Progress on Plant-Level Components for Nuclear Fuel Recycling: Commonality[*]

Valmor F. de Almeida

Oak Ridge National Laboratory
Oak Ridge, TN 37831-6181, USA

Letter Report ORNL/LTR-2011/176

15 August 2011

[*] Also available by request to `dealmeidav@ornl.gov`.

Abstract

Progress made in developing a common mathematical modeling framework for plant-level components of a simulation toolkit for nuclear fuel recycling is summarized. This ongoing work is performed under the DOE Nuclear Energy Advanced Modeling and Simulation (NEAMS) program which has an element focusing on safeguards and separations (SafeSeps). One goal of this element is to develop a modeling and simulation toolkit for used nuclear fuel recycling. The primary function of the SafeSeps simulation toolkit is to enable the time-dependent coupling of separation modules and safeguards tools (either native or third-party supplied) that simulate and/or monitor the individual separation processes in a separations plant. The toolkit integration environment will offer an interface for the modules to register in the toolkit domain based on the commonality of diverse unit operations.

This report discusses the source of this commonality from a combined mathematical modeling and software design perspectives, and it defines the initial basic concepts needed for development of application modules and their integrated form, that is, an application software. A *unifying mathematical theory of chemical thermomechanical network transport* for physicochemical systems is proposed and outlined as the basis for developing advanced modules. A program for developing this theory from the underlying first-principles continuum thermomechanics will be needed in future developments; accomplishment of this task will enable the development of a modern modeling approach for plant-level models. Rigorous, advanced modeling approaches at the plant-level can only proceed from the development of reduced (or low-order) models based on a solid continuum field theory foundation. Such development will pave the way for future programmatic activities on software verification, simulation validation, and model uncertainty quantification on a scientific basis; currently, no satisfactory foundation exists for considering these activities.

The exploitation of mathematical commonality for unit operations with the intent of developing generic and comprehensive software for nuclear reprocessing has not been considered to this author's knowledge. Past attention has been given to plant-level processes on an individual and isolated basis, which has led to various models and corresponding codes implementing non-systematic approaches based on elementary principles of chemical processing. This practice has built an initial knowledge base, but it was not fruitful in producing a lasting and extensible simulation capability. In contrast, a common, rigorous, mathematical modeling framework for all plant-level operations, as proposed here, has a tremendous practical and theoretical value because it will reduce the software implementation work, creates a well defined modeling standard to compare past and future models, and more importantly, opens the doors

for scientific considerations of simulation fidelity; the latter has an obvious beneficial impact in supporting experimental validation programs. Therefore, the proposed framework is likely to generate a solid foundation for modeling plant-level processes for physicochemical nuclear (and non-nuclear) applications.

Demonstration of concrete module implementation is the subject of future communications including prototypes for several modules, namely, voloxidation, dissolver, digester, accountability tank, and solvent extraction. Unit-operation commonality is a key aspect to be explored for a successful implementation of these modules aimed at realizing various flowsheets and plant configurations.

Contents

List of Figures

1 Used fuel recycling toolkit development

Past communications (de Almeida et al., 2009, 2010b,a, 2011) described a top-down path for design and implementation of the SafeSeps fuel recycling toolkit software aiming at obtaining early functionality while preparing the foundation for supporting hierarchical modeling and simulation fidelity. Some of these ideas are restated here for completeness.

An integrated recycling plant is envisioned as a collection of all physicochemical processes needed to transform used nuclear fuel into new fuel and waste forms. Therefore, the plant will co-manage nuclear materials in the same facility, and the quality of the separated materials must match the input specifications for the product and waste downstream processes. In view of the breadth of processes involved, it is beneficial to begin the task of implementing the toolkit by considering a transient plant-level simulation capability that spans the entire plant. This capability would enable early feasibility analysis of existing and emerging recycling technologies and more importantly, would help identify areas for focused research and development. To ensure the longevity of the simulation capability, models should be modular, extensible, hierarchical, and aligned with a long-term vision to operate, control, and safeguard the recycling plant. To help prioritize and organize the task of designing modules and implementing corresponding codes, a plant can be viewed as a three-tier set of modules (fig. 1.1).

The top tier (plant-level) is a collection of *reduced* (sec. 5) models that provides throughput analysis, scheduling impacts, and output compositions. Models at this level generally summarize significant sections of the separations plant or collections of sections. These high-level, tightly-coupled models (*i.e.*, zero-dimensional models for which elapsed time is the principal independent variable) can use the simulation environment to invoke supporting simulations at other levels of (finer) detail not captured in a reduced model, such as those described by field-based chemical process models. Depending on the level of detail, reduced models require the smallest number of computer cycles from the pool of computers attached to the problem-solving environment, *i.e.*, a few processors to several dozens of processors.

In the middle tier (unit operations), macroscopic process models are designed to represent a specific unit operation or group of operations within the facilities. Models at this level are modular and may actually consist of several interchangeable models for the same separations plant section at varying levels of detail: for instance, modules for a space-time-averaged, one-dimensional model versus a two-dimensional fluid flow model of a chemical reactor. The models are typically based on well-founded physical approximations that meet the accuracy requirements of the analysis. For example, classical continuum thermomechanics with coupled chemistry models that are well calibrated and applicable for the systems at hand are representatives of the unit

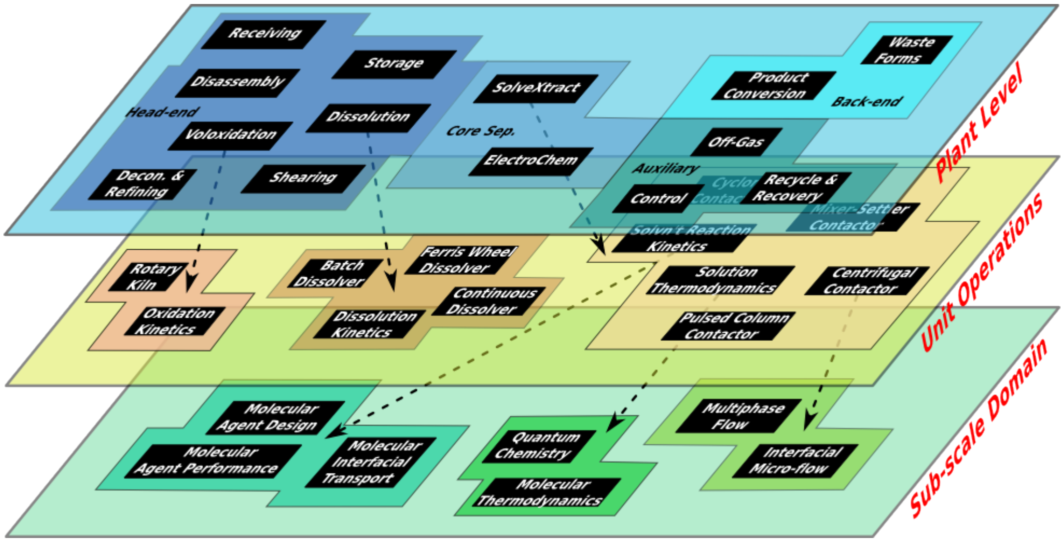

Fig. 1.1: *Tiered grouping of models and codes for a generic recycle plant. Top-down implementation approach guides development and determines research priority on lower tiers.*

