Computational Issues in Hybrid Multizonal/Computational Fluid Dynamics Models

F. Bezzo, S. Macchietto, and C. C. Pantelides
Centre for Process Systems Engineering, Imperial College London, South Kensington Campus, London SW7 2BY, United Kingdom

DOI 10.1002/aic.10383
Published online February 23, 2005 in Wiley InterScience (www.interscience.wiley.com).

Hybrid multizonal/computational fluid dynamics (CFD) models provide a means of introducing a more realistic description of fluid mechanics and mixing phenomena within process models. The solution of the CFD submodel implicitly defines a function relating one or more of the variables in the multizonal model (the function outputs) in terms of another subset of the variables (the function inputs). This paper is concerned with the accurate and efficient evaluation of this function using local approximate models (LAMs) that may have a general mathematical structure, or be based on physical correlations. Practical issues relating to the robustness of the solution of hybrid models are also considered, and a general architecture for the software interface between the multizonal and CFD submodels is presented. The effectiveness and efficiency of the overall approach are tested by two applications relating, respectively, to a stirred-tank chemical reactor fitted with a cooling jacket and to a stirred-tank bioreactor. © 2005 American Institute of Chemical Engineers *AIChE J,* 51: 1169–1177, 2005
Keywords: process simulation, CFD, multizonal modeling, local approximate models

Introduction

In a recent article,[1] the authors presented a rigorous mathematical framework for constructing hybrid multizonal/computational fluid dynamics (CFD) models. (In the interests of simplicity of notation in this article, we treat the output Y as a scalar quantity, rather than a vector. This does not result in any loss of generality because the various outputs of the CFD submodel can be approximated independently of each other.) Such models can be viewed as a generalization of the widely used multizonal (or "multicompartment") models that divide the system of interest (such as a chemical reactor or a crystallizer) into a network of well-mixed zones. The important feature of hybrid models is that the flow rates between adjacent zones are computed by a CFD submodel that constitutes total mass and momentum balances only. All other phenomena are included in the well-mixed zone models. The CFD submodel may also be used to compute fluid-mechanical quantities that are necessary for the characterization of these other phenomena (such as the turbulent energy dissipation rate, which affects the rate of nucleation in crystallization processes).

The above work generalizes and unifies much of the work on hybrid multizonal/CFD models presented in earlier literature (see, for example, Urban and Liberis,[2] Bauer and Eigenberger,[3,4] Alexopoulos et al.,[5] Zauner and Jones,[6] Rigopoulos and Jones,[7] Bezzo et al.[8]). However, a problem common to all these hybrid models is the cost associated with the CFD calculations. More specifically, the CFD submodel is essentially embedded within a multizonal process model as a set of functions of the form

$$Y = F(\mathbf{x}) \tag{1}$$

where $\mathbf{x} \in \mathcal{R}^n$ is the set of inputs to the CFD submodel (such as the density and viscosity of the fluid in each zone) and $Y \in \mathcal{R}$ represents any one of its output quantities[1] (such as an

Correspondence concerning this article should be addressed to F. Bezzo at this current address: DIPIC-Università di Padova, via Marzolo 9, 35131 Padova, Italy; e-mail: fabrizio.bezzo@unipd.it.

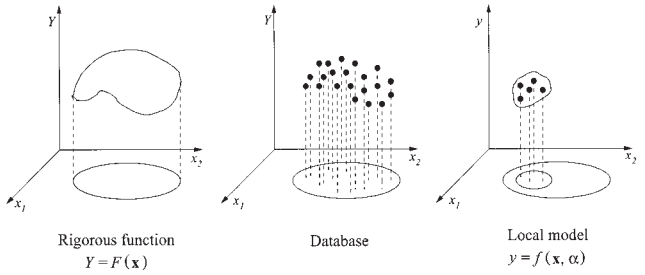

Figure 1. From rigorous to local approximate models.

interzonal flow rate or the volume-averaged turbulent energy dissipation rate for a given zone). During a typical steady-state or dynamic simulation of the multizonal model, the function $F(\cdot)$ may need to be evaluated hundreds of times. Even with some form of "hot-starting" the CFD calculations after the first evaluation, the resulting computational cost may still be prohibitive. Methods such as the in situ adaptive tabulation[9] or the response–surface technique,[10] which are mainly used in reactive flow calculations, are of little use in this case because it is quite impractical to create a priori a comprehensive map of the function $F(\cdot)$ when expensive CFD calculations with large numbers of inputs are involved.

This paper examines ways of addressing the above problem. The basic idea is that a relatively large proportion of the evaluations of the function $F(\mathbf{x})$ are replaced by evaluations of *local approximate models* (LAMs) of the form

$$y = f(\mathbf{x}, \alpha) \tag{2}$$

where the values of the parameters $\alpha \in R^p$ are estimated by fitting the results of evaluations of the original function $F(\cdot)$ carried out earlier during the solution of the hybrid multizonal/CFD model.

In the next section, we present a general algorithm for the derivation and use of LAMs within the context of a hybrid model. The next two sections are concerned with, respectively, LAMs that have a general mathematical structure and LAMs that are based on physical correlations. We then proceed to consider a couple of practical issues relating to the robustness of solution of the hybrid model, and to present a general architecture for the software interface between the multizonal and CFD submodels, combining all the elements presented in the paper. Finally, the overall effectiveness and efficiency of the approach are examined by applying it to two examples presented in earlier literature.

Local Approximate Models

In general, during any solution of the hybrid multizonal/CFD model, we will perform the rigorous CFD calculations several times, gradually building a set ("database") $\mathcal{D}$ of distinct input/output pairs $\{(\mathbf{x}_i, Y_i)\}$. As has already been mentioned, the LAM parameters α are derived by applying an estimation procedure to a subset of the points in $\mathcal{D}$. Thus, a LAM uses a subset of the discrete representation to approximate a portion of the function map (see Figure 1).

