A Dynamic LES Model for Turbulent Reactive Flow with Parallel Adaptive Finite Elements

Jiajia Waters, David B. Carrington, Xiuling Wang and Darrell W. Pepper

Abstract An adaptive finite element method (FEM) is used for the solution of turbulent reactive flows in 3-D utilizing parallel methods for fluid dynamic and combustion modeling associated with engines. A dynamic LES method permits transition from laminar to turbulent flow without the assumptions usually required for turbulent sublayers near wall area. This capability is ideal for engine configurations where there is no equilibrium in the turbulent wall layers and the flow is not always turbulent and often in transition. The developed adaptive FEM flow solver uses "h" adaptation to provide for grid refinement. The FEM solver has been optimized for parallel processing employing the message passing interface (MPI) for clusters and high-performance computers.

Keywords Dynamic LES · Turbulent reactive flow · Mesh adaptation · Parallel implicit solver · Finite element · FEM · CFD · Computational fluid dynamics

Nomenclature

~	Designates a Favre-averaged variable
–	Designates a grid-filtered variable
c	Sound speed (m/s)
C_p	Specific heat capacity at constant P (J/kg.K)
C_{vm}	Vreman fixed SGS eddy viscosity coefficient
C_{DVMG}	Vreman dynamic SGS eddy viscosity coefficient

J. Waters · D. B. Carrington
Los Alamos National Laboratory, T-3 Division, Los Alamos, USA

X. Wang
Purdue University Northwest, Hammond, IN 46323, USA

D. W. Pepper (✉)
NCACM, Department of Mechanical Engineering, University of Nevada
Las Vegas, Las Vegas, USA
e-mail: darrell.pepper@unlv.edu

A. K. Runchal et al. (eds.), *Energy for Propulsion*, Green Energy and Technology,
https://doi.org/10.1007/978-981-10-7473-8_9

$\mathrm{D_j}$	Diffusion coefficient of the jth species $\left(\mathrm{m^2/s}\right)$
D_k	Turbulent diffusion coefficient $\left(\mathrm{m^2/s}\right)$
E	Total internal energy (J/kg)
$f_{k,j}$	Body forces $\left(\mathrm{N/m^3}\right)$
f_{drop}	Body forces related to particulate or droplets in flow $\left(\mathrm{N/m^3}\right)$
H_j	Enthalpy of species j (J)
H_{oj}	Enthalpy of formation (J)
P	Pressure (Pa)
Pr	Molecular Prandtl number
$\mathrm{Pr_{sgs}}$	SGS eddy Prandtl number
Pr_{DVMG}	Vreman dynamic SGS eddy Prandtl number
Q_j	Subtest-scale heat flux vector
q_i	Heat flux vector
Re	Reynolds number
$\tilde{\mathrm{S}}_{\mathrm{ij}}$	Strain rate tensor $\left(\frac{\mathrm{N}}{\mathrm{m^2}}, \mathrm{kg/m\,s^2}\right)$
Sc	Schmidt number
Sct	Subgrid-scale turbulent Schmidt number
T	Temperature (K)
T_{ij}	Subgrid test-scale stress tensor
t_{ij}	Grid-scale (resolved scale) shear stress $\left(\frac{\mathrm{N}}{\mathrm{m^2}}, \mathrm{kg/m\,s^2}\right)$
u_i	Velocity component (m/s)
$\Upsilon_j f_j$	Body force term for the jth component
$\dot{w}^j_{chem}$	Chemical reaction
$\dot{w}^j_{spray}$	Spray evaporation

Greek Symbols

$\partial \mathrm{t}$	Discrete time step size (s)
κ	Coefficient of thermal conductivity (W/m K)
ρ	Density $(\mathrm{kg/m^3})$
Υ^j	Mass fraction (jth species) $\left(\frac{\rho^j}{\rho}\right)$
τ_{ij}	Subgrid-scale stress tensor
μ	Fluid viscosity (Pa s)
μ_{sgs}	Turbulent eddy viscosity

1 Introduction

A number of challenges found in combustion modeling are related to developing methods that handle the numerical requirements of coupled physical and chemical processes. These requirements related to engines are pronounced, with the highly transient three-dimensional (3-D) dynamics ranging over flow regimes with evaporating fuel sprays, the ignition of gases, and the subsequent chemical reactions. In addition, heat transfer and species transport modeling are critical to the success of the models accuracy. When dealing with multiple phases in the liquid sprays, the process will get even more complicated to track the dynamically evolving interfaces. Our efforts in developing accurate methods and models to calculate such flows in engine cylinders, including the effects of turbulence, improve the current Los Alamos National Laboratory's KIVA [1] suite of codes with the creation of an hp-adaptive FEM all flow regime solve (KIVA-hpFE), where h is associated with mesh adaptation and p is the order of the basis function for FEM. Among the adaptation families, the hp-adaptive FEM is one of the best mesh-based algorithms. The computational mesh is automatically refined and unrefined based on the change of flow features; the shape function order is dynamically controlled by the computational error. In this study, we only focus on h-adaptation (mesh refinement).

The most widely used approach for modeling turbulent flow, Reynolds-averaged Navier–Stokes (RANS), is implemented using the two-equation k-ω modeled by Wilcox [2]. Time averaging methods produce a mean value for turbulent variables, kinetic energy, dissipation rate, and turbulent viscosity and do not capture flow structures in unsteady turbulence. The two-equation models generally use a law-of-the-wall relation for emulating the turbulent boundary layer, but this law is not appropriate at varying times during the cycle of an engine. For example, the flow is nearly always perpendicular to the surface near the piston.

A large eddy simulation (LES) model provides a method for solution of unsteady turbulence intensity, varying turbulent viscosity, and dynamic flow structures. In many modern combustion devices, the LES method is more widely adopted to study combustion [3–7]. A LES model solves the spatially averaged Navier–Stokes equations using a grid-based filtering process. In LES, the larger eddies are directly resolved at the grid resolution and eddies smaller than the grid are modeled. Using a dynamic Vreman approach [8] as implemented in the finite element method (FEM), guarantees vanishing subgrid-scale (SGS) dissipation for various laminar shear flows and thus there is no need to use any wall-damping functions in simulating boundary layer flows. These features of this type of LES system are well suited for wall-bounded shear flows [9]. Hence, the dynamic Vreman LES is capable of modeling various flow regimes, laminar, transitional, and turbulent flows simultaneously. More details of the LES implementation can be found in Waters et al. [10].

