Proceedings of the ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering
OMAE2013
June 9-14, 2013, Nantes, France

OMAE2013-11000

EFFICIENT COMPUTATION AND MODELING OF VISCOUS FLOW EFFECTS IN COMFLOW

Henri J.L. van der Heiden*
Computational Mechanics and Numerical Mathematics
Institute of Mathematics and Computing Science
University of Groningen
Groningen, 9747 AG
Email: h.j.l.van.der.heiden@rug.nl

Peter van der Plas
Arthur E.P. Veldman
Roel W.C.P. Verstappen
Roel Luppes
Computational Mechanics and Numerical Mathematics
Institute of Mathematics and Computing Science
University of Groningen
Groningen, 9747 AG

ABSTRACT

In offshore applications, details of viscous flow effects can become relevant when predicting e.g. drag forces on the columns of oil drilling rigs, or the flow around a semisubmersible in figure 1. This motivates a novel approach for efficiently simulating viscous flow effects at high Reynolds numbers with the CFD simulation tool ComFLOW.

In ComFLOW, the Navier–Stokes equations can be solved for one-phase and for two-phase flow. The equations are discretized second-order in space, and second-order in time. An Improved Volume-of-Fluid (IVOF) algorithm is used for free-surface advection and reconstruction [1, 2].

Modeling viscous flow effects in high Reynolds number flows requires a turbulence model that provides accurate results on coarse grids. We pursue to achieve a high local grid resolution in a computationally efficient manner. Both approaches are tested for flows around a square cylinder: grid refinement at Reynolds numbers 10 and 100, and the turbulence model at Reynolds number 22,000.

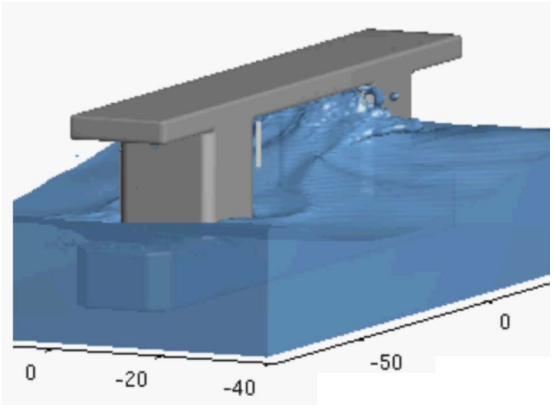

FIGURE 1: Snapshot of a simulation of free-surface flow around on a semisubmersible

1 DISCRETIZATION OF THE NAVIER–STOKES EQUATIONS

An excellent model for incompressible fluid flow is provided by the Navier-Stokes equations. The set of equations consists of the continuity equation

$$\mathscr{M}\mathbf{u} = 0, \tag{1}$$

*Corresponding author.

where $\mathscr{M} = \nabla\cdot$ is the divergence operator, and the momentum equation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathscr{C}(\mathbf{u},\mathbf{u}) + \mathscr{G}p - \nu\mathscr{D}\mathbf{u} = \mathbf{f}, \tag{2}$$

based on the convection operator $\mathscr{C}(\mathbf{u},\mathbf{v}) = \mathbf{u}\cdot\nabla\mathbf{v}$, the pressure gradient operator $\mathscr{G} = \nabla$, the diffusion operator $\mathscr{D}(\mathbf{u}_h) = \nabla\cdot\nabla\mathbf{u}_h$ and forcing term $\mathbf{f}$. The kinematic viscosity is denoted by ν.

The continuity equation (1) is discretized at the 'new' time level $n+1$ to give

$$M\mathbf{u}_h^{n+1} = -M^\Gamma\mathbf{u}_h^{n+1} \tag{3}$$

where M acts on the internal of the domain and M^Γ acts on the boundaries of the domain.

Convection and diffusion are discretized explicitly in time (we write D for the discrete diffusion operator and $C(\mathbf{u}_h)$ for the discrete convection operator.) The divergence and the pressure gradient are discretized at the new time level. For simplicity, in the discussion below, we use forward Euler time integration. If we denote the diagonal matrix containing the fluid volumes of the momentum cells by Ω, the discretized momentum equation is given by

$$\Omega\frac{\mathbf{u}_h^{n+1}-\mathbf{u}_h^n}{\Delta t} = -C(\mathbf{u}_h^n)\mathbf{u}_h^n + D\mathbf{u}_h^n - G\mathbf{p}_h^{n+1}, \tag{4}$$

where Ω is a diagonal matrix containing the fluid volumes of the momentum cells. The discrete convection operator is *skew-symmetric*, i.e.

$$C(\mathbf{u}_h^n) + C(\mathbf{u}_h^n)^* = 0. \tag{5}$$

Finding the solution to the system of equations (3) & (4) is split in two steps. First an auxiliary variable $\mathbf{u}_h^*$ is defined by the equation

$$\Omega\frac{\mathbf{u}_h^*-\mathbf{u}_h^n}{\Delta t} = -C(\mathbf{u}_h^n)\mathbf{u}_h^n + D\mathbf{u}_h^n. \tag{6}$$

Using this variable in (4) gives

$$\Omega\frac{\mathbf{u}_h^{n+1}-\mathbf{u}_h^*}{\Delta t} = -G\mathbf{p}_h^{n+1}. \tag{7}$$

Substitution of equation (7) in the continuity equation (3), gives rise to the following system of equations

$$\Delta t M\Omega^{-1}G\mathbf{p}_h^{n+1} = M\mathbf{u}_h^* + M^\Gamma\mathbf{u}_h^{n+1}, \tag{8}$$

which is often referred to as the discrete pressure Poisson equation, as it can be viewed as a discretization of the equation $\mathscr{M}\circ\mathscr{G}\mathbf{p}_h = \mathscr{M}\mathbf{u}_h$. Note, however, that we are not directly discretizing the composed operator $\mathscr{M}\circ\mathscr{G}$ here, but its separate parts $\mathscr{M}$ and $\mathscr{G}$. Hence, of sole importance is the accuracy of the discretization of the divergence and gradient operators $\mathscr{M}$ and $\mathscr{G}$, respectively. This should be kept in mind when assessing the accuracy of the method.