The recognition of the benefits of using LAMs instead of a more expensive rigorous model is not new. For instance, they have been widely investigated as a means of reducing the cost of rigorous thermodynamics calculations within process simulation. Leesley and Heyen[11] developed an interpolation scheme to avoid complex algorithms to estimate the phase equilibrium constants in vapor–liquid equilibrium calculations. Chimowitz et al.[12] and Macchietto et al.[13] approached the problem of process design and simulation by proposing simple models locally approximating rigorous and expensive tensor property (TP) property models. Parameters were recalculated using recursive updates similar to those developed in Åström et al.[14] for self-tuning regulators. More recently, Støren and Hertzberg[15] demonstrated the usefulness of local thermodynamics models to accelerate dynamic optimization calculations.

In general, the application of the LAMs considered in this paper involves the following steps:

Given an input vector $\mathbf{x}$:

(1) Based on a appropriate set of criteria, determine a subset $\bar{\mathcal{D}}$ of the points in $\mathcal{D}$ that can be used in the construction of the LAM.
(2) IF the subset $\bar{\mathcal{D}}$ contains the minimum necessary number p of points, THEN
  (a) Using the points in $\bar{\mathcal{D}}$, estimate the LAM parameters α.
  (b) Evaluate the LAM to determine $y = f(\mathbf{x}, \alpha)$.
  (c) If y is sufficiently close to the values of the outputs Y corresponding to the points in $\bar{\mathcal{D}}$, then RETURN output y.
(3) LAM is not applicable:
  (a) Solve the CFD submodel to evaluate $Y = F(\mathbf{x})$.
  (b) Add the point $(\mathbf{x}, Y)$ to the database $\mathcal{D}$ for use in future LAMs.
  (c) Optionally, use the point $(\mathbf{x}, Y)$ to adjust the criteria used at step 1.
  (d) RETURN output Y.

The criteria used in step 1 for determining which points in $\mathcal{D}$ should be used will generally depend on the particular form of LAM being used, and so does the minimum number of points p that are necessary to allow the construction of the LAM (cf. step 2). Unless otherwise stated, a point $(\mathbf{x}_i, Y_i)$ is deemed to be suitable for the construction of a LAM for a given set of inputs $\mathbf{x}$ provided $\mathbf{x}_i$ and $\mathbf{x}$ are suitably close to each other by satisfying the following condition

$$d_i \equiv \sqrt{\sum_{k=1}^{n} \left[\frac{w_k (x_{ki} - x_k)}{x_{ki}} \right]^2} \leq \delta \tag{3}$$

where x_{ki} and x_k are the kth elements of vectors $\mathbf{x}_i$ and $\mathbf{x}$, respectively, and w_k is a given weight reflecting the relative sensitivity of the CFD submodel's output with respect to the kth input.

The parameter δ controls the eligibility of points $\mathbf{x}_i$ in condition 3. Experience shows that it is generally best to start with a relatively small δ and then increase it if and when appropriate at step 3c of the algorithm. In particular, if a rigorous CFD submodel calculation is performed (step 3a), then we can a posteriori construct a LAM based on a subset $\bar{\mathcal{D}}$ containing the p points in $\mathcal{D}$ that are nearest to $\mathbf{x}$ even if these do not fulfill the criterion expressed in Eq. 3. If the relative difference of the

approximated output y and the corresponding exact value Y is found not to exceed a specified tolerance ε, then this implies that the subset $\bar{\mathcal{D}}$ could, in fact, have been used for the creation of a LAM, although this was prevented by δ being too conservative. Consequently, δ can be increased to $\max_{i:(\mathbf{x}_i,Y_i)\in\bar{\mathcal{D}}} d_i$.

To facilitate the parameter estimation step 2a, we limit our attention to LAMs in which all parameters occur linearly. This allows the determination of their values either by the solution of a linear system (if the absolutely minimal number of points are used) or, more reliably, by the solution of a linear least-squares problem (if more than the minimum number of points are used). Recursive least-squares methods are not considered in this paper: although they do not require large databanks, an initialization step for which disturbances are introduced is needed for the variance–covariance matrix to be determined in a robust manner.[16] In our case, the cost of CFD calculations would make this initialization phase too computationally expensive.

Step 2c also deserves some additional explanation. Any LAM is essentially an interpolant to the function $F(\cdot)$. Consequently, we need to treat with caution any output prediction produced by such a LAM that differs widely from the output values in the set $\bar{\mathcal{D}}$ from which the LAM was constructed. In practice, we can do this by calculating the mean $\bar{Y}_{\bar{\mathcal{D}}}$ and standard deviation $\sigma_{\bar{\mathcal{D}}}$ of the outputs Y of all points in set $\bar{\mathcal{D}}$. If the LAM output y is in the range $[\bar{Y}_{\bar{\mathcal{D}}} - 3\sigma_{\bar{\mathcal{D}}}, \bar{Y}_{\bar{\mathcal{D}}} + 3\sigma_{\bar{\mathcal{D}}}]$, then it is accepted as a sufficiently accurate output value. Otherwise, the algorithm proceeds to execute step 3, which involves the complete CFD submodel solution. Such cases typically indicate that the LAM may have "overstretched" itself by making use of points in $\mathcal{D}$ that are too far away from the input of interest $\mathbf{x}$ to yield good predictions; thus it is also appropriate to reset the tolerance δ to its initial (small) value.

Mathematically Based Local Approximate Models

In this section, we consider LAMs that are purely mathematical in nature, that is, they do not attempt to take into account the physical characteristics of the dependency of a particular output Y of the CFD submodel on its inputs $\mathbf{x}$. These LAMs are generally applicable, irrespective of the physical quantity that they have to approximate. Their disadvantage is that they may not be as precise as LAMs based on physical understanding.