operations tier. Typically, the geometry of the unit operation equipment is taken into consideration where a partition of the domain is required. Macroscopic models can also use decision logic in the simulation environment to call other process models or models at other levels of detail. The requirement for computing power at this level may be on the order of tens to hundreds or thousands of processors, depending on the particular model. Another role of the models in the middle tier is to provide a reference for their reduced plant-level counterparts (sec. 5.2). For instance, there are numerous ways a non-equilibrium plant-level model can be created. The middle tier models help provide a standard for applying averaging methods to obtain consistent, low-order models at the plant-level tier.

In the bottom tier (sub-scale domain), first-principles models can be developed to address specific computer-intensive calculations needed to accurately describe time-dependent, three-dimensional (or higher dimensional) systems. For example, these models could include solid thermomechanics with microstructure, turbulent fluid flow, micro and nanoflows, molecular dynamics, molecular mechanics, quantum mechanics calculations, and/or radiation transport if the added complexity is necessary to adequately model the system of interest. Computing power requirement at this level could be the highest available, up to hundreds of thousands of processors.

2 Technical objectives

The foregoing prompts a natural top-down approach to initiate design and implementation of codes and modules. Therefore, the purpose of the FY 2011 work has been to continue the development of a realistic, flexible, and effective simulation capability for a used nuclear fuel recycling facility that can be employed to evaluate the performance of alternative process flow-sheet options based on characteristics such as throughput, nonproliferation metrics, environmental and economic impact, and public policy and regulatory compliance. The models in the simulation will provide engineering insight at the appropriate level of detail for the evaluation of alternative flowsheets and/or plant configurations, including experimental support, process and equipment design and engineering, safety analysis, and material accountability.

The effort is aimed at the design and implementation of the toolkit software through a formal iterative process in parallel to the development and/or leverage of stand-alone modules for simulation of separation processes. The work is being implemented in concurrent steps, namely,

- assessment, where existing codes and modules are evaluated by separations and simulations experts,
- integration, where a modular software integration architecture is developed to allow for efficient use of existing codes and incorporation of new advanced codes; the subject of future reports (de Almeida, 2011) and
- advanced modeling, where detailed models of processes are created for which improved accuracy is necessary.

The objective of the work is to define a common modeling architecture for plant-level models and their corresponding modular software realization. In addition, the work is aimed at testing the architecture by prototyping prioritized models for aqueous reprocessing. The following sub-section describes advances in characterizing the behavior and architecture of the toolkit modules from a physicochemical modeling and software perspectives.

3 Toolkit plant-level module development

From a toolkit software perspective the plant-level tier consists of a non-empty family of modules (units of simulation software) denoted (see Noll, 1987, for notation)

$$M := (M_i \mid i \in \mathbb{N}^{\times}), \tag{3.1}$$

and an integration environment for composition of an application; this will be elaborated in section 4.4. A family is a generalization of the concept of a set which is used here in the mathematical description of an application. The application is the software product corresponding to a user-defined recycling technology via *time-coupling* of a set of modules (sec. 4.4). A module is a particular software realization of a mathematical model used for simulating a significant separation process. There are two important mathematical modeling aspects associated to a software module that need to be elaborated:

- *The external behavior of the model.* The abstract definition of the “exterior” of the model is of practical importance to develop two kinds of modules:

  - native modules developed and contributed by the SafeSeps group, and
  - third-party modules used or developed by users of the SafeSeps toolkit but not contributed to the toolkit set.

  The external behavior of the model is dictated by the necessary and sufficient data the model needs and is required to provide in order to be coupled to other models. This external view of the model is instrumental to implement a desirable feature in the modules of the toolkit, namely, to create or preserve the *stand-alone* functionality of the modules. That is, *both native and third-party modules should be usable outside the toolkit integration environment.* This is to accommodate the interest of users in utilizing modules without adopting the environment. A typical scenario is when (industrial) users already have their simulation framework and are only interested in using either a single or a few SafeSeps modules. However, this is not to incur duplication of work in implementing a stand-alone user interface for each SafeSeps developed module. *On the contrary, the toolkit environment is also viewed as providing a thin interface layer for stand-alone modules.* Furthermore, the characterization of the external behavior of a model also helps retrofit existing modules for usage in the toolkit environment.

- *The internal structure of the model.* The importance of mathematical commonality at the plant-level tier among disparate physicochemical processes cannot be emphasized enough. On the practical side, it reduces the amount of code implementation at the same time as making the end product more versatile and configurable. In particular, incorporating generic structural elements that will anticipate future changes stemming from parametric coupling of plant-level models and subscale calibrations from the tiers below will produce a higher quality software product. In addition, a common mathematical framework promotes interoperability between internal components, allowing for easy configuration of the module with alternative models of contactor equipment, thermodynamics, and kinetics. This will avoid redundancy of modules provided by the toolkit and will allow for development activities to concentrate on the modeling issues of the sub-components (divide-and-conquer approach). For instance, subsequent efforts will be focused in improving kinetics or contactor equipment modeling as opposed to developing an entire module that uses those components. Last but not least, sub-components may be shared between modules; for instance, gas kinetics in voloxidation may be used in off-gas modules interchangeably. In summary, there is great practical value in considering commonalities at the mathematical modeling level in order to take advantage of abstraction and generality. This exercise has a beneficial impact on the software design method with close analogy to object oriented and generic programing.