2 LARGE EDDY SIMULATION OF TURBULENT FLOW

In order to simulate the high Reynolds number turbulent flows that are associated to offshore applications, large eddy simulation (LES) modeling is necessary. Simply put, the aim of LES is to simulate those scales of motion that can be represented on the computational grid as accurately as possible through modeling of the interaction between resolved and subgrid scales.

In mathematical terms, an approximation of the solution to the original Navier–Stokes equations is sought that is dynamically less complex than the full turbulent solution. Therefore we demand that the *analytical* model separates the scales of motion that can be resolved (larger than a length scale Δ) from the scales of motion that cannot be resolved ($< \Delta$). The latter are commonly called *subgrid scales* as Δ is typically associated with the grid size.

In a turbulent flow, the production of small scales takes place through the non-linear convective term. The only mechanism counteracting the production of small scales of motion is diffusion. The equilibrium between production (by convection) and dissipation (by diffusion) of small scales cannot be reached on the computational grid. This consideration gives rise to two modeling options: either the production of subgrid scales is restrained or the dissipation of subgrid scales is increased.

An increase of dissipation of subgrid scales of motion is pursued in eddy viscosity models. Below we will describe the QR (scalar) eddy-viscosity model. Stopping the convective production of subgrid scales beyond length scale Δ through smoothing of the convective term is the strategy of so-called *regularization* turbulence models. We will describe a symmetry-preserving regularization model below.

QR model

Important information of turbulent flow physics is contained in the invariants of the rate-of-strain tensor $S(\mathbf{u}) = \frac{1}{2}\left((\nabla\mathbf{u})^T + \nabla\mathbf{u}\right)$. As shown in [3], the rate of dissipation of the

scales contained in a domain Ω_Δ of size Δ, can be expressed in terms of the second invariant $q(\mathbf{u})$ of $S(\mathbf{u})$ as

$$Q(\mathbf{u}) \equiv \int_{\Omega_\Delta} q(\mathbf{u})\, \mathrm{d}\Omega \equiv \int_{\Omega_\Delta} \frac{1}{2}\mathrm{tr}\, S^2(\mathbf{u})\, \mathrm{d}\Omega. \tag{9}$$

The third invariant, denoted by $r(\mathbf{u})$ can be integrated over the same domain to give

$$R(\mathbf{u}) \equiv \int_{\Omega_\Delta} r(\mathbf{u})\mathrm{d}\Omega \equiv \int_{\Omega_\Delta} -\frac{1}{3}\mathrm{tr}\, S^3(\mathbf{u})\, \mathrm{d}\Omega = \int_{\Omega_\Delta} -\det S(\mathbf{u})\, \mathrm{d}\Omega. \tag{10}$$

When positive, the quantity (10) gives a measure for the convective production of dynamical scales $< \Delta$. When negative, energy is transferred from subgrid-scale structures to the resolved scales. In order to separate the scales properly, a turbulence model should close the transfer of energy in either direction.

The analysis in [3] shows that the minimum amount of eddy-viscosity needed for a solution that contains only dynamical scales $> \Delta$, can be evaluated in terms of the invariants as

$$\nu_{\text{eddy}} = \frac{3}{2}\frac{1}{\lambda_\Delta}\frac{|R(\mathbf{u})|}{Q(\mathbf{u})}, \tag{11}$$

where λ_Δ is the eigenvalue of the discrete diffusive operator corresponding to the scale Δ. Taking Δ identical to the grid size, this eigenvalue is a measure for the dissipation of the smallest *resolvable* scales. We will refer to this model as the QR model.

Note that the classical Smagorinsky turbulence model (see e.g. [4]) is formulated in terms of the invariant $Q(\mathbf{u})$ only, and dissipates energy also on well-resolved (even laminar) scales in the flow.

Regularization model

In order not to interfere with the subtle energetic balance between the convection and diffusion in a turbulent flow on resolved scales, it is important to preserve the symmetries of the Navier-Stokes equations on a discrete level [5]. A symmetry-preserving regularization of the convective term smooths the original convective term while preserving its skew-symmetry. The smoothing takes place through a filter operation $\mathbf{u}_h \mapsto \overline{\mathbf{u}_h}$.

Verstappen applies the filter to the convective term [6] to obtain a family of symmetry-preserving regularization models. The discrete convective term is denoted by $C(\mathbf{u}_h)\cdot\mathbf{u}_h$. The second-order (in terms of the filter length) accurate regularization model from this family is given by

$$C_2(\mathbf{u}_h,\mathbf{u}_h) = \overline{C(\overline{\mathbf{u}_h})\cdot\overline{\mathbf{u}_h}}. \tag{12}$$

Selfadjointness of the filter assures the skew-symmetry of the original convective term.

In discrete terms, the filter operation F is an explicit operation and reads $F\mathbf{u}_h = (1-\tilde{D})\mathbf{u}_h$, where $\tilde{D} = -M^*\alpha^2 M$. Here, α is a strichtly diagonal matrix that contains the local values of the filter length. As before, M is the discrete divergence operator and $-M^*$ the discrete gradient operator. Note how the structure of $\tilde{D}$ resembles the structure of the symmetric discrete diffusion operator $D = -M^*M$, which ensures the selfadjointness of the filter operation.

The local filter length (the entries of α) will depend on the local flow physics. The analysis in [7] shows that an expression for the damping of the convective term length scale Δ can be derived. The damping factor f_2 is a functional of the (local) filter length α, i.e. $f_2 = f_2(\alpha(\mathbf{u}))$. Imposing a balance between the convective production of subgrid scales and the natural diffusive dissipation gives

$$f_2(\alpha)\,\frac{|R(\mathbf{u})|}{Q(\mathbf{u})} = \nu|\lambda_\Delta|, \tag{13}$$

from which the filter length can be determined. Note the resemblance of (13) to (11).