The simplest possible mathematically based LAM is a linear function of the form

$$f(\mathbf{x}) = \alpha_0 + \sum_{k=1}^{n} \alpha_k x_k \tag{4}$$

A general-purpose mathematically based nonlinear LAM can be constructed using the well-established Shepard interpolation scheme.[17] This has been used for many diverse applications such as the transfer of solution data from one CFD computational grid to another[18] and the representation of potential energy surfaces in dynamic calculations for electronic structures by using a limited number of accurate data points.[19]

Here, we use the modified Shepard method,[20] which leads to a more efficient and stable interpolation procedure. Given a set of points $\bar{\mathcal{D}}$ to be used in the LAM, the latter is given by

$$\mathbf{f}(\mathbf{x}) = \frac{\sum_{i\in\bar{\mathcal{D}}} W_i(\mathbf{x}) Q_i(\mathbf{x})}{\sum_{i\in\bar{\mathcal{D}}} W_i(\mathbf{x})} \tag{5}$$

where the nodal functions Q_i satisfy $Q_i(\mathbf{x}_i) = Y_i$. Typically (for example, Renka,[21] Carlson and Foley,[22] Iiguni et al.[23]), bivariate quadratic functions or Taylor series expansions are used to define Q_i. Taylor series expansion cannot be used in our case because the complexity of the CFD submodel renders their derivation practically impossible. Instead, we have used a simplified form of the bivariate quadratic functions, which omits terms involving products of two distinct input variables. (Tests that we carried out with and without these terms showed that the loss of accuracy resulting from their omission is negligible.)

$$Q_i(\mathbf{x}) = Y_i + \sum_{k=1}^{n} [\alpha_{ki}(x_k - x_{ki})^2 + \beta_{ki}(x_k - x_{ki})] \tag{6}$$

The nodal function $Q_i(\mathbf{x})$ represents an estimate of the output at point $\mathbf{x}$ based on information pertaining to point $\mathbf{x}_i$ in the database. Equation 5 combines estimates from different points $\mathbf{x}_i$ by relative weights W_i, defined as

$$W_i(\mathbf{x}) = \left[\frac{\max(0, R_w - \|\mathbf{x} - \mathbf{x}_i\|)}{R_w\|\mathbf{x} - \mathbf{x}_i\|}\right]^2 \tag{7}$$

where R_w is a "radius of influence."

We note that $W_i(\mathbf{x}) = 0$ if the distance of point $\mathbf{x}$ from $\mathbf{x}_i$ exceeds R_w; thus, only points within a sphere of radius R_w of $\mathbf{x}$ contribute to the LAM. Thus, for a given value of the radius R_w, we can easily determine the subset of points i that are to be involved in the LAM. We then need to estimate the coefficients α_{ki} and β_{ki} in the corresponding nodal function $Q_i(\mathbf{x})$. These are determined by solving the least-squares minimization problem

$$\min_{\alpha_{ki},\beta_{ki},k=1,\ldots,n} \sum_{\substack{j:(\mathbf{x}_j,Y_j)\in\bar{\mathcal{D}} \\ j\neq i}} \frac{\max(0, R_q - \|\mathbf{x}_i - \mathbf{x}_j\|)}{R_q\|\mathbf{x}_i - \mathbf{x}_j\|} [Q_i(\mathbf{x}_j) - Y_j]^2 \tag{8}$$

where R_q is another radius of influence.

The radii R_w and R_q may be fixed at given values,[20] or they can be chosen to be large enough to include certain minimum numbers of points.[21] In our implementation, we have set

$$R_q = \frac{1}{2} \max_{\mathbf{x}_i:(\mathbf{x}_i,Y_i)\in\mathcal{D}} \|\mathbf{x} - \mathbf{x}_i\| \tag{9}$$

Given that R_q denotes the radius of influence of the data points on the nodal functions, whereas R_w denotes the radius of influence of nodal functions on the interpolating function, it is recommended that $R_w \leq R_q$. We have chosen $R_w = R_q/\sqrt{2}$.[20]

Physically Based Local Approximate Models

Certain fluid-dynamical quantities that are computed by the CFD submodel are also known to be related to the latter's inputs by well-established and relatively simple correlations. The latter may be viewed as LAMs with adjustable parameters that may be estimated by fitting to rigorous CFD results. Moreover, provided the parameters occur linearly, the algorithm presented earlier in this paper is as applicable to these LAMs as it is to general mathematically based ones.

By necessity, physically based LAMs are available for only specific quantities. Two such examples are described in the rest of this section; they relate, respectively, to the fluid-to-wall heat transfer coefficient in a stirred vessel and to the effective viscosity in a non-Newtonian fluid.

Example I: a LAM for stirred-tank heat transfer coefficient

The standard correlations for the fluid-to-wall heat transfer coefficient in a stirred tank are of the form (see, for example, Fletcher[24])

$$\mathrm{Nu} = C(\mathrm{Re})^{\alpha}(Pr)^{\beta} \tag{10}$$

where Nu is the Nusselt number hD/λ with h, D, and λ being the heat-transfer coefficient, tank diameter, and fluid conductivity, respectively; Re is the Reynolds number $(dN^2\rho)/\mu$, with d, N, ρ, and μ being the impeller diameter, impeller rotation speed, fluid density, and fluid viscosity, respectively; Pr is the Prandtl number $(C_p\mu)/\lambda$, with C_p being the fluid specific heat capacity; and C, α, and β are parameters that depend on the particular vessel being studied.

By taking logarithms of both sides, the above can be reformulated to

$$\log(\mathrm{Nu}) = \gamma + \alpha \log(\mathrm{Re}) + \beta \log(\mathrm{Pr}) \tag{11}$$

where $\gamma = \log C$. This can be used as a LAM with three adjustable parameters, α, β, and γ, occurring linearly therein.

