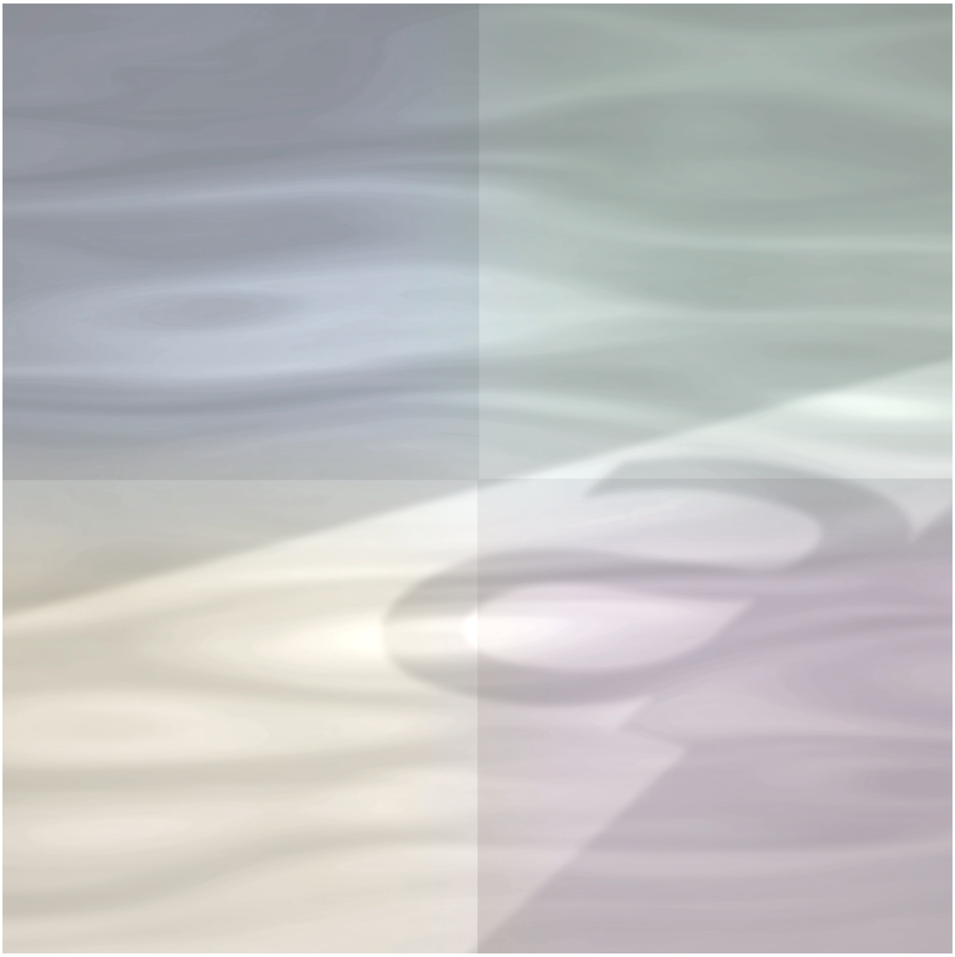
STORM/CFD2000 Theoretical Background



ADAPTIVE RESEARCH

STORM/CFD2000 — Theoretical Background

Printed: December 2002
Printed in the United States of America

ADAPTIVE RESEARCH
A Division of Simunet Corporation.

P.O. Box 7289
Alhambra, California 91802
(800) 326-5155

sales@adaptive-research.com

www.adaptive-research.com

LIMITED WARRANTY

Adaptive Research Division of Simunet Corporation [Adaptive] warrants that for a period of ninety (90) days from the date of purchase the *CFD2000* computer program and documentation will substantially conform to published specifications. Adaptive also warrants that the magnetic media used to transfer the software is free from defects in material and workmanship and that the software is free from substantial programming errors. During this period defective media will be replaced and substantial programming errors in the software will be corrected by Adaptive with no charge.

CUSTOMER REMEDIES

If any magnetic media device supplied with this software fails within ninety (90) days of the original purchase for any reason other than accident, abuse, misuse or misapplication, or substantial programming errors are discovered within ninety (90) days of the original purchase, please return the defective magnetic media device and/or a description and exemplar of the programming error to Adaptive Research, at P.O. Box 7289, Alhambra, California 91802 for a free replacement magnetic media device or for correction of the substantial programming error. Adaptive's and/or its Licensor's/s' entire liability, and your exclusive remedy, is the repair of replacement of magnetic media devices or correction of substantial programming errors that do not meet Adaptive's Limited Warranty that are returned and/or identified to Adaptive no later than one hundred and twenty (120) days after original purchase. This limited warranty is void if failure of the magnetic media device or the programming error has resulted from accident, abuse, misuse, or misapplication.

After the Limited Warranty period expires, or if not covered by the Limited Warranty, Adaptive will replace magnetic media devices and correct substantial programming errors for a fee. For more information, please contact Adaptive Research at P.O. Box 7289, Alhambra, California 91802. This policy will be followed as long as the program is still manufactured by Adaptive.

DISCLAIMER

Except as specified above, Adaptive makes no representations or warranties with respect to the contents of the *CFD2000* software and documentation. Adaptive specifically disclaims any implied warranties of merchantability or fitness of *CFD2000* for any particular purpose. Furthermore, Adaptive reserves the right to make changes from time to time in the contents of *CFD2000* without obligation of Adaptive to notify any persons or organizations of such revisions or changes. This Limited Warranty gives you specific legal rights that vary from state/jurisdiction to state/jurisdiction.

LIMITATION OF LIABILITY

SINCE *CFD2000* IS COMPLEX AND MAY NOT BE ENTIRELY FREE FROM ERRORS, ADAPTIVE ADVISES THAT YOU VERIFY THE DATA PRODUCED BY *CFD2000*. IN NO EVENT SHALL ADAPTIVE BE LIABLE FOR ANY LOSS OF USE, PROFIT, OR REVENUE DUE TO THE USE OF *CFD2000*, OR FOR ANY INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR SPECIAL DAMAGES INCURRED OR SUFFERED DUE TO OR RELATED TO THE USE OF *CFD2000* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO CASE SHALL ADAPTIVE'S LIABILITY EXCEED THE AMOUNT OF THE PURCHASE PRICE. AS SOME STATES/ JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

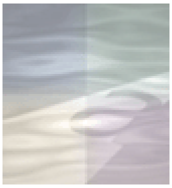
U.S. GOVERNMENT-RESTRICTED RIGHTS

This document is provided with restricted rights. Use, duplication or disclosure by the Government is subject to restrictions as set forth in subdivision (c) (1) (iii) of the Rights in Technical Data and Computer Software Clause 52.227-19, or its current equivalent clause.

COPYRIGHT

TRADEMARKS

STORM, CFD2000, and *Adaptive Research* are registered trademarks of Simunet Corporation.



PREFACE

STORM/CFD2000 — Theoretical Background is one volume of a multi-volume documentation set. In addition to this manual, the following publications are also included as part of the *STORM/CFD2000* software package:

- The *STORM/CFD2000 User Guide* explains the basic concepts of setting up and running cases with *CFD2000*, describes in detail all components of the interface, and explains how to build models using the geometry tools.
- *CFD2000/Fieldview for Windows User Guide* describes the standard three-dimensional visualization tool included with the *STORM/CFD2000* software package.

CONTENTS

Special Terms *(cont.)*

FIGURES

TABLES

Chapter 1

GOVERNING EQUATIONS AND SOLUTION METHODS

This chapter describes the CFD theory that forms the basis of *Storm*, the 3-D Reynolds-averaged Navier-Stokes solver used in *CFD2000*. Individual sections discuss the governing equations, the discretization technique, the solution algorithm, and the solver methodologies employed by *Storm*.

Overall objectives are to:

- Provide insight into the numerical schemes so that you can understand the performance of the code in terms of stability and convergence rate
- Establish signposts that indicate when and how you can intervene in the solution process, and
- Illustrate how the information you provide affects the performance of the code and the quality of the solution.

An understanding of the numerical implementation of *Storm* will help you achieve the ultimate objective—an accurate solution to your flow model.

Governing Equations

Storm is a general purpose computer program designed to numerically solve the Navier-Stokes equations, which consist of conservation equations for mass, momentum, and energy. In addition, it is capable of solving an arbitrary number of general transport equations. The code uses a finite-volume representation of the governing equations, whereby the continuous problem domain is decomposed into multiple control volumes, and the governing equations are applied to individual control volumes and integrated over the entire computational domain. This algebraic equation set is then solved using general and efficient numerical methods to obtain a solution of the engineering system.

In this section we present the basic equation set used by *Storm*. No derivations are presented; for these, the reader may refer to any standard work such as Landau and Lifshitz (1959). We start by presenting the differential equations that express the conservation of mass, momentum, and energy within a fluid volume. These equations are intended to be applied to a single-phase fluid—i.e., one that is either entirely gaseous or entirely liquid. Interactions between the fluid and any suspended particulate material of a different phase are accounted for through source terms that appear in each of the equations.

Conservation of Mass

The conservation of mass within a small, fixed fluid volume is expressed

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = S_{m,p} \tag{1}$$

where ρ is the local fluid density, t is time; x_i is the position vector in the i^{th}-coordinate direction, u_i is the i^{th}-fluid velocity component; and $S_{m,p}$ represents the rate per unit volume at which mass is transferred to the fluid by any of a number of sources—for example, evaporation of particulates. By default, $S_{m,p}$ is set to zero (deactivated) unless the user explicitly activates one of the *Storm* source term models.

Conservation of Momentum

Newton's second law applied to the fluid passing through a small, fixed volume leads to the following expression

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_i} - \frac{\partial p}{\partial x_i} + \rho B_i + S_{u_i,p} \tag{2}$$

where B_i represents the component of the total body force per unit volume (e.g., gravity) exerted on the fluid in the i^{th}-coordinate direction; p is the local thermodynamic pressure; $S_{u_i,p}$ is the momentum source/sink term; and τ_{ij} is the viscous stress tensor given by

$$\tau_{ij} = \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \tag{3}$$

Here μ is the dynamic viscosity of the fluid due to laminar (molecular) diffusion, and δ_{ij} is the Kronecker delta function ($\delta_{ij} = 1$ if $i = j$; $\delta_{ij} = 0$ if $i \neq j$). Techniques used by Storm to model μ are discussed in Chapter 2, pages 30-38.

Implicit in Eq. (3) is the conventional assumption of a linear relationship between the tangential stress and the rate-of-strain tensor (Stoke's hypothesis). *Storm* further assumes that the so-called bulk viscosity (or "second viscosity") is negligibly small—an appropriate simplification for nearly all gaseous flows, and most low density liquids (Landau and Lifshitz, 1959).

Conservation of Energy

The first law of thermodynamics applied to the fluid passing through a small, fixed volume leads to the following conservation equation

$$\frac{\partial \rho H}{\partial t} + \frac{\partial(\rho u_i H)}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\frac{\kappa}{C_p}\frac{\partial H}{\partial x_i}\right] + \frac{\partial p}{\partial t} + u_i\frac{\partial p}{\partial x_i} + \Phi + Q + S_{H,p} \qquad (4)$$

where H is the local density-weighted mean static fluid enthalpy; Φ is the Stokes molecular dissipation function defined as

$$\Phi = \left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu\frac{\partial u_k}{\partial x_k}\delta_{ij}\right]\frac{\partial u_i}{\partial x_j} \;; \qquad (5)$$

Q is the rate per unit volume at which heat is added to the fluid; κ is the thermal conductivity; C_p is the mass-weighted mean specific heat at constant pressure; and $S_{H,p}$ is the enthalpy source/sink term.

For multi-component fluids, *Storm* assumes that the enthalpy H represents the sum of the enthalpies of each individual component. Thus, in general,

$$H = \sum_{n=1}^{N} Y_n H_n \qquad (6)$$

where N is the total number of fluid components in the mixture, Y_n is the density-weighted mean mass fraction of the n^{th} species, and H_n is the local component enthalpy as defined by the expression

$$H_n = H_{n,0} + \int_{T_0}^{T} C_{p,n} dT \tag{7}$$

where $C_{p,n}$ is the specific heat at constant pressure for component n; $H_{n,0}$ is a prescribed reference enthalpy valid at temperature T_0; and T is the local fluid temperature.

Conservation of Species Mass Fraction

The conservation of mass fraction Y_n for component n of an N-component fluid mixture is expressed

$$\frac{\partial \rho Y_n}{\partial t} + \frac{\partial(\rho u_i Y_n)}{\partial x_i} = \frac{\partial}{\partial x_i}\left[D_n \frac{\partial Y_n}{\partial x_i}\right] + \Omega_n + S_{Y_n,p} \tag{8}$$

where D_n and $D_{n,T}$ are the species mass diffusivities (SI units: kg/m/s) for component *n* due to laminar and turbulent motions, respectively; $S_{Yn,p}$ is the mass source of species *n* due, for example, to evaporation from the particulate phase (inactive by default) or chemical reactions within the flow.

Conservation of Turbulence Quantities

The conservation equations for turbulent kinetic energy k (SI unit: m^2/s^2) and the turbulent kinetic energy dissipation rate ε (m^2/s^3) are, respectively:

$$\frac{\partial \rho k}{\partial t} + \frac{\partial(\rho u_i k)}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_T}{Pr_k}\right)\frac{\partial k}{\partial x_i}\right] + \mu_T G - \rho\varepsilon + +S_{k,p} \tag{9}$$

and

$$\frac{\partial \rho \varepsilon}{\partial t} + \frac{\partial(\rho u_i \varepsilon)}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_T}{Pr_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_i}\right] + \frac{\varepsilon}{k}\left(C_1 \mu_T G - C_2 \rho\varepsilon\right) + S_{\varepsilon,p} \tag{10}$$

where C_1 and C_2 are dimensionless model constants; Pr_k and Pr_e are the turbulent Prandtl numbers for kinetic energy and dissipation, respectively; $S_{k,p}$ and $S_{e,p}$ are source terms for the kinetic energy and turbulent dissipation; and

$$G = \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \frac{\partial u_i}{\partial x_j} - \frac{1}{\rho^2} \frac{\partial \rho}{\partial x_j} \frac{\partial p}{\partial x_j} - \frac{2}{3} \left(\frac{\rho k}{\mu_T} + \frac{\partial u_i}{\partial x_j} \right) \frac{\partial u_j}{\partial x_j} \qquad (11)$$

is the turbulent production rate (SI unit: s^{-2}). Values for the various dimensionless constants used in Eqs. (9) and (10) are discussed in Chapter 2, pages 44-46.

