Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 219 (2006) 943–975
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

KIVA-4: An unstructured ALE code for compressible gas flow with sprays

David J. Torres [a,*], Mario F. Trujillo [b]

[a] Theoretical Division, Fluid Dynamics Group (T-3), Los Alamos National Laboratory, MS B216, Los Alamos, NM 87545, USA
[b] Applied Research Laboratory at the Pennsylvania State University, State College, PA 16804, USA

Received 6 February 2006; received in revised form 3 July 2006; accepted 7 July 2006
Available online 30 August 2006

Abstract

The KIVA family of codes was developed to simulate the thermal and fluid processes taking place inside an internal combustion engine. In this latest version of this open source code, KIVA-4, the numerics have been generalized to unstructrured meshes. This change required modifications to the Lagrangian phase of the computations, the pressure solution and fundamental changes in the fluxing schemes of the rezoning phase. This newest version of the code inherits all the droplet phase capabilities and physical sub-models of previous versions. The integration of the gas phase equations with moving solid boundaries continues to employ the successful arbitrary Lagrangian–Eulerian (ALE) methodology. Its new unstructured capability facilitates grid construction in complicated geometries and affords a higher degree of flexibility. The numerics of the code, emphasizing the new additions, are described. Various computational examples are performed demonstrating the new capabilities of the code.
Published by Elsevier Inc.

Keywords: Unstructured meshes; Arbitrary Lagrangian–Eulerian; Fluid dynamics; Spray; Internal combustion engines

1. Introduction

The computation of the fluid mechanics in an internal combustion engine poses serious challenges to the researcher from both the underlying physics, which include spray dynamics, chemical reactions, liquid impingement, turbulence, etc. but also complicated moving geometries with potential topological changes. It is this latter dynamic geometrical aspect which motivated code developers at Los Alamos National Laboratory to implement an arbitrary Lagrangian–Eulerian (ALE) methodology into their simulation strategy [7] when developing a series of computational fluid dynamics codes called KIVA. These codes integrated the essential underlying physics models (spray, combustion, turbulence) and moving boundaries in a computational fluid dynamics code to simulate internal combustion engines. While KIVA was designed and applied mainly in engines, its capabilities can and have been applied in other applications.

* Corresponding author. Tel.: +1 505 667 2638; fax: +1 505 665 5926.
E-mail address: dtorres@lanl.gov (D.J. Torres).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2006.07.006

Nomenclature

a	pressure gradient scaling parameter
A_o	switch (1 or 0) to turn turbulence on or off
$\mathbf{A}_\mathrm{f}$	cell face area vector
c_v, c_p	specific heats at constant volume and pressure
D	gas mass diffusion coefficient
$\mathbf{F}^{\mathrm{spray}}$	spray momentum transfer term
$\mathbf{g}$	gravity
k	turbulent kinetic energy per unit mass
p	pressure
h	specific enthalpy
I	specific internal energy
K	thermal conductivity
M	mass of cell
M_v	mass of control volume surrounding a vertex
Pr_t	turbulent Prandtl number
$\dot{Q}^{\mathrm{chem}}$	chemical source term
$\dot{Q}^{\mathrm{spray}}$	spray source term
R_o	universal gas constant
$\bar{R} = R_o \sum_m \frac{Y_m^\mathrm{B}}{W_m}$	gas constant
Sc_t	turbulent Schmidt number
t	time
T	temperature
$\mathbf{u}$	velocity
$\tilde{\mathbf{u}}$	turbulent gas velocity fluctuation
$\mathbf{v}_\mathrm{p}$	particle velocity
V	volume
$\dot{W}^{\mathrm{spray}}$	spray source term in turbulence equations
W_m	molecular mass of species m
Y_m	mass fraction of species m
ϵ	turbulent dissipation rate
ϕ_D	variable implicitness parameter
ϕ_p	variable implicitness parameter for pressure
ρ	density
ρ_m	density of species m
$\dot{\rho}^{\mathrm{spray}}$	spray source term
$\dot{\rho}_m^{\mathrm{chem}}$	chemical source term for species m
$\boldsymbol{\sigma}$	viscous stress tensor
μ	coefficient of viscosity
λ	coefficient of viscosity
Δt	time step

Subscripts

c	cell
f	face
m	species
p	particle
t	turbulent
v	vertex control volume

Superscripts
A, B, C stage A, B or C
n time level n

The first version of KIVA was capable of computing transient compressible flow dynamics with fuel sprays and combustion in relatively simple two- and three-dimensional geometries [5,4]. KIVA-II [3] made much of the temporal differencing in KIVA implicit. While advection remained explicit with a subcycled time step, advection was made more accurate with an improved upwinding scheme. A k–ϵ turbulence model was also incorporated. KIVA-3 [1] added the capability of using a block-structured mesh where multiple blocks of cells could be patched together to construct a mesh. Software to generate block-structured meshes and post-processing visualization tools were included with the KIVA-3 package. The code was enhanced by a procedure (snapping) used to remove or add layers of cells during piston movement. KIVA-3V [2] added vertical or canted valves and a particle-based liquid wall film model. Due partly to its open source distribution, KIVA-3V has been used extensively by universities and industry to perform engine simulations and to serve as a platform for physics sub-model development [14,19–21]. Throughout the versions of the KIVA code, the ALE method has continued to be used due to its attractive feature of integrating efficiently fluid transport equations with moving solid boundaries. The strengths of this method in handling dynamic boundaries are evidenced by its use in tracking free surface flows [6] or in handling fluid–solid interactions [11,10].

In this paper, we discuss the latest version of the KIVA codes, KIVA-4 [15,16], which generalizes the computational grid from structured to unstructured. This required modifications in the calculation of geometric related quantities to compute diffusional terms and pressure in the Lagrangian stage of the calculations. In addition, required modifications to the rezoning stage (cell and momentum fluxing) were implemented. These constitute fundamental changes in the numerics of the code. Unstructured grids (compared to structured grids) provide an easier route in the discretization of physical domains as complicated as internal combustion engines. The unstructured grids can be composed of a variety of elements including hexahedra, prisms, pyramids, and tetrahedra. In the development of KIVA-4, particular emphasis was placed on keeping KIVA-4 comparable in computational efficiency to KIVA-3V. KIVA-4 maintains the full generality of previous versions in particular in the use of the latest physics sub-models for multicomponent vaporization, wall impingement, Lagrangian liquid film movement, drop breakup, etc. The snapping procedure which accommodates piston and valve motion is also implemented with some restrictions on an unstructured grid. The restricted unstructured grid starts with an unstructured grid in a two-dimensional slice normal to the cylindrical axis. Logically equivalent layers of this slice are stacked to fill up the cylindrical volume.

Many of the equations in the KIVA-II report [3] are repeated here for completeness. The exception are the equations governing spray dynamics which required little modification. Section 2 describes the governing equations, Sections 3–5 detail the numerical implementation of the equations making special emphasis on the new implementations. A summary of the equations solved in a complete numerical cycle is provided in Section 6. Various examples of numerical calculations and their comparisons to analytical results are included in Section 7.

Table 1
Turbulence constants

Constants	Standard KIVA	RNG k–ϵ
c_{ϵ_1}	1.44	1.42
c_{ϵ_2}	1.92	1.68
c_{ϵ_3}	−1.0	Eq. (55)
Pr_ϵ	1.3	0.72
Pr_k	1.0	0.72
c_μ	0.09	0.085
c_s	1.5	1.5

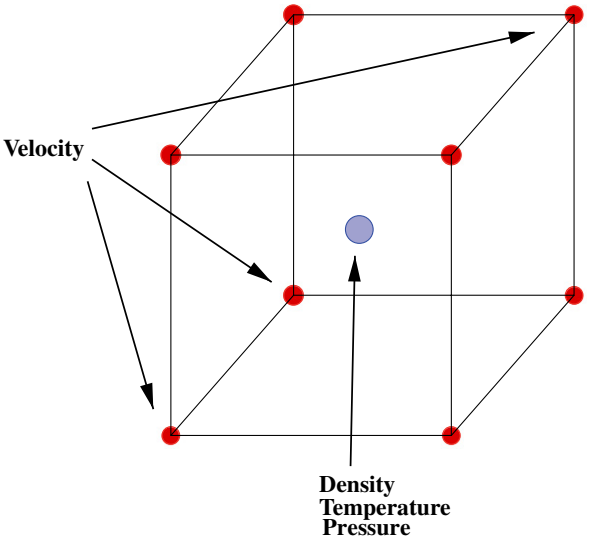


Fig. 1. KIVA's variable placement in a staggered mesh.

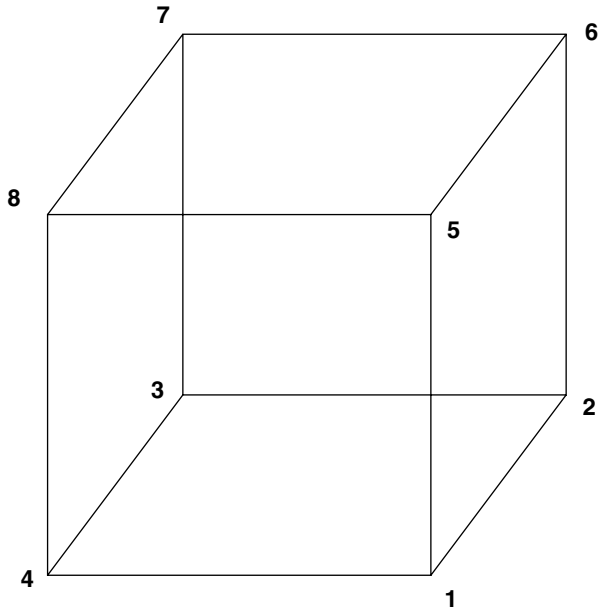


Fig. 2. Indexing convention for a cell.

KIVA-4 should be available for distribution through the Energy Science and Technology Software Center in the fall of 2006.

2. Governing equations

KIVA-4 solves the following conservation equations presented in integral form:

- conservation of mass for species m

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_V \rho_m \,\mathrm{d}V = \int_S \left[\rho D \nabla\left(\frac{\rho_m}{\rho}\right)\right]\mathrm{d}\mathbf{A} + \int_V \dot{\rho}_m^{\mathrm{chem}}\,\mathrm{d}V + \int_V \dot{\rho}_m^{\mathrm{spray}}\,\mathrm{d}V; \tag{1}$$

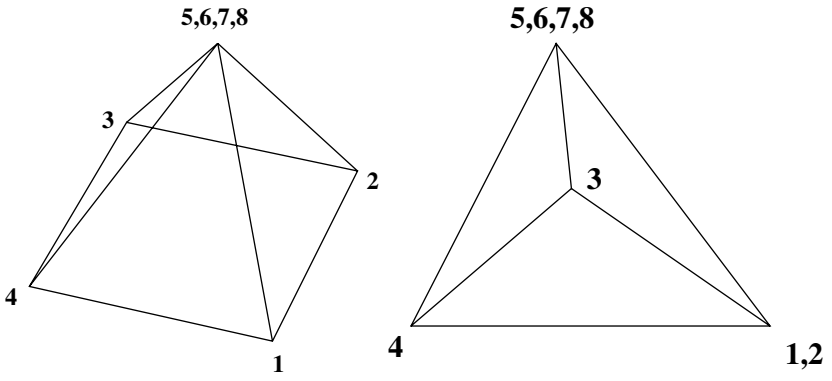


Fig. 3. Pyramid (left) formed and tetrahedron (right) formed by degenerating cell nodes.

- conservation of mass (this equation is automatically satisfied when all chemical species above are solved)

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_V \rho \,\mathrm{d}V = \int_V \dot{\rho}^{\mathrm{spray}} \,\mathrm{d}V; \tag{2}$$

- conservation of momentum

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_V \rho\mathbf{u} \,\mathrm{d}V = -\int_S \left[\frac{1}{a^2}p + A_o\frac{2}{3}\rho k\right]\mathrm{d}\mathbf{A} + \int_S \boldsymbol{\sigma}\cdot\mathrm{d}\mathbf{A} + \int_V \mathbf{F}^{\mathrm{spray}}\,\mathrm{d}V + \int_V \rho\mathbf{g}\,\mathrm{d}V; \tag{3}$$

- conservation of energy

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_V \rho I \,\mathrm{d}V = \int_V -p\nabla\cdot\mathbf{u}\,\mathrm{d}V + \int_V (1-A_o)\boldsymbol{\sigma}:\nabla\mathbf{u}\,\mathrm{d}V + \int_S \left[K\nabla T + \rho D\sum_m h_m \nabla\left(\frac{\rho_m}{\rho}\right)\right]\mathrm{d}\mathbf{A} + \int_V A_o\rho\epsilon\,\mathrm{d}V + \int_V \dot{Q}^{\mathrm{chem}}\,\mathrm{d}V + \int_V \dot{Q}^{\mathrm{spray}}\,\mathrm{d}V; \tag{4}$$

and the k–ϵ turbulence equations

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_V \rho k \,\mathrm{d}V = -\int_V \frac{2}{3}\rho k\nabla\cdot\mathbf{u}\,\mathrm{d}V + \int_V \boldsymbol{\sigma}:\nabla\mathbf{u}\,\mathrm{d}V + \int_S \left[\left(\frac{\mu}{Pr_k}\right)\nabla k\right]\mathrm{d}\mathbf{A} - \int_V \rho\epsilon\,\mathrm{d}V + \int_V \dot{W}^{\mathrm{spray}}\,\mathrm{d}V, \tag{5}$$

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_V \rho\epsilon \,\mathrm{d}V = -\int_V \left(\frac{2}{3}c_{\epsilon_1} - c_{\epsilon_3}\right)\rho\epsilon\nabla\cdot\mathbf{u}\,\mathrm{d}V + \int_S \left[\left(\frac{\mu}{Pr_\epsilon}\right)\nabla\epsilon\right]\cdot\mathrm{d}\mathbf{A} + \int_V \frac{\epsilon}{k}\left[c_{\epsilon_1}\boldsymbol{\sigma}:\nabla\mathbf{u} - c_{\epsilon_2}\rho\epsilon + c_s\dot{W}^{\mathrm{spray}}\right]\mathrm{d}V, \tag{6}$$

where Pr_k, Pr_ϵ, c_{ϵ_1}, c_{ϵ_2}, c_{ϵ_3} and c_s are turbulence constants defined in Table 1. KIVA-4 also provides for an RNG (ReNormalization Group) model for turbulence which is described in Section 4.6. In Eq. (3), a is the pressure gradient scaling (PGS) parameter which is normally set to 1.

The droplet phase employs a stochastic method for Lagrangian particle dynamics [3]. The full treatment of this phase including physical sub-models accounting for wall impingement, vaporization, collisions, and aerodynamic breakup is directly inherited from the previous versions of the code [2,17].

3. KIVA's ALE scheme

KIVA-4 uses an ALE method. The conservation equations are solved in three stages. In stage A, the influence of the spray, wall film particles and chemical reactions on gas quantities is computed. Stage B solves the

governing equations in Lagrangian form using a finite volume scheme. The Lagrangian description automatically accounts for the advective terms without having to explicitly discretize them. Stage C is the Eulerian or rezoning stage, in which the grid is moved to new locations. The new locations of the grid are usually chosen to preserve grid quality but they also can be chosen to effect local grid refinement in regions of interest. Fluxes of mass, momentum, energy and turbulence quantities are exchanged during this rezoning stage.

KIVA-4 staggers its variables. All variables except velocity are located at cell-centers. Velocity is located at the vertices of the cell. See Fig. 1. Let N_i denote the indices of the nodes forming cell i. KIVA-4 (like KIVA-3V) uses the cell indexing convention shown in Fig. 2 (only the subscripts of N_i are shown in Fig. 2).

However in KIVA-4 (unlike KIVA-3V) nodes indices can be equal. For example, if $N_5 = N_6 = N_7 = N_8$, the top face of the cell degenerates into a point, forming a pyramid. If $N_5 = N_6 = N_7 = N_8$, and $N_1 = N_2$, a tetrahedron is formed. See Fig. 3. We refer to cells which have collapsed edges (or equal node indices) as degenerate cells or elements.

4. Numerical schemes

In this section, the solution of the governing equations (1)–(6) in stage A and stage B is presented. Specifically the algorithm for updating density, velocity, temperature and pressure is described. A finite volume scheme requires one to define a control volume with which to compute volume and surface integrals. In updating density, temperature and pressure, which are cell-centered quantities, the control volume coincides with the cell volume. In updating the velocity field, which is defined at all vertices, the control volume consists of portions of computational cells that share a given vertex (see Section 4.2). A detailed description follows for the solution of each respective field.

4.1. Density

The stage A density of species includes contributions from chemistry and spray evaporation and is solved at constant volume:

$$\frac{\rho_m^{\mathrm{A}} - \rho_m^n}{\Delta t} = \dot{\rho}_m^{\mathrm{chem}} + \dot{\rho}_m^{\mathrm{spray}}. \tag{7}$$

Eq. (7) assumes that all evaporated mass from fuel droplets in a cell is distributed uniformly within that cell (which incidentally causes the spray evolution to be dependent on grid resolution). Once the stage A species densities have been calculated, stage A density and mass fractions are calculated

$$\rho^{\mathrm{A}} = \sum_m \rho_m^{\mathrm{A}} \quad \text{and} \quad Y_m^{\mathrm{A}} = \frac{\rho_m^{\mathrm{A}}}{\rho^{\mathrm{A}}}. \tag{8}$$

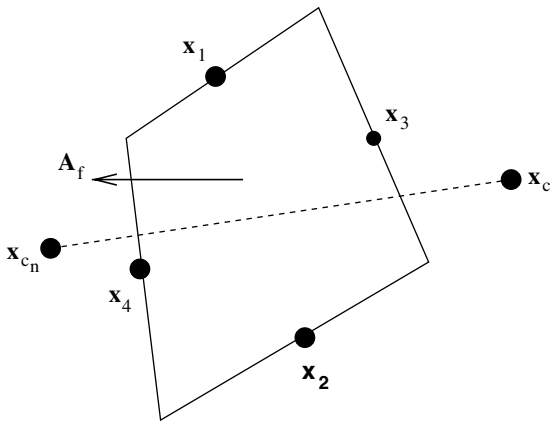


Fig. 4. Edge conventions in calculating geometric coefficients.