We can also derive an alternative local model that is more closely related to the methods used to compute the heat-transfer coefficient within CFD packages. In particular, the heat-transfer coefficient h_c in a cell c adjacent to the vessel wall can be expressed in terms of the following correlation[25]

$$h_c = \frac{\rho_c C_{p_c} C_\mu^{0.25} k_c^{0.5}}{T_c^*} \tag{12}$$

where C_μ is an empirical constant related to viscosity; k_c is the turbulent kinetic energy per unit mass in cell c

$$T_c^* = \mathrm{Pr}_t\left[\frac{1}{\kappa}\log(Ey_c^*) + f(\mathrm{Pr}_c)\right]$$

with

$$f(\mathrm{Pr}_c) = \frac{\pi/4}{\sin(\pi/4)}\left(\frac{A}{\kappa}\right)^{0.5}\left(\frac{\mathrm{Pr}_c}{\mathrm{Pr}_t} - 1\right)\left(\frac{\mathrm{Pr}_t}{\mathrm{Pr}_c}\right)^{0.25} \qquad y_c^* = \frac{\rho C_\mu^{0.25} k_c^{0.5} y_c}{\mu}$$

A, κ, E are constants; Pr_t is the Prandtl turbulent number (assumed to be constant); $\mathrm{Pr}_c = C_{p_c}\mu_c/\lambda_c$; and y_c is the distance from the center of cell c to the wall.

After some algebraic manipulations and taking into account the relation between kinetic energy per mass unit and velocity (that is, $k_c^{0.5} \propto v_c$), the following expression can be obtained:

$$\mathrm{Nu}_c = \frac{A_1 \mathrm{Re}_c \mathrm{Pr}_c}{A_2 + A_3\left[\frac{1}{\kappa}\log \mathrm{Re}_c + f(\mathrm{Pr}_c)\right]} \tag{13}$$

where $A_1 = C_\mu^{0.25}$, $A_2 = \log(E)\mathrm{Pr}_t/k$, $A_3 = \mathrm{Pr}_t\log(EA_1)$, and $\mathrm{Re}_c = \rho v_c y_c/\mu_c$.

Finally, by taking the inverses of both sides and approximating the local (cell) Reynolds and Prandtl numbers by the corresponding vessel quantities, as used in Eq. 10, we obtain

$$\frac{1}{\mathrm{Nu}} = \alpha\frac{1}{\mathrm{Re\ Pr}} + \beta\frac{\frac{1}{\kappa}\log \mathrm{Re} + f(\mathrm{Pr})}{\mathrm{Re\ Pr}} \tag{14}$$

which is linear in the two parameters α and β.

Several tests have demonstrated that this model has the tendency to slightly overestimate Nu. A simpler local model can be derived from it by dropping the first term on the right-hand side, leading to a single-parameter model

$$\frac{1}{\mathrm{Nu}} = \beta\frac{\frac{1}{\kappa}\log \mathrm{Re} + f(\mathrm{Pr})}{\mathrm{Re\ Pr}} \tag{15}$$

Example II: a LAM for non-Newtonian viscosity

The effective viscosity η of a process fluid is an important quantity in many industrial applications. For example, in bioprocesses involving sparging of air into a thick broth, η affects the coefficient for mass transfer of oxygen from the air to the liquid phase.

For non-Newtonian fluids, η is a function of the strain rate $\dot{S}$. For example, in power-law fluids, we have

$$\eta = k\dot{S}^{n-1} \tag{16}$$

where n and k are parameters that depend on fluid characteristics (composition and temperature). Thus, a CFD submodel can readily compute a volume-averaged value of η within each zone in a hybrid multizonal/CFD model.

On the other hand, η can also be estimated by means of general correlations. In the case of stirred-tank reactors, the correlation by Metzner and Otto[26] is of the form

$$\eta = k(aN)^{n-1} \tag{17}$$

where N is the impeller rotation speed and a is a geometric constant. By taking the logarithms of both sides, we obtain a LAM of the form

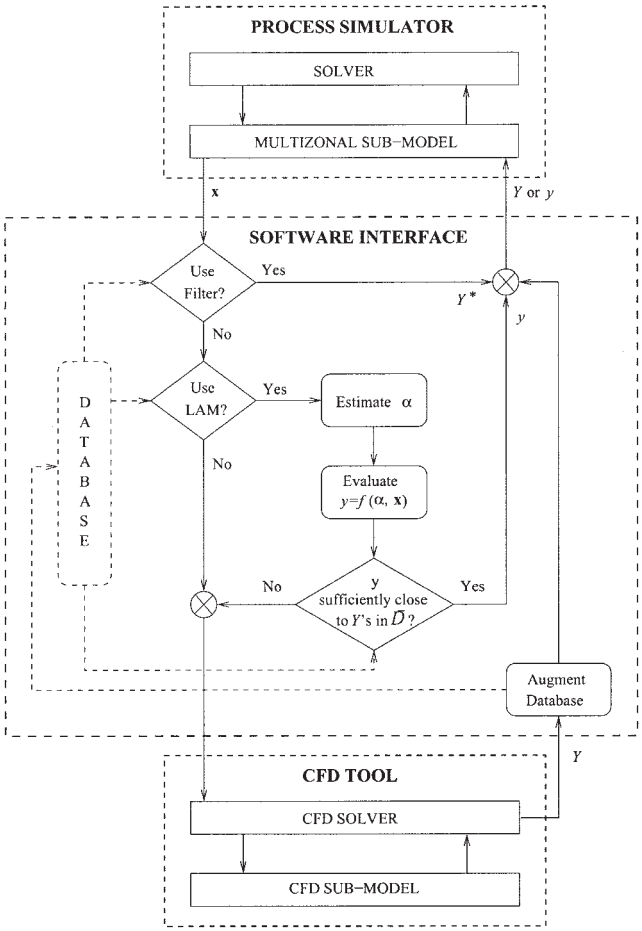

Figure 2. Software architecture for hybrid multizonal/CFD models.

$$\log \frac{\eta}{k} = (n-1)(\log a + \log N) \tag{18}$$

where $\log a$ can be treated as an adjustable parameter to be fitted to rigorous CFD results. A more flexible LAM can be derived by introducing an additional parameter β on the right-hand side

$$\log \frac{\eta}{k} = (n-1)(\log a + \log N) + \beta \tag{19}$$

Ensuring Robustness in Hybrid Calculations

Robustness is a critical issue in simulating a hybrid multizonal/CFD model. In particular, it must be ensured that data coming from the CFD submodel as well as from LAMs do not destabilize the numerical methods of the process simulation package.