Blended model

Another model that we discern is the blended model, in which both the QR eddy viscosity model and the regularization model play a role, depending on the physics of the flow. The transfer of energy from resolved to subgrid scales (i.e. (10) is positive) is modeled by the QR eddy-viscosity model. The backscatter of energy from subgrid scales to resolved scales (i.e. (10) is negative) is prevented by the regularization model.

The mixture of these models allows for a proper separation of resolved and subgrid scales. Only if the energy is transferred to smaller scales of motion, the kinetic energy is dissipated from the resolved flow structures. The model is closed for backscatter by the regularization of the nonlinear convective interaction between subgrid and resolved scales.

Below, the performance of these three different models will be compared for flow around a square cylinder at Reynolds number $\mathrm{Re} = 22{,}000$.

The turbulent boundary layer

Because of the high Reynolds number associated to turbulent flows in offshore applications, it is computationally highly

undesirable to refine the grid to the level at which the boundary layer is resolved. The influence of the turbulent boundary layer on the effective wall-shear stress that the outer flow experiences is modeled by the Werner-Wengle model [8].

3 LOCAL GRID REFINEMENT

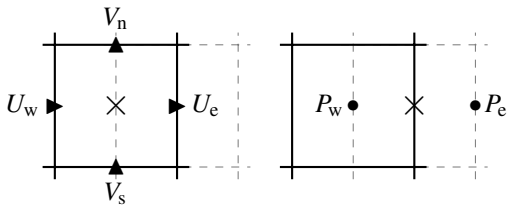

FIGURE 2: Regular discretization stencil for $M\mathbf{u}_h$ (left) and $G\mathbf{p}_h$ (right, horizontal component only) applied at the location indicated with $\times$.

The Navier–Stokes equations are discretized on an Arakawa C-grid as illustrated in figure 2. For brevity the third dimension, which is treated similarly, is omitted. The subscript ℓ is used to indicate the local refinement level, where $\ell = 0$ refers to the unrefined base grid (indexing is discussed in section 3.)

In the regular parts of the grid the divergence operator is discretized as follows (for the subscript convention consult figure 2)

$$M\mathbf{u}_h|_\ell = \Delta y_\ell (U_{\mathrm{e};\ell} - U_{\mathrm{w};\ell}) + \Delta x (V_{\mathrm{n};\ell} - V_{\mathrm{s};\ell}) \tag{14}$$

In order to let the discrete operators satisfy the adjoint condition,

$$G = -M^* \tag{15}$$

the pressure gradient is discretized as $G\mathbf{p}_h = -M^*\mathbf{p}_h$. This gives the following second-order central discretization:

$$\frac{P_{\mathrm{e};\ell} - P_{\mathrm{w};\ell}}{\Delta x_\ell} \qquad \frac{P_{\mathrm{n};\ell} - P_{\mathrm{s};\ell}}{\Delta y_\ell} \tag{16}$$

Refinement approach

A semi-structured approach is followed in which a cell (i, j) at refinement level ℓ is *replaced* by a set of $r_i \times r_j$ smaller cells at refinement level $\ell+1$ having indices $(2i+m, 2j+n)$ at offsets $0 \le m < r_i, 0 \le n < r_j$. The semi-structured indexing system is illustrated in figure 3. On block-shaped refinement regions the method is locally structured, hence the computational efficiency of the original array-based solution methods can be exploited as much as possible. Only at the boundaries of the refinement regions where the actual refinement takes place a new treatment is required.

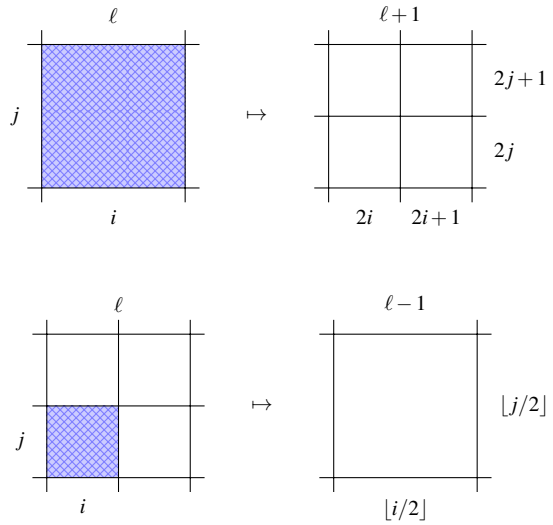

FIGURE 3: Illustration of semi-structured indexing for refinement ratios $r_i = 2, r_j = 2$. *Top*: From coarse to fine indices. *Bottom:* From fine to coarse indices.

For describing the grid layout an auxiliary array is introduced storing only one integer for each potentially occuring cell $(i, j; \ell)$ pointing at the memory location of the subgrid in which it is contained (or null if the cell does not exist). The auxiliary array has a length of $N \times (1 - (r_i r_j)^{\ell_\mathrm{max}})/(1 - r_i r_j)$, where N is the number of cells on the base grid and ℓ_max is the highest refinement level occuring on the grid. Altogether a data structure results that allows for fast and efficient look-up when compared with typical tree-based storage methods, along the lines of [9].

Poisson equation near interfaces

As an example we consider refinement interfaces in the "x=constant"-plane where the refined cells are located to the left of the interface. Five other interface orientations are possible, which are treated similarly. To further simplify discussion we assume a base grid with uniform grid spacings Δx_0 and Δy_0.

In the current discussion we take refinement ratios $r_i = 1$ and $r_j = 2$. In words, no refinement is applied perpendicular to the refinement interface and local refinement is only applied along the refinement interface. For the grid spacings this implies $\Delta y_{\ell+1} = \Delta y_\ell / 2$ and $\Delta x_{\ell+1} = \Delta x_\ell$. Since $r_i = 1$, for the latter, the level subscript will be omitted.

Extending the discretization to the three-dimensional case, non-uniform grids and other refinement directions or refinement

ratios is straightforward.