4 External behavior of a module

In this work, any plant-level separation process module $M_i \in M$ (3.1) behaves externally as follows (fig. 4.1):

1. M_i keeps its own control of how to advance its time t_i,

2. M_i needs a parameter vector $\boldsymbol{p}_i(t_i) \in \mathbb{R}^k$,

3. M_i may *need* one or more coupling vectors $\boldsymbol{q}_{\overrightarrow{j,i}}(t_i) \in \mathbb{R}^{l_i}$ from one or more modules M_j, unless it is a singleton, then no coupling vector is needed,

4. M_i may need to *provide* one or more coupling vectors $\boldsymbol{q}_{\overrightarrow{i,j}}(t_-)$ or $\boldsymbol{q}_{\overrightarrow{i,j}}(t_+)$ (or both) to one or more modules M_j $(i \neq j)$ either in the past, $t_- \leq t_i$ or in the future $t_+ > t_i$; the latter implies a zero-dimensional vector as a response, and

5. M_i is coupled to as many other modules M_j as necessary to form an application.

This external view makes all modules to behave equivalently for all separation processes allowing for the development of a common coupling interface.

4.1 Time evolution

Any module M_i advances time t_i at its own pace and this is an internal feature of the module (sec. 5). Therefore, the module is given a signal to start with a desired final time T and, as it evolves time, it *requests* external data at time t_i. In addition, requests are made by the environment *to* the module at times that may be in the

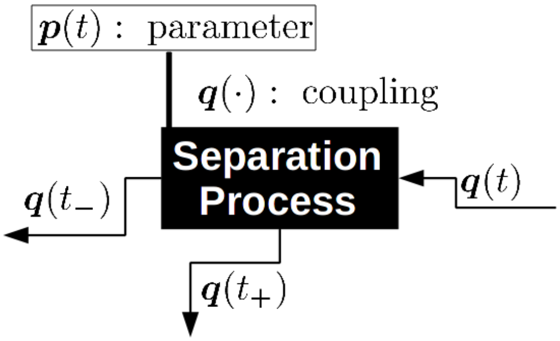

Fig. 4.1: *External view of the behavior of a plant-level separation module. Many, if not all, plant-level processes exhibit this external behavior; others can be treated on a special-case basis. The parameter vector, $\boldsymbol{p}(t)$, is always evaluated at time t of the module. Also at time t, the external coupling data vector, $\boldsymbol{q}(t)$, must be provided upon request by the module. In addition, the module must be able to provide coupling data requested by others at times that precede t, that is, $t_- \leq t$, and at future times $t_+ > t$; this latter case is handled with a trivial response of unavailable data.*

past, t_-, or in the future, t_+, relative to t_i. If the time is in the past, the module will provide historic data, otherwise a trivial response of unavailable data is returned.

Time marching is an internal feature of the module, hidden from the outside, and it is only controlled by the solution method for evolving the state of M_i (sec. 5). *This is an important practical concept which allows for the usage of stand-alone modules.*

Other benefits include: better use of computational cycles by not enforcing a time step for all modules in the application; exploitation of parallelism and/or multithreading; and freedom for usage of time-adaptive solution methods that fit the needs of individual modules.

4.2 Parameter vector

The parameter vector $\boldsymbol{p}_i(t_i)$ collects the configuration parameters values for the ith module M_i. These quantities are needed for operating and configuring the model, for instance, in a rotary dissolver, the rotation rate of the drum, the period of rotation reversal, and the inclination of the dissolver drum are part of $\boldsymbol{p}_i(t_i)$. In the module realization of the model, the parameter vector is data received from the software environment that integrates all modules. A module may induce a change in the parameter vector of other distant module, *i.e.* without a physical connection. This situation is handled via signals. A typical example is the simulation of exceptional conditions that prompt one module to send an alert to an another module to either stop or change its processing configuration. Therefore $\boldsymbol{p}_i(t_i)$ can be modified by a M_j module ($j \neq i$) but this situation takes place indirectly, hence the absence of an additional j sub-index in $\boldsymbol{p}_i(t_i)$ (fig. 4.2).

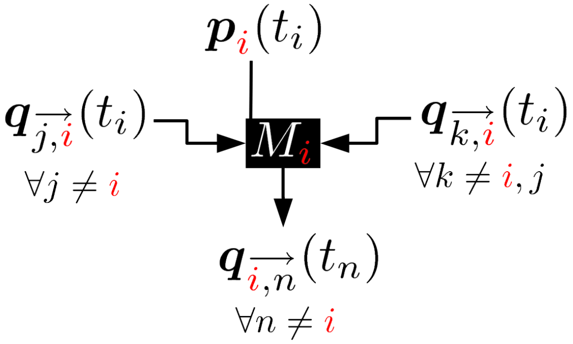
$\boldsymbol{p}_i(t_i)$
$\boldsymbol{q}_{\overrightarrow{j,i}}(t_i)$
$\forall j \neq i$
M_i
$\boldsymbol{q}_{\overrightarrow{k,i}}(t_i)$
$\forall k \neq i, j$
$\boldsymbol{q}_{\overrightarrow{i,n}}(t_n)$
$\forall n \neq i$

Fig. 4.2: *Asynchronous time-dependent data flow into and out of the ith module. At time t_i, the module requests coupling and parameter data. No loop of data (self-connection) is allowed. The coupling data provided by the ith module to any nth module, $\boldsymbol{q}_{\overrightarrow{i,n}}(t_i)$, takes place at the time t_n which may or may not be equal to t_i.*

4.3 Coupling vector

The coupling vector $\boldsymbol{q}_{\overrightarrow{j,i}}(t_i)$ groups external physicochemical quantities provided by M_j modules connected to M_i ($j \neq i$, a module cannot be self-coupled). These quantities constitute a collection of required data by M_i at time t_i. For instance, in a rotary dissolver, many external time-dependent, coupling quantities are part of $\boldsymbol{q}_{\overrightarrow{j,i}}(t_i)$, those are: inlet liquid phase flow rates and their mass concentrations of nitric acid, solid phase mass rate of the feed stream, molar fraction of oxides in the solid phase, surface area of the solid phase, and purge gas flow rate. All these quantities are needed at time t_i during the time evolution of M_i. The time t_i is viewed as the internal time of M_i which advances at its own pace regulated by the arrival of the requested coupling data $\boldsymbol{q}_{\overrightarrow{j,i}}(t_i)$. This represents an asynchronous data coupling, that is, each module has its own independent internal time marching procedure which is balanced indirectly by the request of data.

Every module must make available to its external environment a coupling vector required by connected modules which request data at a particular time. Here $\boldsymbol{q}_{\overrightarrow{i,k}}(t_k)$ denotes the coupling data sent from M_i to M_k at t_k. This *outgoing* information is computed on the fly by the provider module, M_i. Since M_i is currently at t_i, this generation of data can only occur for $t_i \geq t_k$, otherwise M_k receives a message to wait and re-submit its request for $\boldsymbol{q}_{\overrightarrow{i,k}}(t_k)$ later. *This is the indirect mechanism that synchronizes the whole running simulation.* In this paradigm, a module is required to keep not only its current state but the knowledge of the whole state history so the data generation can be made in the past ($t_i > t_k$).

