On the Common Conceptual and Computational Frameworks for Multiscale Modelling

Aidong Yang

Department of Chemical and Process Engineering, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford GU2 7XH, UK

Email: a.yang@surrey.ac.uk

Abstract: Multiscale modelling and simulation has received widespread applications in various scientific and engineering disciplines. Although these applications deal with a diverse range of systems and serve different purposes, they face a number of common challenges with respect to the construction, solution, and computer implementation of multiscale models. The solutions to these challenges could bring common concepts, methods, and tools that can be shared by different disciplines and facilitate new applications of this promising modelling paradigm. This paper attempts to outline the key aspects of these common challenges, identify the important issues of each aspect, and summarise progress to-date. Directions for future work are also suggested towards the formulation of the common conceptual and computational frameworks for multiscale modelling.

Keywords: Multiscale modelling and simulation, model classification, stability and efficiency, error analysis, parameter estimation, software environment

1. Introduction

Mathematical models are tools commonly adopted by virtually all scientific and engineering disciplines. These models are of different natures and possess various levels of complexity. For a given mathematical model, there often exist a number of methods which can be used to solve the model to provide a solution that can be further analysed or utilised. Depending on the nature of the system as well as the purpose or task in question, a model of a right level of complexity and a corresponding right type of solution method should be chosen, so that a solution of an acceptable quality or accuracy can be obtained with the consumption of a reasonable amount of computational effort. Multiscale modelling is concerned with the construction and solution of models which combine “sub-models” of multiple scales of the targeted system in order to gain superior modelling results or
higher computational efficiencies which can hardly be achieved by single scale models. While some of the basic ideas behind multiscale modelling can be traced back to the early days of mathematical modelling, this area of research has gained significant momentum in the last two decades, motivated by the growing challenges in supporting the development of new products, processes and devices or the understanding of complex systems and enabled by the unprecedented increase of computational power available to modellers. This trend is well reflected by a number of recent reviews by researchers from different disciplinary communities, including chemical engineering\textsuperscript{1-7}, computational mechanics\textsuperscript{8,9}, material sciences\textsuperscript{10-12}, and biology and biomedical sciences\textsuperscript{13-16}. An overview of the research communities involved in developing and applying multiscale modelling and simulation is given by Groen et al.\textsuperscript{17}

The widespread application of multiscale modelling in different subject areas suggests the great generality of this promising modelling paradigm. In fact, this attribute has been mentioned in most, if not all of the reviews cited above despite their diverse disciplinary origin, and considerable overlap exists within the types of physical objects and models they studied. Besides, generalisation of the concepts, principles, and modelling approaches indicated in these reviews and by other efforts (e.g. those by chemical engineers\textsuperscript{18-20}), although often made within a specific disciplinary context, has been projected to be applicable in a much wider scope. In addition, there have been several attempts on developing generic frameworks for multiscale modelling not associated with specific disciplines, including the work by E and co-workers\textsuperscript{21} on the heterogeneous multiscale modelling framework, Hoekstra and co-workers\textsuperscript{22-24} on the modelling language and software environment based the framework of Complex Automata, and Yang, Marquardt and co-workers\textsuperscript{25-27} on the conceptualisation of multiscale systems based on general systems theory and the corresponding computer-aided multiscale modelling methodology and tools.

The value of the generalisation of multiscale modelling approaches across the disciplinary boundaries can be viewed at two levels. At the level of concepts and methods, generalised learning can foster the successful new applications of this modelling paradigm in various fields, which may in turn enhance the generalised learning to subsequently benefit future applications. At the level of tools and
implementation, there is potential to establish common model libraries, solvers, software environments and even hardware architectures and facilities that can be shared by different applications, hence making multiscale modelling more efficient and affordable in general. Synthesising existing efforts towards generalised concepts, methods and tools, this paper is aimed to outline the key aspects of common conceptual and computational frameworks and identify and discuss the key issues associated with each aspect. Future research directions will also be suggested.

More specifically, this paper attempts to make a distinction between model construction and model solution and discusses these two aspects separately in Sections 2 and 3, respectively. Although these two aspects are always inter-twined as in any modelling effort, this distinction hopefully allows one to separate the approaches essentially pertaining to the construction or structuring of multiscale models from those more on how the models are solved; in most existing work the approaches of these two categories have been mixed, and the discussions on the issues of the latter aspect (i.e. model solution) are scattered rather sparsely in the literature. Besides, a third aspect of multiscale modelling, namely computer implementation which has been reported rather rarely to-date, will be discussed in Section 4. It should be noted that a comprehensive review of existing generalised concepts and methods for multiscale modelling is beyond the scope of this short paper. Furthermore for multiscale modelling applications in specific disciplines, one is referred to the disciplinary reviews cited earlier.

2. Model construction

In the context of multiscale modelling, model construction involves important decisions such as what scales to be included in a model, how to model each scale, and what links to be placed between the models of individual scales. This section will introduce general concepts of multiscale systems and models and discuss classification and structuring of multiscale models.