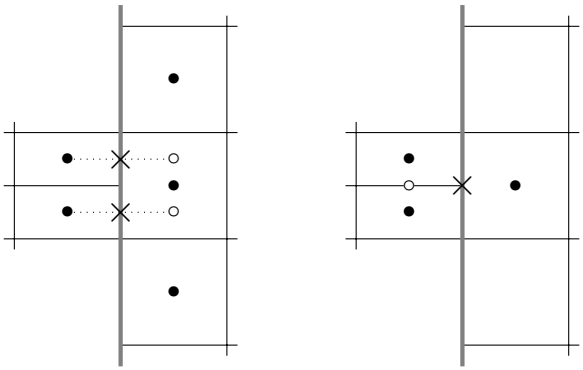

FIGURE 4: *Left*: refinement approach with interpolation of missing pressure variables, *Right*: refinement approach (as followed here) with shifted pressure gradient which is equal for both refined cell faces.

Typically, a large stencil is used for the approximation of missing pressure or velocity variables along the refinement interface. Interpolation of missing variables increases the number of non-zero coefficients in the pressure Poisson matrix, which might result in a non-symmetric matrix, putting higher demands on the solver. Most authors use a non-overlapping interface and apply linear (or even higher-order) interpolation for missing variables on the other side of the interface [10]. Another approach is to apply linear interpolation inside an overlapping interface [11]. In all cases the discretization results in a non-symmetric system of equations.

In the present approach, a compact discretization scheme is designed (in both space and time), which results in a small and symmetric scheme for the discrete composition of M and G. This makes it possible to employ an efficient linear solver. Furthermore, this facilitates the use of adjacent refinement regions as well as the interface discretization near objects and free-surface boundaries.

Discretization in space There are two ways of obtaining a first-order accurate discretization of the pressure gradient. Either by using a linear interpolation for the missing pressure variable outside the refinement region (see left of figure 4) or by slightly shifting the location of the pressure gradient (see right of figure 4). Both approaches result in a first-order accurate discretization scheme introducing an error term that is proportional to respectively $\Delta\frac{\partial^2 p}{\partial y^2}$ and $\Delta\frac{\partial^2 p}{\partial y \partial x}$, where for brevity *we use* Δ *to denote the order of magnitude, omitting any subscripts or products of grid spacings.* However, the first approach results in a relatively large stencil whereas the second approach uses a smaller interpolation stencil consisting of pressure variables that already form part of the regular stencil.

For this reason, the second approach is followed, which can be described as "using a constant pressure gradient along a refined cell face" (see e.g. [12, 13]). Correspondingly, we use a uniform velocity across the entire refined cell face and only place coarse computational velocity variables at the interface.

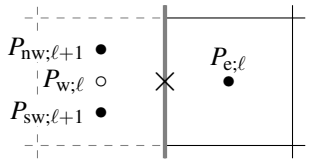

FIGURE 5: Missing variable ($\circ$) for the gradient operator applied at the location indicated with $\times$, together with the variables used for linear interpolation ($\bullet$).

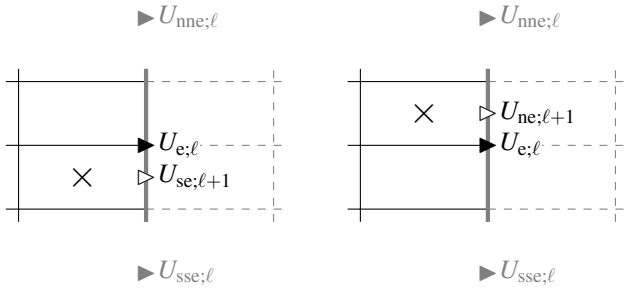

FIGURE 6: Missing variables ($\triangleright$) for the divergence operator applied at the location indicated with $\times$, together with the variable used for constant extrapolation ($\blacktriangleright$) and linear correction ($\blacktriangleright$).

For the missing coarse pressure variable $P_{\mathrm{w};\ell}$ (see figure 5) a simple average of the neighbouring fine pressure values is used. This results in the following discretization of the pressure derivative at the refined cell face

$$\frac{2P_{\mathrm{e};\ell} - P_{\mathrm{sw};\ell+1} - P_{\mathrm{nw};\ell+1}}{2\Delta x} = \frac{\partial p}{\partial x}(\mathbf{x}^u_\ell) + O(\Delta).$$

Note that the approximation for the missing pressure variable is second-order accurate, but one order of accuracy is lost due to the loss of symmetry in the central scheme. Hence the approximation of the pressure gradient across the refinement interface is first-order accurate.

For the discretization of the divergence at the fine side of refinement interfaces, an approximation is needed for the missing

fine velocities. As a first approach, a discretization is obtained by means of the adjointness condition (15), so the divergence operator is defined as the negative transpose of the gradient operator. This implies that missing velocities are simply approximated using constant extrapolation (see figure 6)

$$U_{\mathrm{se};\ell+1} := U_{\mathrm{e};\ell} = u(\mathbf{x}^U_{\mathrm{se};\ell+1}) + O(\Delta),$$
$$U_{\mathrm{ne};\ell+1} := U_{\mathrm{e};\ell} = u(\mathbf{x}^U_{\mathrm{ne};\ell+1}) + O(\Delta),$$

The corresponding divergence operator for interface cells is then given by

$$\left(\overline{M}\mathbf{u}_h\right)_{\mathrm{s};\ell+1} = \Delta y_{\ell+1}\left(U_{\mathrm{e};\ell} - U_{\mathrm{sw};\ell+1}\right) + \Delta x\left(V_{\mathrm{c};\ell+1} - V_{\mathrm{s};\ell+1}\right)$$
$$\left(\overline{M}\mathbf{u}_h\right)_{\mathrm{n};\ell+1} = \Delta y_{\ell+1}\left(U_{\mathrm{e};\ell} - U_{\mathrm{nw};\ell+1}\right) + \Delta x\left(V_{\mathrm{n};\ell+1} - V_{\mathrm{c};\ell+1}\right)$$