Equations of State (Density Models)

The equation of state closes the system of dynamical equations by relating the fluid density to the other thermodynamic variables. The various options available to define density are discussed in Chapter 2, pages 27-30.

General Conservation Equation

The conservation equations presented above constitute the basic set of partial differential equations solved by *Storm*. In principle, each of these equations could be coded separately in discretized form and solved individually to produce a solution. However, careful observation reveals that these equations all have a similar form, indicating that each dependent variable solved by *Storm* obeys the same generalized conservation principle. In particular, if we let ϕ denote the dependent variable, it turns out that all of the governing equations can be reduced to a single convective-diffusive conservation equation of the form

$$\frac{\partial \rho \phi}{\partial t} + \frac{\partial (\rho u_i \phi)}{\partial x_i} = \frac{\partial}{\partial x_i} \left[\Gamma_\phi \frac{\partial \phi}{\partial x_i} \right] + S_\phi \qquad (12)$$

where Γ_ϕ is the effective diffusion coefficient for quantity ϕ; S_ϕ is the net source term; the two terms on the left-hand side are, respectively, the local rate of change of quantity φ (the time term) and the convection term; and the first term on the right is the diffusion term.

Storm exploits the common structure of the governing equations by coding only one conservation equation, Eq. (12), and then deriving solutions for all the dependent variables from it. Consequently, Eq. (12) takes the form of a vector equation with

$$\phi = \begin{Bmatrix} 1 \\ u_1 \\ u_1 \\ u_1 \\ H \\ Y_n \\ k \\ \varepsilon \end{Bmatrix} \quad \Gamma_\phi = \begin{Bmatrix} 0 \\ \mu \\ \mu \\ \mu \\ k / C_p \\ D_n \\ \mu + \frac{\mu_T}{Pr_k} / \\ \mu + \frac{\mu_T}{Pr_\varepsilon} \end{Bmatrix} \quad S_\phi = \begin{Bmatrix} S_{m,p} \\ \frac{\partial \tau_{1j}}{\partial x_1} - \frac{\partial p}{\partial x_1} + \rho B_1 + S_{u_1,p} \\ \frac{\partial \tau_{2j}}{\partial x_2} - \frac{\partial p}{\partial x_2} + \rho B_2 + S_{u_2,p} \\ \frac{\partial \tau_{3j}}{\partial x_3} - \frac{\partial p}{\partial x_3} + \rho B_3 + S_{u_3,p} \\ \frac{\partial p}{\partial t} + u_i \frac{\partial p}{\partial x_i} + \Phi + Q + S_{H,p} \\ \Omega_n + S_{Y_n,p} \\ \mu_T G - \rho\varepsilon + + S_{k,p} \\ \frac{\varepsilon}{k}\left(C_1 \mu_T G - C_2 \rho \varepsilon\right) + S_{\varepsilon,p} \end{Bmatrix} \qquad (13)$$

for the dependent variable vector, the effective diffusion vectors, and the source term vector. Here the first element in each vector corresponds to the continuity equation as given by Eq. (1); the second, third, and fourth elements correspond to the three components of the momentum equation; the fifth element corresponds to the energy equation; the sixth element corresponds to the species mass fraction equation; the seventh element corresponds to the turbulent kinetic energy equation; and the eighth element corresponds to the turbulent kinetic energy dissipation rate equation. Note that when written in this format, the source vector includes all terms in the governing equations that cannot be represented as contributing to either convection or diffusion.

Control Volume Formulation

The finite-volume discretization employed by *Storm* uses an integral form of the general conservation equation Eq. (12). This involves dividing the entire computational domain into a series of small, elementary control volumes (cells) over which the integration is carried out. Curvilinear coordinates (either orthogonal or body-fitted) can be used to define the cells.

General Transport Equation

As stated previously, for a general variable ϕ the transport equation can be written as

$$\underbrace{\frac{\partial \rho\phi}{\partial t}}_{(T)} + \underbrace{\frac{\partial\left(\rho u_i \phi\right)}{\partial x_i}}_{(C)} = \underbrace{\frac{\partial}{\partial x_i}\left[\Gamma_\phi \frac{\partial \phi}{\partial x_i}\right]}_{(D)} + \underbrace{S_\phi}_{(S)} \tag{14}$$

where Γ_ϕ is the effective diffusion coefficient and S_ϕ is the source term. The four terms in this transport equation (as indicated above) will be referred to in subsequent sections as the *Transient* term (T), the *Convection* term (C), the *Diffusion* term (D), and the *Source* term (S).

Coordinate System

All the previous sections have considered the conservation equations to be written for a Cartesian coordinate system. To simulate three-dimensional flows interacting with complicated real-life geometric surfaces, these equations are usually transformed from the Cartesian system (x, y, z) to a generalized non-orthogonal curvilinear coordinate system (ξ, η, ζ). This coordinate system defines the computational plane and is usually referred to as the body-fitted coordinate (BFC) system. The transformation makes it possible to solve the governing equations on a uniformly spaced grid in the computational plane, even though the physical grid is curvilinear in nature.

CFD2000 offers the capability to solve the governing equations in both Cartesian and body-fitted coordinate systems.

Cell Face Nomenclature

Storm requires that every cell comprising the computational domain be hexahedral—i.e., a six-sided, plane-faced volume. A schematic representation of such a cell is shown in Fig. 1.1 below. By convention, each face is labeled according to a right-handed "geographical" scheme in which the "east" face (labeled "e" in Fig. 1.1) is assumed to lie in the general direction of the principal coordinate axis (usually the X-axis) relative to the cell center; the "north" face (n) is assumed to lie in the Y-axis direction; and the "high" face (h) lies above the cell center in the Z-axis direction. Faces labeled w, s, and l (for "west," "south," and "low," respectively) are defined relative to these, and the point at the geometric center of the cell is labeled P. The centers of the neighboring cells surrounding cell P are likewise labeled N, S, E, W, H, and L, as indicated in Fig. 1.1.

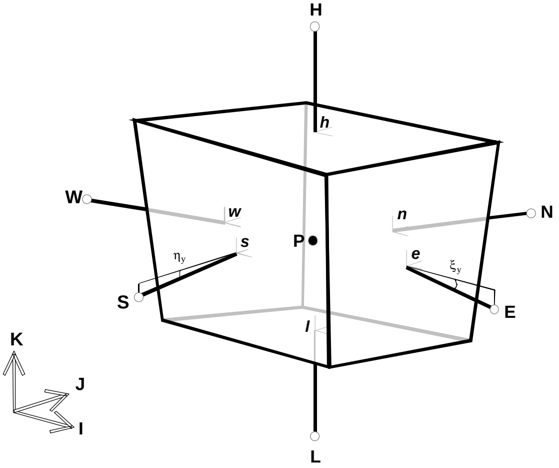


Figure 1.1 Numerical stencil in Storm.

Discretization of the Transport Equation

The governing equation is discretized following the procedure described in Patankar, 1980. Essentially, this consists of integrating the governing equation over the control volume shown in Figure 1.1 (previous page). The volume integrals are converted to surface integrals for the control volume using Green's theorem; the individual fluxes at the faces of each cell are then estimated using appropriate interpolation practices.

Integrating over the volume v of an individual cell and applying Green's theorem, the integral forms of the individual terms in Eq. (12) become

Transient Term:

$$T = \iiint_{v} \frac{\partial \phi}{\partial t} dV = V\left(\phi^{new} - \phi^{old}\right) \Big/ \Delta t \tag{15}$$

Convection Term:

$$C = \iiint_{V} \left(\nabla\left(\rho\vec{V}\phi\right)\right) dV = \iint_{A} \left(\rho\vec{V}\phi\right) dA = \sum_{AllFaces} \rho\vec{V}\phi A \tag{16}$$

Diffusion Term:

$$D = \iiint_{V} \left(\nabla\left(\Gamma\nabla\phi\right)\right) dV = \iint_{A} \left(\Gamma\nabla\phi\right) dA = \sum_{AllFaces} \left(\Gamma\nabla\phi A\right) \tag{17}$$

Source Term:

$$S = \iiint_{V} S_{\phi} dV = VC_{\phi}\left(\phi_{o} - \phi\right) \tag{18}$$

where ϕ^{old}, ϕ^{new} refer to the value of an arbitrary quantity at two time levels; $\vec{V}$ is the velocity vector; V is the cell volume; *dA* is the differential area of a call face; C_{ϕ} is a source linearization coefficient; and ϕ_{o} is a constant.

Interface Fluxes

As is evident from the previous section, the convection and diffusion flux for each variable at the cell face has to be estimated based on the values of the variable at the neighboring cell center. A number of practices are available to interpolate for the interfacial fluxes.

Convection Flux

The convective contribution to the local rate of change of quantity ϕ within the control volume illustrated in Fig. 1.1 can be expressed

$$C = G_e\phi_e - G_w\phi_w + G_n\phi_n - G_s\phi_s + G_h\phi_h - G_e\phi_e \qquad (19)$$

where, for example,

$$G_e = \rho_e u_e A_e \qquad (20)$$

denotes the mass flux of fluid crossing the east cell face; A_e is the face area; ρ_e and ϕ_e are, respectively, the density and the scalar variable value defined at the geometric center of the face; and u_e is the magnitude of the velocity field oriented perpendicular to the cell face (the contravariant velocity). Similar definitions apply to quantities defined at the west, north, south, high, and low faces as well.

Storm offers a selection of schemes for formulation of the convection contribution to the coefficients in the finite-volume equations. The schemes described briefly include:

- Hybrid Scheme
- First-Order Upwind Scheme
- Higher-Order Upwind Schemes

Hybrid Scheme

The hybrid scheme (Spalding, 1972) compares the relative magnitudes of convective and diffusive transport at each cell face to determine whether the neighboring cell is effectively downstream or upstream. The Peclet number (the ratio of diffusive to convective time scales) is then used to determine the coefficient that represents a reasonable approximation to the exact solution of a local, one-dimensional, convection-diffusion flow.

Under this approach, the coefficient for the east neighbor (for example) is

$$G_e = \mathrm{Max}\left(\left| \frac{C_e}{2} \right|, D_e \right) - \frac{C_e}{2} \qquad (21)$$

where C_e is the convective contribution at the east face, defined by Eq. (19), and D_e is the diffusive contribution at the east face, defined

$$D_e = \frac{A_e \Gamma_e}{\delta x_e} \qquad (22)$$

where $(\delta x)_e$ denotes the distance between the cell center and the center of the neighboring “east” cell.

The hybrid scheme is most useful for low-speed flows in which diffusive effects are important. More detailed information about this scheme can be found in Patankar (1980).

First-Order Upwind Scheme

The first-order scheme (Courant, Isaacson, and Rees, 1952) is based upon the assumption that the convected property value at the cell face is that prevailing within the cell on the upwind side. Its stability makes it a desirable choice whenever convection effects fully dominate diffusion effects (i.e., for high Peclet numbers). For example, the east neighbor coefficient is calculated as

$$G_e = \mathrm{Max}\left(-C_e, 0\right) + D_e \tag{23}$$

where C_e and D_e are defined as in the hybrid scheme.

The first-order scheme offers the advantage of being simple and efficient. Of all the schemes included in *Storm* to compute convective fluxes, the first-order upwind scheme uses the fewest computational resources.

Second- and Third-Order Upwind Schemes

In order to reduce the numerical diffusion associated with the first-order upwind scheme, several higher order schemes have been developed. *Storm* offers two: a second-order upwind scheme, and a third-order upwind scheme.

For both higher-order upwind schemes, central differencing is used for the diffusion term. The main difference is in the convection term treatment. Unlike the first-order upwind scheme, which uses only one upstream cell value, higher-order schemes use more points to construct the convection flux as follows:

$$\begin{aligned}\left(\text{higher}-\text{order flux}\right)_e &= \left(\text{first}-\text{order flux}\right)_e \\ &\quad + \frac{(1+\eta)}{4} d\bar{f}_e^{+} - \frac{(1+\eta)}{4} d\bar{f}_e^{-} + \frac{(1-\eta)}{4} d\bar{f}_w^{+} - \frac{(1-\eta)}{4} d\bar{f}_{ee}^{-}\end{aligned} \tag{24}$$

where $d\bar{f}_e^{+}$, $d\bar{f}_e^{-}$, $d\bar{f}_w^{+}$, *and* $d\bar{f}_{ee}$ are the “limited slopes” on the w, e and ee faces, and n is a parameter that determines the order of the scheme. In particular, if η = -1, this scheme is second order; if η = 1/3, the scheme is third order.

For both second- and third-order schemes, extra cells (such as ee in Eq. 24) are brought into the discretized equations, and a “minmod” filler is used to suppress unwanted oscillations (Yee, 1989). To maintain the same bandwidth as that of the first-order scheme, the “limited slopes” are treated explicitly and are combined with the rest of the source terms in the corresponding equations.

Diffusion Flux

The transport coefficient information required for evaluation of the diffusion fluxes is stored at the center of the computational cells. However, diffusion fluxes are required at the cell faces, therefore interpolation is required to evaluate the transport coefficient at those locations.

Storm offers two interpolation options: the arithmetic mean and the harmonic mean.

Arithmetic Mean

This option employs a straightforward linear interpolation of cell-center values to estimate the effective diffusivities at the cell faces. The transport coefficient at the east cell face e, for example, is obtained from the values prevailing at cell center points P and E as follows:

$$\Gamma_e = f_e \Gamma_P + (1 - f_e)\Gamma_E \, , \tag{25}$$

where f_e is a linear interpolation factor. Similar expressions are coded for the evaluation of the transport coefficient at the other cell faces.

The arithmetic mean should be used whenever the spatial variations of the transport coefficient are known to be smooth. Although similar results would be obtained using the harmonic mean (described below) in these situations, the arithmetic mean is preferred due to its simplicity and computational efficiency.

Harmonic Mean

The second interpolation option evaluates the transport coefficient at the east face as

$$\Gamma_e = \left(\frac{1 - f_e}{\Gamma_P} + \frac{f_e}{\Gamma_E} \right)^{-1} \tag{26}$$

This option should be used whenever the transport coefficient exhibits abrupt spatial variations in the domain. This interpolation scheme is somewhat more computationally expensive, but necessary to ensure the correct evaluation of the diffusion fluxes in highly non-uniform flows.