2.1 Conceptual foundation

Common concepts of multiscale systems and models may be defined within the framework of general systems theory (GST). Particularly based on the work of Bunge\textsuperscript{28-29} on the ontology modelling of general systems, Yang and Marquardt\textsuperscript{25} developed a conceptualisation for multiscale models, with
some minor extension introduced more recently by Zhao et al. The conceptualisation defines a set of concepts at the system, component, scale, and inter-scale levels; the key concepts are shown in Figure 1 by a UML class diagram; the exact definitions of these concepts can be found in Yang and Marquardt. Briefly, a system is viewed to consist of components which are either subsystems or atomic components (called A-Components). Components belonging to the same level of decomposition form a scale. Components of the same scale (termed L-Components) can be coupled with each other or with the environment of the scale (called L-environment) which can be different from the environment of the whole system (termed A-environment). Both a system and a scale have a structure (termed A-Structure and L-Structure, respectively). Two scales can be coupled through inter scale links such as aggregation (deriving properties of a coarse scale from those of the components of a fine scale), disaggregation (deriving properties of the components of a fine scale from those of a coarse scale) and mereological connection (connecting fluxes which are modelled as different scales). A system generally has multiple states described via state functions which are classified into coupling-induced ones (i.e. those induced due to the existence of the couplings between this and another system or the environment) and non-coupling-induced ones. These state functions (or phenomena manifested through them) are governed by laws which include internal laws (governing state functions of individual components), intra-scale laws (governing the coupling-induced state functions stemming from an inter-component coupling and other state functions of the components or the environment involved in the coupling), and inter-scale laws (governing inter-scale couplings). This conceptualisation has been formally represented using a set theory based mathematical language with which the key concepts are rigorously defined to avoid ambiguity.
The above conceptual framework is intended to provide a common basis for understanding and developing various multiscale models. A different framework has been proposed by Hoekstra and co-workers\(^\text{22}\) based on cellular automata. A multiscale system in this formalism is modelled by a set of interconnected cellular automata each modelling a single scale; the whole model is referred to as a Complex Automaton (CxA). In a CxA, each single scale is characterised by the space of states, the update rule as well as some external parameters. Modelled after Lattice Gas Automata, the update rule incorporates three operators, namely collision, propagation, and boundary conditions. Mapping to the relevant concepts defined in the GST-based framework, these operators can be regarded as corresponding to internal law, inter-component coupling law, and system-environment coupling law, respectively. Regarding the inter-scale connections, the CxA formalism considers that two scales may be coupled through initial conditions, boundary conditions, and/or collision operator. This differentiation can be interpreted using the concepts defined in the GST-based framework. Particularly, the latter two types of coupling mechanism correspond to the types of state functions involved in an inter-scale connection, coupling induced state functions and non-coupling induced state functions respectively.
From the above discussion, it can be seen that the GST-based framework and the CxA-based framework share a set of comparable concepts. This is not surprising, since cellular automata are commonly used as abstract models of general systems. On the other hand, the GST-based framework focuses more on the conceptual representation of the systems to be modelled, while the CxA-based framework is more directly associated with a computational procedure by which a multiscale model can be realised. Future work on analysis and integration of these and other formalisms has the potential to form a widely accepted common conceptual basis for multiscale modelling. It should be noted that both the GST- and the CxA-based frameworks have already been taken as a basis for developing tools to support multiscale modelling, as will be discussed in Section 4. Besides, the CxA-based framework has recently been extended to integrate not only cellular automata but also other types of models.

2.1.1 Types of scale

In the GST-based formalism, a scale of a system is composed of components belonging to a certain level of decomposition. In general, a system may be decomposed along different dimensions, leading to different types of system scale. These decomposition dimensions or scale types stem from the coordinates of the backdrop against which the state functions or properties of a system are observed. A backdrop commonly has spatial coordinates and a temporal coordinate, while other possible coordinates also exist depending on the nature of the properties to be observed and the system studied. For example, if the system is a population, a property of the population (e.g. number of individuals, population mass, etc.) may depend on a number of individual properties (particle size, age, shape, molecular weight or chain length, etc.) which act as coordinates of the backdrop against which the population is observed.

In principle, each of these backdrop coordinates can become a dimension for decomposing the studied system and be assigned with different levels of granularity leading to coarser and finer system scales of a type corresponding to the backdrop coordinate. In practice, the most frequently encountered examples are spatial and temporal scales with space and time as the corresponding backdrop coordinates, respectively. While the spatial and temporal coarseness can in principle be continuously
refined resulting in an infinite number of resolutions of a backdrop, in reality a finite number of discrete levels of granularity are employed for the purpose of accommodating different scale models; finer resolutions within a discrete scale are employed often in the stage of model solution for numerical purposes, not for introducing different “physics”. These discrete scales usually correspond to the levels of cognition well established in specific domains. For example, the scales of molecule, molecule cluster, particle, particle cluster, and continuum phase are considered by materials science, computational mechanics, and chemical engineering, while the latter deals with further scales including device or process unit, plant and beyond. In systems biology, a different set of scales beyond the molecular level is studied, including sub-cellular, cellular, tissue, organ, and organism. The distinction between these “natural” scales can be viewed as primarily corresponding to the spatial coarseness; the temporal coarseness is closely correlated to the former and handled rather implicitly. This correlation is evident from the frequently quoted “scale diagrams” in which objects of different scales (e.g. molecule, molecule cluster, surface, phase, etc.) are plotted usually diagonally (hence correlatively) in a temporal-spatial coordinate system. For modelling a physical system, explicit and direct treatment of different temporal scales usually occurs when events of significantly different rates exist at the same spatial scale, as in the case of a kinetic Monte Carlo (kMC) model which involves chemical reactions possessing reaction rates of different orders of magnitude. Another rather obvious case where a model is organised primarily along temporal scales is a multiscale or hierarchical characterisation of a discrete event system, as manifested in various hierarchical petri-net models. It was recently shown that the GST-based formalism of multiscale systems can be applied to discrete event systems with some minor extension.