Taking a uniform velocity along the refined cell face is conform the earlier remark of using a uniform pressure gradient along the refined cell face. However, it can be seen that the scheme for the divergence operator is not consistent yet because the velocities in the central difference are not well aligned.

$$\begin{aligned}\frac{1}{\Delta x\Delta y_{\ell+1}}\left(\overline{M}\mathbf{u}_h\right)_{\mathrm{s};\ell+1} &= \frac{\partial u}{\partial x}(\mathbf{x}^P_{\mathrm{s};\ell+1}) + \frac{\partial v}{\partial y}(\mathbf{x}^p_{\mathrm{s};\ell+1}) \\ &\quad + \frac{1}{2}\frac{\Delta y_{\ell+1}}{\Delta x}\frac{\partial u}{\partial y}(\mathbf{x}^u_{\mathrm{e};\ell}) + O(\Delta) \\ \frac{1}{\Delta x\Delta y_{\ell+1}}\left(\overline{M}\mathbf{u}_h\right)_{\mathrm{n};\ell+1} &= \frac{\partial u}{\partial x}(\mathbf{x}^P_{\mathrm{n};\ell+1}) + \frac{\partial v}{\partial y}(\mathbf{x}^p_{\mathrm{n};\ell+1}) \\ &\quad - \frac{1}{2}\frac{\Delta y_{\ell+1}}{\Delta x}\frac{\partial u}{\partial y}(\mathbf{x}^u_{\mathrm{e};\ell}) + O(\Delta)\end{aligned}$$

The inconsistency can be resolved by adding corrections for the observed error terms. In order to satisfy mass conservation it is important that these corrections sum up to zero for each refined cell face. For this we can make use of the symmetry observed in the above error terms and correct the operator $\overline{M}$ with the following linear correction terms:

$$\frac{1}{\Delta x\Delta y_{\ell+1}}\left(M^+\mathbf{u}_h\right)_{\mathrm{s};\ell+1} = -\frac{1}{2}\frac{\Delta y_{\ell+1}}{\Delta x}\left[\delta_y\mathbf{u}_h\right]_{\mathrm{e};\ell} \qquad (18)$$
$$\frac{1}{\Delta x\Delta y_{\ell+1}}\left(M^+\mathbf{u}_h\right)_{\mathrm{n};\ell+1} = \frac{1}{2}\frac{\Delta y_{\ell+1}}{\Delta x}\left[\delta_y\mathbf{u}_h\right]_{\mathrm{e};\ell} \qquad (19)$$

where δ_y is a central differencing operator which is applied along the refinement interface (see figure 6)

$$\left[\delta_y\mathbf{u}_h\right]_{\mathrm{e};\ell} = \frac{U_{\mathrm{nne};\ell} - U_{\mathrm{sse};\ell}}{2\Delta y_\ell} \qquad (20)$$

The divergence operator with correction, i.e. $\overline{M} + M^+$ is now first-order accurate.

To conclude we remark that the correction operator M^+ is similar for other interface orientations. Note that in the three-dimensional case the operator would also include a difference term in the secondary direction tangential to the interface.

Implicit and explicit treatment In line with the discretization in regular parts of the grid it seems most natural to apply the linear interpolation at refinement interfaces *implicitly*, i.e.

$$M\mathbf{u}_h^{n+1} = (\overline{M} + M^+)\mathbf{u}_h^{n+1} = \mathbf{0}, \qquad (21)$$

Writing out the corresponding Poisson equation we get

$$\Delta t\,(\overline{M} + M^+)\left(\mathbf{u}_h^* - \Omega^{-1}(\overline{G} + G^+)\mathbf{p}_h^{n+1}\right) = \mathbf{0} \qquad (22)$$

Note that if the discrete adjointness condition is to be satisfied, the linear correction M^+ needs to be reflected in an additional term $G^+ = -(M^+)^*$ in the pressure gradient operator. Altogether the composition of the operators $\overline{M} + M^+$ and $\overline{G} + G^+$ results in a large implicit discretization stencil for the Poisson equation so it is interesting to search for simplifications. Two considerations can help to reduce the Poisson stencil.

First of all, because the basic operator $\overline{G}$ is already first-order accurate it is possible to remove the contribution G^+. This comes at the price of not satisfying the adjointness condition at refinement interfaces, however problems are also avoided near corners of refinement regions.

This leaves us with the interpolation term $M^+\mathbf{u}_h^{n+1}$ which can be split in an *explicit* part $M^+\mathbf{u}_h^*$ and an *implicit* part $M^+\overline{G}\mathbf{p}_h^{n+1}$. For the interface orientation discussed above the implicit part of the term is proportional to

$$\Delta t\;\Delta y\;\delta_y\,\overline{G}\mathbf{p}_h$$

which by means of Taylor series analysis can be shown to be of the order

$$\Delta t\;\Delta y\;\frac{\partial^2 p}{\partial y\partial x}$$

This implies that by allowing for a first-order error, the linear correction term M^+ at interfaces can be made fully *explicit*

$$(\overline{M} + M^+)\mathbf{u}_h^* - \Delta t\overline{M}\Omega^{-1}\overline{G}\mathbf{p}_h^{n+1} = \mathbf{0} \qquad (23)$$

In particular, the implicit and explicit approaches (23) and (22) are *both* first-order accurate.

Convection and diffusion

Missing velocities that are needed in the convection and diffusion scheme are approximated using (bi-)linear interpolation. Due to this interpolation, the standard second-order discretization schemes for convection and diffusion reduce to first-order accuracy at refinement interfaces. We note that in smooth regions of the solutions this does not pose any problem. Higher-order interpolation could be used to allow for refinement interfaces in less smooth regions. However, the numerical error would then be dominated by the (locally) first-order behaviour of the divergence and gradient operators, which are hard to make both mass conservative and higher-order accurate (as is also observed in [10]).

NUMERICAL RESULTS

In order to investigate the performance of the local grid refinement scheme, two-dimensional simulations of flow around a square cylinder have been performed. All simulations at Reynolds numbers 10 and 100 have been performed using a second-order central discretization.