Pressure-Velocity Collocation

Storm uses a *collocated grid* arrangement, in which all velocity components are stored at the *centers* of each computational cell. Compared to staggered grid schemes, in which flow velocity components are stored at the *faces* of each cell, this approach results in a considerable savings in computational memory requirements. For example, in a three-dimensional problem, a staggered grid scheme requires that the model must store geometric information for *four* sets of control volumes—one main set for the cell-center values, and one for each of the three velocity components. Collocated grids, on the other hand, required only one set, since all variables are carried at cell center.

However, this memory-saving does have its drawbacks. Primary among these is the tendency for the pressure and velocity fields in adjacent cells to become decoupled when using collocated schemes (Patankar, 1980). This results in a so-called “checkerboard” instability, which over time can spoil your solution.

To alleviate the possibility of "checkerboarding" while still retaining the advantages of the collocated scheme, *Storm* uses an approach similar to that introduced by Rhie and Chow (1983), and subsequently refined by Peric *et al.* (1988) and Chen *et al.* (1991). This approach, generally referred to as the *Rhie-Chow method*, employs a specialized formula for computing the cell-face (*contravariant*) velocity components. This formula corrects the usual “2Δξ” interpolated pressure gradient term based on cell-center pressures with a 1Δξ centered-difference form based on the pressures in adjacent cells. For the east-face velocity, the effective formula is

$$u_e = \bar{u} + \bar{A} \left[\frac{p_E - \overline{p_P}}{\Delta x} - \left(\frac{\P p}{\P x} \right) \right] \tag{27}$$

where u_e is the east-face velocity in the ξ-coordinate direction, A is a coefficient that depends on the local fluid state and the grid geometry, and the overbars indicate linear interpolation between the E and P cell centers. This procedure produces a strong pressure-velocity coupling and effectively suppresses the checkerboard instability.

PISO Solution Algorithm

Storm uses the PISO (*P*ressure *I*mplicit with *S*plitting of *O*perators) algorithm developed by Issa (1985) and Issa *et al*. (1991) to solve the coupled system of governing equations. This method, part of a general class of implicit *pressure-based* solution techniques, employs a series of sequential operations at each time step in which the discretized momentum and pressure-based continuity equations are solved in an alternating "predictor-corrector" fashion. This approach offers a firm advantage over many other schemes (most notably, the SIMPLE method [Patankar and Spalding, 1972] and related schemes), in that it achieves a mass- and momentum-balanced, time-accurate solution for the velocity and pressure fields in just a few "passes" per time step, without resorting to either iterative and block-solution techniques.

As implemented in *Storm*, the PISO scheme is carried out in six steps. These are described as follows.

Step 1—First predictor step for momentum

Starting with the u, v, and w velocity components and the pressure field p from the previous time step (or, if the solution is just starting, from the initial conditions), advance the discretized explicit form of the momentum equation one time increment Δt. Then, using the predicted velocity components u*, v*, and w*, advance the energy and species concentration equations one time step as well, if necessary.

Step 2—First predictor step for pressure

The velocity field obtained in the first step will not, in general, satisfy the requirement of mass continuity. Therefore, using the Poisson (elliptic) equation for pressure derived by combining the momentum and continuity equations, predict a pressure field p* that is mass-consistent with u*, v*, and w* everywhere in the domain. Use this pressure and the updated temperature T* derived in Step 1 to update the density as well, if necessary (compressible flow), using whatever equation of state has been selected.

Step 3—First corrector step for momentum

Use the predicted pressure p* and the density ρ* obtained in Step 2 in the discretized momentum equation to obtain a set of corrected velocity components u**, v**, and w**. Then use these corrected fields to update the energy and species conservation fields, if necessary.

Step 4—First corrector step for pressure

Solve the elliptic pressure equation using u**, v**, and w** once more to obtain a corrected mass-balanced pressure field p**. Update the density as well, if necessary, using the same technique as in Step 2.

Step 5—Second corrector step for momentum

Use the corrected u**, v**, and w** fields obtained in Step 3, together with the corrected pressure p** from Step 4 in the momentum equation to arrive at the final velocity field u***, v***, and w***. Also, if the energy and species concentration equations are being solved, use u***, v***, and w*** to update these fields as well.

Step 6—Advance to the next time step

At this point, the velocity components u***, v***, and w***, and the pressure field p** simultaneously satisfy both the mass and momentum balance requirements. Therefore, advance the time step Δt one increment, assign the final fields u***, v***, w*** and p** to the previous (old) time step level, and repeat the entire process from the beginning (Step 1).

Linear Equation Solvers

Finite-volume discretization of the governing equations produces a set of algebraic equations. For a 3-D problem, the size of this system can be quite large. However, because the computational stencil is limited to the cells surrounding a given cell, most of the entries in the system are zero, yielding a sparse matrix.

Strictly speaking, the algebraic equations are not linear in nature, because the coefficients themselves are functions of the dependent variables (for example, the velocity components appear in the convective contributions to the coefficients). The algebraic equations are linearized by “freezing” the coefficients in order to permit the use of linear equation solvers.

Numerous algorithms have been developed for linear sparse systems. They are essentially divided into two categories:

- Direct methods, and
- Iterative methods.

Each approach has advantages and disadvantages, and for every problem there is an optimum choice. *Storm* offers a direct solver algorithm and two iterative solver algorithms: the *A*lternate *D*irections *I*mplicit (ADI) method and the *I*ncomplete *L*ower-*U*pper (ILU) decomposition method.

This section briefly describes each of the methods available.

Direct Method

The direct linear equation solution method solves the linearized system of equations, which in matrix notation can be represented as $AX = B$.

To accomplish this goal, the matrix A is first arranged into a banded format with the bandwidth chosen to minimize the storage and computer execution time required. For example, in a two-dimensional problem with dimensions *idm* by *jdm*, the equations would be ordered:

- In columns when *idm* is smaller than *jdm*
- By rows when *idm* is greater than *jdm*.

Next, the matrix A is expressed by way of a lower-upper decomposition such that $A = LU$.

The solution of $AX = B$ then reduces to a forward elimination, $LY = B$, and a backward substitution, $UX = Y$.

An important consideration in this approach is the fact that for two different systems of equations, the same LU decomposition can be applied as long as the matrix of coefficients A remains unchanged, even if matrix B is different; thus LU need not be recalculated, thereby saving computer time. This advantage is exploited intensively in *Storm* whenever possible. For example, the same LU decomposition applies to both (first and second) pressure corrector steps.

The direct method is very efficient for small bandwidth matrices. It is therefore recommended for most one- and two-dimensional problems. It is also recommended for three-dimensional problems that have small values for any two of the three dimensions (for example, 5 x 5 x 100). For full three-dimensional problems, however, the storage and the computational costs associated with the direct method tend to be prohibitive and its use is not recommended.

Alternating Direction Implicit Method

The Alternating Direction Implicit (ADI) method is a semi-iterative method in which the equations are solved at each step by maintaining full implicitness in one direction at a time, while relaxing the requirement in the other two directions (Peaceman and Rachford, 1955; Douglas, 1955). For each direction a tri-diagonal linear system results that can be solved very efficiently. This method is recommended for all equations solved by *Storm* except for the pressure equation used in the PISO solution algorithm. This limitation arises from the observation that ADI methods often perform poorly for ill-conditioned elliptic systems like that produced by the pressure equation (see, for example, Anderson *et al*., 1984, pp.136-37).

Incomplete LU Factorization Method

The incomplete factorization method (also called the incomplete LU decomposition) attempts to follow the LU decomposition of the direct solver, while maintaining the sparseness of the L and U matrices to save storage (Meijerruk and van der Vorst, 1981). Thus, A is again decomposed as LU, but only the non-zero locations in A are retained in L and U, making the decomposition "incomplete" or "approximate."

For instance, in a two-dimensional situation with a five-point stencil, the matrix A is given by

$$(Au)_{i,j} = a_p u_{i,j} + a_E u_{i+1,j} + a_w u_{i-1,j} + a_N u_{i,j+1} + a_S u_{i,j-1} \tag{28}$$

In this case, the L and U operators are given by

$$(Lu)_{i,j} = b_p u_{i,j} + b_W u_{i-1,j} + b_S u_{i,j-1} \tag{29}$$

$$(Uu)_{i,j} = u_{i,j} + c_E u_{i+1,j} + c_N u_{i,j+1} \tag{30}$$

This method is recommended for solution of the pressure equation for full three-dimensional cases. It typically performs better than ADI because of its greater simultaneous implicitness in all three directions.

Time Step Treatment

Storm uses a time-marching algorithm to obtain a solution. The size of the time step used to advance the calculation is therefore of paramount importance for the stability and convergence characteristics of the code. In this section, the mathematical consequences of user-entered data are discussed.

Stability Consideration

The Strength of Convection Effects

The governing equations of unsteady fluid flow are a mixed set of hyperbolic-parabolic equations. The inviscid part of the equations is hyperbolic, and the time-step size is usually characterized by the nondimensional *Courant-Friedrichs-Lewy* (CFL) number, which specifies the propagation of disturbances in the flow field. The CFL number is defined

$$CFL = \frac{U_c . \Delta t}{\Delta x} \tag{31}$$

where U_c is the local velocity in each cell, *Dt* is the time step, and *Dx* is the cell size. As a result, for a user-specified CFL number, a time step size is associated with each cell in the domain, defined

$$\Delta t_{CFL} = \frac{\Delta x \cdot (CFL)}{U_c} \tag{32}$$

Note that *Dt*$_{CFL}$ is *inversely* proportional to the flow speed and *directly* proportional to the cell size. For one-dimensional flows, CFL = 1 is usually an appropriate choice, whereas for two-dimensional flows, a somewhat smaller value (for example, CFL = $1/\sqrt{2}$ = 0.707) may be required.

Diffusion Effects

The steady-state governing equations are elliptic in the viscous region, and the stability of the scheme is dictated by the nondimensional *Von Neumann Number* (VNN). In *Storm*, this criterion is also used to characterize the time scale of the viscous part of the unsteady equations.

The Von Neumann Number is defined

$$VNN = \frac{\Gamma \cdot \Delta t}{\Delta x^2\, r} \tag{33}$$

where Γ is the effective diffusivity (SI unit: kg/m/s) and ρ is the density. Thus, the time step size based on this criterion is defined

$$\Delta t_{VNN} = \frac{\Delta x^2 \cdot \left(VNN\right) r}{\Gamma} \tag{34}$$

Thus, Dt_{VNN} is directly proportional to the *square* of the local cell size, and inversely proportional to the effective diffusivity.

Time Step Control

To control the size of the time step to be used during the course of the calculation, you can choose from three basic options: fixed time step size, automatic time step size, and local time step size.

Fixed Time Step Size

With this option, you can define a single time step value for the entire calculation, or a set of predetermined values that are applied at intervals that you specify. Select this option if you know in advance the time-step values that will meet the stability and convergence requirements of the flow situation under study.

Automatic Time Step Size

With this option you instruct *Storm* to determine time step values as the solution progresses. *Storm* takes into consideration the local characteristics of the flow to determine the adequate size of the time step according to stability criteria that you specify. In determining the optimum size of the time step, the code takes into account the following factors:

- ***Growth Rate***

As a built-in safety measure, the time step size growth from one time step to the next one is not allowed to exceed 3 per cent. Thus, the new time step size according to this consideration is defined as

$$\Delta t_{growth} = 1.03 \Delta t_{old} \tag{35}$$

- ***Upper Limit***

There is also a user-specified maximum time step size that *Storm* observes in determining the size of the next time step to be adopted. At each cycle, the size for the next time step is set to

$$\Delta t_{new} = Min\left(\Delta t_{CFL}, \Delta t_{VNN}, \Delta t_{growth}, \Delta t_{user}\right) \tag{36}$$

- ***Time Step Factor***

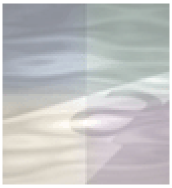
A time factor is used to advance the solution of one variable at a different rate than the one dictated by the time step size described before. Thus, by using numbers less than one but greater than zero, you can slow down the convergence rate of a given group. On the other hand, using factors greater than unity will work as a convergence accelerator within the limits of stability and convergence of the flow.

Because these factors are applied to the true transient term of the conservation equations, they represent a fictitious and hence false time-advancing device. Thus, although the final solution should be independent of these settings, their use may prevent arriving at a time-accurate solution. In other words, these factors should be used mainly for time-independent solutions where time accuracy is not essential.

Although time-accurate solutions are possible for special cases—with careful interpretation—the user should be very cautious when using this technique in attempting time-accurate calculations.

Local Time Step Size

The local time step option is similar to the automatic option, except that rather than using the *same* Δt for all cells in the domain, Δt is allowed to vary on a cell-by-cell basis, based on the *local* CFL or Von Neumann criterion. The local time-step option can yield faster convergence rates for steady-state flows as each cell will be driven towards steady-state with its maximum allowable time step. This option should be used only when time accuracy of the flow solution is not of importance.



Chapter 2

PROPERTY AND PHYSICAL MODELS

Storm can simulate a wide variety of gases and liquids, as well as their thermal interactions with many solids. These materials are distinguished from one another by several parameters that characterize their distinctive properties. These include (for fluids) the density ρ, laminar viscosity μ, the specific heat C_p, the thermal conductivity κ (heat transfer option specified), and the thermal expansion coefficient β (incompressible fluid option specified). The methodologies available in *Storm* for modeling each of these quantities are discussed below.

Density

CFD2000 offers a number of choices to model the density, which may either be specified as a constant value, or prescribed as a function of temperature and pressure, or determined through the use of a custom model. The options available in *CFD2000* are detailed below.

Constant Density

The constant density model may be used for flows where the density is invariant or has very small spatial variations—such as liquids and low-speed, isentropic gas flows. This effectively decouples the density and pressure fields, allowing *Storm* to attain faster convergence.

Ideal Gas Law

The ideal gas law may be used for gas flows where the heat transfer option has been chosen. The ideal gas equation is

$$r = \frac{p + p_{ref}}{R_{GAS}\left(T + T_{ref}\right)} \tag{37}$$

where T_{ref} and p_{ref} define a reference thermodynamic state and R_{GAS} = R^*/m_{GAS} is the specific gas constant, where R* = 8.314 J/K/mole is the universal gas constant and m_{GAS} is the gas molecular weight (kg). The ideal gas law couples the temperature and velocity fields through the density variable. Note that this option is only available when the heat transfer option has been activated in the model.

