approximate projection method shows an excellent correlation with the fourth-order projection method and it is fully divergence-free with third-order global convergence rate.

These pieces, when they are put together, create a highly efficient and scalable codes. To the best of the author knowledge, this scheme is the only non-spectral scheme that can produce a collapsing profile of the first- and second-order statistics of turbulent channel flows up to $Re_\tau = 950$ using approximately one-third of the total grid used by the spectral code. This reduction in number of cells is allowed by the fact that the small scales structures contribute very little to the large scale structures. Ultimately, we expect that the fourth-order scheme can use the grid resolution twice larger than the usual recommended values for turbulent channel flow, and yet deliver accurate results for the first- and second-order statistics.

The current code can easily be used to simulate some classic turbulent flows such as flows over mounted cube, backward/forward facing step, flow over rectangular cavity, mixing layers etc. Due to the excellent scalability of the parallelised version of proposed scheme, we can expect that direct numerical simulations of these classic test cases can be performed by the proposed scheme at higher Reynolds number and expand our understanding of Turbulent flows.

In order for the proposed scheme to be applicable for complex geometries, further developments are needed. This can be in a form of higher-order immersed boundary method or conformal matching grid of Cartesian and curvilinear grids or Cartesian and unstructured grids. This topic could be a very interesting research. The invention of fourth-order scheme could be beneficiary for large-eddy simulations (LES). We have seen clearly that the one-dimensional energy spectra of the fourth-order scheme follows that of the spectral scheme 50% longer than that of the second-order. The local truncation errors are now small. For a well known turbulent channel flow at $Re_\tau = 180$, the DNS of the fourth-order scheme using 32^3 grid cells delivers 5% error in the mean flows profile. What sub-grid scale modelling will improve the solution ? How to adjust the modelling parameter, in comparison to the second-order ? These questions must be answered in order to enable the fourth-order scheme for industrial applications.

Bibliography

[ABC00] Ann S. Almgren, John B. Bell, and William Y. Crutchfield, *Approximate projection methods: Part i. inviscid analysis*, SIAM Journal on Scientific Computing **22** (2000), no. 4, 1139–1159.

[ABM04] Travis Austin, Markus Berndt, and David Moulton, *A memory efficient parallel tridiagonal solver*, Preprint LA-UR-03-4149, 2004.

[ABS96] Ann S. Almgren, John B. Bell, and William G. Szymczak, *A numerical method for the incompressible navier–stokes equations based on an approximate projection*, SIAM J. Sci. Comput. **17** (1996), no. 2, 358–369.

[AG96] P. Arbenz and W. Gander, *Direct parallel algorithms for banded linear systems*, Z. Angew. Math. Mech. **76** (1996), 119–122.

[ARM01] Demuren Ayodeji, Wilson Robert, V., and Carpenter Mark, *Higher-order compact schemes for numerical simulation of incompressible flows, part i: Theoretical development*, Numerical Heat Transfer, Part B **39** (2001), no. 3, 207–230.

[BCM01] David L. Brown, Ricardo Cortez, and Michael L. Minion, *Accurate projection methods for the incompressible navier—stokes equations*, J. Comput. Phys. **168** (2001), no. 2, 464–499.

[Bon91] Stefan Bondeli, *Divide and conquer: a parallel algorithm for the solution of a tridiagonal linear system of equations*, Parallel Computing **17** (1991), no. 4-5, 419–434.

[Bro95] David L. Brown, *Performance of under-resolved two-dimensional incompressible flow simulations*, JCP **122** (1995), no. 1, 165–183.

[Cho68] Alexandre J. Chorin, *Numerical solution of the navier-stokes equations*, Mathematics of Computation **22** (1968), no. 104, 745–762.

[CM04] Saugata Chakravorty and Joseph Mathew, *A high-resolution scheme for low mach number*, INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS **46** (2004), no. 3, 245–261.

[CSM04] Lacor Chris, Smirnova Sergey, and Baelmansb Martine, *A finite volume formulation of compact central schemes on arbitrary structured grids*, Journal of computational physics **198** (2004), no. 2, 535–566.

[Den03] Filippo Maria Denaro, *On the application of the helmholtz-hodge decomposition in projection methods for incompressible flows with general boundary conditions*, International Journal for Numerical Methods in Fluids **43** (2003), no. 1, 43–69.