In summary, the distinction between parameters $\boldsymbol{p}$ and coupling $\boldsymbol{q}$ quantities is necessary in view of the origin of the coupling vector $\boldsymbol{q}$. That is, it is a time-dependent (past and/or future), post-processing operation on the internal state of the module that creates the coupling vector (sec. 5). In contrast, parameter vectors are always evaluated at the same time of the internal state of its module and do not depend on the existence of other modules.

4.4 The application network

This section elaborates on the structures surrounding the external environment of a module necessary for the practical creation of a software application. This is a topic for a future report on the software environment for building applications but some preliminary discussion at this point helps solidify the conceptual behavior of modules.

As mentioned earlier, an application is a software product obtained through composition of modules. In practice, composition is realized by building a network of modules that represents a particular system design (*i.e.* a plant, or a portion of a plant, or a process flowsheet, or even a single separation process). The current work employs a formal network as the basis for creating an application software. This network is not to be confused with the thermomechanical network described later (sec. 5.2), however the repetitive use of a network concept is intentional and a reflection of the many different ways network theory is put in practice today.

The application network is a triple

$$\boxed{(G_{\text{app}}, \varpi_{\text{v}}, \varpi_{\text{a}}),} \tag{4.1}$$

where the first term denotes the network graph, the second and third terms denote two mappings used to *decorate* the graph on the vertices and arcs, respectively. The network graph, G_{app}, is a directed bi-edge graph of modules (or vertices) where directed edges (arcs) are used for assigning the coupling vectors to a data structure (fig. 4.3). Therefore the application network only admits anti-parallel arcs joining any two modules; an arc joining the same module (a loop) is forbidden.

This work defines an application network graph as a triple

$$\boxed{G_{\text{app}} := (V, A, \psi_{\text{G}}),} \tag{4.2}$$

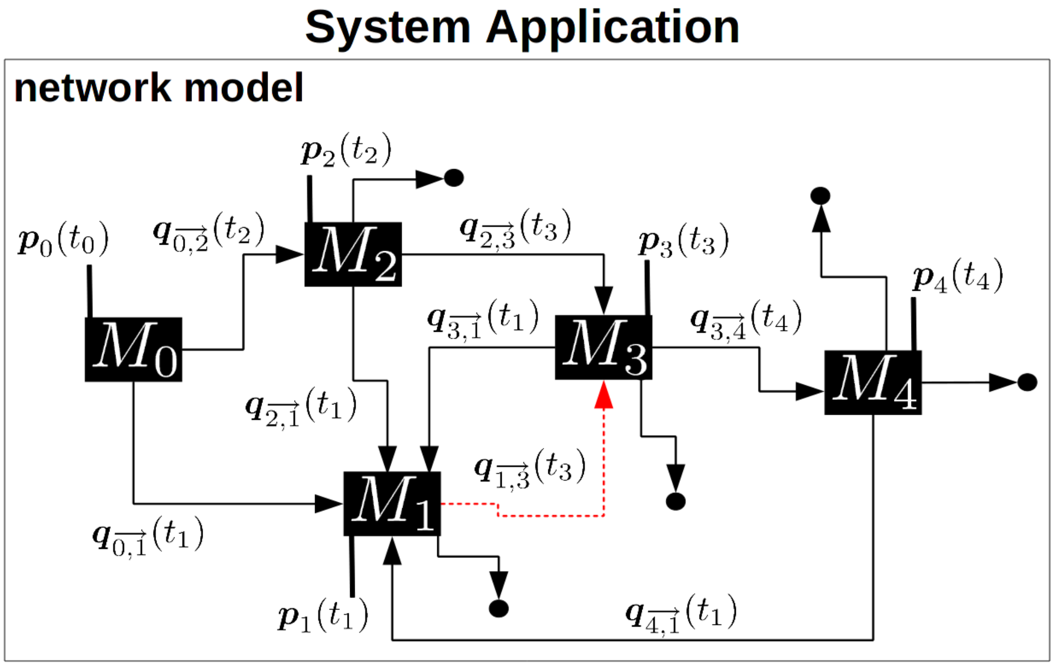
System Application
network model
$\boldsymbol{p}_0(t_0)$
$\boldsymbol{q}_{\overrightarrow{0,2}}(t_2)$
$\boldsymbol{p}_2(t_2)$
$\boldsymbol{q}_{\overrightarrow{2,3}}(t_3)$
$\boldsymbol{p}_3(t_3)$
$\boldsymbol{p}_4(t_4)$
M_0
M_2
$\boldsymbol{q}_{\overrightarrow{3,1}}(t_1)$
M_3
$\boldsymbol{q}_{\overrightarrow{3,4}}(t_4)$
M_4
$\boldsymbol{q}_{\overrightarrow{2,1}}(t_1)$
$\boldsymbol{q}_{\overrightarrow{1,3}}(t_3)$
M_1
$\boldsymbol{q}_{\overrightarrow{0,1}}(t_1)$
$\boldsymbol{p}_1(t_1)$
$\boldsymbol{q}_{\overrightarrow{4,1}}(t_1)$

Fig. 4.3: *A network directed multigraph of modules for a generic application with asynchronous, parallel time evolution. Alternatively, this network could be a simple representation of an aqueous-based process: M_0 for shearing; M_1 for off-gas; M_2 for voloxidation; M_3 for dissolution; M_4 for solvent extraction. The $\boldsymbol{q}_{\overrightarrow{1,3}}(t_3)$ edge data could represent the nitric acid recycle. This establishes a cycle between the off-gas and dissolution modules. This cycle will cause a grid lock during asynchronous time evolution of the system that needs to be handled by the network implementation.*

where

$$\begin{aligned} V &:= \operatorname{Rng} M, && \text{is the modules set,} \\ A &:= (A_k \mid k \in n^{\rceil}),\ \operatorname{Rng} A \subset V \times V, && \text{is a list of arcs in } V \times V, \\ \psi_{\mathrm{G}} &: A_k \in A \to (M_i, M_j) \in V \times V && \text{is the incidence mapping.} \end{aligned}$$

The modules set, V, is a collection of distinct m modules obtained from the family of modules M (3.1). The arc-list, A, of length n, is injective ($A_i = A_j \Rightarrow i = j$). And the incidence mapping, ψ_{G}, builds the graph by associating an ordered pair of modules with each arc in A; ψ_{G} is injective, thus given $(M_i, M_j) \in \operatorname{Rng} \psi_{\mathrm{G}}$ there exists an unique $A_k \in A$ (through the mapping's left inverse). This ordered pair denotes the beginning and ending of the arc which is further associated to (or decorated with) a coupling vector via the network mapping $\varpi_{\mathrm{a}}\colon (M_i, M_j) \in \operatorname{Rng} \psi_{\mathrm{G}} \to \boldsymbol{q}_{\overrightarrow{i,j}} \in \mathbb{R}^{l_j}$, where the i, j indices are obtained through the incidence graph mapping $\psi_{\mathrm{G}}(A_k)$. Similarly, the network vertices are decorated with parameter vectors via the mapping $\varpi_{\mathrm{v}}\colon M_i \in V \to \boldsymbol{p}_i \in \mathbb{R}^{l_i}$. According to the external behavior of modules described in previous sections, arcs are anti-parallel and no more than two arcs exist between two connected modules.