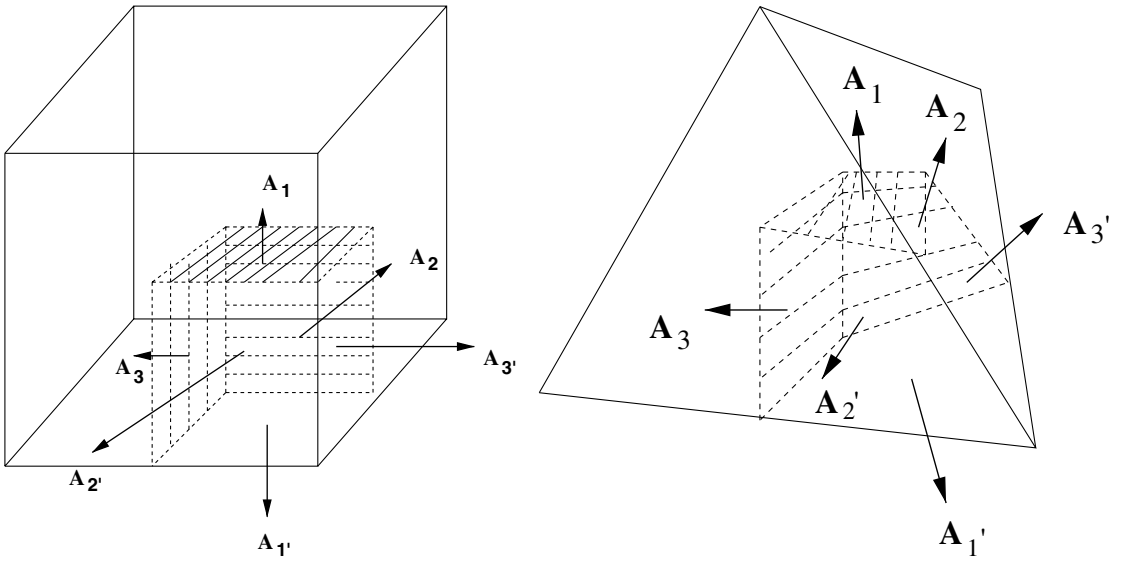


Fig. 5. Portion of the momentum control volume for a hexahedra (left) and a tetrahedron (right).

The Lagrangian stage B species densities are then computed using the finite volume approximation to (1) derived in Appendix A,

$$M^{\mathrm{B}}\frac{Y_m^{\mathrm{B}}-Y_m^{\mathrm{A}}}{\Delta t}=\sum_f(\rho D)_{\mathrm{f}}^n\nabla[\phi_{\mathrm{D}}Y_m^{\mathrm{B}}+(1-\phi_{\mathrm{D}})Y_m^{\mathrm{A}}]_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^n. \tag{9}$$

(Note that stage A already incorporates chemical and spray sources.) In (9), $\mathbf{A}_{\mathrm{f}}$ is a vector which points in the direction of the outward facing cell face normal and whose magnitude is equal to the area of the face. Eq. (9) is approximated using a turbulent Schmidt number $Sc_{\mathrm{t}}=\frac{\mu_{\mathrm{t}}}{\rho D}$,

$$M^{\mathrm{B}}\left(Y_m^{\mathrm{B}}-Y_m^{\mathrm{A}}\right)-\frac{\Delta t}{Sc_{\mathrm{t}}}\left[\sum_f(\mu_{\mathrm{t}}^n)_{\mathrm{f}}(\nabla Y_m^{\mathrm{A}})_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^n+\sum_f(\mu_{\mathrm{t}}^n)_{\mathrm{f}}\nabla[\phi_{\mathrm{D}}(Y_m^{\mathrm{B}}-Y_m^{\mathrm{A}})]_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^n\right]=0. \tag{10}$$

Eq. (10) is solved in every computational cell. The viscosity μ_{t} is computed using

$$\mu_{\mathrm{t}}=\mu_{\mathrm{air}}+A_o c_\mu\rho\frac{k^2}{\epsilon}, \tag{11}$$

where μ_{air} is the laminar air viscosity computed using $A_1T^{\frac{3}{2}}/(T+A_2)$, $A_o c_\mu\rho\frac{k^2}{\epsilon}$ is the turbulent contribution, A_1, A_2 and c_μ are constants, normally set to $1.457\times10^{-5}\ \frac{\mathrm{g}}{\mathrm{cm\ s\ \sqrt{K}}}$, 110 K, and 0.09, respectively. Face viscosities are computed by averaging the cell viscosities adjoining the cell.

The terms

$$\sum_f(\mu_{\mathrm{t}}^n)_{\mathrm{f}}(\nabla Y_m^{\mathrm{A}})_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^n\quad\text{and}\quad\sum_f(\mu_{\mathrm{t}}^n)_{\mathrm{f}}\nabla[\phi_{\mathrm{D}}(Y_m^{\mathrm{B}}-Y_m^{\mathrm{A}})]_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^n$$

or any term of the form $\sum_f(\nabla Q)_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}$ is computed by first calculating geometric coefficients a_{c}, $a_{\mathrm{e}_{12}}$ and $a_{\mathrm{e}_{34}}$ using

$$a_{\mathrm{c}}(\mathbf{x}_{\mathrm{c}_n}-\mathbf{x}_{\mathrm{c}})+a_{\mathrm{e}_{12}}(\mathbf{x}_1-\mathbf{x}_2)+a_{\mathrm{e}_{34}}(\mathbf{x}_3-\mathbf{x}_4)=\mathbf{A}_{\mathrm{f}}, \tag{12}$$

where $\mathbf{x}_{\mathrm{c}}$ is the cell-center, $\mathbf{x}_{\mathrm{c}_n}$ is the cell-center of the neighboring cell across the face, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the centers of opposite edges 1 and 2, and $\mathbf{x}_3$ and $\mathbf{x}_4$ are the centers of opposite edges 3 and 4. See Fig. 4.

In unstructured meshes, the face can be a quadrilateral or a triangle (which necessitates a modified treatment of (12)). If one edge is degenerate, say $\mathbf{x}_4$, we set $\mathbf{x}_4$ to be the face center. If more than one edge

is degenerate, the entire face area is zero. Solving (12) amounts to solving a 3×3 linear system of equations which is accomplished using Cramer's rule. Then $(\nabla Q)_{\mathrm{f}} \cdot \mathbf{A}_{\mathrm{f}}$ is computed using

$$a_{\mathrm{c}}(Q_{\mathrm{c}_n} - Q_{\mathrm{c}}) + a_{\mathrm{e}_{12}}(Q_1 - Q_2) + a_{\mathrm{e}_{34}}(Q_3 - Q_4) = (\nabla Q)_{\mathrm{f}} \cdot \mathbf{A}_{\mathrm{f}}, \tag{13}$$

where Q_{c} is the cell-centered value of Q, Q_{c_n} is the cell-centered value of the neighboring cell across the face, and Q_i are the values of Q on the face edges. The Q_i are calculated by averaging all cells that share the edge. If $\mathbf{x}_4$ is degenerate, we set $Q_4 = \frac{1}{2}(Q_{\mathrm{c}} + Q_{\mathrm{c}_n})$ to be the face value of Q. One should note that in solving (10), there is no net gain of mass, i.e. $M^{\mathrm{B}} = M^{\mathrm{A}}$.

4.2. Momentum

Solution of the momentum equation determines the velocity field, which in our present code is defined at the vertices. In performing this calculation, the control volume employed (denoted here as the vertex control volume V_{v}) is composed of portions of the many cells that share this vertex. See Fig. 5. Specifically, the vertex control volume V_{v} and mass M_{v} are determined by adding cell volume V_{c} and mass contributions M_{c} from each cell sharing the vertex according to

$$V_{\mathrm{v}} = \sum_c \chi_{\mathrm{c}} V_{\mathrm{c}} \quad \text{and} \quad M_{\mathrm{v}} = \sum_c \chi_{\mathrm{c}} M_{\mathrm{c}},$$

where χ_{c} is the reciprocal of the distinct number of vertices a cell owns. For example $\chi_{\mathrm{c}} = \frac{1}{8}$ for a hexahedron and $\chi_{\mathrm{c}} = \frac{1}{4}$ for a tetrahedron. In previous versions of KIVA, χ_{c} was always set to $\frac{1}{8}$.

The Lagrangian discretized version of the momentum equation (3) between stage B and n is given by

$$\begin{aligned}\frac{M_{\mathrm{v}}^{\mathrm{B}}\mathbf{u}^{\mathrm{B}} - M_{\mathrm{v}}^{n}\mathbf{u}^{n}}{\Delta t} = & -\sum_c \sum_{\beta_{\mathrm{c}}} \left[\frac{1}{(a^n)^2}\left(\phi_{\mathrm{p}} p^{\mathrm{p}} + (1-\phi_{\mathrm{p}})p^n\right) + A_0 \frac{2}{3}\rho^{\mathrm{A}} k^{\mathrm{A}} \right]_{\mathrm{c}} \mathbf{A}_{\beta_{\mathrm{c}}}^n + \sum_c \sum_{\beta_{\mathrm{c}}} [\phi_{\mathrm{D}} \boldsymbol{\sigma}(\mathbf{u}^{\mathrm{B}}) \\ & + (1-\phi_{\mathrm{D}})\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{A}})]_{\mathrm{c}} \cdot \mathbf{A}_{\beta_{\mathrm{c}}}^n - \sum_v N_{\mathrm{p}} \frac{4}{3}\pi\rho_{\mathrm{p}} \left[(r_{\mathrm{p}}^{\mathrm{B}})^3 \mathbf{v}_{\mathrm{p}}^{\mathrm{B}} - (r'_{\mathrm{p}})^3 \mathbf{v}'_{\mathrm{p}} \right] + M_{\mathrm{v}}^n \mathbf{g},\end{aligned}$$

where the sum over β_{c} is over all momentum facets corresponding to a given cell c. For instance, these momentum facets in one cell are illustrated by their normal vectors $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$ in Fig. 5. The sum over c is over all the cells which share the common vertex v. For an interior vertex in a structured hexahedral grid there will be 24 such momentum facets since in a structured hexahedral grid eight cells share a vertex. The terms ϕ_{p} and ϕ_{D} are the variable implicitness parameters which allow one to specify the degree of implicitness for a discretization. The momentum transfer due to spray $\int_V \mathbf{F}^{\mathrm{spray}}\,\mathrm{d}V$ is represented by the term

$$-\sum_v N_{\mathrm{p}} \frac{4}{3}\pi\rho_{\mathrm{p}} \left[(r_{\mathrm{p}}^{\mathrm{B}})^3 \mathbf{v}_{\mathrm{p}}^{\mathrm{B}} - (r'_{\mathrm{p}})^3 \mathbf{v}'_{\mathrm{p}} \right]. \tag{14}$$

The quantities r'_{p} and $\mathbf{v}'_{\mathrm{p}}$ denote the particle radius and velocity after droplet aerodynamic breakups, collisions, and gravitational acceleration. The terms $r_{\mathrm{p}}^{\mathrm{B}}$ and N_{p} represent the droplet radius after evaporation and the number of droplets in a parcel. The particle velocity at stage B, $\mathbf{v}_{\mathrm{p}}^{\mathrm{B}}$, is computed with the nearest updated gas velocity $\mathbf{u}^{\mathrm{B}}$ and its turbulent fluctuating component $\tilde{\mathbf{u}}$,

$$\frac{\mathbf{v}_{\mathrm{p}}^{\mathrm{B}} - \mathbf{v}'_{\mathrm{p}}}{\Delta t} = D_{\mathrm{p}}(\mathbf{u}^{\mathrm{B}} + \tilde{\mathbf{u}} - \mathbf{v}_{\mathrm{p}}^{\mathrm{B}}), \tag{15}$$

where D_{p} is the drag function [3]. This implicit calculation of the particle equation circumvents time step limitations due to the strong coupling of gas and droplet velocities. The term D_{p} is computed using

$$D_{\mathrm{p}} = \frac{3}{8}\frac{\rho^n}{\rho_{\mathrm{p}}^{\mathrm{B}}} \frac{|\mathbf{u}^n + \tilde{\mathbf{u}} - \mathbf{v}_{\mathrm{p}}|}{r'_{\mathrm{p}}} C_{\mathrm{D}}^* \tag{16}$$

where the drag coefficient C_{D}^* is defined by $C_{\mathrm{D}}^* = C_{\mathrm{D}}(1 + 2.632 y_{\mathrm{p}})$, C_{D} is defined by

$$C_D = \begin{cases} \frac{24}{Re_p}\left(1 + 1/6 Re_p^{\frac{2}{3}}\right), & Re_p < 1000, \\ 0.424, & Re_p > 1000, \end{cases} \quad Re_p = \frac{2\rho^n r'_p |\mathbf{u}^n + \tilde{\mathbf{u}} - \mathbf{v}_p|}{\mu_{air}(\hat{T})} \tag{17}$$

and $\hat{T} = T_p + \frac{1}{3}(T^n - T_p)$. The term y_p is used to represent the droplet deviation from sphericity. Solving for $\mathbf{v}_p^B$ in (15) and substituting this into Eq. (14) gives

$$\frac{(M_v^B + S_v)\mathbf{u}^B - M_v^n \mathbf{u}^n}{\Delta t} = -\sum_c \sum_{\beta_c} \left[\frac{1}{(a^n)^2}\left(\phi_p p^p + (1-\phi_p)p^n\right) + A_0 \frac{2}{3}\rho^A k^A \right]_c \mathbf{A}_{\beta_c}^n + \sum_c \sum_{\beta_c} [\phi_D \boldsymbol{\sigma}(\mathbf{u}^B) + (1-\phi_D)\boldsymbol{\sigma}(\mathbf{u}^A)]_c \cdot \mathbf{A}_{\beta_c}^n - \frac{\mathbf{R}_v}{\Delta t} + M_v^n \mathbf{g},$$

where

$$S_v = \left(\Delta t \sum_v N_p \frac{4}{3}\pi \rho_p (r_p^B)^3 \frac{\Delta t D_p}{(1 + \Delta t D_p)} \right) \quad \text{and}$$

$$\mathbf{R}_v = \Delta t \sum_v N_p \frac{4}{3}\pi \rho_p \left[(r_p^B)^3 \left(\frac{\mathbf{v}'_p + \Delta t D_p \tilde{\mathbf{u}}}{1 + \Delta t D_p} \right) - (r'_p)^3 \mathbf{v}'_p \right]. \tag{18}$$

The sums in S_v and $\mathbf{R}_v$ are sums over all particles defined to be within the vertex control volume. The calculation of (18) is broken up into two steps from the n to the A stage and from the A to the B stage. The stage A vertex velocities incorporate gravity and an implicit coupling to the spray particles,

$$\frac{(M_v^A + S_v)\mathbf{u}^A - M_v^n \mathbf{u}^n}{\Delta t} = \frac{-\mathbf{R}_v}{\Delta t} + \mathbf{g} M_v^n. \tag{19}$$

The second step is given by

$$(M_v^B + S_v)\frac{\mathbf{u}^B - \mathbf{u}^A}{\Delta t} = -\sum_c \sum_{\beta_c} \left[\frac{1}{(a^n)^2}\left(\phi_p p^p + (1-\phi_p)p^n\right) + A_0 \frac{2}{3}\rho^A k^A \right]_c \mathbf{A}_{\beta_c}^n + \sum_c \sum_{\beta_c} [\phi_D \boldsymbol{\sigma}(\mathbf{u}^B) + (1-\phi_D)\boldsymbol{\sigma}(\mathbf{u}^A)]_c \cdot \mathbf{A}_{\beta_c}^n. \tag{20}$$

To initiate the iteration that will govern the velocity, pressure, and temperature calculations (discussed in Section 4.5), the initial value for pressure above, p^p, is assigned the following extrapolated value from previous time steps

$$p^p = p^{n-1} + \frac{\Delta t^n}{\Delta t^{n-1}}[p^{n-1} - p^{n-2}]. \tag{21}$$

The extrapolated pressure is improved by using the stage B pressure computed in Section 4.4 in subsequent iterations within a cycle. This extrapolated pressure is a reasonable first approximation since in engine calculations the pressure experiences a large temporal variation relative to its spatial variation.

The values for $\mathbf{A}_{\beta_i}$ are not explicitly evaluated. Instead a property of closed surfaces is used. The surface integral over a closed surface is zero

$$\int d\mathbf{A} = 0 \rightarrow \mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3 + \mathbf{A}_{1'} + \mathbf{A}_{2'} + \mathbf{A}_{3'} = 0, \tag{22}$$

where $\mathbf{A}_{i'}$ denotes the partial faces that coincide with the actual cell faces. See Fig. 5. This notation allows one to write for one cell sharing a vertex

$$\sum_{\beta_c} \xi_c \mathbf{A}_{\beta_c} = \xi_c[\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3] = -\xi_c[\mathbf{A}_{1'} + \mathbf{A}_{2'} + \mathbf{A}_{3'}] \equiv -\sum_{\beta'_c} \xi_c \mathbf{A}_{\beta'_c}, \tag{23}$$

where ξ_c is any cell-centered quantity. Furthermore, since $\mathbf{A}_{\beta'_c}$ are actually portions of the actual cell faces $\mathbf{A}_{f_c}$, one can write

$$\sum_{\beta_c} \xi_c \mathbf{A}_{\beta_c} = -\sum_{\beta'_c} \xi_c \mathbf{A}_{\beta'_c} = -\xi_c \sum_f \chi_{f_c} \mathbf{A}_{f_c}, \tag{24}$$

where χ_{f_c} is set equal to one-quarter of the area of the normal cell face for a quadrilateral face and one-third of the area of the normal cell face for a triangular face. Previous versions of KIVA always set χ_{f_c} to $\frac{1}{4}$. Using (24), (20) can be rewrittten in the form with which it is solved

$$\begin{aligned}(M_v^B + S_v)\frac{\mathbf{u}^B - \mathbf{u}^A}{\Delta t} - \sum_c \left[\frac{1}{(a^n)^2}\left(\phi_p p^p + (1-\phi_p)p^n\right) + A_0 \frac{2}{3}\rho^A k^A\right]_c \sum_f \chi_{f_c}\mathbf{A}^n_{f_c} \\ + \sum_c \left[\phi_D \boldsymbol{\sigma}(\mathbf{u}^B) + (1-\phi_D)\boldsymbol{\sigma}(\mathbf{u}^A)\right]_c \cdot \sum_f \chi_{f_c}\mathbf{A}^n_{f_c} = 0,\end{aligned} \tag{25}$$

where $\sum_f \chi_{f_c}\mathbf{A}^n_{f_c}$ again denotes a sum over the three area face vectors of a cell that share a vertex. The viscous stress tensor itself is Newtonian, i.e.

$$\boldsymbol{\sigma} = \mu_t^n\left[(\nabla\mathbf{u}) + (\nabla\mathbf{u})^T\right] + \lambda^n \nabla\cdot\mathbf{u}\mathbf{I}, \tag{26}$$

where λ^n is normally set to $-\frac{2}{3}\mu_t^n$. Gradients of velocity within a cell are evaluated using

$$(\nabla\mathbf{u})_c \approx \frac{1}{V_c}\int_{V_c} \nabla\mathbf{u}\,dV = \frac{1}{V_c}\int_S \mathbf{u}\,d\mathbf{A} = \frac{1}{V_c}\sum_f \mathbf{u}_f\mathbf{A}^n_f. \tag{27}$$

The PGS (pressure gradient scaling) parameter, a, can be used to reduce sound speeds c by a factor of $1/a$, thus effectively lowering the sound speed Courant number, $\frac{c\Delta t}{\Delta x}$. The PGS method should not be used in problems where one wants to accurately track acoustic waves. However, in many applications where acoustic waves do not affect properties of interest, the PGS method can be used to improve the computational efficiency of the code.