[DM01] A. Das and J. Mathew, *Direct numerical simulation of turbulent spots*, Computers & Fluids **30** (June 2001), 533–541.

[EK05] A. Amiri Elhami and S. Hannani Kazemzadeh, *Evaluation of a fourth-order finite-volume compact scheme fore les with explicit filtering*, Numerical Heat Transfer **48** (2005), 147–163.

[GA93] David Gottlieb and Saul Abarbanel, *The stability of numerical boundary treatments for compact high-order finite-difference schemes*, J. Comput. Phys. **108** (1993), no. 2, 272–295.

[Gho96] Sandip Ghosal, *An analysis of numerical errors in large-eddy simulations of turbulence*, J. Comput. Phys. **125** (1996), no. 1, 187–206. MR MR1381809 (96k:76078)

[GPPZ98] Alfonsi Giancarlo, Giuseppe Passoni, Lea Pancaldo, and Domenico Zampaglione, *A spectral-finite difference solution of the navier-stokes equations in three dimensions*, International Journal for Numerical Methods in Fluids **28** (1998), no. 1, 129–142.

[GS97] D. Gaitonde and J. S .Shang, *Optimized compact-difference-based finite-volume schemes for linear wave phenomena*, J. Comput. Phys. **138** (1997), no. 2, 617–643. MR MR1607490 (98j:78005)

[Gul00] J. Gullbrand, *An evaluation of a conservative fourth order dns code in turbulent channel*, Tech. report, Center for Turbulence Research,Stanford University, 2000.

[Heg96] M. Hegland, *Divide and conquer for the solution of banded linear systems of equations*, in Proceedings of the Fourth Euromicro Workshop on Parallel and Distributed Processing, IEEE Computer Society Press, Los Alamitos, 1996, pp. 394–401.

[HJ08] Sergio Hoyas and Javier Jimenez, *Reynolds number effects on the reynolds-stress budgets in turbulent channels*, Physics of Fluids **20** (2008), no. 10, 101511.

[HRT95] J. C. Hardin, J. R. Ristorcelli, and C. K. W. Tam, *ICASE/LaRC Workshop on Benchmark Problems in Computational Aeroacoustics (CAA)*, Workshop held in Hampton, VA, 24-26 Oct. 1994; sponsored by NASA, Washington and Inst. for Computer Applications in Science and Engineering, May 1995, pp. 24–26.

[Hu06] Z. W. Hu, *Wall pressure and shear stress spectra from direct simulations of channel flow*, AIAA Journal **44** (2006), no. 7, 1541–1549.

[KA99] K. C. Kim and R. J. Adrian, *Very large-scale motion in the outer layer*, Physics of Fluids **11** (1999), no. 2, 417–422.

[KL96] J. W. Kim and D. J. Lee, *Optimized compact finite difference schemes with maximum resolution*, AIAA Journal **34** (1996), 887–893.

[KM85] J. Kim and P. Moin, *Application of a fractional-step method to incompressible navier-stokes equations*, Journal of Computational Physics **59** (1985), no. 2, 308–323.

[KMM87] J. Kim, P. Moin, and R. D. Moser, *Turbulence statistics in fully developed channel flow at low reynolds number*, Journal of Fluid Mechanics **177** (1987), 133–166.

[Kni08] R. Knikker, *Study of a staggered fourth-order compact scheme for unsteady incompressible viscous flows*, International Journal for Numerical Methods in Fluids **59** (2008), 1063–1092.

[Kob99] Marcelo H. Kobayashi, *On a class of pade finite volume methods*, Journal of computational physics **156** (1999), no. 1, 137–180.

[Lel92] Sanjiva K. Lele, *Compact finite difference schemes with spectral-like resolution*, J. Comput. Phys. **103** (1992), no. 1, 16–42. MR MR1188088 (93g:76086)

[LPJN93] Josep-Lluis Larriba-Pey, Angel Jorba, and Juan J. Navarro, *Spike algorithm with savings for strictly diagonal dominant tridiagonal systems*, Microprocess. Microprogram. **39** (1993), no. 2-5, 125–128.

[Man04] M. Manhart, *A zonal grid algorithm for DNS of turbulent boundary layers*, Computers & Fluids **33** (2004), no. 3, 435–461.