There is great benefit in representing the composition of an application as a network in view of the existence of excellent graph libraries for handling the associated graph data structure, and numerous available algorithms for operating on the graph. Graph theory has evolved at a fast pace in the past decade with various applications in image processing, networks, the World Wide Web, traffic flow, porous media transport, routing, circuits, chemistry, biology, physics, mathematical financing, gaming, economics, telephone records, and national security; to name a few in addition to journals specialized in graph theory and applications (Jungnickel, 2008; Bondy and Murty, 2008).

Of particular interest here are the network synthesis operations for which one desires to find economic network paths for a particular design, combinatorial optimizations, constrained networks, etc. These by-products from related areas of research can be quickly leveraged in the network defined here (4.1).

Only the external behavior of the modules (vertices) is of interest to the application network, which is a major part of the environment needed by the modules for realizing time coupling. In view of the aforementioned behavior of the modules, the arcs (edges) of the graph need to be directed, and in view of the nature of the intended applications, arcs must be anti-parallel and loop arcs forbidden. Arcs can form a cycle sequence among vertices (modules) and this represents a potential source of grid lock for the asynchronous time evolution of the underlying application system. Thus, the network environment needs to keep track of data requests within a cycle between modules and to break the loop when needed. This can be accomplished expeditiously by using graph search algorithms; this will be discussed in a future communication. One last software feature of practical relevance for modules is the ability to be systematically grouped. This can be achieved by using recursive networks wherein a module itself is composed of a small network of sub-modules. The practical implementation of recursion will require a structure for handling generations of modules through parent-child relations; this is also a topic for future communications.

In summary, the essence of a plant-level module is that

- it advances its own time until a given final time T,
- it may be coupled to other modules through parameter and coupling vectors to form a network (the basis of an application), and
- it regulates its time evolution asynchronously via the network domain when time-coupled with other modules.

5 Internal structure of a module

A module M_i has internal state $\boldsymbol{x}_i(t_i) \in \mathbb{R}^m$ (m here is not to be confused with the cardinality of V in G_{app} (4.2)), that is, a list of all dependent physicochemical variables (unknowns). For example, the state vector of a dissolver module (de Almeida, 2011) contains the following time-dependent, stage-wise quantities: liquid phase flow rate, liquid phase volume, mass density of liquid phase, mass concentration of acid in the liquid phase, mass of the solid phase unconvected with the liquid phase, mass concentration of uranyl nitrate, mass concentration of fission product nitrate, mass concentration of plutonium nitrate, and nitrogen oxide or nitrous oxide mass concentration in the vapor phase. Additional quantities will be added as the model is improved and extended, but the data characterization of the module will most likely remain the same with an expanded state vector and corresponding parameter and coupling vectors.

As stated before (sec. 4.3), every module must make available to its external environment a coupling vector required by connected modules which request data at a particular time (fig. 5.1). Here $\boldsymbol{q}_{\overrightarrow{i,k}}(t_k)$ denotes the coupling data sent from M_i to M_k at t_k. This *outgoing* information is computed on the fly by the provider module, M_i, by either mapping or post-processing its internal state into the requested data: $\boldsymbol{x}_i(t_k) \mapsto \boldsymbol{q}_{\overrightarrow{i,k}}(t_k)$. Since M_i is currently at t_i, this mapping can only occur for $t_i \geq t_k$, otherwise M_i produces a null vector, $\dim(\boldsymbol{q}_{\overrightarrow{i,k}}(t_k)) = 0$, and sends it to M_k as an indication that no data is available. In turn, M_k waits and resubmits its request for data at a later time in the simulation. This mechanism synchronizes the running simulation. In this paradigm, a module is required to keep not only its current internal state $\boldsymbol{x}_i(t_i)$ but the whole state history $\boldsymbol{x}_i(t)\ \forall\ t \le t_i$ so the aforementioned mapping can be made in the past ($t < t_i$).

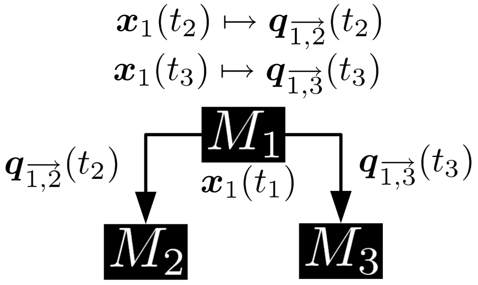
$\boldsymbol{x}_1(t_2) \mapsto \boldsymbol{q}_{\overrightarrow{1,2}}(t_2)$
$\boldsymbol{x}_1(t_3) \mapsto \boldsymbol{q}_{\overrightarrow{1,3}}(t_3)$
M_1
$\boldsymbol{q}_{\overrightarrow{1,2}}(t_2)$
$\boldsymbol{x}_1(t_1)$
$\boldsymbol{q}_{\overrightarrow{1,3}}(t_3)$
M_2
M_3

Fig. 5.1: *Module data coupling. The coupling vector from module 1 to 2 and 3 is computed through a mapping from the internal state of M_1 to the requested data by modules 2 and 3 at times specified by the requesting modules; here t_2 and t_3. For this to happen, M_1 must be at $t_1 \ge t_2$ and $t_1 \ge t_3$ when the requests for data are made.*

The internal structure of a plant-level separation module can be advantageously decomposed into three modeling sub-components and a computing engine (fig. 5.2), namely,

- contacting operations,
- transport balances,
  - mass
  - power
  - energy
  - entropy
- constitutive restrictions
  - thermodynamic constraints
  - kinetic rates
- differential-algebraic equations (DAE) solver.

In what follows, these sub-components will be described.