Below, we consider two issues related to robustness and present appropriate solutions.

Interzonal flow rate reconciliation

CFD packages typically adopt looser convergence criteria than those routinely used by process simulation tools to solve the multizonal model. As a result, the solution of the CFD submodel may not close the mass balance exactly, and this may have an adverse effect on the performance of the solver applied to the multizonal submodel. It is, therefore, generally advisable to apply a reconciliation procedure to the interzonal mass flow rates computed by the CFD submodel.

If $\hat{F}_{zz'}$ denotes the mass flow rate from zone z to z' computed by the CFD submodel, the objective of the reconciliation is to determine a set of corrected flow rates $F_{zz'}$ that are as close to $\hat{F}_{zz'}$ as possible while satisfying the total mass balance constraint in each and every zone z. This can be achieved by solving the following constrained least-squares minimization problem

$$\min_{F_{zz'}} \frac{1}{2} \sum_{z,z'} (F_{zz'} - \hat{F}_{zz'})^2 \tag{20}$$

subject to the linear mass balance constraints

$$\sum_{\forall z' \neq z} F_{zz'} - \sum_{\forall z' \neq z} F_{z'z} = 0 \qquad \forall\ z \tag{21}$$

The optimal solution can be found by introducing a Lagrange multiplier λ_z for each constraint in Eq. 21 and formulating the first-order optimality conditions which, in addition to Eq. 21 include the equations

$$F_{zz'} - \hat{F}_{zz'} + \lambda_z - \lambda_{z'} = 0 \qquad \forall\ z, z' \tag{22}$$

The optimality conditions form a linear system that can be solved for the reconciled flow rates $F_{zz'}$ and the multipliers λ_z.

Note that, in the interests of retaining physical realism, only interzonal flow rates $F_{zz'}$ that have been determined by the CFD submodel as being nonzero (that is, $\hat{F}_{zz'} > 0$) are included in the above reconciliation procedure.

Filtering of CFD results

Loose convergence criteria in CFD packages may cause some fluctuations in the computed results. The resulting noise may adversely affect the numerical methods used for the solution of the multizonal model, such as those arising from the contradictory outputs received by the predictor–corrector steps in a typical time integrator. The problem is usually quite acute during *initialization* of the process simulation model.

A solution to the above problem can be obtained by means of a simple filter: given a vector of inputs $\mathbf{x}$, CFD rigorous calculations are carried out only if

$$\left| \frac{x_k^* - x_k}{x_k^*} \right| > \varepsilon \tag{23}$$

for at least one k, where ε is a preset tolerance and vector $\mathbf{x}^*$ is the vector of inputs used to compute the last rigorous output Y^*. Otherwise, Y is simply returned to be equal to Y^*. Effectively, the filter substitutes the noisy function $Y = F(\mathbf{x})$ with a piecewise constant function.

Table 1. Cases Considered in Application I

Case	Model	Initial Radius δ	No. of Parameters* p
1	Base	N/A	0
2	LAM Eq. 4	5×10^{-2} (Constant)	$n + 1$
3	LAM Eq. 5	5×10^{-2} (Constant)	At least $2n$
4	LAM Eq. 11	5×10^{-2} (Constant)	3
5	LAM Eq. 14	5×10^{-2} (Variable)	2
6	LAM Eq. 15	5×10^{-2} (Variable)	1
7	LAM Eq. 15	5×10^{-2} (Constant)	1

*n = number of inputs (4 or 5).

Software Architecture for Hybrid Multizonal/CFD Models

The coupling of multizonal and CFD submodels within a single hybrid model has both *structural* and *computational* aspects. The former have already been considered in detail in Bezzo et al.[1] in terms of the mathematical definition of the model that describes each well-mixed zone, and the information that needs to be exchanged between the two submodels.

Here, we turn our attention to the computational procedure that actually handles the information flow, and the overall software architecture that integrates the various functions and algorithms introduced in this paper. As shown schematically in Figure 2, there are three major software components: the process simulator, the CFD tool, and the interface between the two.

The process simulator component incorporates the multizonal model defined by the models of the individual zones and their connectivity, and also a numerical solver (such as that for the integration of sets of differential-algebraic equations over time). The interaction between the solver and the multizonal model requires multiple evaluations of the latter's equations, which, in turn, results in requests being issued to the interface for the evaluation of the function $F(\mathbf{x})$ (cf. Eq. 1) for different values of the input vector $\mathbf{x}$.

The interface starts by applying the filter that determines whether simply to return a previously calculated output value Y^* that already exists in the database (cf. Eq. 23). If this is not appropriate, then it checks whether a LAM is applicable in this case by determining a suitable set $\bar{\mathcal{D}}$ of points from the database. If this is true, then the LAM parameters α are estimated, the LAM output (cf. Eq. 2) is evaluated, and its difference from the outputs of the points in $\bar{\mathcal{D}}$ is assessed. If this difference is not excessive, then y is returned to the process model as the output corresponding to the given inputs $\mathbf{x}$.

If a LAM is not applicable, or if its output y is excessively different from the output values in the set $\bar{\mathcal{D}}$ used for the construction of the LAM, the interface issues a call to the CFD tool. By making use of the CFD submodel, the tool computes the output Y corresponding to the specified input $\mathbf{x}$ and returns it to the interface. The latter adds the pair $(\mathbf{x}, Y)$ to the database and then proceeds to return Y as the desired output to the process simulator.

The user primarily interacts with the process simulator component that has overall control of the calculation. The execution of the CFD tool is initiated automatically by the software interface component just before the first solution of the CFD submodel.

The above architecture is very general, being applicable to all hybrid multizonal/CFD models conforming to the general framework of Bezzo et al.[1] In particular, it is independent of the structure of the specific model under consideration in all important respects, including the management of the information flow, the filtering criteria, the implementation of the LAM algorithm, and the additional robustness measures. For instance, the two examples described in the next section are handled in exactly the same way, although one of them uses a single-zone model of a stirred tank, whereas the other one involves a multizonal model of a bioreactor.