4.3. Energy

The first step in the energy calculation occurs from the n-time level to the A stage through a constant volume process where only the spray and chemical reaction sources are considered,

$$\frac{M^A I^A - M^n I^n}{\Delta t} = V^n(\dot{Q}^{chem} + \dot{Q}^{spray}). \tag{28}$$

Then the internal energy is updated to an intermediate t-state accounting for turbulence dissipation and enthalpy diffusion

$$\begin{aligned}M^B\frac{I^t - I^A}{\Delta t} &= A_0 M^B \epsilon^A + \sum_f (\rho D)^n_f \left\{\sum_m h_m(T^n_f)\nabla\left[\phi_D Y^B_m + (1-\phi_D)Y^A_m\right]_f\right\}\cdot\mathbf{A}^n_f \\ &\approx A_0 M^B\epsilon^A + \frac{1}{Sc_t}\sum_m\left\{\sum_f (\mu^n_t)_f h_m(T^n_f)\nabla\left[\phi_D Y^B_m + (1-\phi_D)Y^A_m\right]_f\cdot\mathbf{A}^n_f\right\}.\end{aligned} \tag{29}$$

The temperature T^t is solved implicitly from

$$I^t = \sum_m Y^B_m I_m(T^t), \tag{30}$$

and the specific heat, $c^t_v = \partial I/\partial T|_V$, is then computed from the tabulated values of internal energy in the KIVA-4 fuel libraries. The Lagrangian step from the t-state to the B stage includes the viscous dissipation terms and diffusion terms,

$$\begin{aligned}M^B\frac{I^B - I^t}{\Delta t} \approx -p\sum_f \mathbf{u}_f\cdot\mathbf{A}_f + (1-A_o)\left[\phi_D\boldsymbol{\sigma}^B(\mathbf{u}^B):\nabla\mathbf{u}^B + (1-\phi_D)\boldsymbol{\sigma}^A(\mathbf{u}^A):\nabla\mathbf{u}^A\right]V^n \\ + \sum_f K^n_f\nabla\left[\phi_D T^B + (1-\phi_D)\tilde{T}\right]_f\cdot\mathbf{A}^n_f,\end{aligned} \tag{31}$$

where $\tilde{T}$ is given by

$$\tilde{T} = T^n + \frac{1}{c_{\mathrm{p}}^n}\left[I^{\mathrm{t}} - I^n + p^n\left(\frac{1}{\rho^{\mathrm{A}}} - \frac{1}{\rho^n}\right)\right], \tag{32}$$

and

$$-p\sum_f \mathbf{u}_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}} \approx -\left(\frac{p^{\mathrm{p}}+p^n}{2}\right)\left(\frac{V^{\mathrm{B}}-V^n}{\Delta t}\right). \tag{33}$$

$\tilde{T}$ is an approximation to T^{t} derived by assuming that heat addition occurs at constant pressure. To simplify (31), we have

$$I^{\mathrm{B}} - I^{\mathrm{t}} \equiv c_{\mathrm{v}}^{\mathrm{t}}(T^{\mathrm{B}} - T^{\mathrm{t}})$$

and the equation of state for an ideal gas,

$$V^{\mathrm{B}} = \frac{M^{\mathrm{B}}}{p^{\mathrm{p}}}\bar{R}T^{\mathrm{B}}. \tag{34}$$

Here $\bar{R} = R_o\sum_m \frac{Y_m^{\mathrm{B}}}{W_m}$. It should be noted that the ideal gas approximation is an good assumption under most conditions [18]. Substituting these expressions into (31) and solving for temperature at stage B yields

$$T^{\mathrm{B}} = \left\{ T^{\mathrm{t}} + \frac{p^{\mathrm{p}}+p^n}{2c_{\mathrm{v}}^{\mathrm{t}}}\frac{V^n}{M^{\mathrm{B}}} + \frac{\Delta t}{M^{\mathrm{B}}c_{\mathrm{v}}^{\mathrm{t}}}\left[\frac{1}{Pr_{\mathrm{t}}}\sum_f (c_{\mathrm{p}}\mu_{\mathrm{t}})_{\mathrm{f}}\nabla(\phi_{\mathrm{D}}T^{\mathrm{B}} + (1-\phi_{\mathrm{D}})\tilde{T})_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^n \right.\right.$$
$$\left.\left. +(1-A_o)(\phi_{\mathrm{D}}\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{B}}) : \nabla\mathbf{u}^{\mathrm{B}} + (1-\phi_{\mathrm{D}})\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{A}}) : \nabla\mathbf{u}^{\mathrm{A}})V^{\mathrm{B}}\right]\right\} \Bigg/ \left\{1 + \frac{p^{\mathrm{p}}+p^n}{2c_{\mathrm{v}}^{\mathrm{t}}p^{\mathrm{p}}}\bar{R}\right\}. \tag{35}$$

The turbulent Prandtl number is defined as $Pr_{\mathrm{t}} \equiv \frac{c_{\mathrm{p}}\mu_{\mathrm{t}}}{K_{\mathrm{t}}}$.

4.4. Pressure

The pressure p^{B} is chosen so that the volume V^{B} from the ideal equation of state and the Lagrangian volume V^{LAG} computed from the movement of the cell faces agree. The ideal equation of state (34), rewritten in terms of the pressure p^{B}, is

$$V^{\mathrm{B}} = \frac{M^{\mathrm{B}}}{p^{\mathrm{B}}}\bar{R}T^{\mathrm{B}}. \tag{36}$$

The Lagrangian volume is

$$V^{\mathrm{LAG}} = V^n + \Delta t\sum_f (\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{B}}. \tag{37}$$

The contribution from pressure p^{B} in (37) can be made evident by deriving an expression for $(\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{B}}$.

Consider a thin slice with volume V_{f} centered around face f. Taking the Lagrangian derivative of the inner product of the momentum of this volume with its normal vector gives

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_{V_{\mathrm{f}}}\rho\mathbf{u}\cdot\mathbf{A}_{\mathrm{f}}\,\mathrm{d}V = \frac{\mathrm{D}}{\mathrm{D}t}\left[\mathbf{A}_{\mathrm{f}}\cdot\int_{V_{\mathrm{f}}}\rho\mathbf{u}\,\mathrm{d}V\right] = \frac{D\mathbf{A}_{\mathrm{f}}}{Dt}\cdot\int_{V_{\mathrm{f}}}\rho\mathbf{u}\,\mathrm{d}V + \mathbf{A}_{\mathrm{f}}\cdot\frac{\mathrm{D}}{\mathrm{D}t}\int_{V_{\mathrm{f}}}\rho\mathbf{u}\,\mathrm{d}V. \tag{38}$$

In the numerical approximation of this relation, it is computationally desirable to start at some intermediate time level, t, and progress to the end of the Lagrangian time step, stage B. Explicitly, this numerical approximation is

$$M_{\mathrm{f}}^{\mathrm{B}}\frac{(\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{B}} - (\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{t}}}{\Delta t} = M_{\mathrm{f}}^{\mathrm{B}}\frac{\mathrm{d}\mathbf{A}_{\mathrm{f}}}{\mathrm{d}t}\cdot\mathbf{u}_{\mathrm{f}}^n + \mathbf{A}_{\mathrm{f}}\cdot\frac{[(\int_{V_{\mathrm{f}}}\rho\mathbf{u}\,\mathrm{d}V)^{\mathrm{B}} - (\int_{V_{\mathrm{f}}}\rho\mathbf{u}\,\mathrm{d}V)^{\mathrm{t}}]}{\Delta t}. \tag{39}$$

The time level t is defined as

$$\left(M_{\mathrm{v}}^{\mathrm{B}}+S_{\mathrm{v}}\right)\frac{\mathbf{u}^{\mathrm{t}}-\mathbf{u}^{\mathrm{A}}}{\Delta t}=-\sum_{c}\left[\phi_{\mathrm{D}}\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{B}})+(1-\phi_{\mathrm{D}})\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{A}})\right]_{\mathrm{c}}\cdot\sum_{f}\chi_{f\mathrm{c}}\mathbf{A}_{f\mathrm{c}}^{n}. \tag{40}$$

One should note that $M_{\mathrm{f}}^{\mathrm{B}}=M_{\mathrm{f}}^{\mathrm{t}}$. Since stage A has already accounted for the spray sources and gravity, the difference in momentum between stage t and B should only include gradients of pressure and turbulent kinetic energy. Therefore, Eq. (39) becomes

$$\begin{aligned}M_{\mathrm{f}}^{\mathrm{B}}\frac{(\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{B}}-(\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{t}}}{\Delta t}&=M_{\mathrm{f}}^{\mathrm{B}}\frac{\mathrm{d}\mathbf{A}_{\mathrm{f}}}{\mathrm{d}t}\cdot\mathbf{u}_{\mathrm{f}}^{n}+\mathbf{A}_{\mathrm{f}}\cdot\left[-\int_{V_{\mathrm{f}}}\nabla\left[\frac{1}{a^{2}}p+A_{o}\frac{2}{3}\rho k\right]\right]\mathrm{d}V\\&\approx M_{\mathrm{f}}^{\mathrm{B}}\frac{\mathrm{d}\mathbf{A}_{\mathrm{f}}}{\mathrm{d}t}\cdot\mathbf{u}_{\mathrm{f}}^{n}-V_{\mathrm{f}}\nabla\left[\frac{\phi_{\mathrm{p}}}{(a^{n})^{2}}p^{\mathrm{B}}+\frac{(1-\phi_{\mathrm{p}})}{(a^{n})^{2}}p^{n}+\frac{2}{3}A_{o}\rho^{\mathrm{A}}k^{\mathrm{A}}\right]_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^{n}.\end{aligned} \tag{41}$$

The quantity $(\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{t}}$ is approximated using

$$(\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{t}}\approx\mathbf{u}_{\mathrm{f}}^{\mathrm{t}}\cdot\mathbf{A}_{\mathrm{f}}^{n}, \tag{42}$$

where $\mathbf{u}_{\mathrm{f}}^{\mathrm{t}}$ is an average of the nodal velocities $\mathbf{u}^{\mathrm{t}}$ forming the cell face. Solving for $(\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{B}}$ in (41) and substituting into (37) yields

$$V^{\mathrm{LAG}}=V^{n}+\Delta t\sum_{f}\left[(\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{t}}+\Delta t\frac{\mathrm{d}\mathbf{A}_{\mathrm{f}}}{\mathrm{d}t}\cdot\mathbf{u}_{\mathrm{f}}^{n}-\frac{\Delta t}{\rho_{\mathrm{f}}^{\mathrm{B}}}\nabla\left[\frac{\phi_{\mathrm{p}}}{(a^{n})^{2}}p^{\mathrm{B}}+\frac{1-\phi_{\mathrm{p}}}{(a^{n})^{2}}p^{n}+\left(\frac{2}{3}A_{o}\rho^{\mathrm{A}}k^{\mathrm{A}}\right)\right]_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^{n}\right], \tag{43}$$

where $\rho_{\mathrm{f}}^{\mathrm{B}}$ is the average of the cell densities bordering the cell face. Geometric coefficients are used to evaluate $\nabla[\frac{\phi_{\mathrm{p}}}{(a^{n})^{2}}p^{\mathrm{B}}+\frac{1-\phi_{\mathrm{p}}}{(a^{n})^{2}}p^{n}+(\frac{2}{3}A_{o}\rho^{\mathrm{A}}k^{\mathrm{A}})]_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^{n}$ in contrast to earlier versions of KIVA which use a surface integral formulation over pressure and turbulent kinetic energy in (41), integrating over the faces of the control volume surrounding the face f. The calculation of $\mathrm{d}\mathbf{A}_{\mathrm{f}}/\mathrm{d}t$ is discussed in Appendix B.

One must now find a p^{B} so that V^{B} and V^{LAG}, computed with (36) and (43), agree. However, the presence of pressure p^{B} in the denominator of V^{B} (36) and in the expression for T^{B} (when p^{B} is substituted for p^{p}) in (35) makes the system of equations $V^{\mathrm{B}}=V^{\mathrm{LAG}}$ nonlinear. These problems are remedied by computing V^{B} using a first order Taylor series

$$V^{\mathrm{B}}(p^{\mathrm{B}},T^{\mathrm{B}}(p^{\mathrm{B}}))\approx V^{\mathrm{B}}(p^{\mathrm{p}},T^{\mathrm{B}}(p^{\mathrm{p}}))+\frac{\mathrm{d}V^{\mathrm{B}}}{\mathrm{d}p^{\mathrm{p}}}(p^{\mathrm{B}}-p^{\mathrm{p}})\equiv V_{\mathrm{c}}^{\mathrm{B}}. \tag{44}$$

Using the equation of state (34) and (35), one can derive an expression for the term $\frac{\mathrm{d}V^{\mathrm{B}}}{\mathrm{d}p^{\mathrm{p}}}$,

$$\frac{\partial V^{\mathrm{B}}}{\partial p^{\mathrm{p}}}=-\frac{1}{\gamma^{\mathrm{t}}}\frac{V^{\mathrm{B}}}{p^{\mathrm{p}}}=-\frac{2c_{\mathrm{v}}^{\mathrm{t}}+\bar{R}\left(1-\frac{V^{n}}{V^{\mathrm{B}}}\right)}{2c_{\mathrm{v}}^{\mathrm{t}}+\frac{p^{n}+p^{\mathrm{p}}}{p^{\mathrm{p}}}\bar{R}}\left(\frac{V^{\mathrm{B}}}{p^{\mathrm{p}}}\right), \tag{45}$$

where V^{B} is evaluated at p^{p} in (45). See Appendix C for details.

Defining $V_{\mathrm{c}}^{\mathrm{B}}$ to be the right-hand side of (44), the equation of state volume approximation at p^{B} gives the following relation

$$V_{\mathrm{c}}^{\mathrm{B}}=V^{\mathrm{B}}-\frac{1}{\gamma^{\mathrm{t}}}\frac{V^{\mathrm{B}}}{p^{\mathrm{p}}}(p^{\mathrm{B}}-p^{\mathrm{p}}). \tag{46}$$

A pressure p^{B} is sought so that $V_{\mathrm{c}}^{\mathrm{B}}$ and V^{LAG}

$$|V_{\mathrm{c}}^{\mathrm{B}}-V^{\mathrm{LAG}}|<tol \tag{47}$$

agree to some tolerance.

4.5. SIMPLE method

The semi-implicit pressure linked equations or SIMPLE method is an iteration over the coupled equations solved in Sections 4.2, 4.3, and 4.4. Eq. (25) is solved for velocity $\mathbf{u}^{\mathrm{B}}$, (35) is solved for temperature T^{B} and (47)

is solved for pressure p^{B}. These equations are solved again in successive SIMPLE iterations. Essentially the form of (25), (35) and (47) remains the same except that p^{B} is substituted for p^{p} in (25) and (35). The SIMPLE iteration proceeds if (47) is not satisfied after some number of iterations or if

$$\max_{\mathrm{c}} \frac{|p_{\mathrm{c}}^{\mathrm{B}} - p_{\mathrm{c}}^{\mathrm{p}}|}{p_{\mathrm{dif}}} > tol_{\mathrm{p}} \tag{48}$$

where $p_{\mathrm{dif}} = \max\{\max_{\mathrm{c}}\{p_{\mathrm{c}}\} - \min_{\mathrm{c}}\{p_{\mathrm{c}}\}, p_{\mathrm{dif}}^{o}\}$ and p_{dif}^{o} is a user-defined constant. Here $\max_{\mathrm{c}}\{p_{\mathrm{c}}\}$ and $\min_{\mathrm{c}}\{p_{\mathrm{c}}\}$ refer to the maximum and minimum pressures over the mesh, respectively. Once (47) and (48) have been satisfied, (25) is updated with the latest pressure p^{B}, $V^{\mathrm{B}} \equiv V_{\mathrm{c}}^{\mathrm{B}}$, the densities are updated $\rho_m^{\mathrm{B}} = \rho_m^{\mathrm{A}} \frac{V^n}{V^{\mathrm{B}}}$ (note that $V^n = V^{\mathrm{A}}$), and I^{B} is computed using V^{B}, p^{B} and T^{B} in (31).