In this study, Vreman LES modeling, with the use of adaptive FEM and parallel implicit solution methods, is examined in more detail. LES models, in comparison to the RANS models, require finer density meshes because the modeled flow size depends on the filter size decided by the mesh size. Unstructured CFD algorithms used to model engine combustion typically require large computing resources, typically provided through parallel computer systems. By linking together hundreds and thousands of individual processors (or nodes), parallel computer systems deliver significant enhancements in computational memory, storage, and overall computing speed. In this study, a message passing interface (MPI) is employed to model 3-D turbulent flow in engines. Jimack [11] describes a similar procedure using MPI for

FEM. The parallel system in the KIVA-hpFE code also tracks the motion of spray particle parcels (droplets) from one position to another and from one processor to another [12]. Krylov solvers are used with matrix-vector multiplication and dot product operations for shared element updating and convergence of shared nodes [13].

2 Governing Equations

2.1 Turbulent Flow Modeling with Multi-species

The grid-filtered and Favre-filtered continuity, momentum, energy, and species equations governing the process of large-scale eddies are expressed as

$$\frac{\partial\bar{\rho}}{\partial t}+\frac{\partial(\bar{\rho}\tilde{u}_i)}{\partial x_i} \tag{1}$$

$$\frac{\partial(\bar{\rho}\tilde{u}_i)}{\partial t}+\frac{\partial\left(\bar{\rho}\tilde{u}_i\tilde{u}_j\right)}{\partial x_j}=\frac{\partial\tilde{t}_{ji}}{\partial x_j}-\frac{\partial\bar{p}}{\partial x_i}+\frac{\partial\tau_{ji}}{\partial x_j}+\bar{f}_{drop}+\bar{\rho}\sum_{k=1}^{NumSpecies}\tilde{\Upsilon}_k f_{k,j} \tag{2}$$

where $\tilde{t}_{ij}$ is the stress tensor evaluated using the Stoke's hypothesis as

$$\tilde{t}_{ij}=\mu\left(\frac{\partial\tilde{\mathrm{u}}_{\mathrm{i}}}{\partial\mathrm{x}_{\mathrm{j}}}+\frac{\partial\tilde{\mathrm{u}}_{\mathrm{j}}}{\partial\mathrm{x}_{\mathrm{i}}}\right)-\frac{2}{3}\mu\frac{\partial\tilde{\mathrm{u}}_{\mathrm{k}}}{\partial\mathrm{x}_{\mathrm{k}}}\delta_{ij} \tag{3}$$

$$\begin{aligned}\frac{\partial\tilde{\mathrm{E}}}{\partial\mathrm{t}}&=-\frac{\partial}{\partial x_i}\left(\tilde{\mathrm{E}}\tilde{\mathrm{u}}_{\mathrm{i}}+\mathrm{p}\tilde{\mathrm{u}}_{\mathrm{i}}\right)+\frac{\partial}{\partial x_i}\kappa\frac{\partial\tilde{\mathrm{T}}}{\partial\mathrm{x}_{\mathrm{i}}}-\frac{\partial\left(\mathrm{C}_{\mathrm{p}}\mathrm{q}_{\mathrm{i}}\right)}{\partial\mathrm{x}_{\mathrm{i}}}\\&\quad+\frac{\partial}{\partial\mathrm{x}_{\mathrm{i}}}\left(\mathrm{t}_{\mathrm{ij}}+\tau_{\mathrm{ij}}\right)+\frac{\partial}{\partial\mathrm{x}_{\mathrm{i}}}\left(\bar{\rho}\sum_{\mathrm{j}=1}^{\mathrm{NumSpecies}}\overline{\mathrm{H}}_{\mathrm{K}}(\mathrm{D}_{\mathrm{K}}+\frac{\mu_{sgs}}{\mathrm{Sc}_{\mathrm{t}}})\frac{\partial\tilde{\Upsilon}_{\mathrm{K}}}{\partial\mathrm{x}_{\mathrm{i}}}\right)\\&\quad+\bar{\rho}\sum_{j=1}^{NumSpecies}\tilde{\Upsilon}_j f_j(x_i)\cdot\tilde{u}_i-\sum_{k=1}^{NumSpecies}H_{o,k}w_k\end{aligned} \tag{4}$$

$$\begin{aligned}\frac{\partial\bar{\rho}\tilde{\Upsilon}_j}{\partial t}&=-\frac{\partial}{\partial x_i}\left(\bar{\rho}\tilde{u}_i\tilde{\Upsilon}_j\right)+\frac{\partial}{\partial x_i}\bar{\rho}\left[\left(D_{j,N}+\frac{\mu_{sgs}}{Sc_t}\right)\frac{\partial\tilde{\Upsilon}_j}{x_i}\right]+\bar{\rho}\tilde{\Upsilon}_j f_j(x_i)\\&\quad+\dot{w}_{chem}^{j}+\dot{w}_{spray}^{j}\end{aligned} \tag{5}$$

The SGS stress tensor τ_{ij} and SGS heat flux vector q_i in Eqs. (2) and (4) are defined, respectively, as

$$\tau_{ij}-\frac{1}{3}\tau_{kk}\delta_{ij}=-2\mu_{sgs}\left(\tilde{S}_{ij}-\frac{1}{3}\tilde{S}_{kk}\delta_{ij}\right) \tag{6}$$

$$q_j = -\frac{\mu_{sgs}}{Pr_{sgs}}\frac{\partial \tilde{T}}{\partial x_j} \tag{7}$$

where μ_{sgs} is the SGS viscosity, Pr_{sgs} is the SGS Prandtl number, and $\tilde{S}_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right)$ is the strain rate tensor. Here ~ is a Favre-filtered variable obtained from filtering its grid-filtered component. In this work, the box or top hat filter is applied for the grid-filtered component.

2.2 Mass Conserving Projection Method for Compressible Flow

In order to create a fractional split method, an initial guess for specific momentum is advanced in time, utilizing the expression

$$\Delta U_i^* = -\Delta t \times M_u^{-1}[A_u U_i + K_{\tau u} U_i - F_u]^n \tag{8}$$

where $U_i = \rho u_i$, M_u is the mass matrix, A_u is the advection matrix, $K_{\tau u}$ is the diffusion matrix, and F_u is the source term. The projection method is presented here with solving momentum explicitly, but the same algorithm is used when our governing equations are solved implicitly.

$$\Delta U_i^* = U_i^* - U_i^n \tag{9}$$

The corrected momentum is determined from the estimated momentum and the pressure gradient, given by

$$U^{n+1} - U^* = -\Delta t \frac{\partial P'}{\partial x_i} \tag{10}$$

Changes of density or pressure are determined from solving an implicit pressure/density Poisson equation created as a result of conservation of mass. This leads to the value for P' as shown in the following continuity solution process:

Mass conservation:

$$\frac{\partial \rho}{\partial t} = -\frac{\partial(\rho u_i)}{\partial x_i} = -\frac{\partial U_i}{\partial x_i} \tag{11}$$

Time advancement in discrete terms for continuity is

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} = -\frac{\partial U_i'}{\partial x_i} \tag{12}$$

where

$$U' = \theta_1 U^{n+1} + (1 - \theta_1) U^n \tag{13}$$

with $U = \rho u$.