In order to get a good view of the efficiency gain that is obtained with the local grid refinement approach, the analysis is best performed from a "coarsening" point of view. Near the object the grid resolution is kept constant while the grid is coarsened towards the boundaries of the domain.

Flow around a square cylinder (Re=10)

At a Reynolds number of 10 the flow readily converges to a steady-state solution. The resulting solution is smooth and is not expected to pose any difficulties for refinement interfaces.

The numerical results presented in table 1 show that the drag-force predictions are accurate even on grids that are very coarse close to the boundaries of the domain. Even on a base grid of 30×30 cells and 3 refinement layers the drag-coefficient prediction is merely 0.1% different from the answer obtained on a uniformly refined grid, while the number of grid points has been reduced by almost a factor of 20. The resolution of the grid close to the object is of main importance, and it is seen that good convergence behaviour is obtained when increasing the number of cells around the cylinder.

Flow around a square cylinder (Re=100)

At a Reynolds number of 100 the flow is unsteady, as the flapping shear layer results in an oscillating drag and lift force on the cylinder. This test case clearly provides a more challenging test for the local grid refinement method.

A local grid refinement ratio of 3×3 is used and the results of the locally refined grids are compared to their uniform counterparts. The results shown in table 2 illustrate that the number of grid points can be reduced significantly while useful predictions for the drag and lift coefficients can still be obtained.

cyl.	grid	$\ell_{\max}$	# ptn	C_d
2×4	80×80	0	6k	3.1374
	40×40	1	3k	3.1598
3×6	120×120	0	14k	3.2590
	60×60	1	6k	3.2589
4×8	160×160	0	26k	3.2918
	80×80	1	19k	3.2917
	40×40	2	4k	3.2908
6×12	240×240	0	58k	3.3138
	60×60	2	9k	3.3133
	30×30	3	3k	3.3099
12×24	480×480	0	230k	3.3297
	60×60	3	12k	3.3290
	30×30	4	5k	3.3243

TABLE 1: Drag-coefficient predictions (C_d) for flow around a square cylinder (Re=10) on uniform, and locally refined grids. The column 'cyl.' displays the grid resolution at the boundary of the cylinder, which is constant for level of refinement. The column 'grid' shows the resolution of the base grid to which the local grid refinement is applied. In all cases a refinement ratio of 2×2 is used and $\ell_{\max}$ indicates the number of local refinement regions. In the column '# pnt' the total number of grid points is displayed.

Flow around a square cylinder (Re=22,000)

In order to assess the performance of the turbulence models the flow around a square cylinder at a Reynolds number of $22,000$ is simulated. This classical test case has been the subject of investigation in [14].

The computational domain is taken as suggested in [14]: $[-5.0, 15.0] \times [-7.0, 7.0] \times [0.0, 4.0]$ in streamwise, cross-stream, and spanwise directions, respectively. A square cylinder with diameter $D = 1.0$ is centered at the origin. On the lateral walls free-slip boundary conditions have been imposed, while periodic boundary conditions are applied in the spanwise direction.

The presented results have been obtained on a strongly stretched, coarse grid $[68 \times 64 \times 20]$ with a first off-cylinder-wall grid-spacing of 0.0371. The Werner-Wengle model is applied to model the wall-shear stress in the presence of the boundary layer at the cylinder boundaries. The results from the simulations are compared with the experimental results from Lyn *et al.* [15]. All the averages that are discussed are taken over the spanwise direction *and* over time (for all simulations over at least 13 shedding cycles).

cyl.	grid	ℓ_{max}	#ptn	St	C_d	$C_{l,rms}$
6×12	240×240	0	58k	0.152	1.6275	0.2567
	80×80	1	19k	0.153	1.6273	0.2554
9×18	360×360	0	129k	0.150	1.5687	0.2231
	40×40	2	12k	0.150	1.5687	0.2208
18×36	720×720	0	0.5M	0.150	1.5234	0.2009
	80×80	2	48k	0.150	1.5232	0.1998

TABLE 2: Numerical predictions of the Strouhal number (St), drag force C_d, and the root-mean-square of the lift coefficient $C_{l,rms}$ for flow around a square cylinder (Re=100) on uniform and locally refined grids. In all cases a refinement ratio of 3×3 is used. Again, the 'cyl.' denotes the grid resolution around the cylinder, 'grid' the resolution of the base grid, ℓ_{max} the number of refinements, and '# pnt.' the total number of grid points.

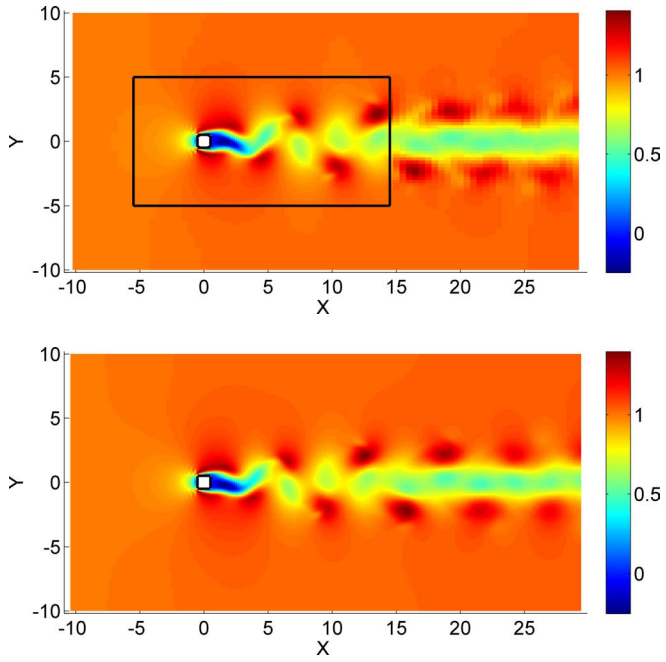

FIGURE 7: Snapshots of the horizontal velocity for flow around a square cylinder (Re=100) simulated on a locally refined grid and corresponding uniform grid consisting of 240×240 points.