Isentropic Gas Law

The isentropic gas law may be used when the system is adiabatic (no heat transfer) but density is a function of pressure. In this case, density is modeled

$$r = r_{abs}\left[\frac{p + p_{ref}}{p_{abs}}\right]^{1/g} \tag{38}$$

where γ is the mass-weighted ratio of specific heats (C_p/C_v) at the reference pressure and temperature, and ρ_{abs}, p_{abs} are the absolute density and pressure for the basic thermodynamic state of the system.

Field Value

When the homogeneous gas phase is a mixture of various gaseous species, as in chemically reacting or multi-species flows, it is necessary to compute an effective molecular weight for the homogeneous phase before applying the ideal gas law. The **Field Value** option automatically computes the effective molecular weight as

$$\frac{1}{M_{eff}} = \sum_{i=1}^{N} \frac{X_i}{M_i} \tag{39}$$

where X_i and M_i are the mass fraction and molecular weight of the i^{th} species, respectively. From the effective molecular weight, density is estimated using the ideal gas law as

$$r = \frac{M_{eff}\, p}{RT} \tag{40}$$

Customized Density Models

In addition to the above-mentioned options available for density, *Storm* also allows the user to define customized models for density. Three options are available from the *CFD2000* interface:

- ***Linear Function***

Density can be described as a linear function of any *Storm* scalar variable

$$r = af + b \tag{41}$$

where a and b are user-defined constants and ϕ is the dependent variable.

- ***Inverse Function***

This option allows the user to define density as an inverse function of a *Storm* scalar variable

$$r = \frac{A}{B + Cf} \tag{42}$$

where A, B and C are user-defined constants and ϕ is the dependent variable.

- ***General User-Defined Model***

You can also define density as an arbitrary user-defined function

$$r = f\left(f_i, \mathbf{x}, t\right), i = 1, \ldots n \tag{43}$$

where ϕ_i is any dependent variable, x denotes the physical space coordinate, and t is time.

The user-defined model can be particularly useful, for example, in situations where the ideal gas law is not valid, such as in high-pressure gas flows. In such cases a second-order virial equation of state of the form

$$r = \frac{p}{\left(R_{GAS}T - ap - bp^2\right)} \tag{44}$$

might be more appropriate, where a and b are fluid-specific coefficients that can be specified directly from the *CFD2000* interface.

Viscosity

The laminar viscosity coefficient controls the rate at which momentum is redistributed within the fluid due to molecular (i.e., diffusive) motions. It is an intrinsic fluid property whose value specifies the correlation between the applied tangential stress τ_{ij} on the fluid and the resulting rate of shear (deformation) $\dot{\Gamma}$.

Newtonian Model

For gases and most pure liquids, the relationship between tangential stress and shear strain is linear, as expressed by Newton's law of viscosity

$$t_{ij} = t_{ji} = m\dot{\Gamma} \tag{45}$$

where the molecular viscosity μ is a function of only the thermodynamic state of the fluid (e.g., p and T), and

$$\dot{\Gamma} = \frac{¶u_i}{¶x_j} + \frac{¶u_j}{¶x_i} (i \neq j) \tag{46}$$

is the rate of shearing strain (or strain rate). Fluids which obey Eq. (45) are said to be *Newtonian*. By default, all of the fluids contained in the *CFD2000* Fluid Material Library are assumed to be Newtonian, and are assigned a constant viscosity μ valid at a single reference temperature.

Fluids for which the shear stress is not directly proportional to the strain rate are known as *non-Newtonian* fluids, discussed in the following section.

Non-Newtonian Models

Many liquids, including plastic melts and some suspensions and solutions, display a more complicated, nonlinear relationship between the applied stress and their observed strain rate. In these fluids, some mechanism other than molecular diffusion contributes to the shearing motion and, as a result, the use of a laminar viscosity independent of the rate of strain is no longer valid. Such fluids are called non-Newtonian fluids. Figure 2.1 below shows the relationship between stress and strain for a Newtonian fluid and some classes of non-Newtonian fluids.

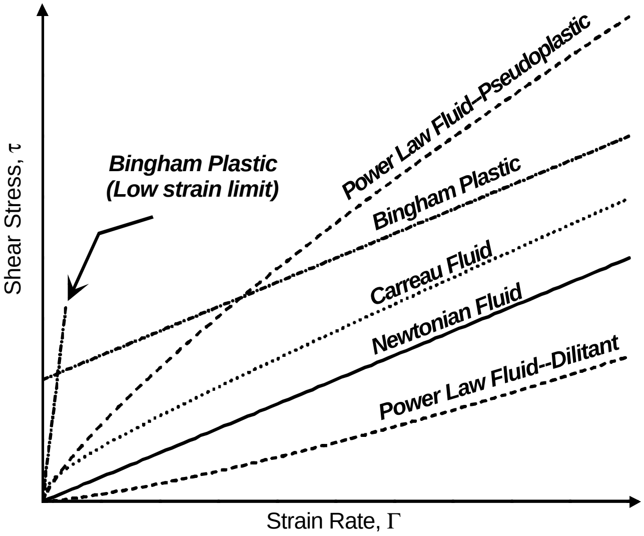


Figure 2.1   Stress-strain relationships for non-Newtonian fluids.

Power Law Model

The power law model (also known as the Ostwald-deWaele, or two-parameter model) is the most general of *Storm*'s non-Newtonian viscosity models. It assumes that the laminar viscosity μ can be expressed as a continuous function of the strain rate according to the formula:

$$m = A\dot{\Gamma}^{B-1} \tag{47}$$

where B is a dimensionless constant called the power-law index, and A is an empirical coefficient known as the consistency factor (SI units: Pa/s^{B} or kg/m/s$^{2\text{-}B}$), which serves as an index of how viscous the fluid is. Note that for B = 1, Eq. (47) reduces to Newton's law (Eq. 45) with A = μ. Thus the deviation of B from unity indicates the degree of non-Newtonian behavior of the fluid. Specifically, when B < 1, the fluid is said to be *pseudoplastic* and the viscosity is found to decrease with increasing strain. On the other hand, when B > 1, the fluid is said to be *dilatant* and the viscosity increases with increasing strain. Dilatant fluids are generally much less common than pseudoplastic ones. Examples of several pseudoplastic fluids involving solid suspensions in water are given in Table 2.1 below.

Table 2.1 Non-Newtonian power law parameter for various fluids at 300K. [a]

	A	**B**
FLUID COMPOSITION (% weight)	**(kg/m/s^{B})**	**--**
4.0% paper pulp in water	20.0	0.575
3.0% carboxymethylcellulose (CMC) in water	9.29	0.566
33% lime in water	7.19	0.171
23.3% Illinois clay in water	5.56	0.229
1.5% CMC in water	3.13	0.554
54.3% cement rock in water	2.51	0.153
0.5% hydroxyethylcellulose in water (293K)	0.84	0.509
0.67% CMC in water	0.304	0.716

[a] *Sources: Bird, Stewart, and Lightfoot, 1960; Tanner, 1985*

Note that when B<1, Eq. (47) predicts infinite viscosity as the strain becomes vanishingly small, and zero viscosity as the strain becomes very large. Despite these limitations, the power law model is nevertheless widely used due to its simplicity, and proves adequate for many common flow problems.

Carreau Model

The Carreau model is based on the observation that many fluids that otherwise display pseudoplastic behaviors at moderate shears become approximately Newtonian when the strain rate is very large or very small [Carreau, 1972; Tanner, 1985]. The functional form of the Carreau model used in *Storm* is

$$m = m_{\infty} + \left(m_{o} - m_{\infty}\right)\left[1 + \left(A\dot{\Gamma}\right)^{2}\right]^{(B-1)/2} \qquad (48)$$

where μ_∞ and μ_0 are the asymptotic viscosities at large and small strain rates, respectively; and A and B are fluid-specific constants determined, for example, by plotting the observed viscosity as a function of strain rate on a log-log plot. The Carreau model is particularly well-suited for certain dilute, aqueous, polymer solutions and melts [e.g., Tiu and Tam, 1989]. Table 2.2 below lists representative model parameters for several Carreau fluids.

Table 2.2 Non-Newtonian Carreau model parameters for various fluids.[a]

	m_{0}	$m_{¥}$	A	B
FLUID	kg/m/s	kg/m/s	s^{-1}	-
Phenoxyl-A at 485K	12400.	0.	7.44	0.728
High-density polyethylene at 443K	8920.	0.	1.58	0.496
5% polystyrene in Aroclor 1242	101.	0.059	0.84	0.380
0.75% Separan-30 in 95/5 mixture (by weight) of water/glycol	10.6	0.010	8.04	0.364

[a] *Source: Tanner, 1985.*

Bingham Fluid Model

Bingham fluids (also called Bingham plastics) include a broad range of real liquids that display a finite yield stress. Common examples include slurries, paints, and suspensions of clays in water. *Storm* uses a model based on the Perzyna hypothesis

$$m = Min\begin{cases} Am_{\infty} \\ m_{\infty} + \dfrac{t_{yield}}{\dot{\Gamma}} \end{cases}, \qquad (49)$$

where A is an arbitrary, dimensionless multiplier supplied by the user (typically, $A = 10^3 \sim 10^5$), μ_{∞} is the viscosity in the limit of very large strain (the fully plastic limit), and τ_{yield} is the yield (threshold) stress for deformation. Equation (49) indicates that the material is essentially Newtonian at very low strain rates (although the viscosity is very high), and becomes plastic once a threshold shear is exceeded (See also Fig. 2.1). This artifice is introduced so that the stress can be modeled as a piecewise continuous function of the strain rate, without the discontinuity inherent in “pure” Bingham plastics at the yield point.

Table 2.3 Non-Newtonian Bingham model parameters for aqueous nuclear fuel slurries.[a]

		VOLUME FRACTION SOLIDS	PLASTIC VISCOSITY[b]	YIELD STRESS
			$m_{¥}/m_{W}$	t_{yield}
SUSPENDED MATERIAL	SOLID CONTENT	-	-	N/m^2
UO_2, Coarse	Thin	0.3	221.	58.2
(D_p = 1.4 mm,	Medium	0.5	8100.	449.
s = 1.7)c	Thick	0.7	297000.	1730.
ThO_2, Very Fine	Thin	0.3	1340.	1420.
(Dp = 0.03 mm,	Medium	0.5	163000.	6590.
s = 2.7)	Thick	0.7	19800000.	18100.
ThO_2, Coarse	Thin	0.3	36.6	42.7
(Dp = 2.4 mm,	Medium	0.5	403.	198.
s = 1.7)	Thick	0.7	4450.	542.

[a] *Source: Bird, Stewart, and Lightfoot, 1960, Table 1.2-1.* [b]*Viscosity relative to heavy water at the same temperature and pressure.* [c]*Lognormal particle size distribution parameters, where D_p is the mean particle diameter and s is the nondimensional standard deviation about the mean.*

Examples of the Bingham parameters and τ_{yield} for two aqueous slurries are presented in Table 2.3 above. Note that these parameters depend not only on the “thickness” of the slurry (expressed here in terms of the volume fraction of the suspended solid particles), but also upon the mean size and the size distribution of the particles themselves.

Customized Viscosity Models

In addition to the above viscosity models, *Storm* also allows the user to define customized models. Three options are available from the *CFD2000* interface: a *linear function, inverse function,* or general *user-defined* model.

- ***Linear Function Model***

Viscosity may be defined as a linear function of a *Storm*-computed or stored variable

$$m = af + b \tag{50}$$

where a and b are user-defined constants and ϕ is the dependent variable.

- ***Inverse Function Model***

Viscosity may also be described as an inverse function of any dependent variable

$$m = \frac{a}{b + cf} \tag{51}$$

where a, b and c are user defined constants and ϕ is the dependent variable.

- ***General User-Defined Model***

Under this option, viscosity may be defined as an arbitrary user-defined function

$$m = f\left(f, x, t\right) \tag{52}$$

where ϕ is any dependent variable, x denotes the physical space coordinate, and t is time.

A constant viscosity approximation is appropriate for many fluids and flow conditions. However, for gas flows in which large temperature fluctuations are expected, it may be more appropriate to employ the user-defined laminar viscosity option and account for the temperature dependence explicitly. For example, one could use Sutherland's equation and compute the laminar viscosity according to the relation

$$m(T) = m\left(T_{ref}\right)\frac{T_{ref} + A}{T + A}\left(\frac{T}{T_{ref}}\right)^{3/2} \qquad (53)$$

where μ (T_{ref}) is the viscosity at the reference temperature T_{ref} and A is a gas-specific constant (Sutherland's constant) with units of absolute temperature.

Specific Heat

The specific heat at constant pressure C_p is an intrinsic material property (SI units: J/kg/deg K) which must be specified whenever the heat transfer option is selected. *CFD2000* provides various options to prescribe C_p.

Constant Value

By default, *Storm* assumes that C_p is constant and uniform throughout the flow, with a value either specified by the user or consistent with the fluid selected from the Fluid Material Property Library.

Field Value

When the homogeneous gas phase is a mixture of various gaseous species—as in chemically reacting flows or multi-species flows—it is necessary to compute an effective specific heat for the homogeneous phase. The **Field Value** option automatically computes the effective specific heat as:

$$C_{p,\ eff} = \sum_{i=1}^{N} X_i \ C_{p,i} \qquad (54)$$

where x_i and $C_{p,i}$ are, respectively, the mass fraction and the specific heart of the i^{th} species.

General User-Defined Specific Heat Model

However, for flows in which the temperature varies over a wide range, it is recommended that the user supply a customized expression for C_p. *Storm* allows for C_p to be defined as an arbitrary function of the form

$$C_p = f\left(f, \bar{x}, t\right) \tag{55}$$

where, for example, ϕ may represent temperature and pressure. Temperature-dependent C_p models for various gases can be found, for example, in Van Wylen and Sonntag [1976] or Potter and Somerton [1993].

Thermal Conductivity

The thermal conductivity κ is an intrinsic material property (SI units: w/m/deg K). *CFD2000* provides various options to define the thermal conductivity of a fluid or solid. In many practical cases, the thermal conductivity is either a constant or a function of temperature.