We define $P' = \theta_2 P^{n+1} + (1 - \theta_2) * P^n$ and $\Delta P = P^{n+1} - P^n$. The final specific momentum as obtained with the explicit corrector defined previously by

$$\Delta U_i = U^{n+1} - U^n = \Delta U^* - \Delta t \frac{\partial P'}{\partial x_i} = \Delta U^* - \Delta t \left(\theta_2 \frac{\partial \Delta P}{\partial x_i} + \frac{\partial P^n}{\partial x_i} \right) \tag{14}$$

The finite element form in matrix form is given by

$$\{\Delta \mathbf{U}_i\} = \left\{\Delta \mathbf{U}^*\right\} - \Delta t \left[\mathbf{M}_u^{-1}\right] \left(\theta_2 [\mathbf{G}] \{\Delta p_i\} [\mathbf{G}] \left\{p_i^n\right\}\right) \tag{15}$$

The final specific momentum is obtained using the corrector,

$$\left\{\mathbf{U}_i^{n+1}\right\} = \{\Delta \mathbf{U}_i\} + \left\{\mathbf{U}_i^n\right\} \tag{16}$$

The mass velocity or momentum is solved and velocity is extracted

$$u^{n+1} = U^{n+1} / \rho^{n+1} \tag{17}$$

Density is recovered from the equation of state. The speed of sound, c, is calculated by

$$c = \sqrt{\gamma R T}. \tag{18}$$

where R is the specific gas constant, for air 287 J/kg K, or it can be determined by a mass-averaging process.

When it is incompressible flow, we use artificial compressibility β for c.

$$\beta = \max\left(\in, u_{conv}, u_{diff}\right) \tag{19}$$

where $\in$ is a small constant to ensure β in Eq. (19) is not approaching zero. $u_{conv} = |U| = \sqrt{u_i u_i}$ and $u_{diff} = \frac{\nu}{h}$, where h is the element size and ν is the kinematic viscosity. The time step for the artificial compressibility method may become limited as $\beta \to \infty$. Therefore, pseudo time-stepping is needed in the incompressible region.

2.3 *Dynamic Vreman SGS LES Model*

Vreman [9] developed a fixed model coefficient SGS model where the SGS viscosity is determined as

$$\mu_{sgs} = \bar{\rho} C_{vm} \Pi^g \tag{20}$$

in which

$$\Pi^g = \sqrt{\frac{B_\beta^g}{\alpha_{ij}\alpha_{ij}}} \tag{21}$$

$$\alpha_{ij} = \frac{\partial \tilde{u}_j}{\partial x_i} \tag{22}$$

$$\beta_{ij} = \sum_{m=1}^{3} \Delta_m^2 \alpha_{mi}\alpha_{mj} \tag{23}$$

$$B_\beta^g = \beta_{11}\beta_{22} - \beta_{12}^2 + \beta_{11}\beta_{33} - \beta_{13}^2 + \beta_{22}\beta_{33} - \beta_{23}^2 \tag{24}$$

where $C_{vm} = 0.07$ and the SGS Prandtl number $\mathrm{Pr}_{\mathrm{sgs}} = 0.4$.

If $\alpha_{ij}\alpha_{ij}$ or B_β^g is approaching zero, then $\Pi^g = 0$. This is equivalent to running laminar flow.

It is not realistic to assume the coefficient to be constant throughout the simulation. The development of the dynamic subgrid-scale model (DSGS) model captures significant progress in the subgrid-scale modeling of non-equilibrium flows. The DSGS model coefficient is calculated from the energy of the smallest resolved scale (grid size), instead of setting a priori parameters. The DSGS is obtained by two-filter processes: in the first filter, we used the grid size Δ, where the filtered expressions are given by (1)–(4). In this portion, the SGS Reynolds stress is included. By adding a test filter $\hat{\Delta} = 2\Delta$ to the filtering Eqs. (1)–(4) leads to the subtest-scale stress tensor T_{ij} and subtest-scale heat flux vector Q_{j}:

$$T_{ij} - \frac{1}{3}T_{kk}\delta_{ij} = -2\mu_{sgs}\left(\hat{\tilde{S}}_{ij} - \frac{1}{3}\hat{\tilde{S}}_{kk}\delta_{ij}\right) \tag{25}$$

and

$$Q_j = -\frac{\mu_{sgs}}{Pr_{sgs}}\frac{\partial \hat{\tilde{T}}}{\partial x_j}, \tag{26}$$

We define $\mu_{sgs} = \bar{\rho} C_{DVMG}\Pi^t$ and $Pr_{sgs} = Pr_{DVMG}$. Under the aid of the Germano identity [14] and a least-squares error minimization technique of Lilly [15], the coefficients C_{DVMG} and Pr_{DVMG} are calculated as

$$C_{DVMG} = \frac{\langle L_{ij}M_{ij}\rangle_V}{\langle M_{ij}M_{ij}\rangle_V} \tag{27}$$

and

$$Pr_{DVMG} = \frac{\left\langle M_j^\theta M_j^\theta \right\rangle_V}{\left\langle L_j^\theta M_j^\theta \right\rangle_V}, \tag{28}$$

More details regarding the filtering system of this dynamic LES method is provided in Water et al. [10].

3 Adaption Methodology

The computational mesh is automatically refined and unrefined based on the change of flow features. The smooth flow region usually associates with small computational error, and the fast-changing flow region usually associated with large computational error, as discussed in Waters et al. [10]. Mesh adaptation starts with an initial coarse mesh. A local element refinement indicator is defined to determine if a local refinement for an element is needed, i.e.,

$$\xi_i = \frac{\|e\|_i}{\bar{e}_{avg}} \tag{29}$$

when $\xi_i > 1$ the element is refined; when $\xi_i < 1$ the element is unrefined.

The local relative percentage error of any single element is $\|e\|_i$ while the average element error is defined as $\bar{e}_{avg}$. The gradient of the momentum in each element is used to calculate the error estimate, e.g., $e = \nabla U$. There are different levels of refinement. When an element is refined once, this is labeled a level one refinement, refining again produces a level two for refinement, and so on. The adaptation process is illustrated in Fig. 1. All physical nodes are vertexes of the quadrilaterals. Virtual nodes are only added physically for visualization. More details can be found in Wang and Pepper [16].