2.2 Model classification

The class of a multiscale model is often important as it to a large extent determines the types and levels of mechanistic knowledge the model can accommodate, the easiness of constructing and running the model, and related to these two the level of accuracy or fidelity and computational efficiency this model can potentially achieve. In the past, a number of classifications have been suggested; the GST-based conceptualisation has been used to derive rigorous definitions of model
classes reported in the literature\textsuperscript{25}. Using the terminology of the GST-based conceptualisation, three different sets of classification can be applied to elucidate the differences between the model classes introduced previously.

Set I: classification by \textit{component-scale relation} (Table 1). Referring to a system with a coarse scale and a fine scale, this relation is about at which scale each component of the system is modelled, as a collection of \textit{laws} that govern its \textit{phenomena} or the change of its \textit{state functions} at the corresponding scale. Most multiscale model classes have been defined essentially according to this relation. Note that some model classes were introduced in the literature together with some of those included in Table 1 (e.g. “hierarchical integration” by Pantelides\textsuperscript{18} and “parallel integration” by Ingram et al.\textsuperscript{19}) but are not considered here, because they do not focus on the relation between components and scales.

<table>
<thead>
<tr>
<th>Model classes</th>
<th>I-a: \textit{All components modelled at both scales}</th>
<th>I-b: \textit{All components modelled at coarse scale, some at fine scale}</th>
<th>I-c: \textit{Some components modelled at both scales while the others only at fine scale}</th>
<th>I-d: \textit{Components modelled only at fine scale}</th>
<th>I-e: \textit{Some components modelled at coarse scale while others primarily or purely at fine scale}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms used in literature</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
E et al.\textsuperscript{21} | Type B | Type C | Type A | 
Hoekstra et al.\textsuperscript{22} | Hierarchical coupling | | Sub-domain coupling | 
Ingram et al.\textsuperscript{19} | Embedded | Simultaneous | Multi-domain | 
Vlachos\textsuperscript{1} | Multigrid-type | | Onion-type | 
Pantelides\textsuperscript{18} | Parallel or Serial | Simultaneous | |

Set II: classification by \textit{phenomena-scale relation} (Table 2). This is a possibility that the set of interesting phenomena of a system has un-correlated coarseness of the temporal and spatial scales. Under this circumstance, a phenomenon may be modelled at a finer temporal scale but at a coarser spatial scale, or vice versa; the allocation of different phenomena to the most appropriate scales gives rise to another way of classifying multiscale models. This type of classification has been proposed by Hoekstra et al.\textsuperscript{22} through a scale-separation map, on which five model classes are differentiated.
depending on how separate temporal and spatial scales have been applied to different phenomena or processes.

Table 2. Multiscale model classes - Set II: by phenomena-scale relation.

<table>
<thead>
<tr>
<th>Model classes</th>
<th>II-a: All phenomena modelled at the same temporal/spatial scales (resulting in single scale models)</th>
<th>II-b: All phenomena modelled at different temporal scale but at the same spatial scale</th>
<th>II-c: All phenomena modelled at different spatial scale but at the same temporal scale</th>
<th>II-d: Phenomena modelled at different but correlated temporal and spatial scales</th>
<th>II-e: Phenomena modelled at different and un-correlated temporal and spatial scales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regions on the scale separation map of Hoekstra et al.</td>
<td>Region 0</td>
<td>Region 1</td>
<td>Region 2</td>
<td>Region 3.1</td>
<td>Region 3.2</td>
</tr>
</tbody>
</table>

Set III: classification by scale-scale relation (Table 3). Focusing on how different scales are related to each other within a multiscale model, a frequently studied aspect is on the information flows between connected scales where a distinction can be made between one-way flows and two-way flows. In addition, two model classes have been proposed where different scales are coupled not by data flows but rather by certain trans-scale mechanisms. The “Type-D” models by E et al.\(^{21}\) are for problems which exhibit self-similarity between different scales, consequently model parameters obtained from simulation at one scale may be applied to the model of a different scale which shares a common model structure with the former scale. One example is the modelling of transport on a percolation network where a parameter for predicting effective conductivity is first estimated via microscopic simulations and subsequently applied to the macroscopic level\(^{21}\). The other class is termed “variational” or “analytical” models by Li et al.\(^{20}\) as part of the EMMS (Energy Minimisation Multi-Scale) framework, which refers to the cases where the modelling of different scales (and mechanisms) of a system includes a stability condition, possibly applying to multiple scales, as an important means to quantify the key behaviour of the whole system. One example of such stability conditions is the minimisation of the energy consumed for the suspension and transportation in a gas-solid flow system\(^{20}\).
Table 3. Multiscale model classes - Set III: by scale-scale relation.