Case Studies

Two applications will be considered to analyze the effectiveness of using LAMs in the dynamic simulation of hybrid multizonal/CFD models. In both cases, the multizonal model is implemented in the gPROMS® v2.0 process modeling tool,[27] whereas the CFD submodel is implemented in Fluent® v4.5.[28] All computations are carried out on an SGI® Onyx® workstation.

Application I

The first test will consider the semibatch esterification reactor originally described in Bezzo et al.[29] The tank is fitted with a double impeller on a centered shaft and a jacket that is used for either cooling or heating. The tank is modeled as a single well-mixed zone, constituting the mass and energy balance equations, and is also coupled with appropriate models of the jacket and the control system.

Because this example involves just one zone, the only output Y of the tank's CFD submodel is the process side heat-transfer coefficient h. This is determined by solving the mass and momentum balances within the tank for given density ρ and viscosity μ of the tank's contents. The specific heat capacity C_p and thermal conductivity κ of the fluid are also required to compute the required heat transfer coefficient (cf. Eq. 12). All four physical properties are computed by the zone submodel from the composition and temperature of the tank's contents. The CFD submodel's input vector $\mathbf{x}$ is defined as $\{\rho, \mu, C_p, \kappa, N\}$, where N is the speed of impeller rotation.

We consider seven distinct cases as listed in Table 1. The base case is a dynamic process simulation that resorts to rigorous CFD calculations whenever it needs to evaluate the function $h(\rho, \mu, C_p, \kappa, N)$; no LAM is used in this case. Cases 2–7 make use of the

Table 2. Performance of Differemt LAMs under Constant Impeller Speed in Application I

Parameter	Base	Mathematically Based LAMs		Physically Based LAMs			
Case	1	2	3	4	5	6	7
A	—	G	G	M	G	G	G
ζ	1.0	12.9	8.3	6.7	37.6	51.2	16.8
$PS\%$	0.03	1.72	2.84	0.44	2.35	2.60	0.80
N_{cfd}	1664	178	203	226	46	35	97

Table 3. Performance of Different LAMs under Variable Impeller Speed in Application I

Parameter	Case 1	Case 2	Case 6
A	—	G	G
ζ	1	3.4	4.0
$PS\%$	0.03	0.60	0.26
N_{cfd}	1950	158	104

LAMs indicated in the second column of the table. The parameter δ used in condition 3 is initially set to the value shown in the third column of the table; thereafter, it is either kept constant or varied, as discussed earlier in this paper.

The performance of the various LAMs is assessed according to the following criteria:

(1) The *accuracy* (A) of the simulation results is classified as *good* (G) if the simulated heat transfer coefficient does not differ by more than 1% from the base case at any point during the dynamic process trajectory; as *medium* (M) if it does not differ by more than 5% from the base case; and as *bad* (B) if it differs by more than 5% from the base case.

(2) The *CPU reduction* achieved by the use of LAMs is assessed by two different parameters. The acceleration factor ζ for a particular case is defined as

$$\zeta = \frac{\text{Total CPU time for base case}}{\text{Total CPU time using LAM}} \tag{24}$$

The parameter $PS\%$ is defined as the ratio

$$PS\% = \frac{\text{Process simulation CPU time}}{\text{Total CPU time}} \tag{25}$$

In addition to the time spent by the process simulator component, the numerator includes the time consumed by the software interface component (that is, effectively all non-CFD computations).

(3) The *numbers of rigorous CFD submodel solutions* (N_{cfd}) are also compared.

As a first test, the impeller rotation rate N is kept constant; thus, there are only four real input variables to the CFD submodel. During the simulation, process viscosity changes by about 50%. Table 2 outlines the performances of different local models.

Case 4 achieves relatively poor accuracy; in fact, the LAM's sensitivity to the inputs $\mathbf{x}$ was found to be significantly different from that of the rigorous CFD model. Nonetheless, the results in Table 2 demonstrate the great effectiveness of the other physical models. Even with a constant δ (Case 7), the local model (Eq. 15) produces an acceleration in calculation time of almost 17 times the base case. When the tolerance is automatically adjusted (Case 6), the simulation becomes over 50 times faster than the base case while still retaining good accuracy.

Another interesting result is the good performance of the simple linear LAM (Case 2). The nonlinear LAM (Case 3) is also both effective and accurate.

In all these simulations, the process simulator CPU time represents a small percentage (well under 3%) of the overall computational time. As expected, CFD calculations are the principal burden, each requiring 30–120 s of CPU time. Thus, the acceleration factor ζ is essentially a direct reflection of the reduced number of rigorous CFD evaluations, as shown in the last row of Table 2.

The base case and the two best performing LAMs (Cases 2 and 6) were also tested in a more demanding simulation where the impeller speed N (in rpm) is varied depending on the temperature T (in K) according to

$$N = \begin{cases} 90 & \text{if } T \le 320 \\ 90 + 0.75(T - 320) & \text{if } T > 320 \end{cases}$$

This test is designed to test the effectiveness and robustness of using local models, given that the impeller speed drastically affects the fluid flow behavior.

Table 3 summarizes the results of this second test. The use of local models decreases computational time by a factor of 3–4 over the base case, which still represents a substantial acceleration, although the benefit of using a LAM is smaller than before. A closer examination of the results obtained reveals that, although only a limited number of rigorous CFD calculations are required, each of these requires longer computations (100–500 s CPU time) because hydrodynamic conditions tend to change significantly from one calculation to the next. However, these large changes in hydrodynamics do not seem to affect the predictive ability of the LAMs.