Constant Value

Storm assumes thermal conductivity, by default, to be uniformly distributed with a constant value either supplied by the user, or consistent with the particular fluid/solid chosen for the simulation. Values of κ for the fluids in the Fluid Material Property Library are valid at a single reference temperature and pressure—usually 300K and 1 atm.

Prandtl Number

CFD2000 also offers the option to specify a Prandtl number Pr for the flow. If this option is chosen, then the thermal conductivity is computed from the Prandtl number as

$$k = \frac{mC_p}{\mathrm{Pr}} \tag{56}$$

Note that this option requires that the **Heat Transfer** option be specified (and, therefore, that C_p be defined).

Field Value

When the homogeneous gas phase is a mixture of various gaseous species, as in chemically reacting flows, it is necessary to compute an effective thermal conductivity for the homogeneous phase. The **Field Value** option auto-matically computes the effective thermal conductivity of the fluid based on the molecular weights and constitution of the individual species in the mixture as

$$k_{eff} = \sum_{i=1}^{N} X_i k_i \tag{57}$$

where X_i and κ_i are the mass fraction and thermal conductivity of the i^{th} species, respectively.

Thermal conductivity usually varies with temperature, so in cases where the temperature varies significantly, the user may prefer to activate one of the model options from the **Thermal Conductivity** options subpanel and prescribe a simple temperature dependent model. *CFD2000* offers the following choices:

- ***Linear Function***

Thermal conductivity can be defined as a linear function of any *Storm* scalar variable

$$k = Af + b \tag{58}$$

where A and β are constants and ϕ is the scalar variable. Temperature-dependent conductivity models for various gases and liquids can also be found, for example, in Weast (1987).

- ***Inverse Function***

Thermal conductivity may also be defined as an inverse function of a *Storm* scalar variable as

$$k = \frac{a}{b + fc} \tag{59}$$

where a, b and c are user-defined constants and ϕ is the scalar variable.

- ***General User-Defined Conductivity Model***

Thermal conductivity may also be defined as an arbitrary user-defined function

$$k = f\left(f, x, t\right) \tag{60}$$

where ϕ is any dependent variable, x denotes the spatial coordinate, and t is time.

Thermal Expansion Coefficient

Storm assigns fixed default values for this parameter valid at 300 K for most fluids. The user may specify a value for the thermal expansion coefficient or decide to use the value for a particular fluid type from the Fluid Material Property Library.

Ratio of Specific Heats

The ratio of the specific heats γ is a key parameter for compressible flows, and must also be specified before any of the isentropic density models can be used.

Constant Value

The user may either specify a constant value for γ, or decide to use the prescribed value for a given fluid (gas) from the Fluid Material Property

Library. Note that if the **Compressibility** option has not been activated, γ can still be specified as a parameter from the **Isentropic Gas Law Density Model** subpanel.

Field Value

When the homogeneous gas phase is a mixture of various gaseous species, as in chemically reacting flows or multi-species, it is necessary to compute an effective ratio of the specific heats for the homogeneous phase. The **Field Value** option automatically computes this effective ratio as

$$g_{p,eff} = \sum_{i=1}^{N} X_i g_i \tag{61}$$

where X_i and γ_i are the mass fraction and the specific heat ratio for the i^{th} species, respectively.

General User-Defined Model

The ratio of specific heats may be defined as an arbitrary function as

$$g = f(f, x, t) \tag{62}$$

where ϕ is any scalar variable, x is the physical space variable, and t is time.

CFD2000 Materials Property Library

CFD2000 includes a built-in Material Property Library that contains a database of the various thermophysical quantities needed in CFD modeling. Table 2.4 below lists the fluids (gas and liquid) available in this library. Note that for most of the gases, three different models are offered. These correspond to the principal density models available in *CFD2000*—namely, the constant density, the ideal gas, and the isentropic flow models.

Table 2.4 Fluid materials in the CFD2000 Material Property Library.

MATERIAL	FORMULA	STATE	NUMBER OF MODELS
Air	Mixture	Gas	3
Ammonia	NH_3	Gas	3
Argon	Ar	Gas	1
Butane	C_4H_{10}	Gas	1
Carbon Dioxide	CO_2	Gas	3
Carbon Monoxide	CO	Gas	3
Engine Oil (SAE 30, unused)	-	Liquid	1
Ethylene	C_2H_4	Gas	2
Ethylene Glycol	$C_2H_6O_2$	Liquid	1
Freon 12	CCl_2F_2	Liquid	1
Glycerin	$C_3H_8O_3$	Liquid	1
Helium	He	Gas	3
Hydrogen	H_2	Gas	3
Mercury	Hg	Liquid	1
Methane	CH_4	Gas	3
Neon	Ne	Gas	3
Nitrogen	N_2	Gas	3
Oxygen	O_2	Gas	3
Steam	H_2O	Gas	2
Water	H_2O	Liquid	4

TURBULENCE MODELING

Turbulence modeling in *CFD2000* is performed using a two-equation k-ε model. This model solves transport equations for the turbulence kinetic energy *k*, and the dissipation rate *e*. The turbulent shear stresses in the Reynolds-averaged Navier-Stokes equations are then modeled using the Boussinesq hypothesis with an appropriate relation for the eddy or turbulent viscosity, based on the computed values of *k* and *e*.

The turbulent viscosity coefficient μ_T plays a role similar to the laminar viscosity in *Storm*, except that it controls the rate at which momentum is redistributed due to *turbulent* eddy motions rather than by molecular diffusion. As such, it is not an intrinsic fluid property, but is rather a space- and time-dependent quantity whose value depends entirely on the local turbulent characteristics of the flow.

k-e Turbulence Model

The k-ε turbulence model is one of several two-equation models that have developed over the years. It is probably the most widely and thoroughly tested of them all (Nallasamy, 1987). Based on simple dimensional arguments concerning the relationship between the size and the energetics of individual eddies in fully developed, isotropic turbulence, the model employs the following diagnostic equation for the turbulent viscosity (Launder and Spalding, 1974)

$$m_T = \frac{C_m r k^2}{e} \tag{63}$$

where C_μ is a dimensionless model constant, ρ is the local fluid density, and k and ε are the *specific turbulent kinetic energy* (SI units: m^2/s^2) and *turbulent kinetic energy dissipation rate* (SI units: m^2/s^2), respectively. These quantities are in turn computed using a pair of auxiliary transport equations of the form

$$\frac{\P\, rk}{\P\, t}+\frac{\P\left(ru_i k\right)}{\P\, x_i}=\frac{\P}{\P\, x_i}\left[\left(m+\frac{m_t}{\mathrm{Pr}_k}\right)\frac{\P\, k}{\P\, x_i}\right]+m_T G-re+S_{k,\,p} \qquad (64)$$

and

$$\frac{\P\, re}{\P\, t}+\frac{\P\left(ru_i e\right)}{\P\, x_i}=\frac{\P}{\P\, x_i}\left[\left(m+\frac{m_T}{\mathrm{Pr}_e}\right)\frac{\P\, e}{\P\, x_i}\right]+\frac{e}{k}\left(C_1 m_T G-C_2\, re\right)+S_{e,\,p} \qquad (65)$$

where C_1 and C_2 are additional dimensionless model constants; Pr_k and Pr_ε are the turbulent Prandtl numbers for kinetic energy and dissipation, respectively; $S_{k,p}$ and $S_{\varepsilon,p}$ are source terms for the kinetic energy and turbulent dissipation; and

$$G=\left(\frac{\P u_i}{\P x_j}+\frac{\P u_j}{\P x_i}\right)\frac{\P u_i}{\P x_j}-\frac{1}{r^2}\frac{\P r}{\P x_j}\frac{\P p}{\P x_j}-\frac{2}{3}\left(\frac{rk}{m_T}+\frac{\P u_i}{\P x_j}\right)\frac{\P u_j}{\P x_j} \qquad (66)$$

is the turbulent production rate. Values for the various dimensionless constants used in Eqs. 1-4 are given in Table 2.5 below.

Table 2.5 Dimensionless constants for Storm k-epsilon turbulence model.

Cm	C_1	C_2	Pr_k	Pr_e
0.09	1.44	1.92	1.0	1.3

These values are identical to those recommended by Launder *et al.*, 1972, based on an evaluation of several plane turbulent free jet and mixing layer simulations. As such, they represent a good “consensus” parameter set, and can be assumed to provide a model accuracy in the range from about 10% to 50%, depending on the flow (Launder and Morse, 1979).

Despite the wide applicability of the k-ε turbulence model, the user should nevertheless be aware of its limits. In particular, it should be kept in mind that all two-equation turbulence models—the k-ε model included—are valid only in situations where the assumed linear relationship between turbulent shearing stresses and the resultant mean strain rate (as in Eq. 45, for example) is appropriate (i.e., where the primary influence of the turbulent mixing is a downgradient or diffusive momentum transport). Although this is usually the case for most turbulent flows, there are situations, especially in highly complex flows involving asymmetric wakes, strong convection, or combustion, where this assumption can break down locally and invalidate the k–ε approach (Nallasamy, 1987).

A second caveat stems from limitations inherent in the transport equations used to predict k and ε. Eqs. 64 and 65 are in fact only strictly valid at high Reynolds number—that is, for fully developed turbulence—and for turbulence that is isotropic (independent of direction). As a result, the *Storm* k–ε turbulence model cannot be expected to be equally applicable to all flow regimes, and cannot in particular be expected to accurately simulate the entire transition from laminar to turbulent flow (e.g., immediately adjacent to walls or other solid boundaries), or the turbulent mixing characteristics of rapidly swirling or strongly stratified flows.

In turbulent flows, the ability of the fluid to transmit heat internally is enhanced somewhat due in part to the fact that turbulent eddies generally have length scales that are much larger than the molecular mean free path. Consequently, whenever the **Turbulent Flow** option is activated, an additional quantity known as the turbulent thermal conductivity κ_T is defined for use in the thermodynamic energy equation. In *Storm*, κ_T is defined relative to κ according to the expression

$$k_T = \frac{k}{\mathrm{Pr}_T} \tag{67}$$

where Pr_T is the *turbulent Prandtl number*. By default, *Storm* assumes Pr = 0.9 for all fluids independent of turbulent strength.

General User-Defined Model

In addition to the standard k-epsilon model, *CFD2000* offers the ability to define a custom model of the form,

$$m_T = f\left(f,\mathbf{x},t\right) \tag{68}$$

where ϕ is any scalar variable in *Storm*, x is the spatial variable, and t is time.

POROUS MEDIUM MODELING

Fluid flows through porous material are very common in industrial processes. The porous medium is categorized as *isotropic* if resistance to the flow does not vary with the flow direction, and *anisotropic* if the flow resistance depends on flow direction. Porous regions are also commonly used to model flows in which geometry is otherwise too complicated to resolve with a practical number of grids such as filtered screens.

In *Storm*, the momentum equation for flow through a porous medium has the following form in the i-direction:

$$\frac{\partial\left(\rho u_i\right)}{\partial t} + \nabla\cdot\left(\rho V u_i\right) + \left[\frac{\mu}{\kappa_i} + \frac{C_i\,\rho\left|\vec{V}\right|}{\sqrt{\kappa_i}}\right]u_i = \nabla_i p + \nabla\cdot\mu_i + \rho f_i\,, \qquad (69)$$

where κ_i is the _direction-dependent permeability of the porous medium and C_i is the nonlinear momentum loss coefficient in the i-direction. If C_i and κ_i are the same for all directions, the porous region is isotropic—otherwise, it is anisotropic. If the transient, convection, diffusion, and body force terms in Eq. 69 are dropped, the equation reduces to

$$-\nabla_i p = \left[\frac{\mu}{\kappa_i} + \frac{C_i\,\rho\left|\vec{V}\right|}{\sqrt{\kappa_i}}\right]u_i \qquad (70)$$

which is the anisotropic form of the Forcheimer-Brinkman model. If $C_i = 0$ and κ_i is isotropic, the result is Darcy's equation,

$$-\nabla_i p = \left[\frac{\mu}{\kappa_i}\right]u_i \qquad (71)$$

Note that the off-diagonal terms of the permeability tensor have been assumed to be zero. Thus, the principal axes of the permeability tensor should coincide with the grid directions.

CHEMICALLY REACTING FLOWS

The coupling between chemical reactions and fluid flows in complex geometries is of interest to users wishing to simulate gas turbine combustors, chemical reactors, and power generation devices such as internal combustion engines and rocket propulsion systems. When considering such a simulation, two factors should be considered:

- The speed of the reaction, and
- The characteristic time scale of the flow itself.

Reacting flows are classified according to the relative magnitudes of these time scales, and these analysis dictates which type of chemical reaction model should be used. *CFD2000 Version 3.0* features six different types of chemical reaction models:

- A multi-step, *finite-rate* chemistry model
- A *mixture fraction* model
- An *instantaneous* chemistry model
- An *equilibrium* model
- A “*frozen*” reaction model, and
- A *chemical vapor deposition* model.

Brief descriptions of each of these models and their typical uses are given below.

Finite Rate Chemistry Model

The *CFD2000 finite rate chemistry model* is the most generally applicable of the six choices offered under Version 3.0. It is particularly suited for situations in which the chemical reactions cannot be completed over the duration of a single computational time step, or when the range of reaction time scales is very large. In such situations, the time required for molecular collisions within the flow must be taken into account, and reaction models based on the theory of chemical kinetics must be developed. Mathematically, this requires the solution of a set of conservation equations which describe the transport of species within the fluid, coupled with source terms derived from chemical kinetics models that describe the production and destruction of species within the flow. These kinetic source term models, however, cause the equations to become nonlinear and highly "stiff", necessitating the use of a specialized numerical treatment to solve the species equations (Curtics and Hirschfelder, 1952).

CFD2000 uses an "operator splitting" solution technique (Rizzi and Bailey, 1965; Eklund *et al.,* 1986; Chen *et al.,* 1994) to solve the finite-rate species conservation equations. This technique, which effectively overcomes much of the difficulty introduced by the stiffness of the governing equations, splits the solution at each discrete model time interval into a two-step predictor-corrector process. In the predictor phase, effective reaction rates, based on an application of the law of mass action and the Arrhenius formula, are computed for the current model time step. Using these effective rates, the species conservation equations are integrated one step forward in time (using a fully implicit integration scheme) with the convection and diffusion terms *omitted.* The result is a tentative prediction of the species concentrations at the new time step. This prediction is then itself updated (the corrector step) by integrating the conservation equations once more, this time with convection and diffusion *included*. Any heat released or absorbed during the reactions is then added to the total flow energy, the time step is advanced one increment, and the process repeats.