<table>
<thead>
<tr>
<th>Model classes</th>
<th>Scales coupled by inter-scale information flow</th>
<th>Scales coupled by trans-scale mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>III-a: Information passed only from one scale to another</td>
<td>III-b: Information passed between two scales in both directions</td>
<td>III-c: Scales coupled by common or shared mechanism</td>
</tr>
<tr>
<td>E et al.(^{21})</td>
<td>Sequential</td>
<td>Concurrent</td>
</tr>
<tr>
<td>Ingram et al.(^{19})</td>
<td>Serial</td>
<td></td>
</tr>
<tr>
<td>Li et al.(^{20})</td>
<td>Correlative</td>
<td></td>
</tr>
<tr>
<td>Pantelides(^{18})</td>
<td>Serial</td>
<td>Parallel</td>
</tr>
</tbody>
</table>

For a specific multiscale model, it can be classified separately along each of the three dimensions mentioned above and subsequently attributed to a model class from each of the three sets. As these model classes generally entail different knowledge/data requirements, levels of accuracy and computational requirements, the decision in a given application should depend on both the nature of the system being modelled and the purpose of modelling. An empirical study was carried out by Ingram et al.\(^{19}\) where several alternative model classes were applied to model a chemical reactor. These classes led to different levels of model complexity, lengths of execution time, and simulation results. While it is difficult to draw general conclusions about the suitability of different model classes from this kind of empirical studies, some general implication of model class selection can be discussed:

- Within Set I, if the characterisation of the components of the system increasingly relies on the models at the fine scale, fewer macroscopic assumptions or approximations will be required which potentially can improve the accuracy of the model. On the other hand, the computational burden is likely to increase as demanded by running the fine-scale models. Class I-d essentially represents the extreme of this shift. For the other classes, different positions exist within each class with respect to the split of the modelling effort between the two different scales. For example, in Class I-e, the split is controlled by the portion of the computational domain on which fine (or coarse)-scale models are adopted. To make the best use of a fine-scale model (in terms of the optimal balance between accuracy and computational load), a general principle would be
applying it only to the critical regions (e.g. cracks and dislocations in material modelling) where the coarse-scale model becomes inappropriate.\textsuperscript{21}

- With respect to Set II, modelling different phenomena at several spatial scales may be required for accurately capturing the underlying physics, which however will generally require the handling of the stiffness of the resulting model which can cause difficulties for numerical solution. When multiple spatial resolutions are involved, the improvement of accuracy by involving finer scales needs to be considered in balance with the increased computational burden as well as the computational consequences of connecting these spatial scales by information aggregation and disaggregation (see Section 3.3 for further discussions).

- Within Set III, the selection of Class III-c will be dictated by whether a mechanistic understanding exists which leads to any trans-scale mechanisms to form part of the multiscale model. Between the two classes distinguished by the inter-scale information flows, Class III-a, with one-way flow is suitable when the model of one scale (normally the fine scale) can be executed to produce information needed by the other scale, without the need for any information from the latter. Class III-b, on the other hand, applies to the cases where one scale is modelled to provide information needed by another scale but based on some quantities from the latter. Computationally, the solution of a Class III-a model does not require an “on-line” connection between the two scales as the model of the information-supplying scale can be solved separately before the solution of the model of the information-requesting scale; there is no need to repeat the solution of the former model in the course of the solution of the latter. In contrast, a multiscale simulation involving two-way information coupling (i.e. with a Class III-b model) normally requires the repeated execution of the fine-scale model while the coarse-scale model is solved if the macroscopic quantities from the coarse scale that the fine-scale model depends upon change during the simulation, possibly as a function of the quantities computed and supplied by the fine scale. Thus, the computational load of a Class III-b model can be much higher than that of a Class III-a model particularly when the fine-scale model is computationally complex.
In addition to what is covered by the above three sets of model classes, some model classes are named after the difference between the coupled models, e.g. discrete-continuum models, stochastic-deterministic models, etc. The possible model classes of this kind depend on the classification of single scale models (Figure 2). Combinations of these different model types lead to the heterogeneity of multiscale models\textsuperscript{21} which often has important computational implications.

![Figure 2. Classification of single-scale models.](image)

### 2.3 Establishing model structure

A general procedure for establishing the structure of a multiscale model should describe how individual scales are introduced, characterised, and connected. Regarding the sequence for introducing different scales into a model, three options can be distinguished\textsuperscript{19}: bottom-up (starting with the finest scale and subsequently introducing coarser scales), top-down (starting with a macroscopic scale and subsequently introducing finer scales as needed), and middle-out (starting with a mid-level abstraction where richest data and knowledge are available and subsequently introducing finer and coarser scales). The choice over these options largely depends on the modelling goal, the knowledge available, as well as the level of affordable effort for model building.

An issue closely related to the sequence of introducing different scales is what scales are most relevant for a given modelling task. For example, to model a biological body, one needs to decide on whether models for organs, tissues, cells, and/or molecular processes should be included. When modelling a chemical reactor, the selection of modelling targets could be between bulk phases, particles, reactive surfaces, and molecule clusters. The decision of scale selection may be considered
within the existing framework of model selection. Conventionally, the field of model selection offers a set of methods that quantify the distance between a model and the reference data according to various criteria\textsuperscript{35}. Furthermore, statistical approaches have been developed to optimise the design of experiments to best reveal the distance between alternative models, hence facilitating model discrimination\textsuperscript{36}. These model selection approaches however are most useful for dealing with statistical or semi-empirical models, whilst multiscale models are sought after often for the purpose of incorporating more first-principle knowledge and hence reducing the empirical extent. Furthermore, the focus of conventional model selection lies mainly on model accuracy, while the aspect of computational effort needs to be heavily considered when choosing from the alternatives of a multiscale model.