4 Implicit Solution Method

Developing an implicit solution scheme allows for a larger time step size and maintaining the accuracy of the overall system of model equations. In this system, the diffusive terms such as turbulence and other stresses, are moved to the left-hand side of the equation and integrated into a matrix equation. The advection and source terms are for the load vector. The method is developed by starting with the momentum Eq. (2) without the pressure gradient term to get the intermediate velocity u_i^*. For simplicity, we drop all of the superscripts and source terms.

$$\rho^n u_i^* - \Delta t \times \left(\frac{\partial t_{ij}^*}{\partial x_j} - \frac{\partial \tau_{ij}^*}{\partial x_j} \right) = \rho^n u_i^n - \Delta t \times \frac{\partial (\rho^n u_i^n u_j^n)}{\partial x_j} \tag{30}$$

where $t_{ij}^* = \mu \left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k^*}{\partial x_k} \delta_{ij}$ and $\tau_{ij} - \frac{1}{3} \tau_{kk} \delta_{ij} = -2\mu_{sgs} \left(\tilde{S}_{ij} - \frac{1}{3} \tilde{S}_{kk} \delta_{ij} \right)$, where the strain rate tensor is given by $S_{ij}^* = \frac{1}{2} \left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i} \right)$.

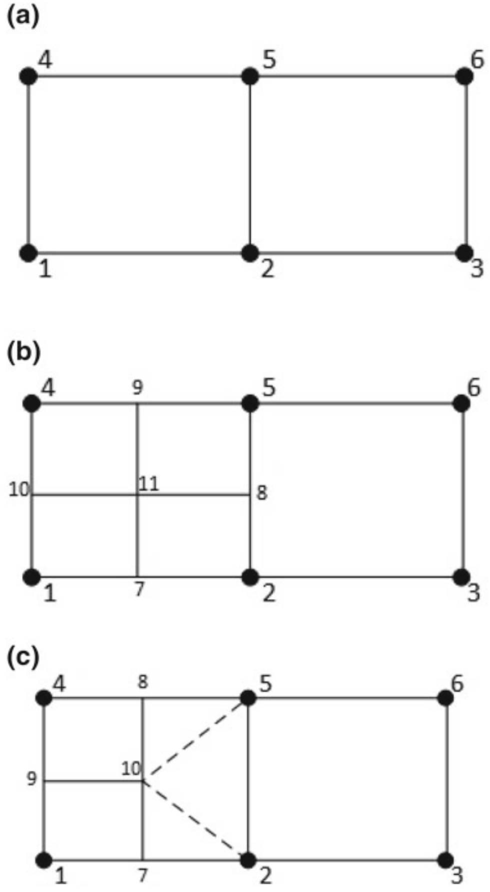


Fig. 1 Adaptation process

Using the mass conserving projection method described previously results in u_i^{n+1} by multiplying u_i^* by ρ^n to form U_i^*, hence $\Delta U_i^* = U_i^* - U_i^n$, as usual in the semi-implicit projection.

After determining the pressure as stated earlier, the specific internal energy is solved again in implicit form:

$$E^{n+1} - \Delta t \times \frac{\partial}{\partial x_i}\left(\frac{\kappa}{C_v}\frac{\partial E^{n+1}}{\partial x_i}\right) - \Delta t \times \frac{\partial}{\partial x_i}\left(\frac{C_p \mu_{sgs}}{\mathrm{Pr}_{sgs}\, C_v}\frac{\partial E^{n+1}}{\partial x_i}\right)$$
$$= E^n - \Delta t \times \frac{\partial}{\partial x_i}\left(E^n u_i^{n+1} + P^{n+1} u_i^{n+1}\right) + \Delta t \times \frac{\partial}{\partial x_i}\left(t_{ij}^{n+1} + \tau_{ij}^{n+1}\right) \qquad (31)$$

Here $E^{n+1} = \rho^n e^{n+1}$ and e^{n+1} is the internal energy, and we can get temperature T^{n+1} with $T^{n+1} = \frac{e^{n+1}}{C_v}$.

The solution to the species transport equations follows in similar manner:

$$\rho^n \Upsilon_j^{n+1} - \Delta t \times \frac{\partial}{\partial x_i} \rho^n \left[\left(D_{j,N} + \frac{\mu_{sgs}}{Sc_t} \right) \frac{\partial \Upsilon_j^{n+1}}{\partial x_i} \right] = \rho^n \Upsilon_j^n - \Delta t \times \frac{\partial}{\partial x_i} (\rho^n u_i^{n+1} \Upsilon_j^n) \tag{32}$$

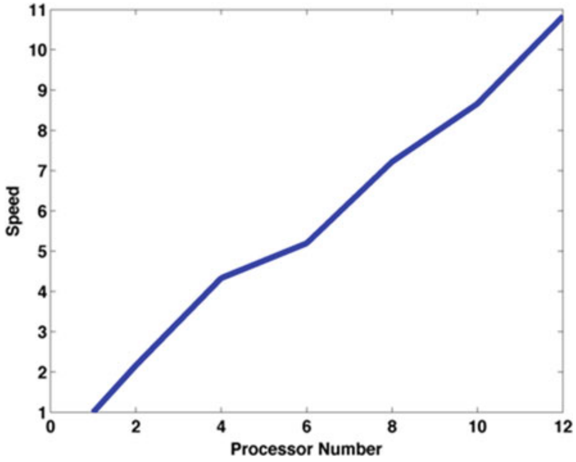
For compressible flow, the sound speed is given by $c = \sqrt{\gamma RT}$, where γ and R are aggregated properties determined from the different species mass concentrations. The aggregation process is described in Carrington [17]. The implicit system is not fully implicit because we want to keep the matrix as symmetric as possible for large-scale parallel calculation to reduce computational time, especially for preconditioning.

5 Parallel Solver System

Choice of either a semi-explicit or an implicit solver system can be selected, both of which use the message passing interface (MPI) paradigm for parallel communication; truly a requirement for 3-D simulations and LES. The linear equation system uses the preconditioned conjugate gradient (PCG) solver package having various Krylov solvers. The user supplies matrix-vector multiplication and dot product operations. Our in situ preconditioning methods use SSOR or SOR, since no off-block information is required during the distributed solution of the matrix equations that might require a block Jacobi scheme. The global system is solved by a Beam-Warming method, checking for global convergence, and evoking the additive Schwartz preconditioning system. Communication for preconditioning requires the matrix-vector multiple and vector dot product; both requiring collocations, that is, a gathering and reduction processes facilitated by MPI. An array utilizing an array of globally shared nodes stored on mother processor. The Beam-Warming iteration process provides for the global convergence of the equation system as developed and described by Carrington [18]. For the semi-explicit process, the PCG linear equation solver is used for the pressure; for the implicit method, the linear equation solver is used for all the equations. Subcylcing of the species transport equations is recommended since there are many replications for each species and can be easily threaded onto processing units by nesting in OpenMP calls, usually this only requires one explicit step, however. Details regarding this MPI implementation for FEM and its efficiency are given in Waters and Carrington [19], showing nearly 30 times speed-up (strong scaling test) over the serial implementation of the overall CFD solver, KIVA-hpFE and is shown to be super-linear.