4.6. Turbulence

The turbulence equations are solved after the phase B kinematic quantities have been computed. The turbulent kinetic energy is first updated with the spray contributions,

$$\frac{M^{\mathrm{B}}k^{\mathrm{A}} - M^n k^n}{\Delta t} = \dot{W}^{\mathrm{spray}} V^n. \tag{49}$$

The diffusion terms in the k equation are solved using

$$\begin{aligned}\frac{M^{\mathrm{B}}k^{\mathrm{B}} - M^{\mathrm{B}}k^{\mathrm{A}}}{\Delta t} = &-\frac{2}{3}\rho^{\mathrm{B}}\frac{V^{\mathrm{B}} - V^n}{\Delta t}\left[(1-f)k^n + fk^{\mathrm{B}}\right] + V^n\left[\phi_{\mathrm{D}}\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{B}}) : \nabla\mathbf{u}^{\mathrm{B}} + (1-\phi_{\mathrm{D}})\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{A}}) : \nabla\mathbf{u}^{\mathrm{A}}\right] \\ &+ \sum_f \frac{(\mu_{\mathrm{t}}^n)_{\mathrm{f}}}{Pr_k}\nabla\left[\phi_{\mathrm{D}}k^{\mathrm{B}} + (1-\phi_{\mathrm{D}})k^{\mathrm{A}}\right]_{\mathrm{f}} \cdot \mathbf{A}_{\mathrm{f}}^n - M^{\mathrm{B}}\frac{\epsilon^n}{k^n}k^{\mathrm{B}},\end{aligned} \tag{50}$$

where

$$f = \begin{cases} 1, & V^{\mathrm{B}} - V^n > 0, \\ 0, & V^{\mathrm{B}} - V^n \leqslant 0. \end{cases} \tag{51}$$

Similarly, the turbulence dissipation rate is first updated with the spray contributions,

$$\frac{M^{\mathrm{B}}\epsilon^{\mathrm{A}} - M^n\epsilon^n}{\Delta t} = c_s\dot{W}^{\mathrm{spray}}V^n\frac{\epsilon^{\mathrm{A}}}{k^n}. \tag{52}$$

The diffusion terms in the ϵ equations are solved using

$$\begin{aligned}\frac{M^{\mathrm{B}}\epsilon^{\mathrm{B}} - M^{\mathrm{B}}\epsilon^{\mathrm{A}}}{\Delta t} = &-\left(\frac{2}{3}c_{\epsilon_1} - c_{\epsilon_3}\right)\rho^{\mathrm{B}}\frac{V^{\mathrm{B}} - V^n}{\Delta t}\left[(1-f)\epsilon^n + f\epsilon^{\mathrm{B}}\right] + c_{\epsilon_1}\frac{\epsilon^n}{k^n}V^n\left[\phi_{\mathrm{D}}\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{B}}) : \nabla\mathbf{u}^{\mathrm{B}}\right. \\ &\left. + (1-\phi_{\mathrm{D}})\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{A}}) : \nabla\mathbf{u}^{\mathrm{A}}\right] + \sum_f \frac{(\mu_{\mathrm{t}}^n)_{\mathrm{f}}}{Pr_\epsilon}\nabla\left[\phi_{\mathrm{D}}\epsilon^{\mathrm{B}} + (1-\phi_{\mathrm{D}})\epsilon^{\mathrm{A}}\right]_{\mathrm{f}} \cdot \mathbf{A}_{\mathrm{f}}^n - c_{\epsilon_2}M^{\mathrm{B}}\frac{\epsilon^n}{k^n}\epsilon^{\mathrm{B}}.\end{aligned} \tag{53}$$

In the RNG (ReNormalization Group) k–ϵ equations, the term $c_{\epsilon_1}\frac{\epsilon^n}{k^n}V^n[\phi_{\mathrm{D}}\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{B}}) : \nabla\mathbf{u}^{\mathrm{B}} + (1-\phi_{\mathrm{D}})\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{A}}) : \nabla\mathbf{u}^{\mathrm{A}}]$ in (53) is replaced by

$$(c_{\epsilon_1} - \tilde{\tau})\frac{\epsilon^n}{k^n}V^n\left[\phi_{\mathrm{D}}\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{B}}) : \nabla\mathbf{u}^{\mathrm{B}} + (1-\phi_{\mathrm{D}})\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{A}}) : \nabla\mathbf{u}^{\mathrm{A}}\right], \tag{54}$$

where

$$\tilde{\tau} = \tau\frac{\left(1 - \frac{\tau}{4.38}\right)}{(1 + .012\tau^3)}, \quad \tau = \frac{k^n}{\epsilon^n}\varsigma$$

and

$$\varsigma = \sqrt{\frac{1}{\mu_{\mathrm{t}}^n}\left(\phi_{\mathrm{D}}\boldsymbol{\sigma}^{\mathrm{B}} : \nabla\mathbf{u}^{\mathrm{B}} + (1-\phi_{\mathrm{D}})\boldsymbol{\sigma}^{\mathrm{A}} : \nabla\mathbf{u}^{\mathrm{A}}\right) + \frac{2}{3}(\nabla\cdot\mathbf{u}^{\mathrm{B}})^2}.$$

In addition c_{ϵ_3} is replaced by

$$\begin{cases} .41333 + 0.06899\tau\tilde{\tau} - \hat{\tau}, & \nabla \cdot \mathbf{u}^{\mathrm{B}} \geqslant 0, \\ .41333 - .06899\tau\tilde{\tau} - \hat{\tau}, & \nabla \cdot \mathbf{u}^{\mathrm{B}} < 0, \end{cases} \tag{55}$$

where $\hat{\tau} = \frac{2}{3} c_\mu \tilde{\tau} \nabla \cdot \mathbf{u}^{\mathrm{B}} \frac{k^n}{\epsilon^n}$. The RNG equivalent of the constants c_μ, Pr_k, c_{ϵ_1}, c_{ϵ_2} and Pr_ϵ in (11), (50), (53) and (54) assume different values than in the standard k–ϵ model. See Table 1.

Since one cannot resolve the gradients in velocity and temperature near the wall at large Reynolds numbers due to resolution requirements, wall functions are used to set velocities and modify the cell internal energy for cells adjacent to a wall.

4.7. Conjugate residual method

The coupled system of equations described in Section 4.5 is solved with the SIMPLE algorithm. Individually the Lagrangian equations for stage B quantities, namely species density (10), velocity (25), temperature (35), pressure (47), turbulent kinetic energy (50) and turbulent dissipation rate (53) are solved using the conjugate residual method [12]. We briefly describe the method with the linear system $\mathbf{Bx} = \mathbf{b}$ where $\mathbf{B}$ denotes a matrix.

The solution to the linear system $\mathbf{Bx} = \mathbf{b}$ is obtained by performing the following steps.

1. Calculate the current residual $\mathbf{r}_j = \mathbf{b} - \mathbf{Bx}_j$.
2. Calculate $\mathbf{s}_j = \mathbf{Pr}_j$, where $\mathbf{P}$ is a preconditioning matrix taken to be the Jacobi preconditioning matrix in KIVA
$$p_{ij} = \begin{cases} 1/b_{ij} & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$
Here p_{ij} and b_{ij} denote the entries of the matrices $\mathbf{P}$ and $\mathbf{B}$. Initialize $\mathbf{q}_0 = \mathbf{s}_1$.
3. Orthogonalize $\mathbf{s}_j$ with respect to $\mathbf{q}_{j-1}$,
$$\mathbf{q}_j = \mathbf{s}_j - \kappa_j \mathbf{q}_{j-1} \quad \text{where } \kappa_j = \frac{(\mathbf{PBs}_j \cdot \mathbf{Bq}_{j-1})}{(\mathbf{PBq}_{j-1} \cdot \mathbf{Bq}_{j-1})}.$$
4. Find the component $\delta\mathbf{x}_{j+1}$ of $\mathbf{x} - \mathbf{x}_j$ in the direction of $\mathbf{q}_j$,
$$\delta\mathbf{x}_{j+1} = \alpha_j \mathbf{q}_j, \quad \alpha_j = \frac{(\mathbf{Pr}_j \cdot \mathbf{Bq}_j)}{(\mathbf{PBq}_j \cdot \mathbf{Bq}_j)}.$$
5. Calculate $\mathbf{x}_{j+1} = \mathbf{x}_j + \delta\mathbf{x}_{j+1}$ and return to step 1. The iteration is stopped when $|\delta\mathbf{x}_j|_i < \epsilon_{\mathrm{tol}}(\max\{\mathbf{x}_0\} - \min\{\mathbf{x}_0\})$ for each component i of the vector $\delta\mathbf{x}_j$ where ϵ_{tol} is some user defined tolerance.

5. Fluxing schemes

After stage B, the vertices have moved following a Lagrangian trajectory as evidenced by the governing equations (1)–(6). In stage C, vertices are moved from their stage B locations to their locations at the next global time step, $n + 1$. This entails the calculation of all fluxes relevant to this translation. In calculations consisting of stationary domains the location for all vertices will be the same at the beginning of each time step, and the rezoning step C will simply consist of flux calculations corresponding to the reverse of the Lagrangian B translation. For dynamic calculations, e.g. internal combustion engines, the evolution of all vertices is controlled by the user through a prescribed algorithm aimed at maintaining grid quality at the very least.

When vertices move, cell faces move, and fluxes of mass, internal energy and turbulence quantities need to be computed between cells. The faces of the momentum control volume surrounding a vertex also change and vertices will need to exchange momenta.

5.1. Cell fluxing

There are two types of advection schemes provided in KIVA-3V: partial donor cell differencing (PDC) and quasi-second-order upwind (QSOU) differencing. These are also used in KIVA-4.

Fig. 6 shows a cell whose right face is shown at the Lagrangian vertex locations $\mathbf{x}^{\mathrm{B}}$ and at the new time level $n+1$ locations, $\mathbf{x}^{n+1}$. The other faces of the cell will also be displaced. However, only the displacement of the right face is shown here for simplicity. Define the volume δV_{f} to be equal to the volume between the area of face $A_{\mathrm{f}}(\mathbf{x}^{\mathrm{B}})$ and the area of face $A_{\mathrm{f}}(\mathbf{x}^{n+1})$. By KIVA's convention, if the cell volume increases when moving from the $\mathbf{x}^{\mathrm{B}}$ Lagrangian locations to the $\mathbf{x}^{n+1}$ locations, the volume $\delta V_{\mathrm{f}} > 0$ is considered positive with respect to that cell. The volume change can be computed directly from the Lagrangian vertices $\mathbf{x}^{\mathrm{B}}$ and the new locations $\mathbf{x}^{n+1}$. The volume change can also be computed by observing that

$$\delta V[A_{\mathrm{f}}(\mathbf{x}^{n}) \to A_{\mathrm{f}}(\mathbf{x}^{n+1})] = \delta V[A_{\mathrm{f}}(\mathbf{x}^{n}) \to A_{\mathrm{f}}(\mathbf{x}^{\mathrm{B}})] + \delta V_{\mathrm{f}}, \tag{56}$$

where $\delta V[A_{\mathrm{f}}(\mathbf{x}^{n}) \to A_{\mathrm{f}}(\mathbf{x}^{n+1})]$ is the volume swept by the face moving from the time level n locations to the time level $n+1$ locations and $\delta V[A_{\mathrm{f}}(\mathbf{x}^{n}) \to A_{\mathrm{f}}(\mathbf{x}^{\mathrm{B}})]$ is the volume swept by the face moving from the time level n locations to the Lagrangian locations. Substituting $\delta V[A_{\mathrm{f}}(\mathbf{x}^{n}) \to A_{\mathrm{f}}(\mathbf{x}^{\mathrm{B}})] = (\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{B}}\Delta t$ yields

$$\delta V_{\mathrm{f}} = \delta V[A_{\mathrm{f}}(\mathbf{x}^{n}) \to A_{\mathrm{f}}(\mathbf{x}^{n+1})] - (\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{B}}\Delta t, \tag{57}$$

which is the equation KIVA-4 uses to compute δV_{f}. The term $(\mathbf{u}\cdot\mathbf{A})_{\mathrm{f}}^{\mathrm{B}}$ is computed in Section 4.4 and saved for use in (57). In Fig. 6, the cell will gain mass $\rho\delta V_{\mathrm{f}} > 0$. Similarly, the cell will gain quantities, $\rho q\delta V_{\mathrm{f}} > 0$, where q is specific internal energy, turbulent kinetic energy k or turbulent length scale $L = \frac{k^{\frac{3}{2}}}{\epsilon}$. The turbulent length scale is fluxed rather than ϵ because ϵ generally has steeper gradients.

Let us now discuss cell fluxing exclusively with quantities $\rho q\delta V_{\mathrm{f}}$ since the fluxing of $\rho_m\delta V_{\mathrm{f}}$ is a special case where $q = Y_m$. A key quantity in the flux calculation is the determination of the term $(\rho q)_{\mathrm{f}}$, the value of ρq at the face. PDC and QSOU differ in how $(\rho q)_{\mathrm{f}}$ is approximated. Let $(\rho q)_{\mathrm{c}}$ and $\mathbf{x}_{\mathrm{c}}$ refer to the cell-centered quantity (ρq) and the cell-center respectively for which δV_{f} is positive. (In the case $\delta V_{\mathrm{f}} < 0$, one must merely locate the cell opposite the face for which $\delta V_{\mathrm{f}} > 0$ and compute the fluxes from this opposite cell's perspective.) In addition, let $(\rho q)_{\mathrm{c}_n}$ and $\mathbf{x}_{\mathrm{c}_n}$ refer to the cell-centered quantity and the cell-center of the neighboring cell across face f. See Fig. 7.

For PDC differencing

$$(\rho q)_{\mathrm{f}}^{\mathrm{B}} = \frac{1}{2}(\rho q)_{\mathrm{c}_n}^{\mathrm{B}}(1+\alpha_0+\beta_0 C) + \frac{1}{2}(\rho q)_{\mathrm{c}}^{\mathrm{B}}(1-\alpha_0-\beta_0 C), \tag{58}$$

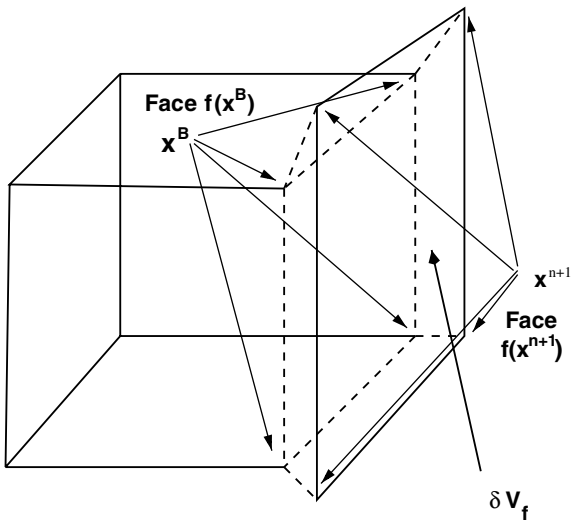


Fig. 6. Cell control volume change.

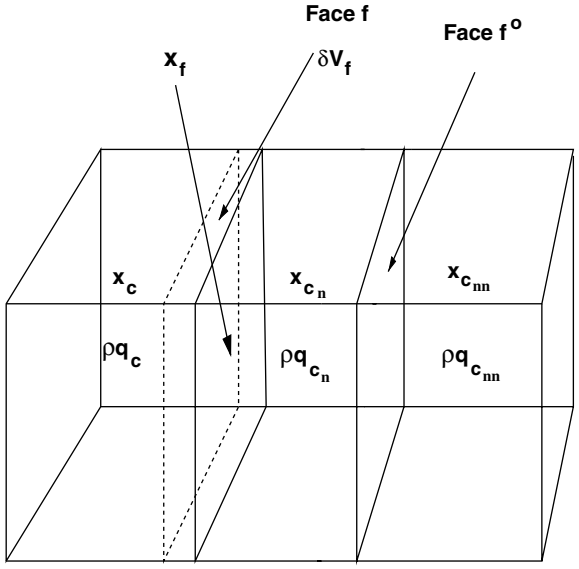


Fig. 7. Neighboring cells for gradient calculation.

where

$$C = \frac{2|\delta V_{\mathrm{f}}|}{V_{\mathrm{c}} + V_{\mathrm{c}_n}} \quad \text{and} \quad 0 \leqslant \alpha_0 + \beta_0 C \leqslant 1. \tag{59}$$

When $\alpha_0 = 1$ and $\beta_0 = 0$, the method defaults to full upwind differencing.

For QSOU differencing a gradient is first computed for the face,

$$\frac{\mathrm{d}(\rho q)}{\mathrm{d}f} = \mathrm{sign}((\rho q)^{\mathrm{B}}_{\mathrm{c}} - (\rho q)^{\mathrm{B}}_{\mathrm{c}_n}) \min\left(\frac{|(\rho q)^{\mathrm{B}}_{\mathrm{c}} - (\rho q)^{\mathrm{B}}_{\mathrm{c}_n}|}{|\mathbf{x}^{n+1}_{\mathrm{c}} - \mathbf{x}^{n+1}_{\mathrm{c}_n}|}, \frac{|(\rho q)^{\mathrm{B}}_{\mathrm{c}_n} - (\rho q)^{\mathrm{B}}_{\mathrm{c}_{nn}}|}{|\mathbf{x}^{n+1}_{\mathrm{c}_n} - \mathbf{x}^{n+1}_{\mathrm{c}_{nn}}|}\right). \tag{60}$$

If $(\rho q)^{\mathrm{B}}_{\mathrm{c}} - (\rho q)^{\mathrm{B}}_{\mathrm{c}_n}$ and $(\rho q)^{\mathrm{B}}_{\mathrm{c}_n} - (\rho q)^{\mathrm{B}}_{\mathrm{c}_{nn}}$ differ in sign, $\frac{\mathrm{d}(\rho q)}{\mathrm{d}f}$ is set to zero. The quantities $(\rho q)^{\mathrm{B}}_{\mathrm{c}_{nn}}$ and $\mathbf{x}_{\mathrm{c}_{nn}}$ refer to the cell-centered quantities across face f^o. See Fig. 7. The face f^o is the face opposite face f in the cell whose cell center is $\mathbf{x}_{\mathrm{c}_n}$. The cell $\mathbf{x}^{n+1}_{\mathrm{c}_{nn}}$ is computed by finding the cell whose angle with $\mathbf{x}^n_{\mathrm{f}} - \mathbf{x}^n_{\mathrm{c}_n}$ is largest,

$$\mathbf{x}^{n+1}_{\mathrm{c}_{nn}} = \min_{\mathbf{x}^n_{\mathrm{c}_j}} \left(\frac{\mathbf{x}^n_{\mathrm{f}} - \mathbf{x}^n_{\mathrm{c}_n}}{|\mathbf{x}^n_{\mathrm{f}} - \mathbf{x}^n_{\mathrm{c}_n}|} \cdot \frac{\mathbf{x}^n_{\mathrm{c}_j} - \mathbf{x}^n_{\mathrm{c}_n}}{|\mathbf{x}^n_{\mathrm{c}_j} - \mathbf{x}^n_{\mathrm{c}_n}|}\right), \tag{61}$$

where $\mathbf{x}^n_{\mathrm{f}}$ is the face center of the face between cells whose cell centers are $\mathbf{x}_{\mathrm{c}}$ and $\mathbf{x}_{\mathrm{c}_n}$. For hexahedral grids, the opposite face f^o can be approximated by using the hexahedral geometry (i.e. the opposite face of a right face is a left face, the opposite face of a bottom face is a top face). Appendix D describes how to improve the calculation of $\frac{\mathrm{d}(\rho q)}{\mathrm{d}f}$. For QSOU differencing, the face value of ρq is computed using

$$(\rho q)^{\mathrm{B}}_{\mathrm{f}} = (\rho q)^{\mathrm{B}}_{\mathrm{c}_n} + \frac{\mathrm{d}(\rho q)}{\mathrm{d}f}|\mathbf{x}^{n+1}_{\mathrm{f}} - \mathbf{x}^{n+1}_{\mathrm{c}_n}|\left(1 - \frac{|\delta V_{\mathrm{f}}|}{V^{\mathrm{B}}_{\mathrm{c}_n}}\right),$$

where $\mathbf{x}^{n+1}_{\mathrm{f}}$ is the center of the face evaluated at time $n+1$. For both PDC and QSOU differencing, $(\rho q)_{\mathrm{f}}\delta V_{\mathrm{f}}$ is exchanged between cells

$$\begin{aligned}(\rho q V)^{n+1}_{\mathrm{c}} &= (\rho q V)^{\mathrm{B}}_{\mathrm{c}} + (\rho q)^{\mathrm{B}}_{\mathrm{f}}\delta V_{\mathrm{f}},\\ (\rho q V)^{n+1}_{\mathrm{c}_n} &= (\rho q V)^{\mathrm{B}}_{\mathrm{c}_n} - (\rho q)^{\mathrm{B}}_{\mathrm{f}}\delta V_{\mathrm{f}}.\end{aligned} \tag{62}$$

Eq. (62) is solved for every non-solid face in the mesh.