A broader set of model selection criteria was proposed by Verheijen\textsuperscript{37}, which includes the mechanistic appropriateness of the elected model elements for the target system and phenomena, the range of model validity, and model reusability in addition to model accuracy and computational effort. Applying to multiscale models, what is particularly relevant is the determination of the importance of various scales within a given context based on a mechanistic understanding. In some areas, a certain scale might be viewed as inherently more important or fundamental than the others, as argued by Walker and Southgate\textsuperscript{38} for the special status of cells as a critical level of abstraction of biological systems in a middle-out approach to multiscale model building. More generally, the relative importance of different scales would depend on the modelling goal and the particulars of the target system. For example, for predicting the overall reaction rate in a heterogeneous catalytic reactor, the importance of the porous structure of the catalyst pellet depends on the magnitude of the intrinsic chemical reaction rate\textsuperscript{39}.

To summarise, the most appropriate model structure for a given modelling problem should be determined considering both model quality and computational implications. In the future a guiding framework for choosing from candidate model structures may be developed by extending existing methods for model selection to allow the balance between model accuracy and computational effort and also to incorporate a generic yet practical procedure for assessing the relevant importance of
different scales. The development in this area could eventually realise the vision of “adaptive multiscale modelling”\textsuperscript{40}. It should be noted that the computational effort involved in a multiscale simulation is determined by both the model structure and the method for solving the model; the latter is the subject of the next section.

3. Model solution

As indicated earlier, the way how a multiscale model is constructed will affect how the model can or should be solved. Therefore, it can be beneficial in certain cases to select the model structure and the solution approach simultaneously. In this section, we will solely focus on the computational aspects of the solution of multiscale models after they are already constructed. These aspects, namely efficiency, stability and accuracy are inevitably coupled with each other. Therefore, the separate discussion of these aspects in this section does not imply that they ought to be treated in an isolated way. Furthermore, two different (and extreme) cases should be distinguished when examining the applicability of different techniques for improved model solution. In the first case, the content of each single scale model (e.g. equations) involved in a multiscale simulation is known and even changeable by the multiscale modeller, so that mathematical manipulation and detailed numerical analysis are feasible on these models. In the second case, each single scale is modelled by a simulator which only makes the input and output data structure available to a multiscale simulation; the model itself can only be viewed as a black box. In this case, techniques available for improving the simulation can be rather limited.

3.1 Computational efficiency

In general, coupling a coarse scale model with a fine scale model will make the numerical simulation computationally more demanding than simulation with a coarse scale model alone. Ignoring the “overhead” required for coupling two single-scale models (data exchange etc.), the computational time of a multiscale simulation is determined by two factors, namely the number of runs of each model and the time required for solving each model once. These two factors may be improved by efficient algorithms at both single- and multi-scale levels and reduction of the complexity of single-scale models.
Firstly, model solution algorithms may be improved at the single scale level, which may achieve the desired convergence or complete the intended simulation period with fewer evaluations of not only the model at this scale but possibly also that at another scale which is coupled to the former (e.g. when the latter is a fine-scale model which computes certain quantities for the former, coarse-scale model). In the past, significant progress has been made in developing adaptive solution approaches for (deterministic) ODE/DAE\textsuperscript{41} and PDE models\textsuperscript{42}, which are nowadays routinely used for modelling continuum scales. Temporal and spatial coarse graining techniques, on the other hand, have been proposed for stochastic simulation such as kMC (kinetic Monte Carlo) for modelling finer scales\textsuperscript{1}.

Secondly, special algorithms may be developed to target on the inter-scale coupling. One example concerns the content (particularly the level of noise) of the information flow passed from a fine scale to a coarse scale, which may upset the solver of the coarse-scale model leading to slower solution. Therefore, applying certain signal processing algorithms to condition the information flow exchanged between two single scale models can be beneficial\textsuperscript{1,21}. Note that noisy signals between two scales can also be a source of numerical instability, an issue to be discussed in Section 3.2. A similar issue pertains to the disaggregation of data from a coarser scale to a finer scale, which may bring significant discontinuities leading to a slower solution process or even a solver failure at the finer scale. Smoothing or interpolation techniques may be applied to render the reconstruction of smoother profiles to ease the problem\textsuperscript{43}. As a different example, an algorithm may be adopted which allows an adaptive choice of the size of the domain over which a fine-scale model is solved to supply data required by a coarse-scale model, as speculated by E et al\textsuperscript{21}. While proposals of this kind, aimed to address coupled simulations, are interesting and promising, the development in this area is still rather insignificant compared to the algorithms available for single scale models.

In addition to adopting efficient algorithms, reducing the complexity of single scale models involved in a multiscale simulation is another avenue for improving computational efficiency. In this respect, both model order reduction and model simplification can be considered; their applications in general chemical process modelling were comprehensively reviewed by Marquardt\textsuperscript{44}. Model order reduction is a well-developed area offering a range of methods to reduce the complexity of differential
equations based models and has been used in multiscale modelling\textsuperscript{45}. Models can also be simplified without changing their order, by approximating complex functions with computationally simpler ones.