Essentially, the super-linear behavior of the system results from the local integration of each element and the additive property of integrands. A section of the mesh used for the FEM discretization is shown in Fig. 3. Only those elements dissected are considered overlapped. Some information about the material properties and primitive variables required for integration on an overlapping element must be gathered. This

Fig. 2 Speed up improvement

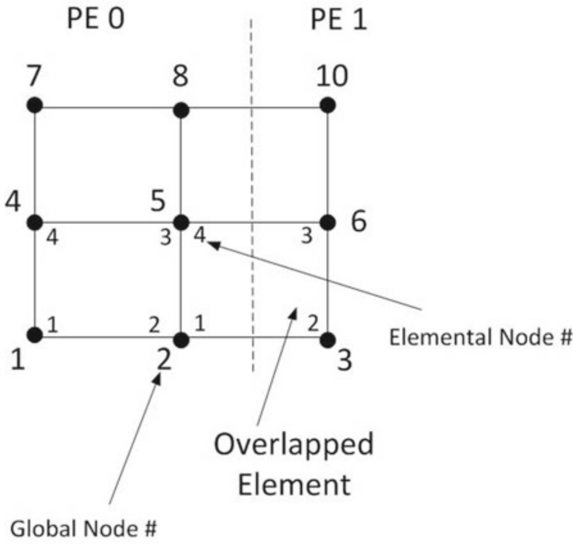


is a simple message passing gather, as performed by a Message Passing Interface (MPI). Jimack and Touheed [20] give a brief introduction about using MPI for FEM. Only the material and primitive variables changing in time need to be part of the communication. Values at each node are determined by integration over elements, on which this node resides. The integration over an overlapped element, shown in Fig. 2, requires gathering values whenever a node is off processor. Therefore, mostly all information that is required for each processor is as follows:

1. Which elements are overlapped, and placed into a 1-D array of shared element s having dimension of number of shared elements.
2. How many nodes should be received from each processors—this is a 1-D array with the dimension of number of processors.
3. How many nodes should be sent to the other processors—this is 1-D array with the dimension of number of processors.
4. Which nodes should be sent to which processor—this is an 2-D array of shared nodes between domains in the global domain, called global_sharednode.
5. What ordering of nodes and elements is present for the global numbering, a mapping from local to global numbering and is called NEWORDER and NEWORDER_E.

In the explicit case, the solution is merely updated at new time value at n + 1 and no matrix equation needs to be solved except for the pressure Poisson equation (as described in the next paragraph on implicit procedures). Elements that are needed to be integrated are elements residing on this processor and the shared elements found in the shared element array. We only update nodes on each processor when the value of the array NEWORDER is not zero and is less than the number of nodes on that processor. After all nodes on that processor are integrated or updated that information is passed into the shared nodes array and distributed to their corresponding processors (e.g. in Fig. 3, nodes 2, 5, and 8 will be passed to processor (PE) 1

Fig. 3 Overlapped nodes



and 3, 6 and 10 need to be passed to PE 0). This information is found in the array global_sharednode and the process occurs via MPI using an all gather command for vector arrays (MPI_Allgatherv). The procedure is applicable to the momentum predictor and corrector steps, along with all transport equations, provided both the viscous and convection terms are stated explicitly.

For the implicit solution processes for pressure or any of the implicit matrices produced, the PCG linear equation solver package of Krylov solvers with user-supplied matrix-vector multiplication and dot product operations is incorporated. In addition, the PCG package provides for user-developed equation preconditioning and overall equation convergence. In this case, a Beam-Warming method is used for the solution to the equations using an additive Schwartz preconditioning system. Only the communication for preconditioning requires the matrix-vector multiplication and dot product routines in KIVA-hpFE. These systems require the collocation, i.e., gathering (GATHERV) and reducing (ALLREDUCE) processes from MPI and employing the 1-D array of shared nodes in the global domain called global_sharednode. The Beam-Warming iteration process provides for the global convergence of the equation system.

The MPI code is implemented in Fortran and has been tested for 3-D flow over a cylinder with 50,720 elements and 57,508 nodes. Figure 2 shows the super-linear speed improvement with MPI incorporated into the semi-implicit algorithm.

6 Numerical Results

6.1 Implicit Solver for 3-D Flow Over a Cylinder

Turbulent flow over a circular cylinder for Reynolds numbers 1.2e+05 (determined by the cylinder diameter and inlet velocity) is investigated here using the dynamic Vreman LES with implicit solver. For simplicity, we assume all species consist of air. The dimension of the domain is

$$x \in [-2.75\,\mathrm{m}, 2.75\,\mathrm{m}],\ y \in [-1.375\,\mathrm{m}, 1.375\,\mathrm{m}],\ z \in [-0.275\,\mathrm{m}, 0.275\,\mathrm{m}].$$

Inlet velocity $\mathrm{U} = 9.0$ m/s, $\mathrm{V} = 0.0$ m/s, and $\mathrm{W} = 0.0$ m/s, and the cylinder diameter is 0.2 m. Here we want to show the efficiency of our implicit solver; therefore, we used a mesh with 114 K nodes and 104 K elements. The flow domain is decomposed into eight subdomains which are assigned to eight processing elements. The domain decomposition is done by ParMETIS, which is an MPI-based parallel library that implements a variety of algorithms for partitioning unstructured graphs, meshes, and for computing fill-reducing orderings of sparse matrices [21]. The domain decomposition is shown in Fig. 4a and the simulation mesh set up is as Fig. 4b. Since no reactions are occurring for the problems described here, the source terms in the governing equations are zero. We use free outflow and no slip as boundary conditions on all walls, and expect 3-D effects from the side walls (along Z direction). The wake starts in a symmetric fashion, as seen in Fig. 5a, then the symmetry breaks down: one eddy becomes larger, as shown in Fig. 5b. Secondary eddies form downstream where the boundary layer separates from the cylinder surface. This is the onset of the eddy separation process, as shown in Fig. 3c. Asymmetry continues to develop in higher Re flows, and eddies are shed from the cylinder, producing the familiar flow pattern of periodic vortex shedding. At later times, the vortices in the wake generate large recirculation zones on the leeward side of the cylinder and develop into the oscillatory von Karman vortex street, as shown in Fig. 5d, e. These results agree with Mustto and Bodstein [22]. Figure 5 shows the instantaneous velocity contour plot at meridional slice of $\mathrm{Z} = 0$ for different time where the vortex shedding is well captured. The Reynolds number is close to the critical Reynolds number and under the influence of 3-D side well effect, we experience some turbulence in the downstream after the cylinder and vortex shedding starts at a further distance compared to small Reynolds number, e.g., $\mathrm{Re} = 1000$. Figure 6 shows the turbulent viscosity and local Mach number. In order to compare the pressure coefficient with the experimental data from Merrick and Bitsuamlak [23], we have the same set up as the experiments shown in [23]. The comparison is shown in Fig. 7. The detachment of the boundary layer flow from the cylinder occurs at $\theta = 85°$, which is approximately 5° different from that described in the literature where $\theta = 80°$ [24]. This difference may be a result of the 3-D side-wall effects (Taylor–Gortler vortex) in the narrow domain with no-slip walls. Overall, results match the experimental values for the coefficient of pressure, shown in Fig. 7. The implicit solver allows for a time step