Your *CFD2000 Version 3.0* installation includes a library of several pre-defined finite rate reaction models. As indicated in Table 2.6, most of these reactions involve at least two "steps" (individual reactions), and a few require thirty or more. Required inputs for these models include, for each reaction, the individual reaction equations, the reaction collision coefficient A, the collision factor exponent n, the activation energy E, and the values of any third body coefficients M. You can inspect the values of these inputs for any of the predefined models directly from the *CFD2000* interface, or by referring to the **react1.lib** text file included with your installation. Users wishing to add their own finite rate models to the chemical reaction library can do so by appending the relevant stoichiometric and kinetic data (in the proper format) to this file, and by adding the appropriate thermodynamic data (heat capacities, enthalpies, Gibbs energies, etc.) to file **species.lib**. Contact Adaptive Research for assistance.

Table 2.6. Finite rate Chemical Reaction Library models.

	Number of-		
REACTION NAME	**REACTION STEPS**	**SPECIES**	**ELEMENTS**
H_2 + O2	2	4	2
H_2 + Air	2	5	3
H_2 + Air	7	7	3
$H_2 + O_2$	7	6	2
H_2 + Air	9	7	3
$H_2 + O_2$	9	6	2
H_2 + Air	32	13	3
C_2H_4 + Air	10	10	4
$C_2H_4 + O_2$	10	9	3
H_2 + Air	16	10	3
$CH_4 + O_2$	45	15	3
Air Dissociation	32	12	2
Air Dissociation	11	7	2
$CH_4 + O_2$	4	6	3
CH_4 + Air	4	7	4
$C_3H_8 + O_2$	4	6	3
C_3H_8 + Air	4	7	4
Liquid $C_{12}H_{26}$ + Air	1	5	3
Liquid $C_{13}H_{28}$ + Air	1	5	3
Rarefied Gas	10	9	4
$SiH_4 + H_2$	2	4	2
$Si_4 + H_2$	20	24	2
NOx Formation	5	11	4

Mixture Fraction Model

The *CFD2000 mixture fraction* model is a specialized model intended primarily for oxidation (e.g., combustion) simulations involving reactant gases injected from one or more nozzles into a common domain. Its use is limited to *single-step* reactions with *finite* (non-zero) rates that are nevertheless fast enough to be completed within one model (fluid) time step.

The name "mixture fraction" stems from the fact that this model treats the influx of gases at each inlet boundary as a mixture of two generic components: (1) the reactant gas (or *fuel*), which itself may be composed of a mixture of several chemical species, and (2) the carrier gas (typically, air). These components are assumed to enter the model domain completely mixed, with the relative *mass fraction* of fuel to air crossing the inlet boundary per unit time designated by the user-specified quantity *F*. Thus, F—along with the mass flux *Mdot* (or velocity) and the temperature—constitutes an extra boundary condition that must be set for each inlet whenever the mixture fraction model is used.

NOTE In its delivered configuration, *CFD2000 Version 3.0* allows fuel to enter the model domain through at most *two* inlets when using the mixture fraction model. However, this restriction can be easily relaxed through the use of user coding. Contact Adaptive Research for assistance.

Once the inlet boundary conditions have been set, the mixture fraction model employs a two-step, predictor-corrector technique similar to that used in the finite rate chemistry model, although the steps are applied in reverse order. In the first step, the local concentration of fuel at each grid location within the model domain is determined by solving a *single* species conservation equation for F with all sources and sinks set to zero. Once F is known, the local concentration of each species is calculated by referring to the local density and the relative mass fractions of each component as dictated by the reaction stoichiometry. This provisional prediction is then revised (corrected) by temporarily "freezing" the effects of convection and diffusion and applying the reaction chemistry at each grid point. (Note, however, that unlike the finite rate chemistry model, the mixture fraction model assumes that the reaction can be completed in its entirety within a single time step.) Any heat released during the reaction is then added to the total flow energy, the time step is advanced one increment, and the process repeats.

Your *CFD2000* installation includes a built-in library of three mixture-fraction, fuel-air reaction models (Table 2.7). As in the finite rate model, each of these reactions is defined by the three Arrhenius coefficients *A*, *n*, and *E*, as well as its case-specific reaction equations. Additions to this model library can be made by appending your specific "fast chemistry" reaction data to file **react4.lib**. Contact Adaptive Research for assistance.

Table 2.7. Mixture fraction, instantaneous, and equilibrium reaction library models.

	Number of …		
REACTION NAME	**REACTION STEPS** [a]	**SPECIES**	**ELEMENTS**
H_2 + Air	1	5	3
CH_4 + Air	1	7	4
C_3H_8 + Air	1	7	4

[a] *Mixture fraction and instantaneous models only.*

Instantaneous Reaction Model

When the chemical reactions in the flow are so fast relative to the model time step that the reaction time is effectively zero, the *instantaneous reaction* model can be applied.

The instantaneous model is similar in many ways to the mixture fraction model. Both are intended primarily for fast oxidation simulations, and both require that the relative mass fraction of fuel to air at each inlet be prescribed through a boundary condition F. However, unlike the mixture fraction approach, the instantaneous model discards the kinetic source terms altogether and instead relies entirely upon the stoichiometric formula to obtain the proper balance of fuel and reaction products. Such reactions are always carried to completion (stoichiometric balance) at each grid point as long as there is sufficient fuel present after the first (predictor) time step. If, however, the reactant mass is not stoichiometric, then the reaction process is limited by the exhaustion of the fuel species and the yield is diminished proportionally.

The *CFD2000* chemical reaction library contains three built-in instantaneous reaction models. These model reactions are the same ones available in the mixture fraction model library, and are based on identical stoichiometry (Table 2.7). The only difference is that no kinetic (Arrhenius) coefficients are provided in the input file **react2.lib**.

Equilibrium Chemistry Model

The *equilibrium chemistry* model can be used for reactions that are able to proceed in both directions (i.e., reactant to product, or product to reactant), and that are fast enough that they can be treated as effectively instantaneous. Mixtures characterized by such reactions tend to maintain a chemical equilibrium everywhere within the fluid based on a state of minimized Gibbs free energy. As a result, the products of such reactions can be determined solely by the thermodynamic state variables (for example, temperature and pressure) of the system.

The *CFD2000* equilibrium chemistry model computes, for each species and at each grid point in the model domain, a *chemical potential function* based on the local temperature, pressure, and species concentration (determined from the mixture fraction *F).* These potential functions are, in turn, incorporated into a system of *N* nonlinear algebraic mass conservation equations, where N is the number of individual atomic elements involved. Assuming that the cumulative molar concentration of elements in the mixture remains constant, these equations are then solved and a provisional estimate of the updated species concentrations is obtained. Local temperatures and effective gas constants are then recalculated, and the whole system is solved again. The process then repeats from the beginning and continues until convergence is obtained.

Your chemical reaction library contains three built-in reactions based on the equilibrium chemistry approach. These are the same three fuel-air reactions found in both the mixture fraction and the instantaneous chemistry model libraries (Table 2.7). User-supplied additions to the equilibrium chemistry library can be appended to file **react3.lib**.

Frozen Reaction Model

The frozen reaction model is essentially a multi-species mixing model in which all reaction rates are artificially set to infinity, effectively eliminating ("freezing") the chemical component of the fluid flow. This model can be used to compute the concentrations of up to twenty individual components within the model domain, as well as the effective (mass-weighted) heat capacity and specific heat ratio of the bulk mixture. Required inputs are the mass fluxes of each species at one or more inlets (a boundary condition), plus the appropriate thermodynamic data for each species (included in **species.lib**).

Chemical Vapor Deposition Model

Chemical vapor deposition (CVD) is a process in which chemical reactions in a gas and on the surface of an adjacent solid substrate are used to induce the growth of a thin, solid film directly onto the surface. Its most common application is in the electronic component industry, where it used to form the thin silicon wafers that comprise most semiconductor devices and integrated circuits. But it is also finding increased use as a general technique for creating high temperature ceramic coatings (Galasso, 1991).

As implemented in *CFD2000,* the chemical vapor deposition model contains two distinct components: (1) a finite-rate, multi-step, gaseous-phase chemistry model, and (2) a *surface reaction* model. These models must be specified individually by the user, and they must be compatible with each other.

Currently, *CFD2000* includes just two gaseous phase reaction models suitable for CVD, both of which are intended for simulations of silicon deposition from a mixture of silane (SiH_4) and molecular hydrogen. By selecting either of these models, *CFD2000* automatically activates an additional term in the species conservation equation that simulates the effect of thermal-gradient driven mass diffusion (the *Soret* term). Kinetic and stoichiometric data for both CVD reaction models is contained in the file **react1.lib** (the finite rate chemistry input file), and the Soret term is modeled in the user-accessible Fortran file **ucvddif.f**.

CFD2000 also contains a library of models for several types of surface reactions, each of which can be classified as being either a *catalytic* or a *combustion* type reaction (Table 2.8). In catalytic reactions, the solid substrate acts only to induce (catalyze) the gas phase reaction and remains chemically inert. (Such reactions are also sometimes referred to as adsorption reactions.) On the other hand, in combustion surface reactions, the substrate actually participates in reaction itself, such as in the case where an oxidizing gas passes over a solid fuel surface.

Table 2.8 Surface reaction library models.

		Number of-		
REACTION NAME	**TYPE**	**REACTION STEPS**	**SPECIES**	**ELEMENTS**
Copper + N_2	Catalytic	1	2	1
Copper + O_2	Catalytic	1	2	1
Copper + Air	Catalytic	2	4	2
Carbon + O_2	Fast combustion	1	4	2
Carbon + O_2	Slow combustion	1	2	2
SiH_4 deposition	Catalytic (CVD)	2	4	2

In most CVD reactors, deposition is initiated by heating the substrate—either by thermal conduction or by the dissipation of high frequency radio energy. Thus the types of surface reactions relevant for CVD modeling clearly fall into the catalytic category. Currently, only one CVD surface reaction model of this type is available from the *CFD2000* reaction library. This model—a two-step, four-species, finite rate catalytic model—is intended for use in simulations of Si deposition from silane gas. As such, it serves as the appropriate complement to either of the two SiH_4+H_2 finite-rate, gaseous phase CVD models available from the library.

Input data for all *CFD2000* surface reaction models is contained in file **react5.lib**. This file contains, for each model, the appropriate kinetic and stoichiometric data for each reaction. It also includes, for each of the two steps involved in the CVD surface reaction model, a pair of additional coefficients used to model the temperature-dependent *sticking coefficient*. Users wishing to modify any of this data may do so by editing **react5.lib** or, if an additional surface reaction model is desired, by appending the appropriate data. Contact Adaptive Research for assistance.

LAGRANGIAN PARTICLE TRACKING

Two-phase flows in which discrete particles are dispersed in a continuous fluid occur in many scientific and engineering applications. They are of critical importance in fields such as rocket and turbojet propulsion, chemical processing, and pollution control. Transfer of mass, momentum, and energy between the continuous fluid and the particles plays an important role in determining the characteristics of the flows.

Two-phase flow simulations use either an Eulerian or a Lagrangian approach for modeling the particulate phase. The Eulerian approach treats the particle phase as a second continuous fluid, and both phases interpenetrate. The Eulerian approach is appropriate for flows containing very fine particles with diameters in the micron or sub-micron range (e.g., smoke or powder dust). It has relatively high computational efficiency, especially for monodispersed systems. The Lagrangian approach treats particles as discrete entities interacting with a turbulent or laminar continuous fluid flow and calculates their individual trajectories; the Eulerian approach is appropriate for polydispersed particle sprays. *CFD2000* uses the Lagrangian approach.

The two-phase flow simulation in *CFD2000* is a fully interactive combination of Eulerian fluid and Lagrangian particle flow calculations. Interactions between the continuous phase and the particulate phase are complex. Ordinary differential equations describe the particle trajectories, deriving their coefficients from the conditions in the fluid phase at the particle locations. In continuous phase calculations, the influence of the particles is transmitted through source terms added to the Navier-Stokes equations. A global iterative process adjusts the continuous phase source terms and the particle phase coefficients.

Table 2.9 on the facing page shows the allowable combinations of fluid and particle material states. Particle size distributions can be specified, and models for size distribution changes due to evaporation, breakup, or collision and coalescence are provided.

Table 2.9 Material phase combinations.

CONTINUOUS PHASE	PARTICULATE PHASE
Gas	Liquid
Gas	Solid
Liquid	Gas
Liquid	Liquid
Liquid	Solid

Particle Phase

Particles enter the flow at particle injection inlets, which are designated boundaries defined for the continuous flow domain. These boundaries may be interior or exterior to the flow domain and may be of any type; they need not correspond to fluid inlets.

A set of initial characteristics for particles entering the flow is associated with each injection inlet. Particle characteristics are size, velocity, temperature, and material phase and properties. At each injection inlet the particles may have a specified uniform size, or individual sizes may be assigned by sampling on a selected distribution function. Similarly, particle initial velocities may be uniform or may be assigned by sampling on a specified injection velocity profile.

After injection, the motion of each particle is calculated by solving the Lagrangian equation of motion. Optionally, particle temperature may be calculated by solving a heat transfer equation, or by assuming equilibrium with the continuous phase local temperature. Also optionally, changes in size for liquid particles may be calculated by applying models for evaporation, breakup, or collision and coalescence.

Particle Size Distributions

Most engineering multiphase flow applications have particles that are non-uniform in size and shape. Complete descriptions of particle ensemble geometric characteristics require a number of distribution functions. These are obtained experimentally as spectra over discrete measurement bins and are fitted with analytic functions having one or more correlation constants. Specification of the distribution functions and their correlation constant values are sufficient to describe the particle ensemble.

In particle tracking simulations, it is usual to describe the particle ensemble by a single distribution function relating the number of particles to a characteristic size. *CFD2000* allows selection of either the *Rosin-Rammler distribution function* or the χ*-Squared distribution function.* (Uniform and user-defined particle size distributions are also available as additional options.)