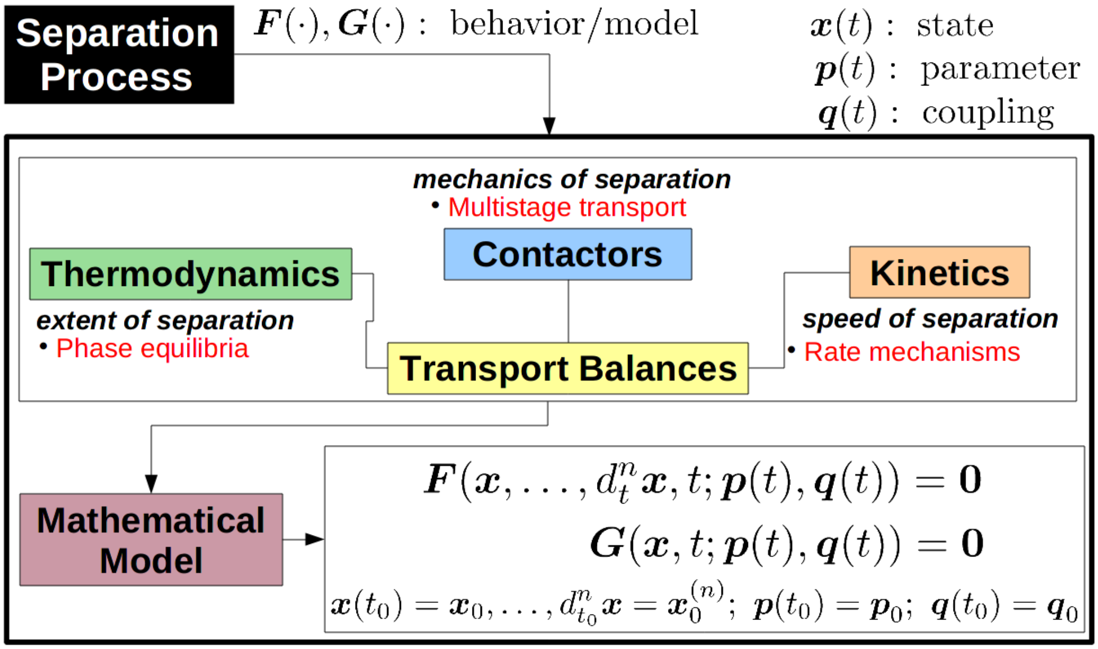

Fig. 5.2: *Internal view of the mathematical structure of a generic plant-level separation module. Many plant-level processes can use the components of this modeling structure; others can be treated on a special-case basis.*

5.1 Contacting operation

This work considers a contacting operation as any physicochemical process that enables the intimate contact of two or more material phases. In this generic view, a vast number of systems can be classified as a contacting operation. For instance, fluid flow in a duct is a contacting operation since the fluid is in contact with the tube wall; if a "gutter" flow regime is the prevalent flow in this example, then a two-phase gas/liquid flow may have to be considered in the description of the contact operation.

While all module sub-components are equally significant, the phase-contacting operation has a central position among the others (fig. 5.2). This is so because the mechanics of the contacting operation is the foundation of the underlying model.

The modeling approach followed here to represent phase-contacting devices employs a general stage-wise cascade network (fig. 5.3) of either physical or logical stages (or both). Numerous models of the *same* contactor equipment can be developed using a network approach, which is highly desirable to support flexibility and modularity. Furthermore, using the associated network graph, the integration of the contactor with other structural components becomes tractable. That is, a contactor can be used with different kinetics, thermodynamics, and balance equations without internal changes.

Alternatively, models of different contactors can be readily plugged into a given module $M_i \in M$ (3.1) allowing for generality and enabling interoperability of different contacting devices; a desirable internal behavior. Therefore, the application of equilibrium and non-equilibrium considerations will be independent of the particular contacting equipment, which has been abstracted into a connected network of stages. *This decomposition will eliminate a common implementation problem of past codes which intertwined all elements and therefore had no natural extensibility property.* For example, a change in kinetics or contactor device in legacy computer programs typically leads to a total overhaul of the code; moreover, there is no transferability to other seemingly different separation processes. In other words, existing codes for different separation processes share virtually no sub-components.

A stage-wise cascade network model of a contactor device is composed of vertices and their connecting edges which form a graph. A vertex (also called here a stage or node) controls the evolution of an associated subset of state variables (*i.e.*, the state variables associated to the vertex). If the network has only one vertex, then all state variables in the corresponding module are controlled by this singleton. In a sense the state variables of a module are distributed among vertices in the network and later assembled in one vector of all variables for the equation solver (sec. 5.4). The vertex is a lumped representation of the physicochemical contact between material phases; therefore, material streams of individual phases arrive at the vertex, interact, and leave the vertex. The transit of information is represented by directed multi-edges

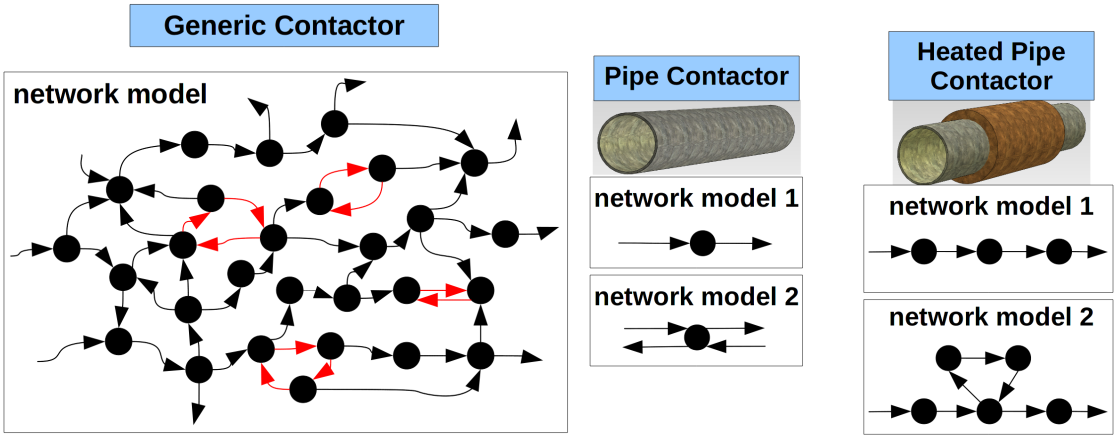

Fig. 5.3: *Stage-wise network model of contactor devices.* ***Left:*** *A generic contactor device is decomposed into a network of vertices. Each vertex represents a stage directly connected to its neighbors. The resulting network is modeled by a directed multi-graph with the possibility of loops and many parallel arcs. The arcs/edges model transport phenomena between stages (alternatively called vertices or nodes) in the equipment.* ***Center:*** *A geometrically simple contactor equipment with a simple one-directional flow (network model 1) and a countercurrent two-phase flow (network model 2). Vertices (stages) represent lumped models of significant domains in the equipment.* ***Right:*** *A heated pipe network model with one-directional flow (network model 1), and a one-directional multispecies flow with wall deposition phenomena (network model 2).*

for each material phase. Note that edges do not represent physical streams because of the abstract nature of the network representation. A vertex may be thought as a control volume in a general form, thus the existence of physical streams is an option, not a basis for the conceptual foundation of the network. In view of the potential phase-change processes taking place at a vertex, material phases are not conserved in general. Therefore, the inflow of a phase does not guarantee an outflow of the same phase.