[MKM99] Moser, Kim, and Mansour, *Direct numerical simulation of turbulent channel flow up to re_τ = 590*, Physics of Fluids **11** (1999), no. 1, 943–945.

[MLVM98] Y. Morinishi, T. S. Lund, O. V. Vasilyev, and P. Moin, *Fully conservative higher order finite difference schemes for incompressible flow*, J. Comput. Phys. **143** (1998), no. 1, 90–124. MR MR1624676 (99a:76100)

[MM98] Parviz Moin and Krishnan Mahesh, *Direct numerical simulation: A tool in turbulence research*, Annual Review of Fluid Mechanics **30** (1998), no. 1, 539–578.

[MSKR02] M. Meinke, W. Schroeder, E. Krause, and T. Rister, *A comparison of second- and sixth-order methods for large-eddy simulations*, Computers & Fluids **31** (2002), 695–718(24).

[MZH85] M.R. Malik, T.R. Zang, and M.Y. Hussaini, *A spectral collocation method for the navier-stokes equation*, JCP **61** (1985), 64–88.

[Nab99] Reinhard Nabben, *Decay rates of the inverse of nonsymmetric tridiagonal and band matrices*, SIAM Journal on Matrix Analysis and Applications **20** (1999), no. 3, 820–837.

[NLF03] Santhanam Nagarajan, Sanjiva K. Lele, and Joel H. Ferziger, *A robust high-order compact method for large eddy simulation*, J. Comput. Phys. **191** (2003), no. 2, 392–419.

[PKP01] J. M. C. Pereira, M. H. Kobayashi, and J. C. F. Pereira, *A fourth-order-accurate finite volume compact method for the incompressible navier-stokes solutions*, Journal of computational physics **167** (2001), no. 1, 217–243.

[PS04] M. Piller and E. Stalio, *Finite-volume compact schemes on staggered grids*, Journal of computational physics **197** (2004), no. 1, 299–340.

[RM91] M. M. Rai and P. Moin, *Direct simulation of turbulent flow using finite-difference schemes*, JCP **96** (1991), 15–53.

[SK78] A. H. Sameh and D. J. Kuck, *On stable parallel linear system solvers*, J. ACM **25** (1978), no. 1, 81–91.

[SSN89] Xian-He Sun, Hong Zhang Sun, and Lionel M. Ni, *Parallel algorithms for solution of tridiagonal systems on multicomputers*, ICS '89: Proceedings of the 3rd international conference on Supercomputing (New York, NY, USA), ACM, 1989, pp. 303–312.

[Sto73] Harold S. Stone, *An efficient parallel algorithm for the solution of a tridiagonal linear system of equations*, Journal of the ACM **20** (1973), no. 1, 27–38.

[Sun95] Xian-He Sun, *Application and accuracy of the parallel diagonal dominant algorithm*, Parallel Computing **21** (1995), no. 8, 1241–1267.

[SW07] Olga Shishkina and Claus Wagner, *A fourth order finite volume scheme for turbulent flow simulations in cylindrical domains*, Computers & Fluids **36** (2007), no. 2, 484–497.

[Vas00] Oleg V. Vasilyev, *High order finite difference schemes on non-uniform meshes with good conservation properties*, J. Comput. Phys. **157** (2000), no. 2, 746–761.

[VV03] R. W. C. P. Verstappen and A. E. P. Veldman, *Symmetry-preserving discretization of turbulent flow*, J. Comput. Phys. **187** (2003), no. 1, 343–368.

[Wan81] H. H. Wang, *A parallel method for tridiagonal equations*, ACM Trans. Math. Softw. **7** (1981), no. 2, 170–183.

[WD01] Robert V. Wilson and Ayodeji O. Demuren, *Higher-order compact schemes for numerical simulation of incompressible flows, part ii: Applications*, Numerical Heat Transfer, Part B **13** (2001), no. 39, 231–255.

[WHS07] Claus Wagner, Thomas Hüttl, and Pierre Sagaut (eds.), *Large-eddy simulation for acoustics*, thirteenth ed., pp. 89–127, Large-eddy simulation for acoustics, 2007.

[Wil80] J. H. Williamson, *Low-storage Runge-Kutta schemes*, J. Comput. Physics **35** (1980), no. 48, 48–56.