The integral quantities of the flow (table 3) show that the predicted Strouhal number is within the experimental range for all of the models under consideration. The drag coefficient C_d is well-predicted by the QR model, while it is slightly overpredicted by both the regularization and the blended model.

model	St	C_d
QR	0.130	2.09
blended	0.134	2.25
$\mathscr{C}_2$	0.135	2.20
exp.	0.133 ± 0.003	2.10

TABLE 3: Strouhal number (St) and drag coefficients C_d for flow around a square cylinder (Re=22,000) for the simulations using the different models and the reported experimental values.

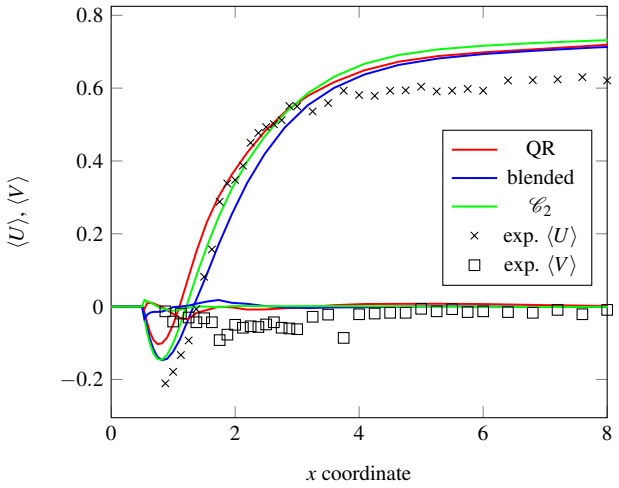

FIGURE 8: Average velocity profiles $< U >$ and $< V >$ along the centerline ($x = 0$). The experimental data is taken from [15].

Centerline profiles The velocity at outflow is overpredicted by all models. This phenomenon has been reported in [15] for other simulations of the same flow. The predicted length of the wake and maximum average backflow of the regularization and blended model correspond best to the reported experimental data. The QR model predicts a wake length and backward flow that is significantly smaller.

The fluctuation profiles in the wake, in figure 9, are all underestimating the experimentally reported fluctuations. This can be expected for the coarse grid that has been used in this simulation. The regularization model approximates the experimentally reported fluctuations best, especially in the region close to the cylinder, where the grid is finest.

Cross-stream profiles In cross-stream direction only small differences between the models are observed in the near-wall region. Note that only 8 grid points have been used to cover the distance between the cylinder wall ($x = 0.5$) and $x = 1.0$. The

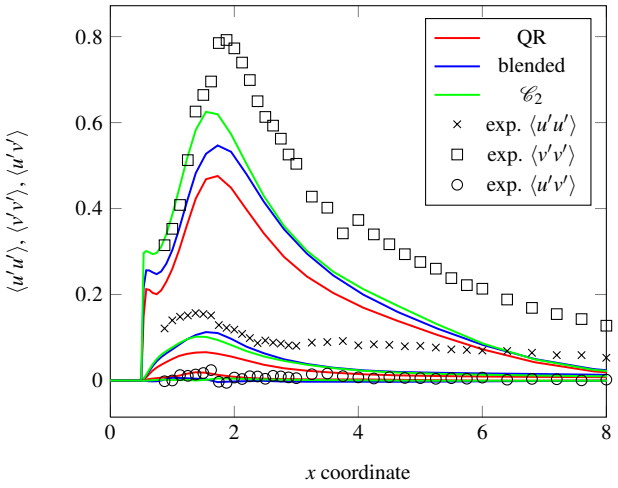

FIGURE 9: Average velocity fluctuations $< u'u' >$, $< v'v' >$, and $< u'v' >$ along the centerline $x = 0$.

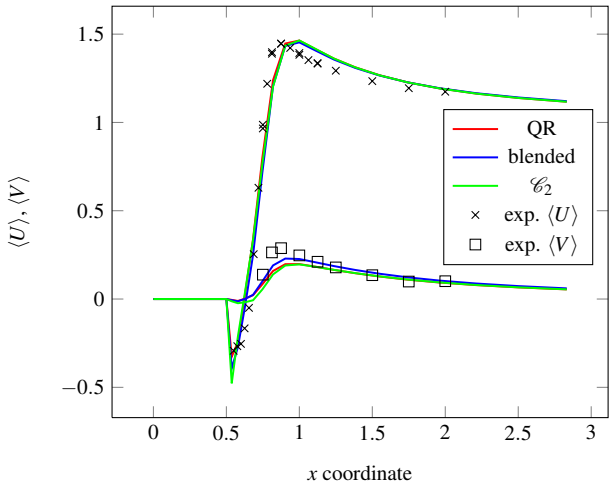

FIGURE 10: Average velocity profiles $< U >$ and $< V >$ along the cross-stream line $y = 0$.

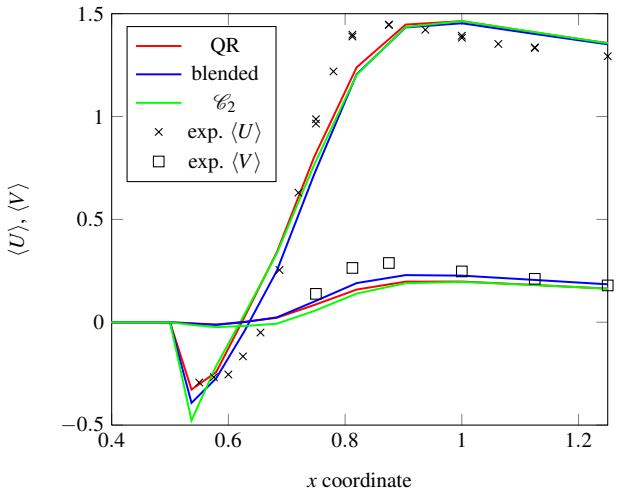

FIGURE 11: Close-up of the near-wall region of the average velocity profiles in cross-stream direction along the line $y = 0$.