An elaboration of the concept of a phase is in order. A phase is a part of a system which is bounded by a *distinguished* macroscopic interphase region. This region is approximated by an interface, that is, a mathematical surface. Hence, a phase is defined by its boundary. In the interior of a phase, chemical composition and physical properties are smoothly varying, while across the interface material properties are sharply varying. In particular, interface discontinuity of properties is a typical assumption. Therefore, in general, a phase is a region of space and it may or not coincide with a material body. If mass transfer is allowed across the interface, the phase is clearly not a material body. The concept of a phase is then similar to the concept of a mathematical closed domain, that is, a set of points in the Euclidean point space including its boundary. This domain is then endowed with additional structure to function as a model of a physical phase.

A phase is called uniform when all material points are isomorphic, that is, the mechanical, thermal, and chemical properties are the same for all points. Roughly, this means that all material points belong to the same material. A uniform phase is called homogeneous if any material point subjected to the same physicochemical process produces an identical response. In other words, the result of a physicochemical process applied to any two material points in a homogeneous phase is indistinguishable. A homogeneous phase is a specialization of a uniform phase. Phases in classical thermodynamics are typically homogeneous phases. A homogeneous phase is called isotropic when its response to a physicochemical process does not depend on its orientation.

Returning to the network model of a contactor, two incoming phases (δ and β in fig. 5.4; left) from neighbor vertices may transform into another phase (say, γ) during their interaction in the kth vertex. In this example, while the α phase is shown as preserved, it may have been substantially reduced in its output in view of the internal processes in the vertex.

Each phase exiting a vertex is represented by a single edge which may be the result of the interaction of multiple incoming phases, each one represented by an individual in-arc. There may be many incoming phases into a vertex with many outgoing phases from a vertex (countercurrent flow example in fig. 5.4; right).

A significant structure in the mathematical representation of a contactor network is the underlying graph and associated graph theory for elaborating the chemical

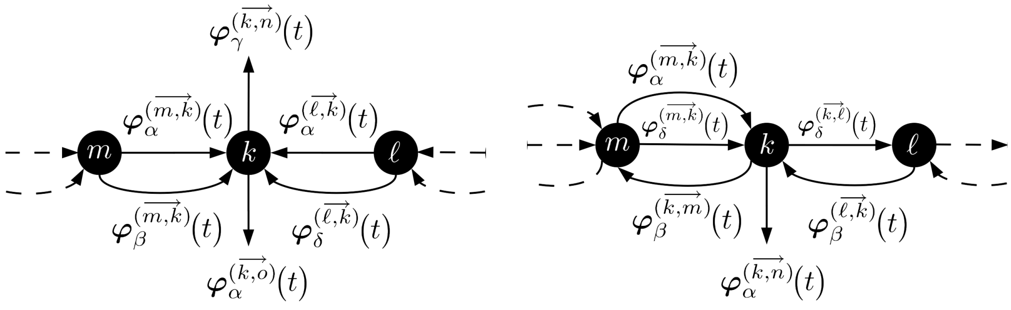
$\boldsymbol{\varphi}_\gamma^{(\overrightarrow{k,n})}(t)$
$\boldsymbol{\varphi}_\alpha^{(\overrightarrow{m,k})}(t)$
$\boldsymbol{\varphi}_\alpha^{(\overrightarrow{\ell,k})}(t)$
m
k
ℓ
$\boldsymbol{\varphi}_\beta^{(\overrightarrow{m,k})}(t)$
$\boldsymbol{\varphi}_\delta^{(\overrightarrow{\ell,k})}(t)$
$\boldsymbol{\varphi}_\alpha^{(\overrightarrow{k,o})}(t)$
$\boldsymbol{\varphi}_\alpha^{(\overrightarrow{m,k})}(t)$
$\boldsymbol{\varphi}_\delta^{(\overrightarrow{m,k})}(t)$
$\boldsymbol{\varphi}_\delta^{(\overrightarrow{k,\ell})}(t)$
m
k
ℓ
$\boldsymbol{\varphi}_\beta^{(\overrightarrow{k,m})}(t)$
$\boldsymbol{\varphi}_\beta^{(\overrightarrow{\ell,k})}(t)$
$\boldsymbol{\varphi}_\alpha^{(\overrightarrow{k,n})}(t)$

Fig. 5.4: *Contactor vertices and their edges phasic data. The flow of phase quantities, $\boldsymbol{\varphi}$, at time t into and out of the kth vertex from its neighbors.* ***Left:*** *Here the β and δ phases transform into the γ phase which is extracted from vertex k. The α phase is partially consumed in the process.* ***Right:*** *A typical countercurrent flow between phases α and β with product extraction of the α phase. In addition the δ phase could represent an offgas stream.*

thermomechanical network model and manipulating its data (in a sense, similar to the application network, sec. 4.4).

This work defines a contactor graph by a triple

$$\boxed{H_{\text{ctor}} := (V, A, \psi_{\text{H}}),} \tag{5.1}$$

where $V := \text{Rng}(S_i \mid i \in \mathbb{N}^{\times})$ is a set of N_{v} vertices representing all stages in the contactor network, $A := (A_k \mid k \in n^{]})$ is a list of arcs of length n in $V \times V$, and $\psi_{\text{H}} \colon A_k \in A \to (V_i, V_j) \in V \times V$ is the incidence mapping that assigns pairs of vertices (stages) to arcs. Since an arc is associated to a material stream, the mapping $(A_k \mapsto \alpha) \colon A \to C$, where C is the set of phase types, associates to each arc a unique phase type. In order to accommodate a variety of contacting operations, H_{ctor} must be a directed multigraph with support for cycles and loop arcs. Thus $\psi_{\text{H}}(A_k) = \psi_{\text{H}}(A_m)$ implies that A_k and A_m are parallel arcs. In addition, ψ_{H} is not invertible in general.