The *Sauter Diameter*, which is defined as the ratio of the particle volume to the particle area, is commonly used as the characteristic size in engineering applications. However, some published data refer to actual particle diameter, and either of the two sizes may be specified in *CFD2000*. In either case, the primary constant defining a specific particle ensemble is the mean particle characteristic size. This is defined as

$$\overline{D} = \frac{\int_0^\infty D f(D) dD}{\int_0^\infty f(D) dD} \tag{72}$$

where D is characteristic particle size, f(D) = dF(D)/dD and F(D) is the cumulative number of particles in the distribution having sizes between zero and D.

For a liquid particle, the Sauter Diameter is determined by particle deformation and varies in accordance with forces acting on the particle.

Rosin-Rammler Distribution

The Rosin-Rammler distribution function can be expressed in terms of the cumulative fraction of total particle volume occupied by particles with sizes between zero and D as

$$\frac{dQ}{dD} = \frac{qD^{q-1}}{X^{q}} \exp\left[-\left(\frac{D}{X}\right)^{q}\right] \tag{73}$$

where

$$\frac{X}{\overline{D}} = \Gamma\left(1 - \frac{1}{q}\right) \tag{74}$$

$\overline{\mathrm{D}}$ is the mean particle size for the distribution, and X and q are correlation constants related to $\overline{\mathrm{D}}$ through the function Γ. This volume distribution is converted into a number distribution through

$$\frac{dN}{dD} = \frac{p}{6} D^{-3} \frac{dQ}{dD} \tag{75}$$

χ-Squared Distribution

The normalized number distribution function for the χ-Squared distribution is

$$f(D) = \frac{D^{3}}{3\overline{D}^{4}} \exp\left(-\frac{D}{\overline{D}}\right) \tag{76}$$

and the corresponding cumulative distribution function is

$$F(D) = 1 - \exp\left(-\frac{D}{\overline{D}}\right)\left[1 + \frac{D}{\overline{D}} + \frac{1}{2}\left(\frac{D}{\overline{D}}\right)^{2} + \frac{1}{6}\left(\frac{D}{\overline{D}}\right)^{3}\right] \tag{77}$$

Equation of Motion

The equation of motion for a particle in a fluid continuum with variable velocity (Soo, 1967) is

$$\frac{p}{6} d_p^3 r_p \frac{dv_i}{dt} = \frac{p}{8} d_p^2 r C_D \left| v_i - u_i \right| (v_i - u_i) - \frac{p}{6} d_p^3 \frac{\P p}{\P x_i} + \frac{p}{12} d_p^3 r \left(\frac{du_i}{dt} - \frac{dv_i}{dt} \right) + \frac{3}{2} d_p^2 \sqrt{prm} \int_{t_0}^{t} dt \, \frac{\frac{du_i}{dt} - \frac{dv_i}{dt}}{\sqrt{t-t}} + \frac{p}{6} d_p^3 r_p F_{bi} \tag{78}$$

where the five terms on the right hand side in order are drag force, pressure gradient force, virtual mass force, the "Basset" term for the effect of non-uniform flow, and the gravity or other body force. Here d_p is the particle diameter, ρ_p is the particle density, v_i is the ith component of the particle velocity, u_i is the ith component of the fluid velocity, C_d is the drag coefficient, p is the pressure in the continuous phase, τ is the particle relaxation time, and F_{bi} is the body force per unit mass on the particle.

Usually the drag and gravity forces dominate the particle motion, and the approximate form of the equation of motion is

$$\frac{dv_i}{dt} = \frac{U_i + u_i' - v_i}{t} + F_{bi} \tag{79}$$

where U_i is the ith component of fluid velocity, and u_i' is the fluctuating ith component of the fluid velocity. Here τ, the particle relaxation time, is defined by

$$t^{-1} = \frac{3}{8} \frac{r}{r_p r_p} C_D \left| U_i + u_i' - v_i \right| \tag{80}$$

where r_p is the particle radius, and the drag coefficient C_D is calculated as

$$C_D = \begin{cases} \frac{24}{\mathrm{Re}_p}\left(1+\frac{1}{6}\mathrm{Re}_p^{2/3}\right) & \text{if } \mathrm{Re}_p \le 1000 \\ 0.424 & \text{if } \mathrm{Re}_p > 1000 \end{cases} \tag{81}$$

where Re_p is the particle Reynolds number, defined as

$$\mathrm{Re}_p = \frac{\left|U_i + u_i' - v_i\right| r d_p}{m} \tag{82}$$

The particle position x_i is determined by integrating the equation of motion to obtain the velocity v, then integrating the equation

$$\frac{dx_i}{dt} = v_i \tag{83}$$

Particle Breakup Models

Storm provides two particle breakup model options: the *Taylor Analogy Breakup (TAB)* model and the *Wave Instability* breakup model.

TAB Model

The TAB model (O’Rourke and Amsden, 1987) is based on an analogy between an oscillating and distorting liquid droplet and a spring-mass system. The spring restoring force is analogous to the surface tension and the external force on the mass is analogous to the gas aerodynamic force. Damping forces due to fluid viscosity are added to the analogy. A deformation parameter *y* is defined as the displacement of the droplet surface from its equilibrium position, divided by the equilibrium radius. The equation for acceleration of the *y* parameter is

$$\frac{d^2y}{dt^2} = \frac{2}{3}\frac{r}{r_p}\frac{(U+u'-v)^2}{r_p^2} - \frac{8a(T_p)}{r_p r_p^3}y - \frac{5m_l(T_p)}{r_p r_p^2}\frac{dy}{dt} \tag{84}$$

where α is the liquid surface tension coefficient and μ_1 is the liquid viscosity, both of which depend on the particle temperature T_p. Integration of the above equation gives the time-dependent distortion

$$y(t)=\frac{We}{12}+\exp\left(-\frac{t}{t_p}\right)\left[y(0)-\frac{We}{12}Coswt+\frac{1}{w}\left(\dot{y}(0)+\frac{y(0)-\frac{We}{12}}{t_p}\right)Sinwt\right] \qquad (85)$$

where

$$We=\frac{r(U+u'-v)}{a}$$ is the Weber number,

$$t_p=\frac{2}{5}\frac{r_p r_p^2}{m_l}$$ is the viscous damping time, and

$$w^2=8\frac{a}{r_p r_p^3}-\frac{1}{t_p^2}$$ is the square of the oscillation frequency.

The droplet oscillation and breakup computations require two normalized particle arrays (*y* for deformation and dy/dt for oscillation). Droplet breakup occurs if and only if deformation exceeds the particle equilibrium radius (y(t)>1). Following breakup, the product drop has the same temperature as the parent drop, and its deformation and oscillation parameters are set to zero.

Wave Instability Model

This model is based on wave stability analyses of liquid particles (“blobs”) ejected from a nozzle (Reitz and Diwakar, 1986). The initial size of the blobs is the nozzle diameter. From the wave dispersion equation for a round jet, the wavelength of the fastest growing wave Λ and the maximum wave growth rate Ω are calculated as

$$\frac{\Lambda}{r_j} = 9.02 \frac{(1 + 0.45Z^{0.5})(1 + 0.40T^{0.7})}{(1 + 0.87We^{1.67})} \tag{86}$$

and

$$\Omega\left(\frac{r_p r_j^3}{a}\right)^{0.5} = \frac{0.34 + 0.38We^{1.5}}{(1 + Z)(1 + 1.4T^{0.6})} \tag{87}$$

Here $Z = We_p^{0.5} / \mathrm{Re}_p$ and $T = ZWe^{0.5}$.

The mean product droplet radius and breakup rate are given by:

$$r_p = \begin{cases} B_0 & \text{if } B_0\Lambda \le r_j \\ \min\left[\left(\frac{3pr_j^2}{2\Omega}\right)^{1/3}, \left(\frac{3r_j^2\Lambda}{4}\right)^{1/3}\right] & \text{if } B_0\Lambda > r_j \end{cases} \tag{88}$$

and

$$\frac{dr_j}{dt} = -\frac{r_j - r_p}{3.726B_1 r_j}\Lambda\Omega \tag{89}$$

Values for the model constants B_0 and B_1 in the initial nozzle atomization process and the subsequent particle breakup process are shown in Table 2.10 on the next page.

Table 2.10 Model constants in wave instability model.

CONSTANT	NOZZLE AUTOMIZATION PROCESS	SECONDARY DROPLET BREAKUP
B_0	0.61	0.61
B_1	10.0	1.73

Particle Collision and Coalescence

Storm uses a particle collision model established by O'Rourke (1981) to calculate the droplet collision and coalescence. Collisions are calculated for particles with locations within a single computational cell in the continuous flow domain. The particles associated with each parcel are assumed to be uniformly distributed throughout the cell.

A collision frequency *n* between a parcel with radius r_1 and another with radius r_2, $r_1 \geq r_2$, is

$$v = \frac{N_2^n}{dV^n} p \left(r_1^n + r_2^n\right)^2 \left|v_1 - v_2\right| \tag{90}$$

where N_2 is the number of particles in parcel 2, $(v_1 - v_2)$ is the relative velocity between parcels, and dV is the volume of the cell. The probability P_n for n collisions is assumed to obey the Poisson distribution

$$P_n = e^{-\bar{n}} \frac{\bar{n}^{n}}{n!} \tag{91}$$

where $\bar{n}$ is the mean value $(\bar{n} = v\Delta t)$ and Δt is the time step. The number of collisions, n, is determined by sampling on the Poisson distribution.

The collision impact parameters are calculated stochastically. If the collision impact is less than a critical impact parameter, all n collisions are treated as coalescence; if the impact is greater, all n collisions are treated as grazing collisions. The critical impact parameter for a specific collision depends on particle radii, relative velocities, and the surface tension coefficient. Grazing collisions, which usually occur between particles of nearly equal size, are calculated between N pairs of particles, where

$$N = \min(N_1^n, N_2^n) \tag{92}$$

Turbulence Modulation Model

Turbulence in the continuous phase is modified by the presence of the particulate phase. This effect is represented in the continuous phase simulations by the source terms $S_{k,p}$ and $S_{\varepsilon,p}$ in the turbulent transport equations for kinetic energy and dissipation, respectively. Shuen (1984) obtained expressions for these source terms by using the momentum equation with the source terms $S_{u_i,p}$ and $S'_{u_i,p}$ representing particulate phase effects on continuous phase momentum and turbulence fluctuation velocity

$$S_{k,p} = \overline{u_i S_{u_i,p}} - U_i \overline{S_{u_i,p}} \tag{93}$$

and

$$S_{e,p} = 2m \overline{\frac{\P u'_i}{\P x_j} \frac{\P S'_{u_i,p}}{\P x_j}} \tag{94}$$

where the turbulence fluctuation velocity u'_i has a Gaussian distribution.

The continuous phase momentum source term has the form

$$S_{u_i,p} = \frac{1}{dV} \sum_{p=1}^{NP} \left[N_p \dot{m}_{ev,p} (v_i)_p - m_p N_p \left(\frac{U_i + u'_i - v_i}{t} \right)_p \right] \tag{95}$$

where dV is the computational cell volume, NP is the total number of particles, and N_p is the number of particles for each computational parcel.

The turbulence source terms are then

$$S_{k,p} = \frac{1}{dV} \sum_{p=1}^{NP} \left[N_p \dot{m}_{ev} \overline{u'_i (v'_i)_p} - m_p N_p \overline{u'_i \left(\frac{u'_i - v'_i}{t} \right)_p} \right] \tag{96}$$

and

$$S_{e,p} = 1.0 \frac{e}{k} S_{k,p} \tag{97}$$

MOVING GRID

When the computational grid moves as a function of time, the grid velocity enters the analysis and must be included in discretizing the governing differential equations. Basically, the grid motion affects the convective fluxes of mass, momentum, energy, and other scalar dependent variables. In integral form, the continuity and the generalized transport equations can be written as follows:

Continuity Equation

$$\frac{d}{dt}\int_V r\,dV + \int_A r\left(\vec{V} - \vec{V}_g\right) dA = 0 \tag{98}$$

General Transport Equation

$$\frac{d}{dt}\int_V r\,f\,dV + \int_A r\,f\left(\vec{V} - \vec{V}_g\right) dA = \int_A \Gamma\nabla f\,dA + \int_V S\,dV \tag{99}$$

where V is an arbitrary moving volume, A is the surface of V, ϕ is any scalar quantity, and $\Gamma\nabla\phi$ and S are the diffusive flux and source terms for the corresponding variable.

In *Storm*, after you specify the characteristics of the grid motion, the mass and other convective fluxes across the cell faces are calculated according to the local fluid flow conditions and grid velocity. The cell volume, face area, and face direction cosines are recalculated at every time step.

CONJUGATE HEAT TRANSFER

In many flow problems, there are solid objects within the computational domain. Though fluid cannot penetrate the solid-fluid interface, heat can be transferred through the interface and conducted inside the solid objects. In this circumstance, mass and momentum equations are solved in the fluid side only, but the energy equation is applied to both the fluid and solid regions. Because the solid-fluid interface requires attention to ensure appropriate conservation of energy, conjugate heat transfer analysis was developed.

Basically, at the solid-fluid interface the following two conditions need to be met:

$$k_f \left. \frac{\partial T_f}{\partial n} \right|_i = k_s \left. \frac{\partial T_s}{\partial n} \right|_i \tag{100}$$

and

$$\left. T_f \right|_i = \left. T_s \right|_i \tag{101}$$

where the subscripts f and s are for fluid and solid respectively, and subscript i indicates interface. The two conditions state that the heat flux and temperature across the fluid-solid interface are continuous. In general, the fluid and solid objects have different specific heats $C_{p,f}$ and $C_{p,s}$. Therefore, enthalpy across the interface is not generally continuous.

Your *CFD2000* installation contains a library of several solid materials that can be used in conjugate heat transfer analyses. For each of these materials included in the Solid Material Property Library, three parameters are stored: the solid density, the specific heat, and the thermal conductivity. Table 2.11 on the facing page lists all of the substances contained in the Solid Material Property Library, as well as the default values for each of the three parameters. As in the case with the Fluid Material Library, you may modify any of these values, or define and store your own material properties in the library.