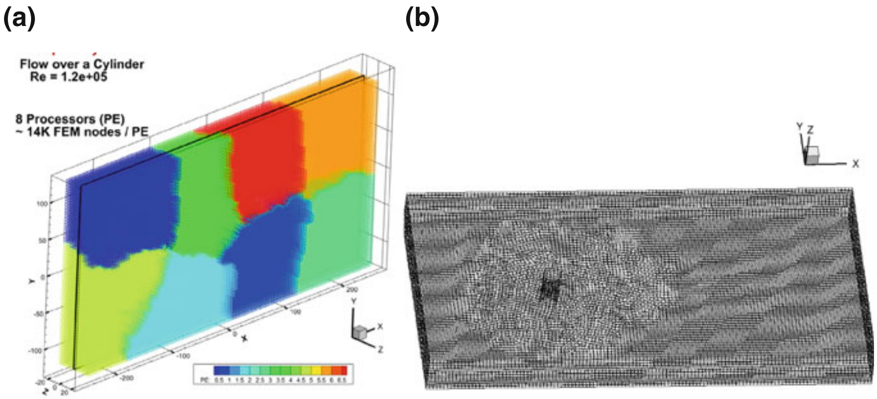
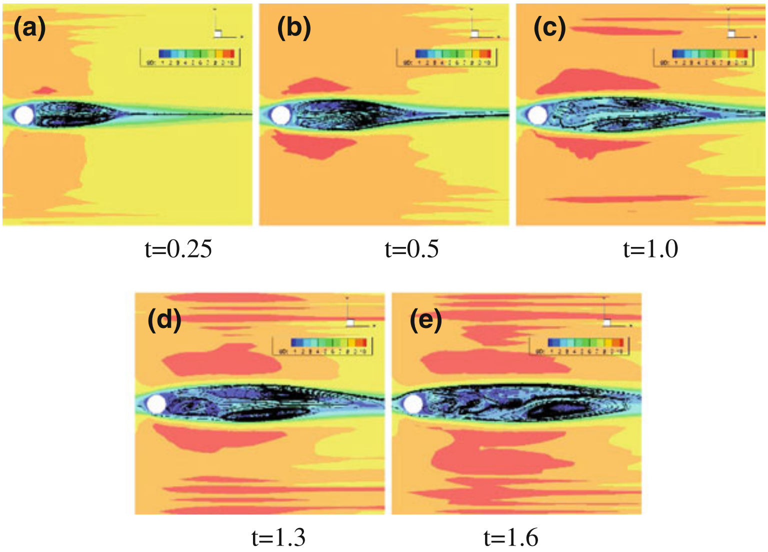


Fig. 4 **a** Simulation domain decomposition onto eight processors (PE is the number of the processor) and **b** mesh set up



t=0.25 t=0.5 t=1.0

t=1.3 t=1.6

Fig. 5 Instantaneous velocity streamlines at different time (seconds) of the simulation, Re = 1.2×10^5 and SD is the velocity magnitude

size of $\Delta t = 1.64 \times 10^{-5}$ while $\Delta t = 1.21 \times 10^{-7}$ is needed for the explicit solver, running this lower than stability limits require. The implicit solver requires more computational effort, taking 2.62 s for each time step compared to only 0.25 s for the explicit solver. However, the implicit solution scheme produces a shorter wall-clock time per calculation by a factor of 10.

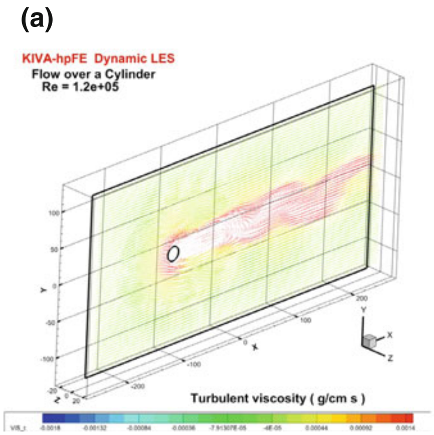
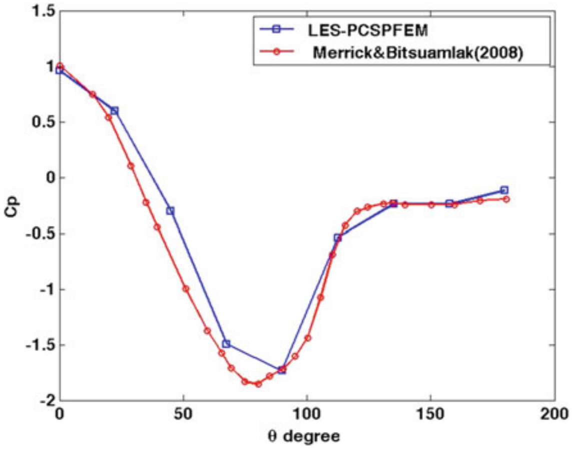


Fig. 6 **a** Turbulent viscosity and **b** local Mach number

Fig. 7 Pressure coefficient comparison with experiment for Re = 1.2×10^5



6.2 3-D Flow Over a Backward-Facing Step Using Adaptive FEM and LES

For this test case, 3-D flow over a backward-facing step is examined using the h-adaptive system and LES. The Reynolds number for this test case is 28,000 (determined by the step height) using air as the fluid at 296 K° with Prandtl number (Pr) of 0.71. The step height (H) is 0.025 m. The outflow boundary is modeled with a zero-gradient applied for momentum, which is located 30 step height downstream from the expansion. No-slip boundary conditions are applied on the walls except for the Z direction, which receives a periodic boundary condition to remove 3-D effects from the side walls. The 2-D version of the model is discussed in Waters et al. [10]. Here, we present the comparison of LES and RANS model with the help of the h-