5.2. Momentum fluxing

The momentum fluxing calculation must calculate fluxes of momentum through each facet of the vertex control volume surrounding a vertex. For illustration purposes a portion of the vertex control volume described here is shown in Fig. 5. Conservation of momentum is accomplished by exchanging momentum gained and lost through these facets with neighboring vertices. The algorithm used here for momentum fluxing in unstructured grids is based on facets of the vertex control volume surrounding an edge. It departs significantly from the structured algorithm used in previous versions of KIVA. An edge referenced in this section is defined as the line joining two vertices, for instances nodes 1 and 2 in either Fig. 8 or Fig. 9. The momentum facet corresponding to an edge in an element or cell is the quadrilateral area defined by the following four points: the average of the two node locations on the edge, the centers of the two abutting faces to the edge, and the center of the element. See the left side of Fig. 8. The mass change of the momentum facet δM_{e_c} is computed by determining the mass between the momentum facet when the vertices are located at the Lagrangian locations and the momentum facet computed when the vertices are located at the time $n+1$ locations. See the right side of Fig. 8. The mass change δM_{e_c} is taken to be positive with respect to a vertex if the vertex control volume gains mass as the facet moves from the Lagrangian to the time $n+1$ locations (or in other words, the net movement of the momentum facet from the Lagrangian to the time $n+1$ locations is away from the

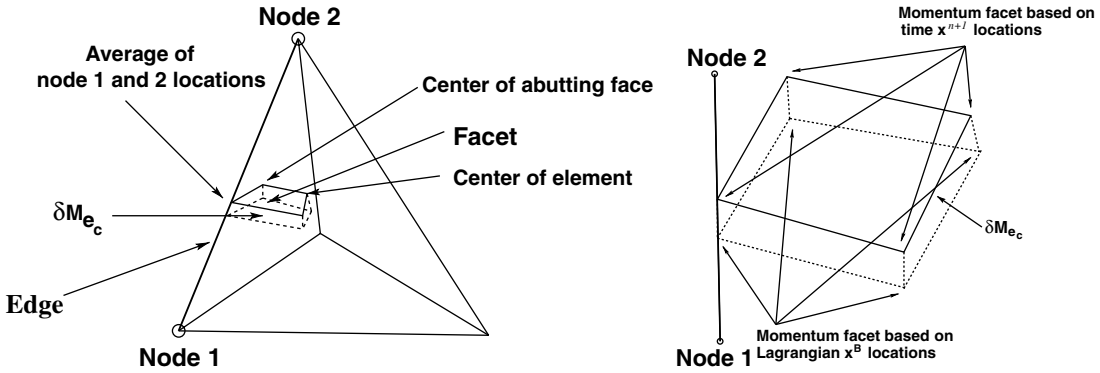


Fig. 8. Change in momentum control volume face in a cell for an edge.

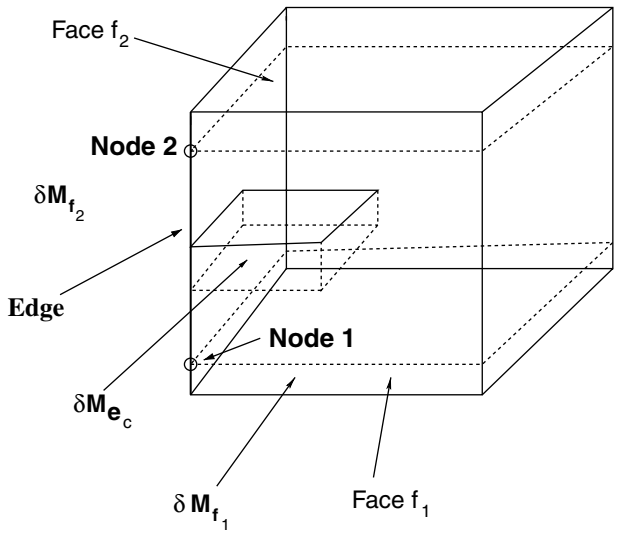


Fig. 9. Change in momentum control volume face in a cell for an edge.

vertex). For example, $\delta M_{e_c} > 0$ for node 1 in Fig. 8. The volume of δM_{e_c} is always a hexahedral volume. The total mass displacement, δM_e, for the momentum facets corresponding to an edge are computed for each non-degenerate edge using

$$\delta M_e = \sum_c \delta M_{e_c} \tag{63}$$

and the mass of the edge, M_e, is computed using

$$M_e = \sum_c \frac{1}{f_{e_c}} M_c, \tag{64}$$

where the sum $\sum_c$ is a sum over all cells sharing the edge. The factor f_{e_c} is computed by first realizing that the edge in question is one of the multiple edges connecting two non-adjacent faces (e.g. f_1 and f_2 in Fig. 9) of a cell. One of the faces may be degenerate. The factor f_{e_c} is the total number of edges connecting the two faces. For hexahedral meshes $f_{e_c} = 4$ and for tetrahedral meshes $f_{e_c} = 3$.

For hexahedral meshes, δM_{e_c} can be approximated using $\delta M_{e_c} = \frac{1}{2}[\frac{1}{4}\delta M_{f_2} + \frac{1}{4}\delta M_{f_1}]$, where δM_{f_i} refers to the change in the mass of cell face i when the cell face moves from its Lagrangian locations to its time $n+1$ locations. See Fig. 9. In Fig. 9, the dashed lines are used to represent the Lagrangian $\mathbf{x}^B$ locations and the solid lines are used to represent the $\mathbf{x}^{n+1}$ locations. The quantities δM_{f_1} and δM_{f_2} are considered positive with respect to (say node 1) if the net movement of their faces is in the direction of $\mathbf{x}_2 - \mathbf{x}_1$, where $\mathbf{x}_2$ is the location of the other node forming the edge. For example, $\delta M_{f_1} < 0$ and $\delta M_{f_2} > 0$ for node 1 in Fig. 9.

Besides the mass associated with the fluxing, the velocity $\mathbf{u}_e$ needs to be provided to complete the momentum calculation. This velocity is representative of all non-degenerate momentum facets corresponding to an edge and its mass displacement δM_e. The PDC and QSOU schemes differ in how the $\mathbf{u}_e$ in the mass flux for an edge is approximated. For both schemes, choose node 1 to be the node such that $\delta M_e > 0$ with respect to its location. For partial donor cell differencing (PDC),

$$\mathbf{u}_e = \frac{1}{2}\mathbf{u}_2^B(1 + \alpha_0 + \beta_0 C') + \frac{1}{2}\mathbf{u}_1^B(1 - \alpha_0 - \beta_0 C'), \tag{65}$$

where

$$C' = \frac{|\delta M_e|}{M_e}, \quad 0 \leqslant \alpha_o + \beta_o C \leqslant 1. \tag{66}$$

For the quasi-second-order upwind (QSOU) differencing scheme, first find the vertex $\mathbf{x}_3^{n+1}$ connected to $\mathbf{x}_2^{n+1}$ whose angle with the vector $\mathbf{x}_1^{n+1} - \mathbf{x}_2^{n+1}$ is largest,

$$\mathbf{x}_3^{n+1} = \min_{\mathbf{x}_v^{n+1}} \left(\frac{\mathbf{x}_1^{n+1} - \mathbf{x}_2^{n+1}}{|\mathbf{x}_1^{n+1} - \mathbf{x}_2^{n+1}|} \cdot \frac{\mathbf{x}_v^{n+1} - \mathbf{x}_2^{n+1}}{|\mathbf{x}_v^{n+1} - \mathbf{x}_2^{n+1}|} \right). \tag{67}$$

Compute for each component (u) of velocity $\mathbf{u}$,

$$\frac{\mathrm{d}u}{\mathrm{d}s} = \begin{cases} \operatorname{sign}(u_1^B - u_2^B) \min\left(\frac{|u_1^B - u_2^B|}{|\mathbf{x}_1^{n+1} - \mathbf{x}_2^{n+1}|}, \frac{|u_2^B - u_3^B|}{|\mathbf{x}_2^{n+1} - \mathbf{x}_3^{n+1}|} \right) & \text{if } (u_1^B - u_2^B)(u_2^B - u_3^B) > 0, \\ 0 & \text{if } (u_1^B - u_2^B)(u_2^B - u_3^B) < 0, \end{cases} \tag{68}$$

where u_3^B refers to a component of velocity at node 3. Appendix D describes (at additional cost) how to improve how $\frac{\mathrm{d}u}{\mathrm{d}s}$ is computed. One component of the edge velocity is then computed using

$$u_e = u_2^B + \left(\frac{\mathrm{d}u}{\mathrm{d}s} \right) \frac{|\mathbf{x}_2^{n+1} - \mathbf{x}_1^{n+1}|}{2} \left| 1 - \frac{|\delta M_e|}{M_e} \right|.$$

Then for both the partial donor cell differencing and QSOU, momentum is exchanged between the two vertices using,

$$\begin{aligned} (\mathbf{u}M_v)_1^{n+1} &= (\mathbf{u}M_v)_1^B + \mathbf{u}_e \delta M_e, \\ (\mathbf{u}M_v)_2^{n+1} &= (\mathbf{u}M_v)_2^B - \mathbf{u}_e \delta M_e. \end{aligned} \tag{69}$$

5.3. Subcycling of fluxing schemes

The cell fluxing and momentum fluxing equations are subcycled. Specifically a smaller time step Δt_c is used in solving the equations in Sections 5.1 and 5.2 where

$$\Delta t_c \leqslant f_{con}\Delta t^n \min_f \left|\frac{V_c}{\delta V_f}\right| \quad \text{and} \quad n\Delta t_c = \Delta t^n \tag{70}$$

n is an integer, $f_{con} = 0.2$ and the minimum is a minimum over all faces. The subcycled time step is chosen so that the Courant stability condition (in 1D $u\Delta t_c/\Delta x < 1$ is satisfied). Equations in Sections 5.1 and 5.2 are cycled through n times.

6. Summary of equations

KIVA-4 solves the equations in the order KIVA-3V solves them. Eq. (10) is first solved. Then (25), (35) and (47) are solved in order and re-solved in the SIMPLE iteration. Once the pressure criteria is satisfied (48), the turbulence equations (50) and (53) are solved. Vertices are subsequently moved to their time $n + 1$ locations. The cell fluxing and momentum fluxing equations in Sections 5.1 and 5.2 are solved. Then temperature is computed from the fluxed internal energy and pressure is computed from the ideal equation of state and a new cycle is started. The localization of a particle in an unstructured grid and accuracy time step constraints are discussed in Appendices E and F, respectively.

7. Results

To test KIVA-4's new capabilities with unstructured meshing, various examples are presented in this section beginning with the simple 1-D diffusion problem and ending with a full scale engine simulation. KIVA-4 uses a first-order discretization in time which means the temporal error scales as $C(\Delta t)$. The spatial discretization ranges from first- to second-order (Error $\propto (\Delta x)^r$, $1 \leqslant r \leqslant 2$) depending on the grid used and the smoothness of the fields to be fluxed. Specific results confirm these order of accuracy expectations. QSOU differencing is used in all the calculations.

7.1. Diffusion

We test KIVA-4's diffusion accuracy using a one-dimensional diffusion problem $\frac{\partial Y}{\partial t} = \frac{\partial^2 Y}{\partial x^2}$ with boundary conditions $\frac{\partial Y}{\partial x}|_0 = \frac{\partial Y}{\partial x}|_{10} = 0$ on a 10 cm bar with initial mass fraction profile

$$Y(x, t = 0) = \frac{1}{2}\left[1 + \cos\left(\frac{\pi x}{10}\right)\right] \tag{71}$$

which evolves according to the analytical solution

$$Y(x, t) = \frac{1}{2}\left[1 + \cos\left(\frac{\pi x}{10}\right)\exp\left(\frac{-\pi^2 t}{100}\right)\right]. \tag{72}$$

The error is computed with

$$\sqrt{\Delta x \sum_c (Y_c - Y_e)^2}, \tag{73}$$

where Y_c is the numerical mass fraction, Y_e is the exact mass fraction, the sum $\sum_c$ is over all cells and Δx is the cell dimension in centimeters. The error is evaluated at a time of $t = 10$ s. The left and right sides of Fig. 10 show the temporal and spatial convergence, respectively, for both hexahedral and tetrahedral grids.

The temporal convergence test was run with a cell size of .05 cm for hexahedra and a cell size of 6.25×10^{-3} cm for tetrahedra. Cell sizes were determined by finding the minimum cell edge length aligned with the direction of diffusing mass. For example in Fig. 11, the bar length is 10 cm and the cell dimension is 1 cm. These cell sizes were chosen to make the spatial error insignificant when compared to the temporal error. Thus

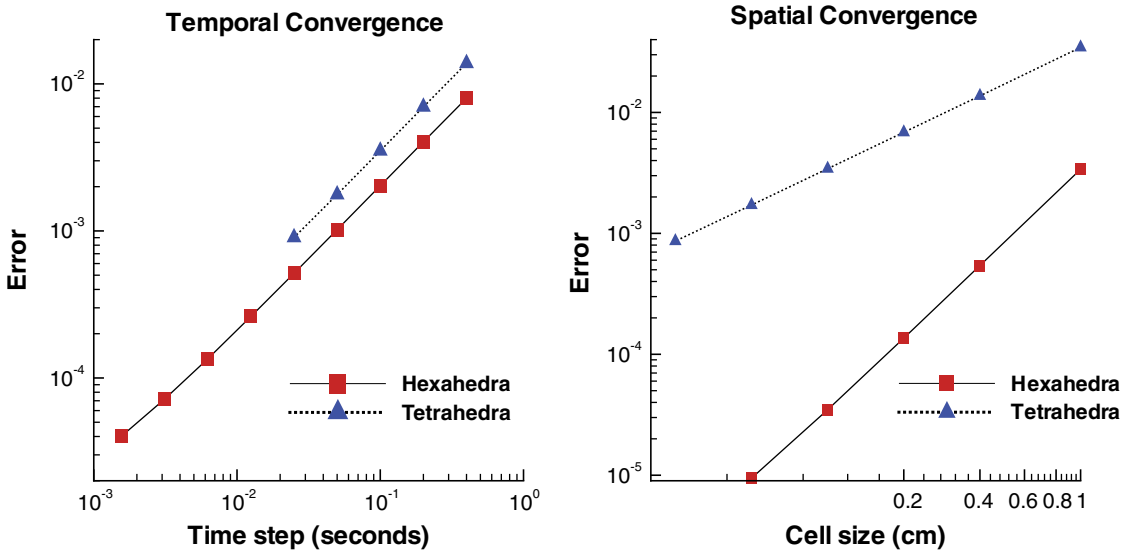


Fig. 10. Temporal convergence (left) and spatial convergence (right) for 1D diffusion problem on a log–log plot.

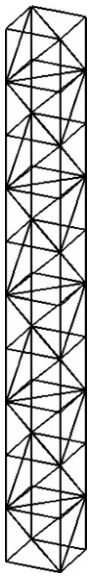


Fig. 11. Edges of tetrahedral grid with cell size of 1 cm in the diffusion problem.

while the left side of Fig. 10 has both spatial and temporal errors, the plot is mostly a reflection of the temporal error. The spatial convergence test was run with a time step of 4.883×10^{-5} s for hexahedra and a time step of 3.125×10^{-3} s for tetrahedra. These time steps were chosen to make the temporal error insignificant when compared to the spatial error. A larger time step can be used for tetrahedra because the spatial errors are larger for tetrahedra. The slope of the temporal convergence least-squares fit line is 0.96 for hexahedra and 0.98 for tetrahedra showing that the order of temporal convergence is 1. The slope of the spatial convergence least-squares fit line is 1.97 for hexahedra and 1.0 for tetrahedra showing that the order of spatial convergence is 2 for hexahedra and 1 for tetrahedra. The cross-section of the tetrahedral grid is scaled as the number of cells increase to maintain aspect ratios between tetrahedral edges.

7.2. Cell fluxing

KIVA-4's cell fluxing algorithm in Section 5.1 is tested by rotating a two-dimensional notched square and a two-dimensional translated Gaussian distribution $Y = \exp(-\frac{1}{2}(x^2 + (y-1)^2))$ a full 360° in a disc. These initial conditions are shown in Fig. 12. The dark and light regions are composed of two different species with the same molecular weight. The velocity is prescribed to be a simple rotation. The radius of the circle is 5 cm. Hexahedral, prism, tetrahedral and pyramidal grids are used. The hexahedral grids used are unstructured O-grids. The results of the rotation for different mesh resolutions for the prism grids are shown in Figs. 13 and 14.

The two-dimensional computation (in the x–y plane) was performed with three-dimensional elements. All cell edges in the hexahedral and prism meshes lie either in the x–y plane or purely in the z-direction. Cell edge lengths that lie in the z-direction are equal to the disc thickness for hexahedral and prism meshes. In the tetrahedral and pyramidal grid, many cell edges contain non-zero components in all three coordinate directions. Regardless of the mesh type, the cell dimension used in Figs. 15 and 16 was computed using $\sqrt{(\text{area of circle})/(\text{number of cells})}$. The error is computed using

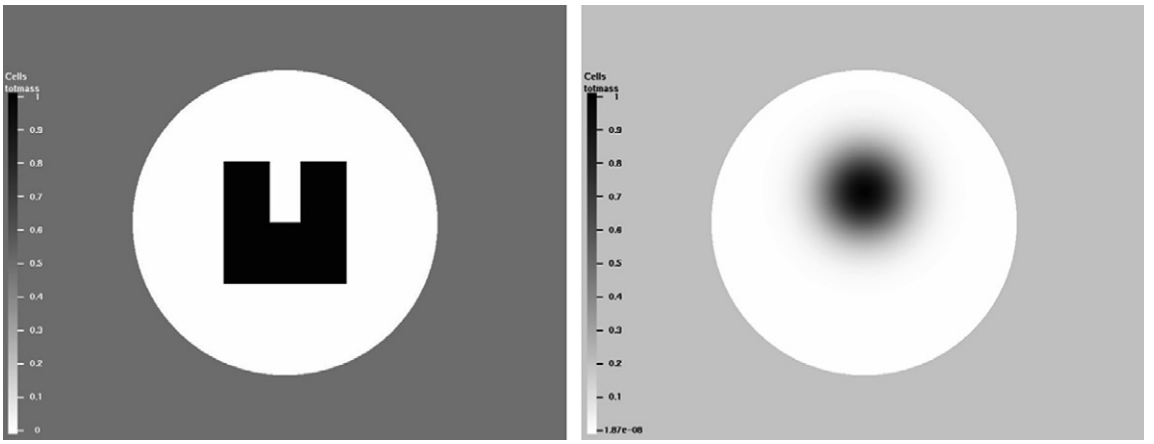


Fig. 12. Notched square (left) and translated Gaussian (right) used as initial conditions for cell-centered advection test.