In multiscale modelling, the simplest treatment following this idea is to retain the solution of a finer scale model and delay its update until necessary, implying a piece-wise constant approximation of the finer scale model (Et al.\textsuperscript{21}; see also Bezzo et al.\textsuperscript{46} where such a treatment was introduced for maintaining robustness of a coarse scale solver which again will be discussed in Section 3.2). Beyond this simplest treatment, other forms of the approximation function (e.g. polynomial) with a moderate computational cost for evaluating and updating (on-the-fly) the function may be employed\textsuperscript{21,46}.

Unlike model order reduction, model simplification usually does not require the knowledge of the structure and the symbolic form of the model being simplified and can establish the approximating function purely based on the input-output data produced by the original model. This is advantageous particularly when a single scale model is available only in a closed-form (e.g. as a piece of computer code). However when applying this approach, lessons learned from the past applications of local or surrogate models in chemical process engineering\textsuperscript{44,47} must be noted with respect to (i) the maintenance of the balance between reducing the computational effort for updating the simplified model and improving its accuracy and (ii) the handling of the discontinuity caused by model update which may create negative effect similar to that of the noisy signals exchanged between two coupled models as mentioned earlier.

3.2 Stability and robustness

The coupling of models of different scales may give arise to mathematical and numerical issues which do not exist when these models are stand-alone.\textsuperscript{18} These issues may impact on the computational efficiency of multiscale simulation (discussed in Section 3.1), the stability of the numerical solutions, or the robustness of the numerical solvers involved. In a simulation that coupled a multizonal (coarser-scale) model and a CFD (finer-scale) model, Bezzo et al.\textsuperscript{46} observed the troubles brought by the noise in the CFD results to the predictor-corrector type of solver for the numerical integration of the multizonal model. They applied a simple piecewise constant filter to the data flow to maintain the robustness of the solver. In multiscale simulations coupling a kMC model with a continuum model, it
has been found that the numerical solution may become unstable when the two models are coupled (i.e. exchanging data) at large time intervals. \textsuperscript{1,48} Methods including averaging the results of multiple kMC runs\textsuperscript{1} and placing a filter on the kMC results\textsuperscript{48} have been applied to suppress the numerical instability of such simulations. Dealing with a different type of multiscale simulation, which coupled molecular dynamics with continuum modelling, Ren\textsuperscript{49} studied the stability and convergence rate with respect to different coupling schemes, such as velocity-velocity, flux-velocity, velocity-flux and flux-flux, and revealed that the stability conditions for these schemes were different.

Towards a generic understanding of the numerical instability problem in multiscale simulation, Rasli et al.\textsuperscript{48} have developed a control theory based analytical framework and concluded that a simulation by two coupled models will become unstable when the joint gain of these two models becomes larger than one. To suppress the instability, a filter is suggested to be placed on the data flow passed from one model to another, which acts against a perturbation component of each model. This perturbation component is viewed as the derivation between the model and the (stable) physical system and hence the source of numerical instability. An approach for designing such a filter was also suggested, based on the nature of the perturbation component of each model. As such information might not be readily available in practice, a more pragmatic approach may be adopted for designing the filter\textsuperscript{50}, following a standard empirical procedure for filter structure selection and parameter tuning\textsuperscript{51}.

### 3.3 Accuracy

If the error of a multiscale simulation is defined as the distance between the output of the simulation and the actual behaviour of the target system, this error is contributed by a number of sources, as indicated in Figure 3. Errors arising from inappropriate models and solvers of individual scales can be reduced by proper model construction and solver selection. In the following, we will focus on the other sources of errors.
To reduce errors in model parameters, suitable parameter estimation techniques are required. In the context of multiscale modelling, one often needs to handle the parameter estimation for stochastic models. As summarised by Vlachos, this requires suitable approaches for sensitivity analysis and effective methods to handle the computational burden associated with stochastic simulation; the latter aspect could benefit from using model simplification techniques as mentioned in Section 3.1. The task of estimating unknown parameters of multiple single-scale models within a multiscale model, however, has rarely been addressed so far. When the experimental data are not sufficiently available at all the scales, it will be impossible to carry out separate parameter estimation for each single scale, and one would have to rely on data for a subset of the scales, usually only the (top) macro scale, to estimate the parameters for all scales. In general (single-scale) modelling, parameter estimation can be done either simultaneously or incrementally. The latter approach, instead of estimating simultaneously and directly all the unknown parameters, employs the structure of conservation relation–flux–constitutive relation of the model and estimates intermediate (e.g. fluxes) and ultimate (e.g. kinetics) unknowns in a stepwise manner, in order to better use the knowledge about the modelled system and to improve computational efficiency and numerical robustness of the parameter estimation process. Such an incremental approach might be adapted for use in multiscale modelling, where the inter-scale “fluxes” could be estimated (subject to identifiability) which are subsequently used to estimate the parameters of relevant scales. The applicability of this approach and a comparison with the simultaneous approach need to be investigated for multiscale modelling.
Analysing and reducing errors arising from the coupling of different models (i.e. those shown in the lower part of Figure 3) is a more special challenge for multiscale modelling. As pointed out by Kulikov et al.\textsuperscript{43}, both the disaggregation of data from a coarser scale to apply to a finer scale and the aggregation of data in the opposite direction introduce errors, due to the resolution gap between the two scales. In disaggregation, errors inevitably occur when the profile of a quantity is reconstructed at the finer scale from its “lumped” value given by the coarser scale model, regardless of the reconstruction method used. In aggregation, the detailed profile of a quantity rendered by the finer scale is lost; the averaged value for the quantity leads to errors when used as an independent variable in any nonlinear functions in the coarser-scale model. Besides, attempts for stabilising multiscale simulations such as the introduction of a filter between two simulation models can also introduce errors to the solution; certain consistency checks\textsuperscript{40} are required to ensure the validity of the simulation result. Similarly, an efficiency-improving measure, such as the use and update of a simplified model “on the fly” to replace a complex original model can also affect the accuracy of the simulation. As discussed in Section 3.1, this requires care in choosing the policy for model update to ensure the expected level of accuracy is not compromised.