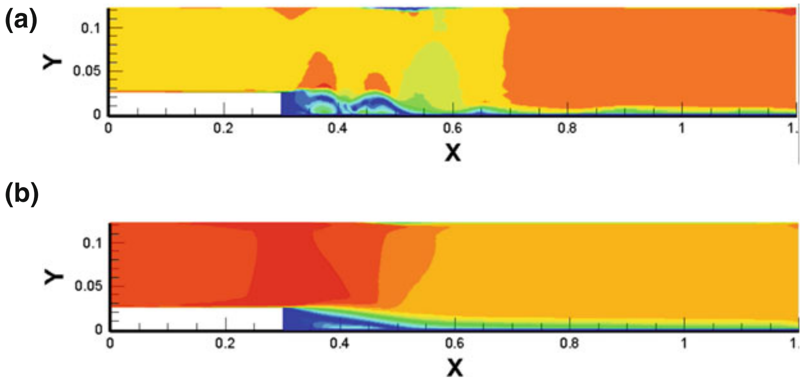


Fig. 8 2-D instantaneous flow magnitude with adaptation by LES (**a**) and RANS, (**b**) at 24 s

adaptation in Fig. 8 where LES in Fig. 8a is better at capturing the instantaneous flow than RANS in Fig. 8b where only the average results are shown. Figure 9 is showing the h-adaptation and the heat flux is well simulated here. In this work, we adapted the mesh twice (level 2 adaptation). 3-D simulation is demonstrated in this study. The initial mesh starts with a coarse mesh having 6,955 nodes and 4,976 elements as shown in Fig. 10a. LES requires a resolved grid in order to evaluate the boundary layer properly so with the use of mesh refinement algorithm, h-adaptation, the grid density is altered automatically in the region of interest. The refinement is limited to this region since the solution is performed on 1 processor, restricting the amount of memory available for more cells and greater refinement. In order to capture the recirculation after the step in the downstream, the initial grid is adapted twice and the final grid has 18,930 nodes and 16,316 elements shown in Fig. 10b. Figure 10c shows the adaptation at a meridional slice at $Z=0$, where the recirculation region has been h-adapted to two levels. Taking the time-averaged velocity allows for the measurement of the mean recirculation length behind the step. Figure 11a shows the reattachment region ranges from −0.25 to −0.07 m, with a secondary eddy being well captured in the corner. The reattachment length is therefore 7.6H, agreeing with the experimental value ~7.2H and previous work by Waters and Carrington [19]. Figure 11b shows the instantaneous flow at $t=0.6$ s, where fully turbulent flow is shown behind the step.

7 Conclusions

A parallel implicit solver with a dynamic 3-D LES model has been embedded in KIVA-hpFE. Flow over a cylinder with $Re=1.2 \times 10^5$ is tested and the numerical results compared with experimental data. The vortex shedding is shown to be accurately captured in the 3-D simulation. Comparing with the use of an explicit solver,

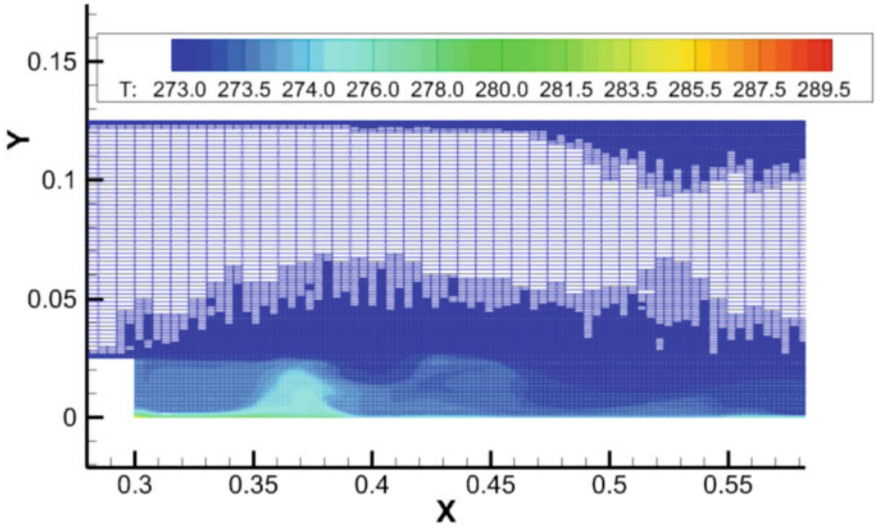


Fig. 9 Shown the adaptation up to level 2 adaptation with temperature contour

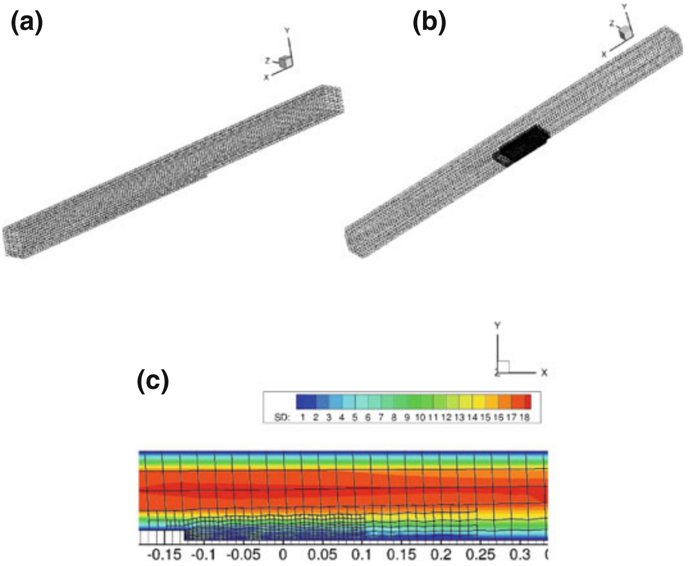
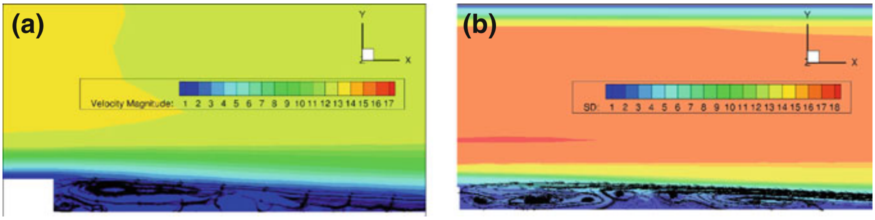