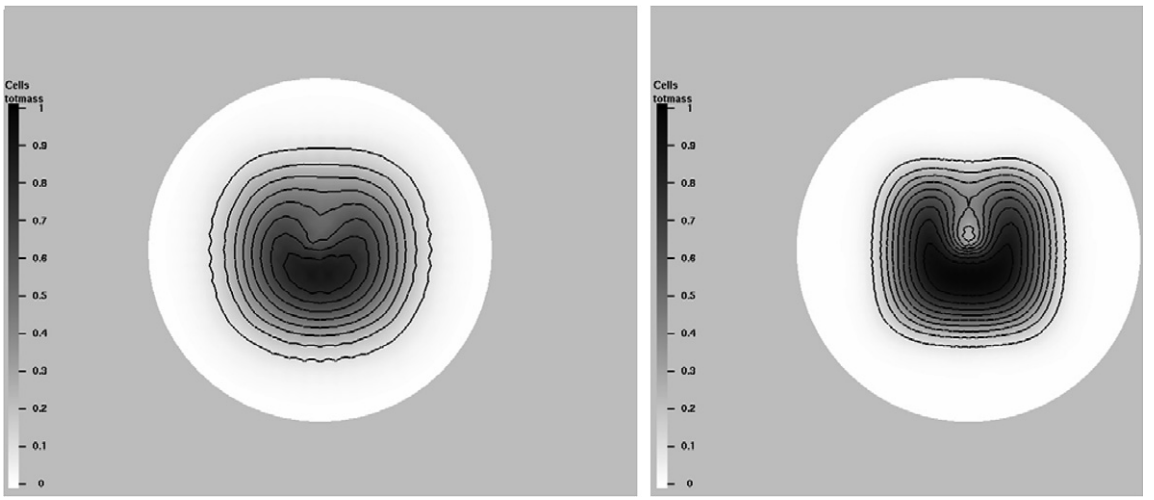


Fig. 13. Rotated notched square with 1456 prism cells (left) and 6000 prism cells (right).

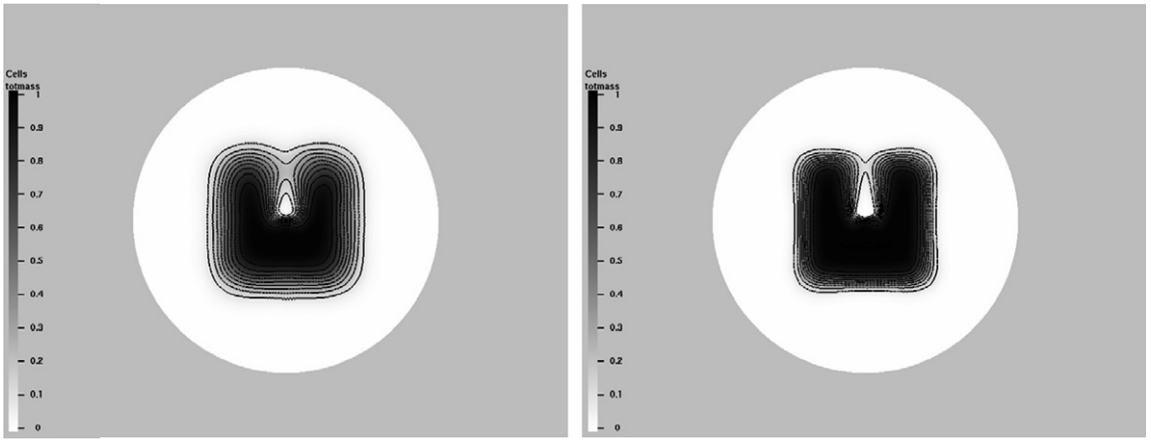


Fig. 14. Rotated notched square with 23,296 prism cells (left) and 88,000 prism cells (right).

$$\frac{\sqrt{\left(\sum_c \left[(\rho_{m_c} - \rho^o_{m_c})V_c\right]^2\right)}}{\sum_c \rho^o_{m_c} V_c},$$

where ρ_{m_c} is the species density of the dark material within a cell after the 360° rotation, $\rho^o_{m_c}$ is the initial species density of the dark material, $\sum_c$ is a sum over all cells, and V_c is the volume of each cell. Computations were run with a time step equivalent to a Courant number of approximately $\frac{1}{5}$. The spatial convergence results of the rotation at different mesh resolutions are shown in Figs. 15 and 16. While there is some spread in the least squares slope of the lines, the convergence appears to be 2nd order for the smooth Gaussian distribution and 1st order for the discontinuous notched square.

7.3. Shock tube

We perform the shock tube problem whose analytical solution is described by Harlow and Amsden [9]. The problem is performed in a 20 cm × .8 cm × .8 cm domain. Adiabatic boundaries conditions are imposed on the

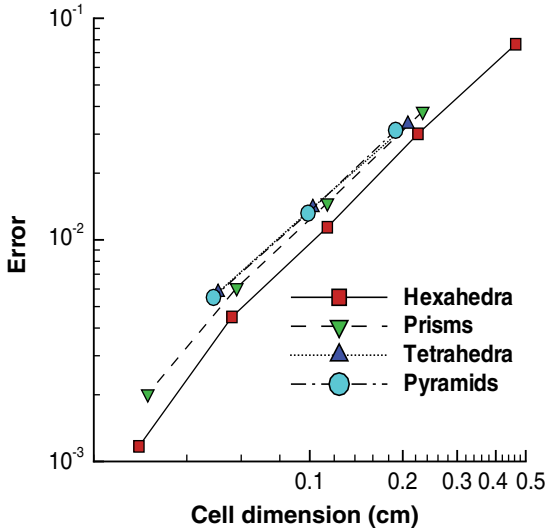


Fig. 15. Spatial convergence test on a log–log plot for the notched square using hexahedra (slope = 1.46), prisms (slope = 1.42), tetrahedra (slope = 1.24), and pyramids (slope = 1.28).

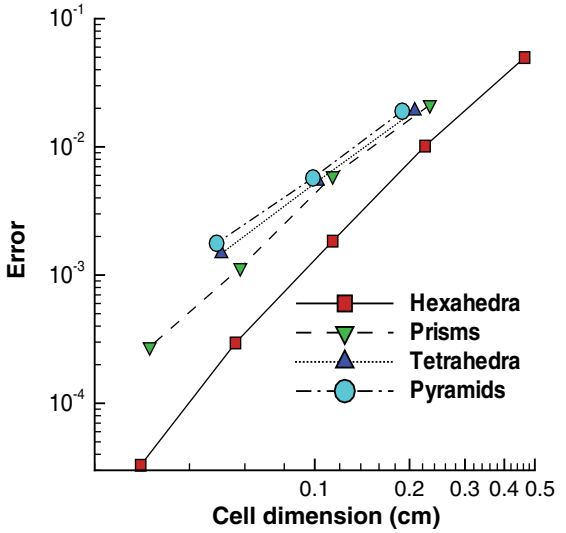


Fig. 16. Spatial convergence test on a log–log plot for the Gaussian distribution using hexahedra (slope = 2.60), prisms (slope = 2.15), tetrahedra (slope = 1.81), and pyramids (slope = 1.75).

temperature. The initial density is .2 g/cm^3 in $0 \leqslant x \leqslant 10$ cm and 0.1 g/cm^3 from 10 cm $\leqslant x \leqslant 20$ cm. The initial pressure is 1.78×10^8 dyn/cm^2 from $0 \leqslant x \leqslant 10$ cm and 8.9×10^7 dyn/cm^2 from 10 cm $\leqslant x \leqslant 20$ cm. The initial temperature is 428 K in the entire domain. Free-slip boundaries conditions are used. The time step used was 4×10^{-7} s and the calculation ran for 450 cycles up to a time of 1.8×10^{-4} s. The initial ratio of specific heats was 1.39. A shock and contact discontinuity propagate toward the right and a rarefaction wave propagates toward the left. Note that while the problem is one-dimensional, calculations are performed in a three-dimensional mesh. Portions of the grids are shown in Fig. 17. In the hexahedral mesh, all cells are oriented advantageously to resolve the shock in contrast to the tetrahedral mesh where all cells could not be aligned to resolve the shock. Analytical and numerical results are compared in Fig. 18. All densities are plotted versus their x-coordinate, irrespective of their y- and z-coordinates. The hexahedral and tetrahedral meshes both use 8000 cells. The hexahedral mesh does a better job of resolving the discontinuities in density. The tetrahedral mesh overshoots the analytical solution at about 17 cm and smooths out the discontinuities. The hexahedral and tetrahedral calculation took 179 s on one processor of a Linux cluster using a Pentium IV XEON 2.2 GHz processor.

7.4. Driven-cavity problem

Computations are performed in a 1 cm × 1 cm box. No-slip boundary conditions are imposed. At the top boundary ($y = 1$ cm), the velocity in the x-direction is forced to move at 1 cm/s. Calculations were performed at a Reynolds number of 1000 with a 40,000 cell prism mesh until a steady-state solution was obtained. The left side of Fig. 19 shows contours of vorticity at a time of 36.25 s and the right side compares the steady-state

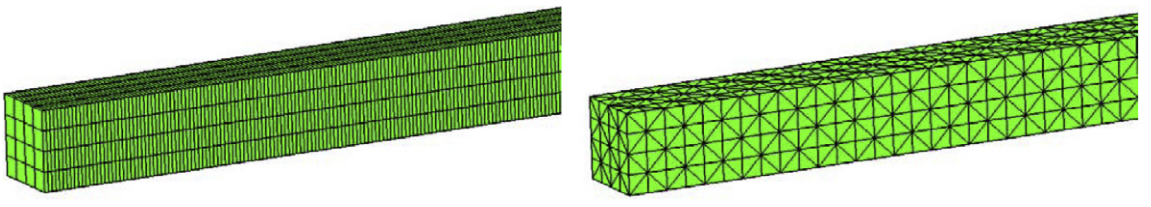
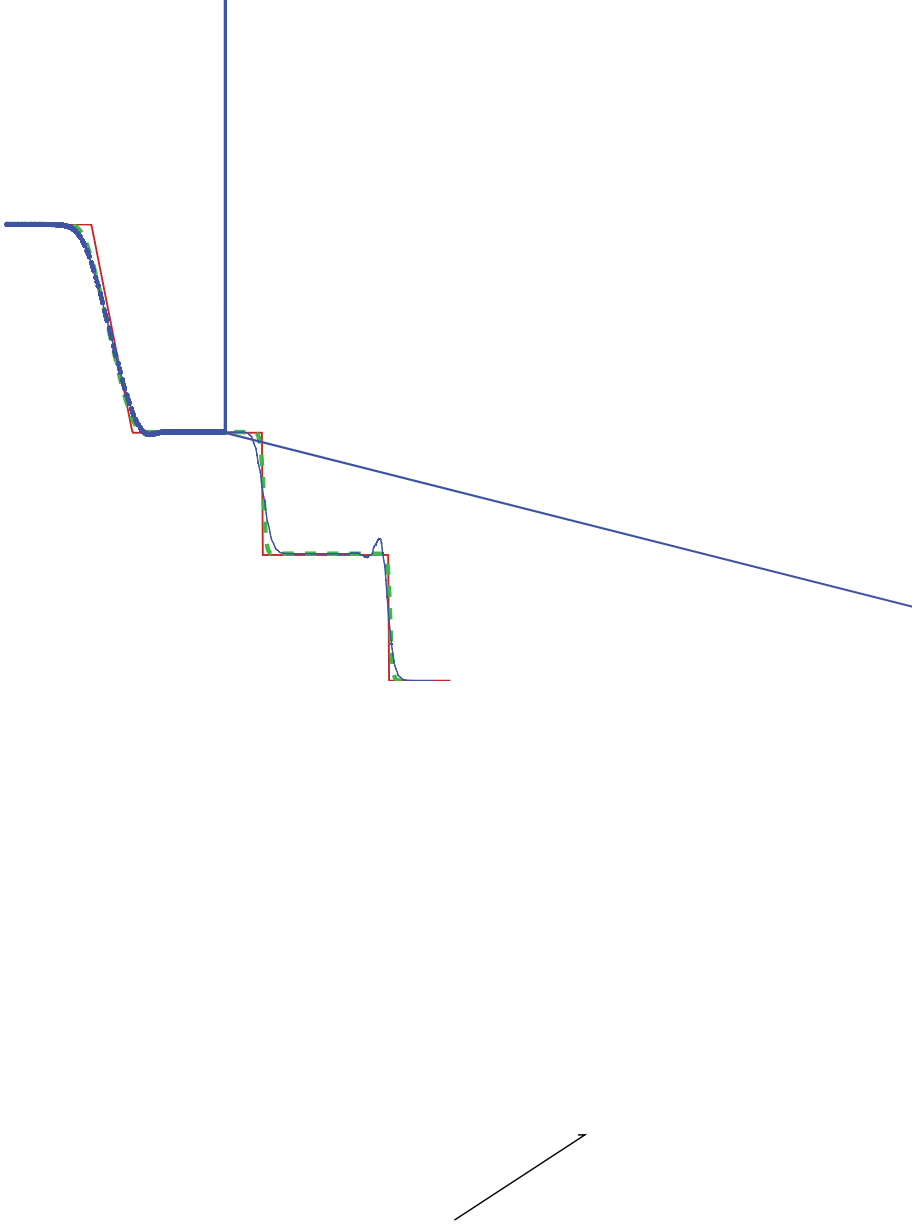


Fig. 17. Portion of hexahedral (left) and tetrahedral (right) meshes used in shock tube problem.



velocity in the x-direction (u velocity) at $x = .5$ cm and the y-direction velocity (v velocity) at $y = .5$ cm with the results of Ghia [8].

7.5. Compression test

A cylinder with a bore of 10 cm and a height of 10 cm is compressed adiabatically to a compression ratio of 40 and expanded back to its original state. The analytical solution and numerical error are shown in Fig. 20. The numerical error is computed using $\frac{100|p_c - p_e|}{p_e}$ for hexahedral (6720 cells), prism (6720 cells) and tetrahedral (6480 cells) grids. The term p_c refers to the average computed pressure and p_e refers to the exact analytical

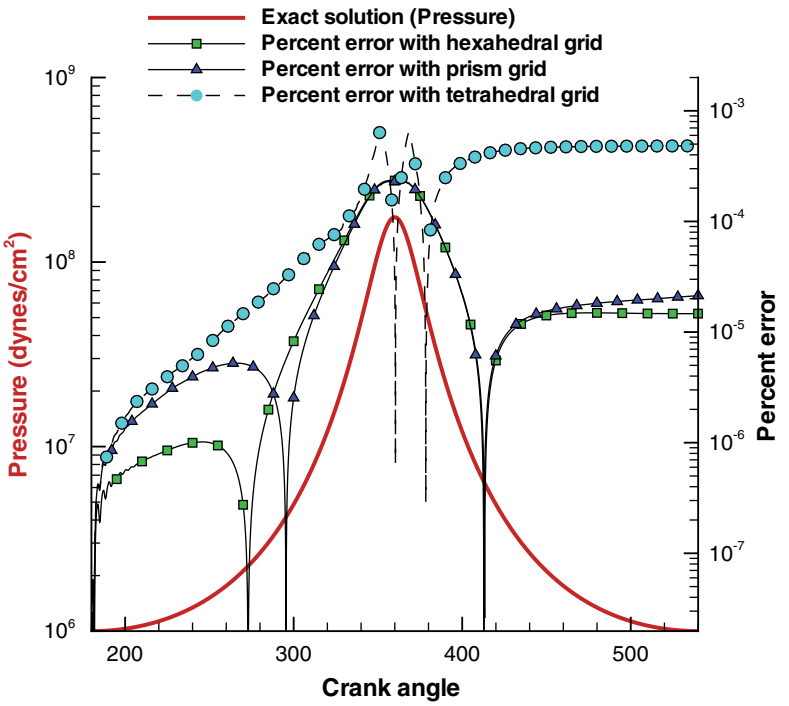


Fig. 20. Pressure history and error history of a compression–expansion stroke for unstructured hexahedral, prism, and tetrahedral meshes.

pressure. All grids (hexahedral, prism and tetrahedral) do an excellent job at matching the exact solution. The exact pressure p_e was calculated using

$$p_e = p_i \left(\frac{V_i}{V} \right)^{\gamma},$$

where $V_i = 250\pi$ cm^3 is the initial volume, V is the volume of the cylinder at a later time, $p_i = 1 \times 10^6$ dyn/cm^2 is the initial pressure and $\gamma = 1.4$ is the ratio of specific heats. The exact pressure is an accurate analytical solution since the sound speed is large compared to the piston speed. At the highest compression $\frac{V_i}{V} = 40$. During compression and expansion, the z-coordinate (the coordinate parallel to the cylinder axis) of nodes is scaled so nodes retain the same relative position between the piston crown and cylinder head. The gas used was nitrogen. Normally γ is a function of temperature, but (for this simulation) the enthalpies of nitrogen in the numerical solution were chosen so that γ remained a constant regardless of temperature so an analytical comparison could be made. The hexahedral mesh took 1381 s and the prism mesh took 1710 s. The tetrahedral mesh took much longer (18,225 s) due to the highly distorted tetrahedra that are created when the cylinder volume is compressed. All calculations were performed on the Pentium IV 2.2 GHz processor.