Application II

Our second application, taken from Bezzo et al.,[8] is a stirred-tank bioreactor for the production of polymer xanthan gum. A population balance zone model is implemented to represent the

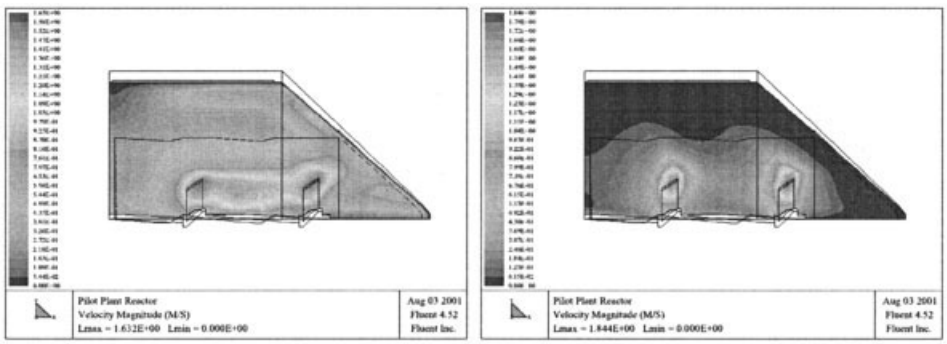

Figure 3. Velocity distribution at the beginning (left) and at the end (right) of the simulation.[8]

Table 4. Effective Viscosity LAMs Considered in Application II

Case	Model	Initial Radius δ	No. of Parameters p
1	LAM Eq. 4	5×10^{-2} (Constant)	3
2	LAM Eq. 18	5×10^{-2} (Constant)	1
3	LAM Eq. 19	5×10^{-2} (Constant)	2

distribution of cell sizes in the reactor, also incorporating gas-to-liquid oxygen mass transfer and biokinetics. The broth rheology is modeled by a non-Newtonian correlation. Networks of between 5 and 25 zones are used.

The CFD submodel solves the total mass and momentum balance equations for given values of the parameters k and n of Eq. 16 to determine the following outputs:

- The mass flow rates $F_{zz'}$ and $F_{z'z}$ between each pair of neighboring zones z and z'.
- The volume-averaged effective viscosity η_z for each zone z.

The interzonal mass flow rates are handled by means of linear LAMs. Although the velocity field changes dramatically (see Figure 3) over the course of the simulation, its variation is rather smooth over time and there is no need for more complicated models. (In fact, for this particular application, satisfactory results may even be obtained by treating the flow rate values as piecewise constant functions of time, which are occasionally updated during the simulation by means of rigorous CFD calculations.)

On the other hand, three different LAMs are tested for the computation of the effective viscosity in each zone (see Table 4). These include the simple linear LAM (Eq. 4) and the two physically based LAMs (Eqs. 18 and 19).

A base case simulation making use of rigorous CFD calculations only was not possible for this application because of the excessive computational effort required. From this perspective, the use of LAMs is already making a positive contribution by allowing a practically feasible simulation to be performed.

Table 5 presents results of simulations carried out with a multizonal submodel involving 20 zones. The "G(ood)" entries in the accuracy (A) row indicate that the results were smooth without any discontinuities or fluctuations that could not be justified on physical grounds.

The total CPU time (CPU_{tot}) and the numbers of CFD submodel solutions are reported as metrics of computational efficiency. As in application I, physically based LAMs are found to perform best. However, in this case, the process simulation calculations account for about 20–25% of the total CPU time, reflecting the much more complex nature of the underlying zone models, which are themselves integro-partial differential equations distributed over the domain of cell sizes.

Table 6 presents data relating to the computational effort required for different numbers of zones in the model (using LAM Eq. 18 for all calculations). The required CFD computational time varies relatively little because the number of necessary updates (attributed to changing physical properties) is not significantly affected by the number of zones. On the contrary, the process simulation CPU time drastically depends on the number of zones because that determines the number of equations and variables in the multizonal submodel. For $nz \geq 20$, the process simulation CPU time contributes substantially to the total computational time.

Table 5. Performance of Different LAMs in a 20-Zone Model for Application II

Parameter	Mathematically Based LAMs	Physically Based LAMs	
Case	1	2	3
A	G	G	G
$PS\%$	21.3	22.3	24.7
N_{cfd}	115	110	104
CPU_{tot} (s)	40,100	35,300	33,200

Concluding Remarks

Hybrid multizonal/CFD models provide a means of introducing a more realistic description of fluid mechanics and mixing phenomena within process models. Early hybrid models of this type performed the CFD calculation either once only (normally, before the start of the simulation) or very infrequently (such as when conditions within the process equipment changed dramatically). However, for many applications of industrial interest, a closer coupling is necessary to preserve the accuracy and reliability of the overall hybrid model, without making arbitrary decisions regarding the required frequency of the CFD calculation.

The above objective can be achieved by viewing the CFD submodel as defining a function $F(\cdot)$ expressing one or more of the variables in the multizonal model (the function outputs, Y) in terms of another subset of the variables (the function inputs, $\mathbf{x}$). Thus, standard numerical solvers (such as differential–algebraic equation integrators) can be applied unaltered to the multizonal process model.

The present paper has focused on the accurate and efficient evaluation of the function $F(\mathbf{x})$ using local approximate models. Both mathematically and physically based LAMs were considered. Unsurprisingly, the latter category has been found to be superior in the two applications tested here; however, their applicability is limited to specific types of outputs Y, whereas the mathematically based LAMs are universally applicable.

Both of the example applications presented here were related to dynamic process simulation. It is worth noting that other model-based applications (such as process optimization) can, in principle, be handled using the same approach with the function $F(\mathbf{x})$ being evaluated by a combination of rigorous CFD and LAM-based calculations. However, more attention may have to be paid to the continuity and differentiability of this function.

Acknowledgments

The authors gratefully acknowledge the financial support of the United Kingdom's Engineering and Physical Sciences Research Council (ESPRC) for process modeling research in the Centre for Process Systems Engineering at Imperial College London under Platform Grant GR/N08636.