Fig. 10 Initial mesh (**a**), final mesh after adaptation (**b**), and (**c**) slice mesh adaptation at z = 0



Average velocity

Instantaneous flow at t=0.6s

Fig. 11 Average velocity streamline and velocity magnitude are the average velocity magnitude, instantaneous flow at t = 0.6 s (**a**) average velocity (**b**) instantaneous flow at t = 0.6 s

a 10x improvement in wall-clock time is achieved for achieving the same results. It should be noted that the parallel MPI implementation provides for ~30x speed-up over serial solution. Overall, the implicit method, when combined with parallelization, essentially yields a 300x speed-up over the explicit serial system. The adaptive, dynamic LES model was also applied to 3-D flow over a backward-facing step and achieved very good results compared with studies reported in the literature with just two levels of grid refinement. Efforts are underway to expand the system to include a parallel hp-adaptive scheme. The hp-adaptive FEM strategy follows a three-step strategy using posterior error estimators based on the L_2 norm and is expected to achieve exponential convergence rates.

Acknowledgements The DOE's Office of Energy Efficiency and Renewable Energy (EERE) Advanced Combustion Program (Gurpreet Singh and Leo Breton) is supporting this effort. Los Alamos National Laboratory, an affirmative action/equal opportunity employer is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the US Department of Energy (DOE) under contract DE-AC52-06NA25396. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

References

1. A.A. Amsden, P.J. O'Rourke, T.D. Butler, KIVA-II, a computer program for chemically reactive flows with sprays. Los Alamos, N.M.: Los Alamos National Laboratory Scientific Report, LA-11560-MS (1989)
2. D.C. Wilcox, *Turbulence Modeling for CFD*, 3rd edn. (DCW Industries Inc, La Canada CA, 2006)
3. P.E. Desjardins, S.H. Frankel, Two dimensional Large Eddy Simulation of soot formation in the near field of a strongly radiating non-premixed acetylene-air jet flame. Combust. Flame **119**, 121–133 (1999)
4. O. Colin, F. Ducros, D. Veynante, T. Poinsot, A thickened flame model for large eddy simulations of turbulent premixed combustion. Phys. Fluids **12**, 1843–1863 (2000)
5. C. Angelberger, F. Egolfopoulos, D. Veynante, Large Eddy Simulations of chemical and acoustic effects on combustion instabilities. Flow Turbul. Combust. **65**, 205–222 (2000)

6. H. Pitsch, Duchamp, L. de la Geneste, Large Eddy simulation of premixed turbulent combustion using a level-set approach, in *Proceedings of the Combustion Institute*,vol. 29 (2002) (in press)
7. C.D. Pierce, P. Moin, Progress-variable approach for large eddy simulation of non-premixed turbulent combustion. J. Fluid Mech. **504**, 73–97 (2004)
8. G. E Lau, G.H. Yeoh, V. Timchenko, J.A. Reizes, Application of dynamic global-coefficient subgrid-scale models to turbulent natural convection in an enclosed tall cavity. Phys. Fluids (1994-present) **24**, 094105 (2012)
9. A.W. Vreman, An eddy-viscosity subgrid-scale model for turbulent shear flow: algebraic theory and applications. Phys. Fluids **16**, 3670–3678 (2004)
10. J. Waters, D.B. Carington, D.W. Pepper, An adaptive finite element method with dynamic LES for turbulent reactive flows. Comput. Therm. Sci. Int. J. **2940–2503**(8–1), 57–71 (2016)
11. P.K. Jimack, N. Touheed, Developing parallel finite element software using MPI. HPC Comput. Mech. 15–38 (2000)
12. J. Waters, D.B. Carrington, Modeling turbulent reactive flow in internal combustion engines with an LES in a semi-implicit/explicit finite element projection method*, in Proceedings of the ASME 2016 Internal Combustion Fall Technical Conference*, ICEF2016, 9–12 Oct 2016, Greenville, SC, USA (accepted)
13. W.D. Joubert, G.F. Carey, PCG: a software package for the iterative solution of linear systems on scalar, vector and parallel computers, in *Society for Industrial and Applied Mathematics (SIAM) Proceedings of the 6th SIAM Conference on Parallel Processing for Scientific Computing*, 22–24 Mar, Norfolk, Virginia (1993)
14. M. Germano, U. Piomelli, P. Moin, W.H. Cabotm, A dynamic subgrid-scale eddy viscosity model. Phys. Fluids A **3**(7), 1760–1765 (1991)
15. D.K. Lilly, A proposed modification of the Germano subgrid-scale closure method. Phys. Fluids A **4**(3), 633–635 (1992)
16. X. Wang, D.W. Pepper, Application of an hp-adaptive FEM for solving thermal flow problems. AIAA J. Thermophys. Heat Transf. **21**(1), 190–198 (2007)
17. D.B. Carrington, X. Wang, D.W. Pepper, A Predictor-Corrector Split projection method for turbulent reactive flow. Comput. Therm. Sci. **5**(4), 333–353 (2013)
18. D.B. Carrington, A parallel first-order spherical harmonics (P1) matrix-free method for radiative transport, in *Numerical Heat Transfer, Part B: Fundamentals*, vol. 53 (Taylor and Francis, 2008), pp. 1–21
19. J. Waters, D.B. Carrington, A parallel large eddy simulation in a finite element projection method for all flow regimes. Numer. Heat Transf. Part A Appl. **70**(2), 117–131 (2016)
20. P.K. Jimack, N. Touheed, Developing parallel finite element software using MPI. High Perform. Comput. Comput. Mech. 15–38 (2000)
21. G. Karypis, V. Kumar, MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0. 2009, http://www.cs.umn.edu/~metis, University of Minnesota, Minneapolis, MN
22. A.A. Mustto, G.C.R. Bodstein, Subgrid-scale modeling of turbulent flow around circular cylinder by mesh-free vortex method. Eng. Appl. Comput. Fluid Mech. **5**(2), 259–275 (2011)
23. R. Merrick, G. Bitsuamlak, Control of flow around a circular cylinder by the use of surface roughness: a computational and experimental approach, http://www.ihrc.fiu.edu/wpcontent/uploads/2014/03/MerrickandBitsuamlak_FlowAroundCircularCylinders.pdf. Accessed 2008
24. T. Kawamura, T. Nakao, M. Takahashi, M. Hayashi, K. Murayama, N. Gotoh, Synchronized vibrations of a circular cylinder in cross flow at supercritical reynolds numbers. ASME. J. Press. Vessel Technol. **125**(1), 97–108 (2003)