7.6. Gresho problem

We perform the Gresho problem to test our momentum fluxing algorithm. In the problem, an inviscid vortex is allowed to spin in a two-dimensional 1 cm × 1 cm box. Free-slip boundary conditions are imposed. Since it is expected that the numerics will incur some level of dissipation, the quality of the numerical method is determined by how much of the initial kinetic energy is retained. The problem is initialized with a circumferential velocity

$$u_\theta = \begin{cases} 5r, & 0 < r < 0.2, \\ 2 - 5r, & 0.2 \leqslant r < 0.4, \\ 0, & 0.4 \leqslant r, \end{cases} \tag{74}$$

where r is the radial distance. See Fig. 21 for the initial velocity vectors and velocity magnitude contours. The calculation is run to a time of 3.0 s which is about 2.4 rotations. Fig. 22 shows a plot of the velocity vectors and velocity magnitude in a 2000 cell tetrahedral grid at 3.0 s. The amount of kinetic energy retained as a function of the number of cells used in the mesh is plotted for hexahedral, prism, pyramidal, and tetrahedral meshes in Fig. 23. The fraction of kinetic energy retained increases as one increases the mesh resolution.

However, there is a dramatic increase in kinetic energy dissipation when transitioning from hexahedral meshes to prism, pyramidal and tetrahedral meshes. This dissipation is due to the KIVA-4 staggering of variables as described in Section 3. Velocity is not co-located at cell-centers with all other field variables (density, pressure, temperature). This staggering makes the Gresho problem stiff for meshes with degenerate elements and leads to increased dissipation. A solution is to co-locate all variables, either at cell-centers or at nodes. For example, O'Rourke and Sahota [13] perform the same problem and report a retention of .74 with a hexahedral grid (400 cells) and .63 with a tetrahedral grid (2400 cells) with a co-located nodal scheme. These values are also plotted on Fig. 23.

7.7. 3D engine with vertical valves

We simulate a 3D engine with vertical valves. While no analytical solution exists, we can compare with KIVA-3V. All features of a full engine calculation (without combustion) are included: spray, wall film, valve and piston movement. In addition, a procedure known as snapping is used in the calculation. Snapping involves removing or adding layers of cells when the piston is moving up and down. Snapping also exchanges which layers of cells serve as the solid valve surfaces. One should note the improved geometry around the port and cylinder perimeters in the unstructured hexahedral grid in Fig. 24 compared to the structured hexahedral grid which has four nearly triangular cells around each of its ports and cylinder. The lower left runner is a pressure inflow boundary whose outermost face is held at 2 bars. The concentrations of the species that enter

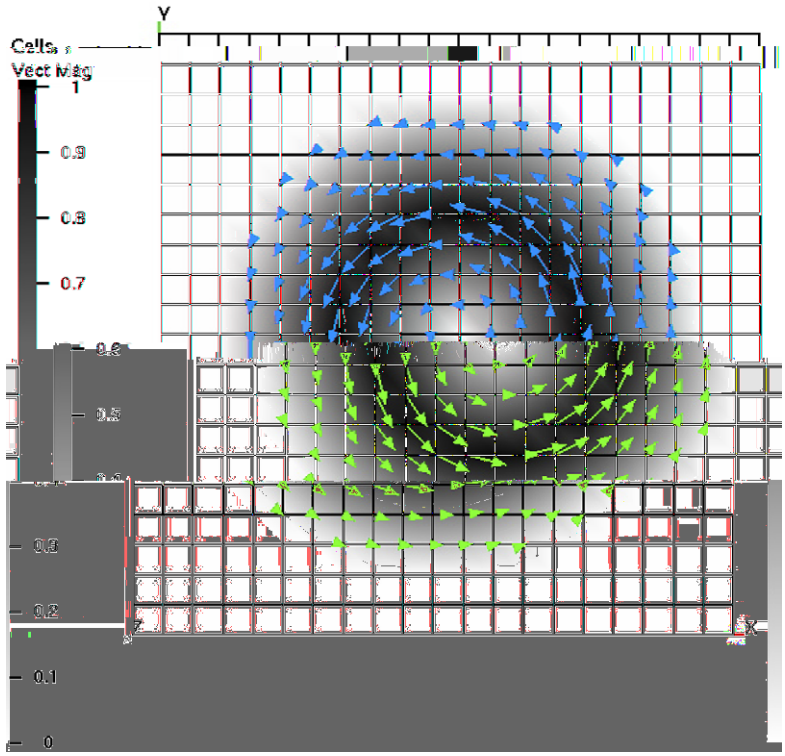


Fig. 21. Initial plot of velocity vectors and velocity magnitude for the Gresho problem.

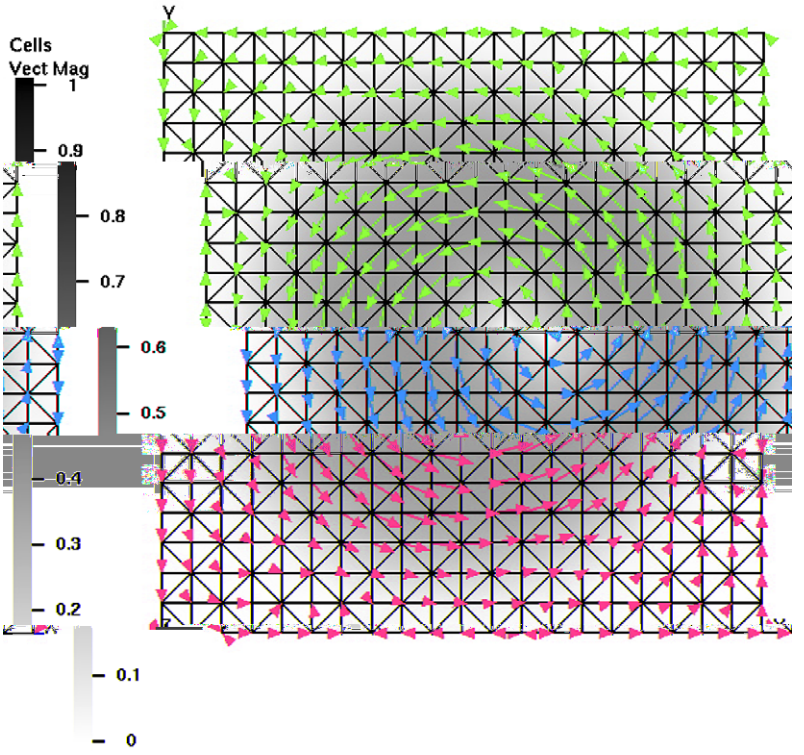


Fig. 22. Plot of velocity vectors and velocity magnitude for the Gresho problem at 3.0 s using a 2000 cell tetrahedral grid.

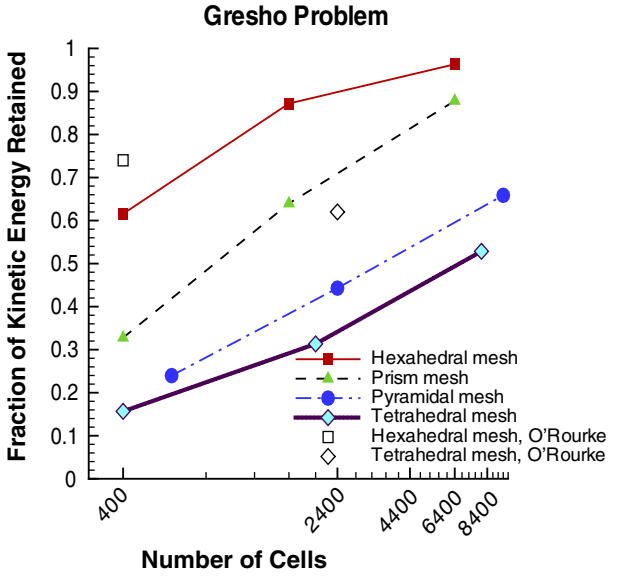


Fig. 23. Initial fraction of kinetic energy retained as a function of the number of cells used in a mesh for hexahedral, prism, pyramidal and tetrahedral meshes. O'Rourke and Sahota's results are also plotted.

the inflow boundary are 11% iso-octane, 19% oxygen and 70% nitrogen by mass. The ambient density is computed using $\rho_{\text{ambient}} = \frac{p_{\text{ambient}} W_{\text{avg}}}{R_o T_{\text{ambient}}}$ where $p_{\text{ambient}} = 1 \times 10^6 \frac{\text{dyn}}{\text{cm}^2}$, $T_{\text{ambient}} = 300$ K and $R_o = 8.3143 \times 10^7 \frac{\text{erg}}{\text{mole K}}$. The ambient density is then isentropically compressed to a density at the inflow boundary using

$$\rho_{\text{inflow}} = \rho_{\text{ambient}} \left(\frac{p_{\text{cell}}}{p_{\text{ambient}}} \right)^{\frac{1}{\gamma}},$$

where p_{cell} is the pressure of the cell bordering the pressure inflow boundary. An inflow temperature is also computed at the pressure inflow boundary using $T_{\text{inflow}} = \frac{p_{\text{cell}} W_{\text{avg}}}{\rho_{\text{inflow}} R}$. Values of turbulent kinetic energy and length scale are specified to be 100 $\frac{\text{cm}^2}{\text{s}^2}$ and 1 cm, respectively, at the pressure inflow boundary. The upper right runner is a pressure outflow boundary whose outermost face is held at 1 bar. The left runner and vertical port are initially filled with a fuel (11% iso-octane, 19% oxygen and 70% nitrogen by mass) held initially at 1 bar and 300 K. The cylinder (bore = 14 cm), right runner and right port are initially filled with mostly air (1%

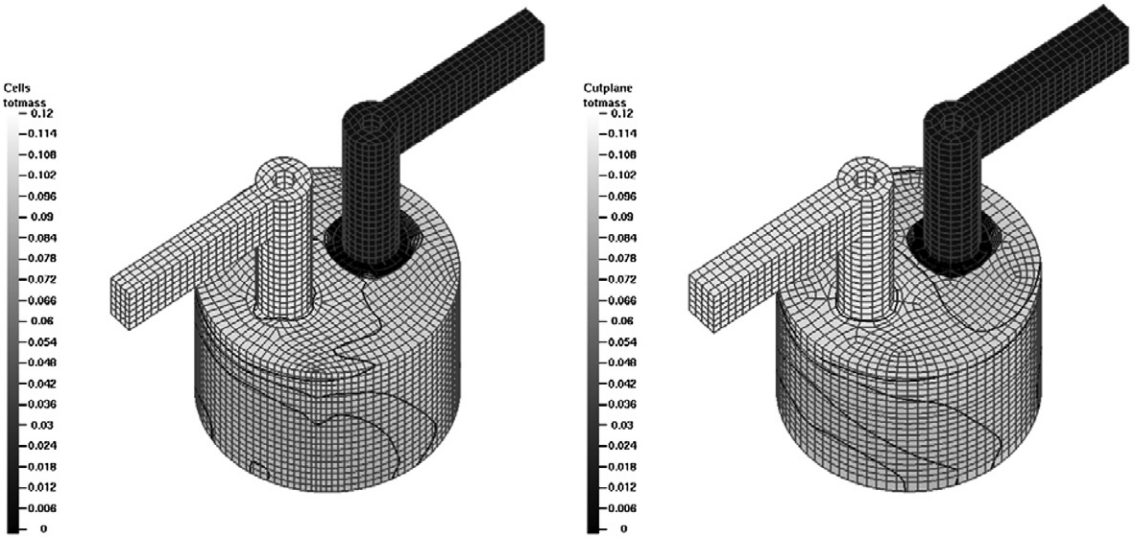


Fig. 24. Fuel contours in a structured grid (left) computed with KIVA-3V and fuel contours in an unstructured grid (right) computed with KIVA-4.

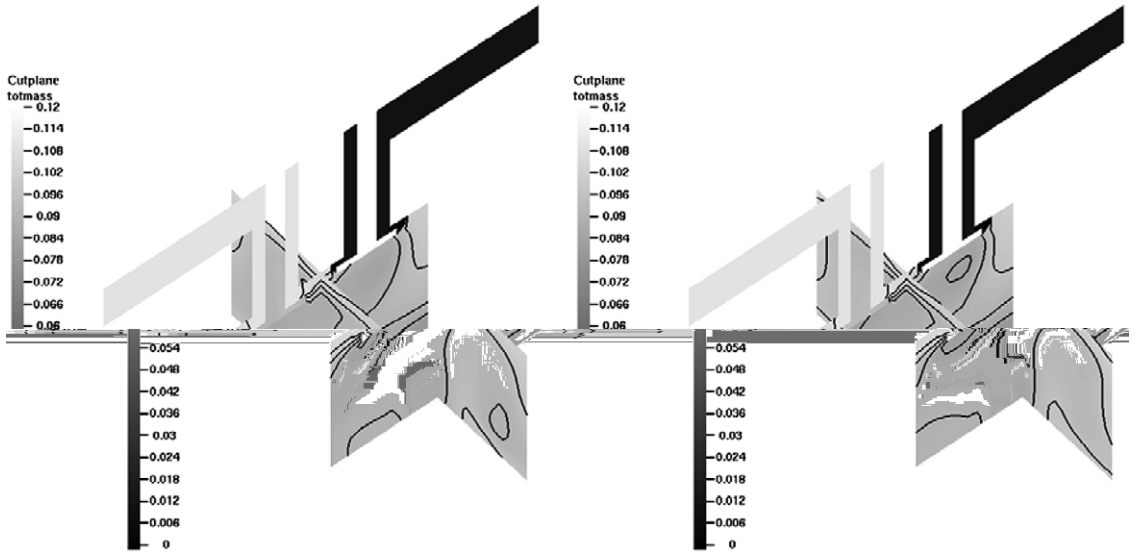


Fig. 25. Cross-sectional fuel contours in structured grid (left) computed with KIVA-3V and fuel contours in unstructured grid (right) computed with KIVA-4.

iso-octane, 29% oxygen and 70% nitrogen by mass) at 300 K and 1 bar. The engine speed is 1000 RPM. 2000 iso-octane fuel spray particles (whose radii are 37.9 μm and total mass is 14.4 mg) are injected at 0 crank angle degrees (CAD) for 10 CAD with a 74° hollow cone angle. In our CAD convention, 0 CAD corresponds to the highest point of the piston and 180 CAD corresponds to the lowest point of the piston. The structured grid has 18,080 cells and 21,021 nodes. The unstructured hexahedral grid has 16,000 cells and 18,349 nodes. The calculation is run from 0 CAD to 180 CAD. KIVA-3V required 1090 s and 1038 cycles to perform the calculation on the structured grid. KIVA-4 required 1362 s and 1009 cycles to perform the calculation on the structured grid. Thus KIVA-4 is 25% slower than KIVA-3V in the structured 3D engine calculation. Much of the reduced computational efficiency (20%) is due to the fact that KIVA-4 uses Fortran 90 modules and dynamic memory allocation. The unstructured engine calculation required 1069 s to run using KIVA-4. The unstructured calculation also benefits from the improved geometry of its mesh by eliminating the near triangular cells on the perimeter of its cylinder and ports.

Figs. 24 and 25 show the fuel contours from these calculations at 180 CAD. The fuel distribution compares reasonably well in both simulations despite some differences in the fuel contours.

8. Conclusion

In this new version of KIVA, the solution of the transport equations coupled with sprays and chemical reactions has been implemented in an unstructured grid framework capable of handling hexahedra, prisms, tetrahedra, and pyramids. This required modifications to the Lagrangian stage of the calculations for all geometrical dependent quantities. Fundamental changes were implemented in the new rezoning scheme, particularly in the momentum fluxing and the pressure solution. The code has been successfully tested with various examples ranging from a 1-D diffusion problems to full scale engine computations with moving boundaries. The unstructured capability of this code allows for easier and more flexible grid construction which would directly benefit applications in complex geometries.

Acknowledgments

The authors acknowledge the support of the Office of Freedom Car and Vehicle Technologies of the Department of Energy. This research was conducted by the US Department of Energy under contract W-7405-ENG-36 (LA-UR-05-9286). We also gratefully acknowledge the many helpful suggestions and advice of Peter O'Rourke and Manjit Sahota of the Theoretical Division and Margaret Hubbard at Los Alamos National Laboratory.

Appendix A. Mass

Start with

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_V \rho_m \,\mathrm{d}V = \int_S \left[\rho D \nabla\left(\frac{\rho_m}{\rho}\right)\right]\mathrm{d}\mathbf{A}.$$

Approximate the time derivative and set the control volume to be a cell volume which moves with the fluid velocity,

$$\frac{(\rho_m)^{\mathrm{B}}V^{\mathrm{B}} - (\rho_m)^{\mathrm{A}}V^{\mathrm{A}}}{\Delta t} = \sum_{f=\text{cell faces}} \left[\rho D \nabla\left(\frac{\rho_m}{\rho}\right)\right]_{\mathrm{f}} \cdot \mathbf{A}_{\mathrm{f}}.$$

Use the definition of mass fraction $Y_m = \rho_m/\rho$.

$$\frac{(Y_m)^{\mathrm{B}}M^{\mathrm{B}} - (Y_m)^{\mathrm{A}}M^{\mathrm{A}}}{\Delta t} = \sum_f (\rho D \nabla Y_m)_{\mathrm{f}} \cdot \mathbf{A}_{\mathrm{f}}.$$

Choose a semi-implicit formulation and use $M^{\mathrm{B}} = M^{\mathrm{A}}$ since mass within a cell does not change due to diffusion

$$M^{\mathrm{B}}\left(\frac{Y_m^{\mathrm{B}}-Y_m^{\mathrm{A}}}{\Delta t}\right)=\sum_f\left[(\rho D)^n\nabla[\phi_{\mathrm{D}}Y_m^{\mathrm{B}}+(1-\phi_{\mathrm{D}})Y_m^{\mathrm{A}}]\right]_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^n. \tag{A.1}$$

Eq. (A.1) is approximated in the form

$$M^{\mathrm{B}}\left(Y_m^{\mathrm{B}}-Y_m^{\mathrm{A}}\right)-\frac{\Delta t}{Sc_{\mathrm{t}}}\left[\sum_f(\mu_{\mathrm{t}}^n)_{\mathrm{f}}(\nabla Y_m^{\mathrm{A}})_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^n+\sum_f(\mu_{\mathrm{t}}^n)_{\mathrm{f}}\nabla[\phi_{\mathrm{D}}(Y_m^{\mathrm{B}}-Y_m^{\mathrm{A}})]_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^n\right]=0, \tag{A.2}$$

where Sc_{t} is the turbulent Schmidt number $Sc_{\mathrm{t}}=\frac{\mu_{\mathrm{t}}}{\rho D}$. The equation is solved using a conjugate residual algorithm where the residual is the left side of Eq. (A.2).