Table 2.11 Solids in CFD2000 Solid Material Property Library.[a]

MATERIAL	DENSITY kg/m^3	SPECIFIC HEAT J/kg/degK	CONDUCTIVITY W/m/degK
Aluminum (pure)	$2.707x10^3$	897.2	203.
Apple	$0.840x10^3$	3600.	0.513
Asbestos	$0.577x10^3$	1050.	0.16
Banana	$0.980x10^3$	3350.	0.481
Brass	$8.522x10^3$	385.	104.
Brick (common)	$1.600x10^3$	840.	0.69
Bronze	$8.650x10^3$	343.	26.
Chromium	$7.200x10^3$	451.4	93.7
Coal (anthracite)	$1.300x10^3$	1260.	0.26
Concrete	$2.240x10^3$	880.	1.13
Copper (pure)	$8.954x10^3$	384.9	384.
Corkboard	$0.160x10^3$	1700.	0.043
Glass-Pyrex	$2.640x10^3$	800.	1.09
Glass-Window	$2.483x10^3$	770.	1.09
Gold	$18.880x10^3$	129.7	315.
Ice (273 K)	$0.913x10^3$	1930.	2.22
Iron (pure)	$7.897x10^3$	450.2	71.4
Lead	$11.373x10^3$	129.6	34.6
Nickel (pure)	$8.906x10^3$	443.4	90.3
Paper	$0.930x10^3$	2500.	0.13
Paraffin	$0.900x10^3$	2890.	0.02
Plastic (Acrylic)	$1.185x10^3$	1470.	0.21

Table 2.11 Solids in CFD2000 Solid Material Property Library.[a]

	DENSITY	SPECIFIC HEAT	CONDUCTIVITY
MATERIAL	kg/m^3	J/kg/degK	W/m/degK
Plastic (PVC)	1.714x10^3	1050.	0.092
Platinum	21.450x10^3	132.6	71.6
Rock (Granite)	2.650x10^3	824.	3.4x10-3
Rock (Shale)	2.73x10^3	753.	1.9x10-3
Rubber(hard, 273 K)	1.200x10^3	2000.	0.156
Sand	1.515x10^3	800.	0.027
Silicon	2.330x10^3	713.9	148.
Silver (pure)	10.524x10^3	234.5	417.
Soil (coarse)	2.050x10^3	1.84	0.52
Stanless Steel (AISI-34	7.910x10^3	460.	13.8
Tin (pure)	7.304x10^3	217.7	64.1
Wood (oak)	0.820x10^3	2400.	0.28
Wood (pine)	0.500x10^3	2800.	0.17
Zinc (pure)	7.144x10^3	388.6	111.

[a] *Sources: Edwards et al., 1979; Chase et al., 1985; Weast, 1987; Ekert and Drake, 1972; Meyers, 1971.*

RADIATION

CFD2000 uses a six-flux radiation model to simulate radiative heat transfer. Basically, it solves the following transport equation in the i-direction:

$$-\frac{d}{dx}\left[\frac{1}{a+s}\frac{d}{dx_i}R_i\right]=S_u \qquad (102)$$

where a is the absorption coefficient, s is the scattering coefficient, and

$$S_u=\begin{cases} abT^4-aR_i \quad for1-D \\ \left(\frac{s}{2}\right)R_j+abT^4-\left(a+\frac{s}{2}\right)R_i \quad for2-D \\ \left(\frac{s}{3}\right)\left(R_j+R_k\right)+abT^4-\left(a+\frac{2}{3}s\right)R_i\ for3-D \end{cases} \qquad (103)$$

with *b* as the Stefan-Boltzmann constant, which is 5.6696e-8 $\frac{w}{m^2k^4}$.

Due to the absorption of radiant energy, the above radiation fluxes result in a net source for the energy equation, which is

$$\begin{array}{l} 2aR_i-2abT^4R_j \quad for1D \\ 2a\left(R_i+R_j\right)-4abT^4 \quad for2D \\ 2a\left(R_i+R_j+R_k\right)-6abT^4\ for3D \end{array} \qquad (104)$$

SPECIAL TERMS

In general, it is not necessary to solve the set of governing equations specified by Eqs. 1-4 in Chapter 1 in their entirety for all classes of flows. Depending on the type of flow, some of these terms may be neglected. By default, *Storm* does not include some of the influences available as “switch-on models” from the *CFD2000* interface. These are listed and detailed below.

Swirl Flow

In certain classes of axisymmetric flows, the swirl component of velocity may have a non-zero velocity, though there are no gradients in the circumferential direction. In these situations, the problem may still be computed in two dimensions by solving the transport equations for the three components of velocity for a 2-D, axisymmetric geometry. This can be performed by activating the **Swirl Flow** option.

Compressibility

For most low-speed flows, the tendency (time) term in the continuity equation has a negligible influence on the overall solution and by default, *Storm* does not include this term in the governing equations. However, in the case of high speed (M>0.3), compressible flows, this is an important contribution and hence needs to be included in the simulation. This compressibility term can be explicitly activated from the *CFD2000* **Analysis Specifications** panel for such flows.

Viscous Dissipation

In certain classes of flows, there is a source term in the energy equation due to viscous stresses. By default this effect, called viscous dissipation is not included in the energy. It has to be explicitly activated when viscous dissipation effects are significant.

The *Brinkman number*, a non-dimensional parameter given by the product of the Prandtl and Eckert numbers as

$$Br = \frac{mC_p}{k}\frac{u^2}{C_p \Delta T} = \Pr Ec \qquad (105)$$

is used to determine whether the temperature rise due to dissipation is significant. Qualitatively, the Brinkman number represents the ratio of dissipation effects to fluid conduction effects; a Brinkman number of order unity or larger means that viscous dissipation has to be included in the simulation.

Pressure Work

The third term on the right-hand side of the energy equation (Eq. 4) is often referred to as the *Pressure Work* term. This is usually a significant influence in high-speed flows with large velocity gradients and hence is not included in the energy equation in *Storm* by default. It can however be activated when modeling, for example, compressible flows.

Transient Pressure term

For unsteady flow problems involving moving pressure waves, it is important to include the effects of the unsteady pressure derivative in the energy equation (the second term on the right-hand side of Eq. 4). This term is included in the energy equation in *Storm* whenever the **Unsteady Flow** option is selected.

Body Forces

Body forces in *CFD2000* refer to forces that act on the entire mass of the fluid element such as gravity. *CFD2000* allows for a number of options to include body forces.

Constant

A constant force that acts on the fluid over the entire domain may be specified with this option.

Gravity

The **Gravity** option allows the user to specify the value of acceleration due to gravity in the appropriate unit system to compute the effective body force.

Buoyancy Forces

A body force of the form

$$F = A(\rho - B) \tag{106}$$

where A and B are constants may be specified using the **Buoyancy Force** option.

Boussinesq Model

A special extension of the constant density model is available whenever both the heat transfer and gravity options have been activated. In this situation, convective motions can arise even when the density variations are very small. The user can account for this effect while still retaining most of the advantages of the constant density model by choosing the Boussinesq force option from the gravity boundary condition menu. When this option is selected, density is treated as a constant in the dynamical equations—*except* where it is coupled with the gravitational acceleration. Thus the body force term in the momentum equation (Eq. 1) is cast in terms of a small (linearized) density variation about the basic reference value

$$b_i = g_i r_{ref} b\left(T - T_{ref}\right) \tag{107}$$

where g_i is the component of the gravitational acceleration vector in the i^th^ coordinate direction; ρ_{ref} and T_{ref} are the constant reference density and temperature, respectively; and β is the isobaric *thermal expansion coefficient* defined

$$b = -\frac{1}{r}\left(\frac{\P r}{\P T}\right)_p \tag{108}$$

Values for β have been assigned for every fluid in the *CFD2000* Fluid Material Library. For gases, *Storm* uses the ideal gas relationship

$$b = 1/T \tag{109}$$

where T is in absolute degrees (either Kelvin or Rankine).

REFERENCES

Anderson, D.A., J.C. Tannehill, and R.H. Pletcher, *Computational Fluid Mechanics and Heat Transfer,* McGraw-Hill, New York, 559 pp., 1984.

Bird, R.B., W.E. Stewart, and E.N. Lightfoot, *Transport Phenomena,* John Wiley and Sons, New York, 1960.

Carreau, P.J., “Rheological Equations from Molecular Network Theories,” *Transactions of the Society of Rheology, 16,* 99-127, 1972.

Chase, M.W. *et al.*, JANNAF *Thermochemical Tables*, Third Edition, National Bureau of Standards, Washington, D.C. 1985.

Chen, C.P., Y. Jiang, Y.M. Kim, and H.M. Shang, *A Computer Code for Multiphase All-Speed Transient Flows in Complex Geometries—MAST Version 1.0,* NASA Contract Report, October 1991.

Chen, Z.J., C.P. Chen, and L.C. Chien, *Assessment of a Pressure-based Method for Transient Chemically Reacting Flow*, AIAA Paper 94-0237, January 1994.

Courant, R., E. Isaacson, and M. Rees, “On the Solution of Non-Linear Hyperbolic Differential Equations by Finite Differences,” *Communications on Pure and Applied Mathematics*, 5, 243 pp., 1952.

Curtics, C.F. and J.O. Hirschfelder, “Integration of Stiff Equations,” *Proc. Nat. Acad. Sci., 38*, 235-241, 1952.

Douglas, J., “On the Numerical Integration of $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = \partial u/\partial t$ by Implicit Methods,” *Journal of the Society of Industrial Applied Mathematics,* 3, 42-65, 1955.

Edwards, D.K., V.E. Denny, and A.F. Mills, *Transfer Processes*, Second Edition, McGraw-Hill, New York, 421 pp., 1979.

Eckert, E.R.G., and R.M. Drake, Jr., *Analysis of Heat and Mass Transfer*, McGraw-Hill, New York, 805 pp., 1972.

Eklund, D.R., H.A. Hassan, and J.P. Drummond, *The Efficient Calculation of Chemically Reacting Flow*, AIAA Paper 86-0653, January 1986.

Galasso, F.S., *Chemical Vapor Deposited Materials*, CRC Press, Boston, 1991.

Issa, R.I., "Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting," *Journal of Computational Physics*, 62, 40-65, 1985.

Issa, R.I., B. Ahmadi-Befrui, K.R. Beshay, and A.D. Gosman, "Solution of the Implicitly Discretized Reacting Flow Equations by Operator-Splitting," *Journal of Computational Physics*, 93, 388-410, 1991.

Landau, L.D. and E.M. Lifshitz, *Fluid Mechanics*, 536 pp., Pergamon Press, London, 1959.

Launder, B.E. and A. Morse, *Turbulent Shear Flows I*, 279-294, Springer-Verlag, Berlin, 1979.

Launder, B.E. and D.B. Spalding, "The Numerical Computation of Turbulent Flows, *Computer Methods in Applied Mechanics and Engineering*, 3, 269-289, 1974.

Launder, B.E., A. Morse, W. Rodi, and D.B. Spalding, "The Prediction of Turbulent Free Shear Flows—A Comparison of the Performance of Six Turbulence Models," *NASA Conference on Free Shear Flows*, 361-422, National Aeronautics and Space Administration, NASA Langley Research Center, Hampton, VA, 1972.

Meijerruk, J.A. and H.A. van der Vorst, "Guidelines for the Usage of Incomplete Decompositions in Solving Set of Linear Equations as They Occur in Practical Problems," *Journal of Computational Physics*, 44, 134-155, 1981.

Meyers, G.E., *Analytical Methods in Conduction Heat Transfer*, McGraw-Hill, New York, 508 pp., 1971.

Nallasamy, M., “Turbulence Models and Their Applications to the Prediction of Internal Flows: A Review,” *Computers and Fluids, 15,* 151-194, 1987.

O’Rourke, P.J., *Collective Drop Effects on Vaporizing Liquid Sprays,* Los Alamos Scientific Laboratory Report LA-9069-T, 1981.

O’Rourke, P.J. and A.A. Amsden, *The TAB Method for Numerical Calculation of Spray Droplet Breakup,* SAE Technical Paper 870289, 1987.

Patankar, S.V., *Numerical Heat Transfer and Fluid Flow,* 197 pp., Taylor and Francis, 1980.

Patankar, S.V. and D.B. Spalding, “A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows,” *International Journal of Heat and Mass Transfer, 15,* 1787, 1972.

Peaceman, D.W. and H.H. Rachford, “The Numerical Solution of Parabolic and Elliptic Differential Equations,” *Journal of the Society of Industrial Applied Mathematics,* 3, 28-41, 1955

Peric, M., R. Kessler, and G. Scheuerer, “Comparison of Finite-Volume Numerical Methods with Staggered and Collocated Grids,” *Computers and Fluids,* 16 (4), 389-403, 1988.

Potter, M.C. and C.W. Somerton, *Theory and Problems of Engineering Thermodynamics,* Shaum’s Outline Series, New York, 355 pp., 1993.

Reitz, R.D. and R. Diwakar, *Effect of Drop Breakup on Fuel Sprays,* SAE Technical Paper 860469, 1986.

Rhie, C.M., and W.L. Chow, “Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Separation,” *AIAA Journal*, 21 (11), 1525-1532, 1983.

Rizzi, A.W and H.E. Bailey, “Split Space Marching, Finite-Volume Method for Chemically Reacting Supersonic Flow,” *AIAA Journal,* 14, 621-628, 1965.

Shuen, J.S., *A theoretical and Experimental Investigation of a Dilute Particulate Turbulent Gas Jet*, Ph.D. Dissertation, Pennsylvania State University, May 1984.

Soo, S.L., *Fluid Dynamics of Multiphase Systems*, Blaisdell Publishing Co., 1967.

Spalding, D.B., "A Novel Finite-Difference Formulation for Differential Expressions Involving both First and Second Derivatives," *Internal Journal of Numerical Methods in Engineering,* 4, 551, 1972.

Tanner, R.I., *Engineering Rheology*, Clarendon Press, Oxford, 451 pp., 1988.

Tiu, C. and K.C. Tam, "A Phenomenological Model for Dynamic Properties of Dilute Polymer Solutions," *Journal of Non-Newtonian Fluid Mechanics*, 33, 165-180, 1989.

Van Wylen, G.J. and R.E. Sonntag, *Fundamentals of Classical Thermo-dynamics,* Wiley and Sons, New York, 1976

Weast, R.C., *CRC Handbook of Chemistry and Physics,* CRC Press, Inc., Boca Raton, 1987.

Yee, H.C., "A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods," National Aeronautic and Space Administration, *NASA TM101088,* 1989.

INDEX