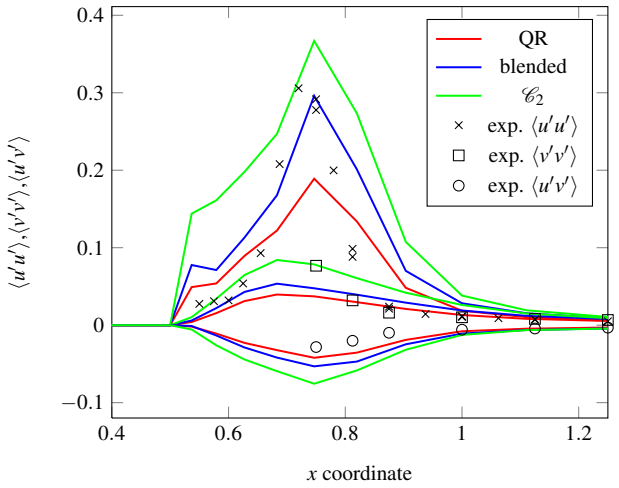

FIGURE 12: Average velocity fluctuations $< u'u' >$, $< v'v' >$, and $< u'v' >$ in cross-stream direction along the line $y = 0$.

experimental fluctuation profiles in the near-side-wall region are best simulated by the blended model. The regularization model tends to overestimate the width as well as the height of the experimental profile, while the QR model tends to underestimate it.

From this test case the picture emerges that the QR model has the tendency to underpredict the fluctuations, which suggests an overdamping of the small scale structures in the flow. The regularization model seems to be capable of capturing the profiles in regions with large gradients, but an overprediction for the $< u'u' >$ and $< v'v' >$ profiles in the near-wall region is observed. The blended model poses a well-performing third option. It does not suffer from the overprediction of the regularization approach, but captures profiles with large gradients significantly better than the QR model.

4 CONCLUSIONS

In this paper, a local grid refinement approach for efficiently simulating turbulent flow is introduced. A compact scheme is presented which reduces the overhead introduced by the refinement interfaces. Although the interface scheme is formally first-order accurate, for typical practical applications the method still provides an effective approach to saving computational time. By restricting refinement interfaces to smooth areas of the flow, as is

common practice, effectively the convergence rate of the original method is maintained. This suggests that also at higher Reynolds numbers it will provide a useful tool for reducing the number of computational grid points without compromising the accuracy.

The presented turbulence models give promising results on a coarse computational grid for a test case involving a separated flow. This gives us hope that also for higher Reynolds numbers, accurate flow predictions can be obtained on relatively coarse grids. Combining turbulence modeling with local grid refinement, will reduce the computational time even further. Therefore, one of the main research outlooks within the ComFLOW project is to combine these approaches and apply them to practical offshore applications, such as the flow around a semisubmersible.

REFERENCES

[1] Kleefsman, K., Fekken, G., Veldman, A., Iwanowski, B., and Buchner, B., 2005. "A volume-of-fluid based simulation method for wave impact problems". *Journal of Computational Physics,* ***206***(1), pp. 363–393.

[2] Veldman, A., Luppes, R., Bunnik, T., Huijsmans, R., Duz, B., Iwanowski, B., Wemmenhove, R., Borsboom, M., Wellens, P., van der Heiden, H., and van der Plas, P., 2011 (paper OMAE2011-49488). "Extreme wave impact on offshore platforms and coastal constructions". In 30th Conf. on Ocean, Offshore and Arctic Eng. OMAE2011.

[3] Verstappen, R., 2011. "When does eddy viscosity damp subfilter scales sufficiently?". *Journal of Scientific Computing,* ***49***(1), pp. 94–110.

[4] Sagaut, P., 2001. *Large Eddy Simulation for Incompressible Flows*. Springer–Verlag.

[5] Verstappen, R., and Veldman, A., 2003. "Symmetry-preserving discretization of turbulent flow". *Computers & Fluids,* ***187***, pp. 343–368.

[6] Verstappen, R., 2008. "On restraining the production of small scales of motion in a turbulent channel flow". *Computers & Fluids,* ***37***(7), pp. 887–897.

[7] Trias, F., and Verstappen, R., 2011. "On the construction of discrete filters for symmetry-preserving regularization models". *Computers & Fluids,* ***40***(1), pp. 139 – 148.

[8] Werner, H., and Wengle, H., 1991. "Large-eddy simulation of turbulent flow over and around a cube in a plate channel.". In 8th Symposium on Turbulent Shear Flows, Springer–Verlag, pp. 155–168.

[9] de Tullio, M., De Palma, P., Iaccarino, G., Pascazio, G., and Napolitano, M., 2007. "An immersed boundary method for compressible flows using local grid refinement". *Journal of Computational Physics,* ***225***(2), pp. 2098–2117.

[10] Minion, M., 1996. "A projection method for locally refined grids". *Journal of Computational Physics,* ***127***(1), pp. 158–178.

[11] Liu, Q., 2011. "A stable and accurate projection method on a locally refined staggered mesh". *International Journal for Numerical Methods in Fluids,* ***67***(1), pp. 74–92.

[12] Uzgoren, E., Singh, R., Sim, J., and Shyy, W., 2007. "Computational modeling for multiphase flows with spacecraft application". *Progress in Aerospace Sciences,* ***43***(4), pp. 138–192.

[13] Losasso, F., Gibou, F., and Fedkiw, R., 2004. "Simulating water and smoke with an octree data structure". In ACM Transactions on Graphics (TOG), Vol. 23, ACM, pp. 457–462.

[14] Rodi, W., Ferziger, J., Breuer, M., and Pourquiée, M., 1997. "Status of large eddy simulation: Results of a workshop". *Journal of Fluids Engineering,* ***119***(2), pp. 248–262.

[15] Lyn, D., Einav, S., Rodi, W., and Park, J., 1995. "A laser-doppler velocimetry study of ensemble-averaged characteristics of the turbulent near wake of a square cylinder". *Journal of Fluid Mechanics,* ***304***(12), pp. 285–319.