As pointed out by E et al.\textsuperscript{21}, systematic quantification of the errors of multiscale simulation is an important task for both accuracy and stability concerns. Targeting on the distance between the result of a multiscale simulation and that of an effective macroscopic model, E et al.\textsuperscript{21} developed theoretical error analysis for several specific Type B problems in heterogeneous multiscale modelling (i.e. those involving a microscopic model to provide quantities required by a macroscopic model that effectively models the entire computational domain). Defining a model error as the distance between the solution of a multiscale model and that of a single (finest) scale model solved by a multiscale approach, Caiazzo et al.\textsuperscript{53} quantified the error of the simulation of a reaction-diffusion system using a model constructed according to the CxA formalism. Future work is needed in developing generic analytical frameworks which can quantify the errors arising from multiscale simulation and which ideally also isolate those from different sources, in order to effectively assess alternative models and solution approaches.
4. Computer implementation

To carry out multiscale simulation, models and numerical methods need to be implemented in a computer environment. Recent advances in computer technology such as grid computing and high performance computing (HPC) have brought unprecedented computing power to modellers. Demanding fine-scale or multiscale simulations have been run successfully on grids or HPC systems with arrays of CPUs and GPUs, producing impressive results. Leveraging these advanced computing infrastructures, a special challenge to the multiscale modelling community is to develop software systems that effectively support the modellers in the development and execution of high-quality simulation. In this section, we will first review relevant efforts arising from individual disciplines, and then discuss the progress and issues in developing domain-independent software environments and tools.

4.1 Domain-specific developments

When multiscale modelling was practised in the early years, the pioneers from various disciplines adopted mainly ad-hoc software solutions. The widespread application of this model paradigm, however, has motivated the development of several modelling environments which can be used to construct a range of models and simulations addressing the needs of specific communities. In materials science, Doi reported a modelling platform for predicting the properties of polymers, which integrated simulation engines for molecular dynamics, reptation dynamics (calculating rheological responses), interfacial dynamics, and micro-fluidics and elastics. Data were exchanged between these simulation engines in structures following a standard data format. The system provided services for the users to operate these engines and to analyse the data flows.

In biology and biomedical sciences, a notable effort is the initiative combining the Physiome project by the International Union of Physiological Sciences and the Virtual Physiological Human (VPH) Project by the EU. In this effort, a mechanism is developed to support the reuse and integration of physiological models across all the scales ranging from protein to whole body. It particularly highlights the standardised expression of mathematical models by means of two standard mark-up languages, namely CellML (for biophysically based lumped parameter models) and FieldML (for
spatially varying fields). A CellML model can be annotated using biological and biophysical concepts defined in the CellMLBio ontology. Models expressed in these standard languages can be integrated and executed by simulation packages that support these standards. Several other multiscale simulators exist in systems biology\textsuperscript{15}, including those of agent-based modelling which is a modelling approach not included in the VPH/Physiome work.

In chemical engineering, multiscale modelling has started making impact on commercial process modelling software, particularly in the area of coupling (coarse-level) process simulation with (fine-level) equipment simulation via CFD models\textsuperscript{46,59}. Morales-Rodríguez and Gani\textsuperscript{60} reported an environment for developing and solving chemical engineering models that could be applied to multiscale systems (e.g. fuel cells). Kulikov et al.\textsuperscript{43} performed multiscale simulation of crystallisation and fluid dynamics using CHEOPS, an open simulation platform for integrating diverse process models and numerical solvers\textsuperscript{61}. The development of CHEOPS and the improvement of openness and interoperability of commercial simulators for process engineering have been based on the component-based software architecture and standard software interfaces, with CAPE-OPEN as the most notable standardisation initiative in this area\textsuperscript{62}. Developed originally for the general task of integrating models, the CAPE-OPEN standards (http://www.colan.org/), like the similar ones developed in other fields (e.g. OPENMI in environmental sciences\textsuperscript{63} or AMUSE in astrophysics, see http://www.amusecode.org), may be extended to better support multiscale modelling.