Appendix B. $d\mathbf{A}_f/dt$

To compute the term $\frac{d\mathbf{A}_{\mathrm{f}}}{dt}$, let us refer first to Fig. 26. For a planar quadrilateral, one can compute the area of the face using

$$\mathbf{A}_{\mathrm{f}}=\frac{1}{4}[(\mathbf{x}_1+\mathbf{x}_2)-(\mathbf{x}_3+\mathbf{x}_4)]\times[(\mathbf{x}_2+\mathbf{x}_3)-(\mathbf{x}_1+\mathbf{x}_4)].$$

Differentiating,

$$\begin{aligned}\frac{d\mathbf{A}_{\mathrm{f}}}{dt}=&\frac{1}{4}\left[\left(\frac{d\mathbf{x}_1}{dt}+\frac{d\mathbf{x}_2}{dt}\right)-\left(\frac{d\mathbf{x}_3}{dt}+\frac{d\mathbf{x}_4}{dt}\right)\right]\times[(\mathbf{x}_2+\mathbf{x}_3)-(\mathbf{x}_1+\mathbf{x}_4)]+\frac{1}{4}[(\mathbf{x}_1+\mathbf{x}_2)-(\mathbf{x}_3+\mathbf{x}_4)]\\&\times\left[\left(\frac{d\mathbf{x}_2}{dt}+\frac{d\mathbf{x}_3}{dt}\right)-\left(\frac{d\mathbf{x}_1}{dt}+\frac{d\mathbf{x}_4}{dt}\right)\right].\end{aligned}$$

and substituting velocities $\mathbf{u}_i^n$ for the temporal derivatives of the vertex locations $\frac{d\mathbf{x}_i}{dt}$ yields,

$$\begin{aligned}\frac{d\mathbf{A}_{\mathrm{f}}}{dt}=&\frac{1}{4}\left[\left(\mathbf{u}_1^n+\mathbf{u}_2^n\right)-\left(\mathbf{u}_3^n+\mathbf{u}_4^n\right)\right]\times[(\mathbf{x}_2+\mathbf{x}_3)-(\mathbf{x}_1+\mathbf{x}_4)]+\frac{1}{4}[(\mathbf{x}_1+\mathbf{x}_2)-(\mathbf{x}_3+\mathbf{x}_4)]\\&\times\left[\left(\mathbf{u}_2^n+\mathbf{u}_3^n\right)-\left(\mathbf{u}_1^n+\mathbf{u}_4^n\right)\right].\end{aligned}$$

Appendix C. Pressure

Here we derive (45). Solve (34) for T^{B} and substitute the result in (35),

$$\begin{aligned}\frac{V^{\mathrm{B}}}{\frac{M^{\mathrm{B}}}{p^{\mathrm{p}}}\bar{R}}=&\left\{T^{\mathrm{t}}+\frac{p^{\mathrm{p}}+p^n}{2c_{\mathrm{v}}^{\mathrm{t}}}\frac{V^n}{M^{\mathrm{B}}}+\frac{\Delta t}{M^{\mathrm{B}}c_{\mathrm{v}}^{\mathrm{t}}}\left[\frac{1}{Pr_{\mathrm{t}}}\sum_f(c_{\mathrm{p}}\mu_{\mathrm{t}})_{\mathrm{f}}\nabla(\phi_{\mathrm{D}}T^{\mathrm{B}}+(1-\phi_{\mathrm{D}})\tilde{T})_{\mathrm{f}}\cdot\mathbf{A}_{\mathrm{f}}^n\right.\right.\\&\left.\left.+(1-A_o)(\phi_{\mathrm{D}}\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{B}}):\nabla\mathbf{u}^{\mathrm{B}}+(1-\phi_{\mathrm{D}})\boldsymbol{\sigma}(\mathbf{u}^{\mathrm{A}}):\nabla\mathbf{u}^{\mathrm{A}})V^{\mathrm{B}}\right]\right\}\Bigg/\left\{1+\frac{p^{\mathrm{p}}+p^n}{2c_{\mathrm{v}}^{\mathrm{t}}p^{\mathrm{p}}}\bar{R}\right\}.\end{aligned} \tag{C.1}$$

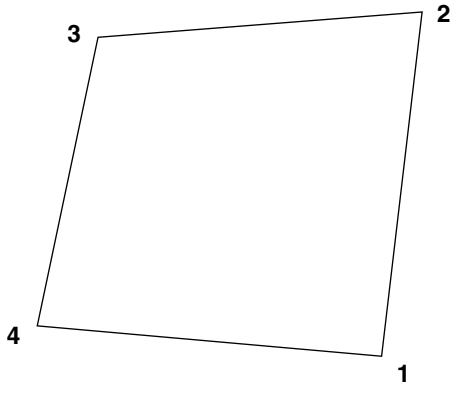


Fig. 26. Nodes on a face.

Neglecting diffusion and dissipation terms, (C.1) can be rearranged in the form

$$p^{\mathrm{p}} V^{\mathrm{B}} \left[\frac{1}{M^{\mathrm{B}} \bar{R}} + \frac{1}{2M^{\mathrm{B}} c_{\mathrm{v}}^{\mathrm{t}}} \right] + \frac{p^n V^{\mathrm{B}}}{2M^{\mathrm{B}} c_{\mathrm{v}}^{\mathrm{t}}} = T^{\mathrm{t}} + \frac{p^{\mathrm{p}} + p^n}{2c_{\mathrm{v}}^{\mathrm{t}}} \frac{V^n}{M^{\mathrm{B}}}. \tag{C.2}$$

Differentiating (C.2) with respect to $\frac{\partial}{\partial p^{\mathrm{p}}}$ yields

$$\left(p^{\mathrm{p}} \frac{\partial V^{\mathrm{B}}}{\partial p^{\mathrm{p}}} + V^{\mathrm{B}} \right) \left[\frac{1}{M^{\mathrm{B}} \bar{R}} + \frac{1}{2M^{\mathrm{B}} c_{\mathrm{v}}^{\mathrm{t}}} \right] + \frac{p^n}{2M^{\mathrm{B}} c_{\mathrm{v}}^{\mathrm{t}}} \frac{\partial V^{\mathrm{B}}}{\partial p^{\mathrm{p}}} = \frac{V^n}{2c_{\mathrm{v}}^{\mathrm{t}} M^{\mathrm{B}}} \tag{C.3}$$

which can be rewritten in the form of Eq. (45).

Appendix D. Potential improvements

One can improve the computation of the derivative in (60) (at additional cost) by using three cells and solving a 3×3 linear system of equations. Find three cells (with cell centers $\mathbf{x}_{\mathrm{c}_i}$) whose angle with $\mathbf{x}_{\mathrm{f}} - \mathbf{x}_{\mathrm{c}_n}$ is largest

$$\mathbf{x}_{\mathrm{c}_i} = \min_{\mathbf{x}_{\mathrm{c}_j}} \left(\frac{\mathbf{x}_{\mathrm{f}} - \mathbf{x}_{\mathrm{c}_n}}{|\mathbf{x}_{\mathrm{f}} - \mathbf{x}_{\mathrm{c}_n}|} \cdot \frac{\mathbf{x}_{\mathrm{c}_j} - \mathbf{x}_{\mathrm{c}_n}}{|\mathbf{x}_{\mathrm{c}_j} - \mathbf{x}_{\mathrm{c}_n}|} \right). \tag{D.1}$$

Then solve the 3×3 linear system for the coefficients of the vector $\mathbf{e}_{\mathrm{c}_n}$

$$\begin{aligned} (\mathbf{x}_{\mathrm{c}_1} - \mathbf{x}_{\mathrm{c}_n}) \cdot \mathbf{e}_{\mathrm{c}_n} &= (\rho q)_{\mathrm{c}_1} - (\rho q)_{\mathrm{c}_n}, \\ (\mathbf{x}_{\mathrm{c}_2} - \mathbf{x}_{\mathrm{c}_n}) \cdot \mathbf{e}_{\mathrm{c}_n} &= (\rho q)_{\mathrm{c}_2} - (\rho q)_{\mathrm{c}_n}, \\ (\mathbf{x}_{\mathrm{c}_3} - \mathbf{x}_{\mathrm{c}_n}) \cdot \mathbf{e}_{\mathrm{c}_n} &= (\rho q)_{\mathrm{c}_3} - (\rho q)_{\mathrm{c}_n}, \end{aligned} \tag{D.2}$$

where $(\rho q)_{\mathrm{c}_i}$ refers to the cell-centered quantity (ρq) whose cell center is $\mathbf{x}_{\mathrm{c}_i}$. Replace the term

$$\frac{(\rho q)_{\mathrm{c}_n} - (\rho q)_{\mathrm{c}_{nn}}}{|\mathbf{x}_{\mathrm{c}_n} - \mathbf{x}_{\mathrm{c}_{nn}}|}$$

in (60) with

$$\frac{\mathbf{x}_{\mathrm{f}} - \mathbf{x}_{\mathrm{c}_n}}{|\mathbf{x}_{\mathrm{f}} - \mathbf{x}_{\mathrm{c}_n}|} \cdot \mathbf{e}_{\mathrm{c}_n}.$$

Similarly one can improve the computation of (68). Find three vertices (with locations $\tilde{\mathbf{x}}_i$) whose angle with $\mathbf{x}_1 - \mathbf{x}_2$ is largest

$$\tilde{\mathbf{x}}_i = \min_{\mathbf{x}_{\mathrm{v}}} \left(\frac{\mathbf{x}_1 - \mathbf{x}_2}{|\mathbf{x}_1 - \mathbf{x}_2|} \cdot \frac{\mathbf{x}_{\mathrm{v}} - \mathbf{x}_2}{|\mathbf{x}_{\mathrm{v}} - \mathbf{x}_2|} \right). \tag{D.3}$$

Then solve the 3×3 linear system for the coefficients of the vector $\mathbf{e}_u$

$$\begin{aligned} (\tilde{\mathbf{x}}_1 - \mathbf{x}_2) \cdot \mathbf{e}_u &= \tilde{u}_1 - u_2, \\ (\tilde{\mathbf{x}}_2 - \mathbf{x}_2) \cdot \mathbf{e}_u &= \tilde{u}_2 - u_2, \\ (\tilde{\mathbf{x}}_3 - \mathbf{x}_2) \cdot \mathbf{e}_u &= \tilde{u}_3 - u_2, \end{aligned} \tag{D.4}$$

where $\tilde{u}_i$ is a component of the velocity at locations $\tilde{\mathbf{x}}_i$. Replace the term

$$\frac{u_2 - u_3}{|\mathbf{x}_2 - \mathbf{x}_3|}$$

in (68) with

$$\frac{\mathbf{x}_2 - \mathbf{x}_3}{|\mathbf{x}_2 - \mathbf{x}_3|} \cdot \mathbf{e}_u.$$

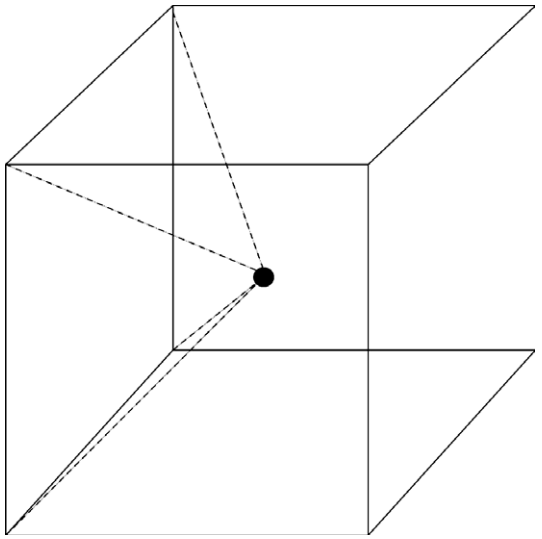


Fig. 27. Pyramidal volume generated with a particle and a face of a cell.

Appendix E. Time step constraints

While KIVA-4 does not have any stability constraints since the diffusion terms are differenced implicitly and the advection fluxing is subcycled, the time step is still limited based on accuracy considerations. The first time constraint limits the product of the acceleration and the time step squared by the cell size Δx_{cell},

$$\Delta t_{\text{cell}} = \frac{f_{\text{acc}} \Delta x_{\text{cell}}}{\max\limits_{\text{nodes of cell}} |\mathbf{u}^{\text{B}} - \mathbf{u}^{n}|}$$

where $f_{\text{acc}} = 0.5$. A time step Δt_{acc} constraint for the entire mesh is computed by taking the minimum of Δt_{cell} for all cells in the mesh.

The second time constraint limits the amount of cell distortion that occurs in the Lagrangian stage.

$$\Delta t_{\text{rst}} = \min_{\text{c}} \frac{f_{\text{rst}}}{2\sqrt{e/3} + |\tilde{p}|/3},$$

where

$$e = \frac{1}{3}(\tilde{p}^2 - 3q), \quad \tilde{p} = -s_{mm}$$

and

$$q = s_{11}s_{22} + s_{11}s_{33} + s_{22}s_{33} - s_{23}^2 - s_{13}^2 - s_{12}^2,$$

where $f_{\text{rst}} = 0.6$, $\mathbf{S}$ is the rate of strain tensor and s_{lm} are its entries,

$$\mathbf{S} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\text{T}}).$$

The time step is also limited in its growth rate $\Delta t_{\text{gr}} = 1.02\, \Delta t^n$, by a user defined maximum time step Δt_{mx} and a user defined time step based on maximum crank angle Δt_{mxca}. There are also time step constraints due to chemistry Δt_{cm} and evaporation Δt_{sp}. The final time step Δt^{n+1} for the next cycle is computed by taking the minimum of all of the above time steps,

$$\Delta t^{n+1} = \min(\Delta t_{\text{acc}}, \Delta t_{\text{rst}}, \Delta t_{\text{cm}}, \Delta t_{\text{sp}}, \Delta t_{\text{gr}}, \Delta t_{\text{mx}}, \Delta t_{\text{mxca}}).$$

Appendix F. Finding a particle in an unstructured grid

Moving from a structured grid to an unstructured grid in regards to particles does not require any significant modifications to the algorithm used to find particles. In KIVA-3V, a particle is inside a cell or on a cell boundary if all the face-particle pyramidal volumes are non-negative for all six faces of the cell. Fig. 27 shows the pyramidal volume generated with a particle and a cell face. The vertices of this face-particle pyramidal volume are the particle vertex and the cell face vertices. The volume of the pyramid is found by decomposing it into two tetrahedra.

For unstructured meshes, faces can be triangular or degenerate (have zero area). If the face is triangular, the face-particle volume will be a tetrahedron. However, the algorithm for finding the pyramidal volume still applies. If the face is degenerate, the face-particle volume will be zero. However if the remainder of the face-particle volumes are likewise non-negative, the particle still lies inside the cell. Thus the criteria (if all six face-particle volumes are nonnegative, the particle lies in the element) generalizes to unstructured meshes.

References

[1] A.A. Amsden, KIVA-3: A KIVA program with block-structured mesh for complex geometries, Technical Report, Los Alamos National Laboratory, LA-12503-MS, 1993.
[2] A.A. Amsden, KIVA-3V: A block-structured KIVA program for engines with vertical or canted valves, Technical Report, Los Alamos National Laboratory, LA-13313-MS, 1997.
[3] A.A. Amsden, P.J. O'Rourke, T.D. Butler, KIVA-II: A computer program for chemically reactive flows with sprays, Technical Report, Los Alamos National Laboratory, LA-11560-MS, 1989.
[4] A.A. Amsden, J.D. Ramshaw, L.D. Cloutman, P.J. O'Rourke, Improvements and extensions to the KIVA computer program, Technical Report, Los Alamos National Laboratory, LA-10534-MS, 1985.
[5] A.A. Amsden, J.D. Ramshaw, P.J. O'Rourke, J.K. Dukowicz, KIVA: A computer program for two- and three-dimensional fluid flows with chemical reactions and fuel sprays, Technical Report, Los Alamos National Laboratory, LA-10245-MS, 1985.
[6] H. Braess, P. Wriggers, Arbitrary Lagrangian-Eulerian finite element analysis of free surface flow, Computational Methods in Applied Mechanical Engineering 190 (2000) 95–109.
[7] T.D. Butler, L.D. Cloutman, J.K. Dukowicz, J.D. Ramshaw, CONCHAS: An arbitrary Lagrangian–Eulerian computer code for multicomponent chemically reactive fluid flow at all speeds, Technical Report, Los Alamos National Laboratory, LA-8129-MS, 1979.
[8] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, Journal of Computational Physics 48 (1982) 387–411.
[9] F.H. Harlow, A.A. Amsden, Fluid dynamics, Technical Report, Los Alamos National Laboratory, LA-4700, 1971.
[10] P. Kjellgren, J. Hyvarinen, An arbitrary Lagrangian–Eulerian finite element method, Computational Mechanics 21 (1998) 81–90.
[11] H.H. Nu, N.A. Patankar, M.Y. Zhu, Direct numerical simulations of fluid–solid systems using the arbitrary Lagrangian–Eulerian technique, Journal of Computational Physics 169 (2001) 427–462.
[12] P.J. O'Rourke, A.A. Amsden, Implementation of a conjugate residual iteration in the KIVA computer program, Technical Report, Los Alamos National Laboratory, LA-10849-MS, 1986.
[13] P.J. O'Rourke, M.S. Sahota, A variable explicit/implicit numerical method for calculating advection on unstructured meshes, Journal of Computational Physics 143 (1998) 312–345.
[14] D.P. Schmidt, C.J. Rutland, Reducing grid dependency in droplet collision modeling, Journal of Engineering for Gas Turbines and Power 126 (2004) 227–233.
[15] D.J. Torres, KIVA-4: Validation, rezoning, and remapping, in: 15th International Multidimensional Engine Modeling User's Group Meeting, Detroit, 2005.
[16] D.J. Torres, P.J. O'Rourke, KIVA-4, in: 14th International Multidimensional Engine Modeling User's Group Meeting, Detroit, 2004.
[17] D.J. Torres, P.J. O'Rourke, A.A. Amsden, A discrete multicomponent fuel model, Atomization and Sprays 13 (2003) 131–172.
[18] M.F. Trujillo, D.J. Torres, P.J. O'Rourke, High-pressure multicomponent liquid sprays: departure from ideal behavior, International Journal of Engine Research 5 (3) (2004) 229–246.
[19] J.F. Wiedenhoefer, R.D. Reitz, A multidimensional radiation model for diesel engine simulation with comparison to experiment, Numerical Heat Transfer, Part A (Applications) 44 (2003) 665–682.
[20] Y. Yi, R.D. Reitz, Modeling the primary breakup of high-speed jets, Atomization and Sprays 14 (2004) 53–79.
[21] Y. Zeng, C.F. Lee, A model of multicomponent spray vaporization in a high-pressure and high-temperature environment, Journal of Engineering for Gas Turbines and Power 124 (2002) 717–724.