Notation

d_i = distance of point $\mathbf{x_i}$ from the given input vector $\mathbf{x}$
$\mathcal{D}$ = set of discrete input/output pairs $\{\mathbf{x}_i, Y_i\}$
$\tilde{\mathcal{D}}$ = subset of the points in $\mathcal{D}$ that can be used for the construction of a LAM
$F_{zz'}$ = mass flow rate from zone z to zone z' computed by the CFD submodel
n = number of inputs (length of vector $\mathbf{x}$)
p = number of parameters in a LAM (length of vector α)
$\mathbf{x}$ = inputs to a CFD submodel
Y = output of a CFD submodel

Table 6. Dependency of Computational Effort on Number of Zones for Application II

	Number of Zones				
	5	10	15	20	25
Process simulation CPU time	500	2100	3700	8200	15,300
CFD CPU time	23,700	24,000	24,500	25,000	26,000
Total CPU time	24,200	26,100	28,200	33,200	41,300
No. of multizonal model equations	800	2100	5700	10,000	17,600

y = output of a LAM
z = zone of a multizonal submodel
α = local model parameters
δ = parameter controlling eligibility of points $\mathbf{x}_i$ for the construction of a LAM

Literature Cited

1. Bezzo F, Macchietto S, Pantelides CC. A general methodology for hybrid multizonal/CFD models. Part I: Theoretical background. *Comput Chem Eng.* 2004;28:501-511.
2. Urban Z, Liberis L. Hybrid gPROMS-CFD modelling of an industrial scale crystalliser with rigorous crystal nucleation and growth kinetics and a full population balance. Proc of Chemputers 1999 Conf., Düsseldorf, Germany; 1999.
3. Bauer M, Eigenberger G. A concept for multi-scale modeling of bubble columns and loop reactors. *Chem Eng Sci.* 1999;54:5109-5117.
4. Bauer M, Eigenberger G. Multiscale modeling of hydrodynamics, mass transfer and reaction in bubble column reactors. *Chem Eng Sci.* 2001;56:1067-1074.
5. Alexopoulos AH, Maggioris D, Kiparissides C. CFD analysis of turbulence of non-homogeneity in mixing vessels. A two-compartment model. *Chem Eng Sci.* 2002;57:1735-1752.
6. Zauner R, Jones AG. On the influence of mixing on crystal precipitation processes. Application of the segregated feed model. *Chem Eng Sci.* 2002;57:821-831.
7. Rigopoulos S, Jones A. A hybrid CFD-reaction engineering framework for multiphase reactor modelling: Basic concept and application to bubble column reactors. *Chem Eng Sci.* 2003;58:3077-3089.
8. Bezzo F, Macchietto S, Pantelides CC. General hybrid multizonal/CFD approach for bioreactor modeling. *AIChE J.* 2003;49:2133-2148.
9. Yang B, Pope SB. Treating chemistry in combustion with detailed mechanisms: In situ adaptive tabulation in principal directions—Premixed combustion. *Combust Flame.* 1998;112:85-112.
10. Frenklach M, Packard A, Seiler P, Feeley R. Collaborative data processing in developing predictive models of complex reaction systems. *Int J Chem Kinet.* 2004;36:57-66.
11. Leesley ME, Heyen G. The dynamic approximation method of handling vapor–liquid equilibrium data in computer calculations for chemical processes. *Comput Chem Eng.* 1977;1:109-112.
12. Chimowitz EH, Anderson TF, Macchietto S, Stutzman LF. Local models for representing phase equilibria in multicomponent nonideal vapor–liquid and liquid–liquid systems. 1. Thermodynamics approximation functions. *Ind Eng Chem Process Des Dev.* 1986;25:674-682.
13. Macchietto S, Chimowitz EH, Anderson TF, Stutzman LF. Local models for representing phase equilibria in multicomponent nonideal vapor–liquid and liquid–liquid systems. 3. Parameter estimation and update. *Ind Eng Chem Process Des Dev.* 1986;25:674-682.
14. Åström KJ, Borisson U, Ljung L, Wittenmark B. Theory and applications of self tuning regulators. *Automatica.* 1977;13:457-476.
15. Støren S, Hertzberg T. Local thermodynamics models used in sensitivity estimation of dynamic systems. *Comput Chem Eng.* 1997;21:S709-S714.
16. Soroush M. State and parameter estimations and their applications in process control. *Comput Chem Eng.* 1998;23:229-245.
17. Shepard D. A two dimensional interpolation function for irregularity spaced data. Proc of 23rd National ACM Conf., New York, NY: ACM Press; 1968:517-523.
18. Shen CY, Reed HL, Foley TA. Shepard's interpolation for solution-adaptive methods. *J Comput Phys.* 1993;106:52-61.
19. Nguyen KA, Rossi I, Truhlar DG. A dual-level Shepard interpolation method for generating potential energy surfaces for dynamic calculations. *J Chem Phys.* 1995;103:5522-5530.
20. Franke R, Nielson G. Smooth interpolation of large set of scattered data. *Int J Numer Methods Eng.* 1980;15:181-200.
21. Renka RJ. Multivariate interpolation of large sets of scattered data. *ACM Trans Math Software.* 1988;14:139-148.
22. Carlson RE, Foley TA. The parameter R^2 in multiquadric interpolation. *Comput Math Appl.* 1991;55:291-300.
23. Iiguni Y, Kawamoto I, Adachi N. A nonlinear adaptive estimation method based on local approximation. *IEEE Trans Signal Process.* 1997;45:1831-1841.
24. Fletcher P. Heat transfer coefficient for stirred batch reactor design. *The Chemical Engineer.* London, UK: Institution of Chemical Engineers; 1987;33–37.
25. Launder BE, Spalding DB. *Lectures in Mathematical Models of Turbulence.* Pacific Grove, CA: Academic Press; 1974.
26. Metzner AB, Otto RE. Agitation of non-Newtonian fluids. *AIChE J.* 1957;1:3–10.
27. *gPROMS Introductory User Guide*, v. 2.0. London, UK: Process Systems Enterprise Ltd.; 2001.
28. *FLUENT 4.4 User's Guide*. Lebanon, NH: Fluent, Inc.; 1997.
29. Bezzo F, Macchietto S, Pantelides CC. A general framework for the integration of computational fluid dynamics and process simulation. *Comput Chem Eng.* 2000;24:653-658.

Manuscript received Oct. 31, 2003, and revision received Jul. 19, 2004.