4.2 Domain-independent developments

The developments in various areas towards multiscale modelling and simulation environments share some common themes, such as modularisation and standardisation. As multiscale modelling is a truly generic modelling paradigm, it would be interesting to explore the possibility of developing domain-independent software environments, which can not only be used for the modelling tasks of different disciplinary origin but also enable complex simulations with models spanning several disciplines, a scenario much more likely to occur in multiscale modelling than in conventional (single-scale) modelling. In the following, two initiatives in this direction are described, each corresponding to one of two conceptual multiscale modelling formalisms introduced in Section 2.1.
Building upon the GST-based conceptualisation, Zhao et al. proposed a computer-aided multiscale modelling environment (Figure 4a). In this environment, a conceptual modelling tool supports the modeller in producing a conceptual description of a target multiscale system (termed a conceptual model) in terms of the components of the system, its scales, and connections between components and those between scales. The conceptual model is expressed using concepts defined in a domain-specific ontology which concretises the generic concepts defined in a common ontology for multiscale systems following the GST-based conceptualisation. A conceptual model can be converted into a simulation script by the model realisation tool, which assigns specific modelling tools or simulators to individual scales indicated in the conceptual model as well as specific algorithms to the inter-scale connections, in addition to other detailed simulation settings. The simulation script can then be interpreted by a simulation co-ordinator in a model execution system and carry out the coupled simulation (Figure 4b). Two different simulation modes are distinguished, namely coordinator-driven (i.e. the simulation is initiated and controlled by the co-ordinator) and master tool-driven (i.e. the simulation is initiated and primarily controlled by one of the single scale simulators). In both cases, the simulation co-ordinator, the single-scale simulators, and the algorithms for coupling different scales communicate with each other through a component-based software architecture particularly with CORBA (www.corba.org) adopted as the middleware for the prototype implementation.
Figure 4. The GST-based multiscale modelling and simulation environment, adapted from Zhao et al.\textsuperscript{27} (a): the overall approach; (b): the model execution system.

In the work by Hoekstra and co-workers\textsuperscript{64}, a Multiscale Modelling Language (MML) and a set of software tools have been developed to support multiscale simulation based on the CxA-based formalism (Figure 5). The MML\textsuperscript{65} represents a multiscale model by means of three types of components (submodels, filters that treat data flows between two submodels, and mappers that implement many-to-one connections between submodels) as well as several types of couplings for linking these components. MML-based model specifications can be registered in a repository called MAPPER Memory (MaMe), from which submodels can be selected by the Multiscale Application Designer (MAD) which is used to compose specific simulations. MAD exports the description of a composed simulation in the form of an executable experiment, which is then executed in a system called GridSpace Experiment Workbench (EW). The communication between different models is supported by the multiscale coupling library and environment (MUSCLE) which itself is based on JADE, a Java-based software agents middleware (\url{http://jade.cselt.it/}).
Figure 5. The key components of the CxA-based multiscale modelling and simulation environment, adapted from Falcone et al.\textsuperscript{65}

The above two multiscale modelling and simulation environments share a similar three-stage strategy, in which the modelling output is transformed from conceptual/abstract descriptions to the actual simulation results, via a journey sequentially delivering conceptual model/MML model specification, simulation script/executable experiment, and simulation execution. Several differences between these two solutions can also be noticed. A conceptual model in the GST-based solution is primarily system oriented, specifying the scales and components of the target system possibly with details about the phenomena and laws involved. Such conceptual models are natural (and formal) documentations of the multiscale models realised in the later stage and also hold the potential to enable model generation, following e.g. the approach proposed by Yang et al.\textsuperscript{66} In contrast, a MML model specification in the CxA-based solution more directly outlines the model structure with no intention to reveal detailed system knowledge encapsulated within each sub model. On the other hand, the GridSpace EW in the CxA-based solution offers comprehensive support to model execution in a heterogeneous grid environment, whilst the GST-based solution focuses on offering high-level logics for coordinating the simulations and simply relies on a middleware technology to handle the details pertaining to computing infrastructures. As pointed out by the work on the Common Component Architecture\textsuperscript{67}, a general middleware technology such as CORBA, i.e. the one currently used in the GST-based work, usually does not consider special requirements of high performance scientific computing, hence special features such as those supporting complex data structures and reducing communication overhead need to be incorporated into computing infrastructures to be suitable for demanding tasks such as multiscale simulation.
5. Conclusions and outlook

The development of multiscale modelling and simulation in different disciplines has brought common concepts, model classes and model construction procedures, as well as mathematical and numerical techniques for improving the efficiency, accuracy and stability of simulation experiments. The sharing of these common knowledge and methods between different disciplines may be achieved not only conceptually but also by means of some common computer implementation of models, algorithms, and tools. This paper has attempted to outline the key aspects, issues and notable progress towards common conceptual and computational frameworks. Based on what has been achieved to-date, future research may be pursued along the following directions:

- Towards a common conceptual framework, a unified conceptualisation of multiscale systems and models, shared by the multiscale modelling community across conventional disciplinary boundaries, may be developed to support the systematic understanding of such systems and models and to offer a common language for formulating concepts and methods for model construction and solution;

- Regarding model construction and solution approaches, techniques and analytical frameworks developed in general mathematical modelling such as those of model selection, model order reduction and simplification, error and stability analysis and parameter estimation, may be extended or adapted to address particularly the coupling between models of different mathematical nature and computational costliness;

- In computer implementation, common computational architectures may be developed which will accommodate software environments and libraries of computational components (models, algorithms) encapsulated at various levels of granularity, which are sharable by different domains or applications and are open to the domain/application-specific customisation and extension. These components will particularly include generic multiscale model solvers (or simulation co-ordinators) and algorithms to address common efficiency, accuracy and stability problems pertaining to inter-scale couplings. These software developments are expected to support the deployment and execution of simulations on the state-of-the-art computing platforms.
Acknowledgements

This work is supported by the UK Engineering and Physical Science Research Council (EPSRC) under Grant No. EP/G008361/1. The author is grateful for the contribution of Drs Cheng Jiang and Yang Zhao to this project. He also wishes to thank the anonymous reviewers for their constructive comments and suggestions on the manuscript.

References