A chemical thermomechanical (contactor) network is then built as a triple

$$\boxed{(H_{\text{ctor}}, \boldsymbol{\varpi}_{\text{v}}, \boldsymbol{\varpi}_{\text{a}}),} \tag{5.2}$$

where vertices are decorated by the mapping $\boldsymbol{\varpi}_{\text{v}} \colon V_k \in V \to \boldsymbol{\varphi}^{(k)} \in \mathbb{R}^{l_k}$, with $\boldsymbol{\varphi}^{(k)}$ representing all state variables associated to stage k. Likewise, arcs are decorated with a mapping, $\boldsymbol{\varpi}_{\text{a}} \colon A_k \in A \to \boldsymbol{\varphi}_\alpha^{(\overrightarrow{i,j})} \in \mathbb{R}^{l_i}$, to flow state variables with the i, j indices determined by $\psi_{\text{H}}(A_k)$ and α assigned by $(A_k \mapsto \alpha)$.

As mentioned earlier (sec. 4.4) there is great incentive for a graph description of any network structure. Here, the intended use of a graph is similar to the application network (sec. 4.4) but with many additional features prompted by the underlying physicochemical processes lumped in each vertex of the graph. Since multiple material streams may enter and/or leave each graph vertex, the graph must allow for any number of arcs between vertices. In addition, in view of the abstraction of a physical device as a network, a very large graph may result. In this regard, one interesting identification is the similarity of a network and a conventional discretization (*e.g.*, control volume method, finite differences, finite elements, etc.) of partial differential equations that ultimately model the chemical thermomechanical continuum processes associated to the stages of the network (fig. 5.5). This similarity justifies the view followed in this work that the physicochemical models associated to vertices of the network are *reduced (or lower order)* models of the underlying continuum, and it motivates the development of a rigorous approach for deriving network models based on thermomechanics theory applied to a network system; the topic of the next section.

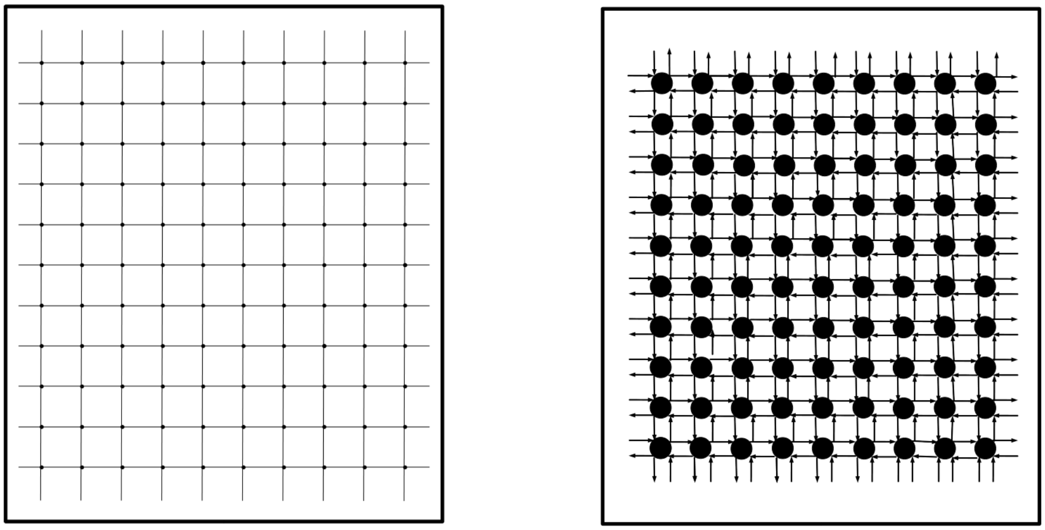

Fig. 5.5: *Mesh discretization and graph analog. A network approach can be utilized to mimic conventional methods for solution of partial differential equations used as models of continuum physics. This identification can be made practical for sufficiently large network graphs. Therefore a network/graph approach is instrumental for the development of the multi-tier framework described earlier (sec. 1) wherein lower tiers implement improved models of physicochemical phenomena. In addition, the similarity of a nodal mesh guides the reduction of partial differential equation models to a derived network model.* ***Left:*** *Nodal based mesh.* ***Right:*** *Corresponding network.*

5.2 Non-equilibrium network transport balances

A theory of network transport for physicochemical processes is the key to unify the modeling of plant-level systems. To this author's knowledge, such a theory has not been developed from first-principles chemical thermomechanics of continua—the only tractable, well-established field theory of macroscopic material transport (Truesdell and Toupin, 1960; Truesdell and Noll, 1965). A full program to develop a network transport theory is not described here; it is still a work in progress. Hence, this section is an outline of the results of the current status of the development.

What makes the development of the theory challenging, even without the inclusion of material behavior, is the consideration of various routes one can take to arrive at the final set of basic governing network equations. This prompts the question of whether a common form for the governing equations can be obtained for any route taken. The task is feasible by observing the limits of a contacting operation. That is, either the phase contact takes place with mixing by dispersing a phase into another or it takes place by direct or indirect contact preserving the continuity of the phases (fig. 5.6).

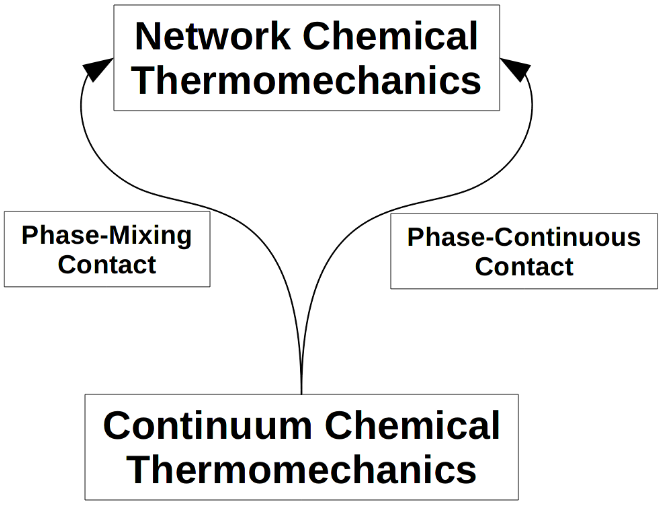
Network Chemical Thermomechanics
Phase-Mixing Contact
Phase-Continuous Contact
Continuum Chemical Thermomechanics

Fig. 5.6: *Paths for network theory and modeling of plant-level processes. The network transport theory derives from up-scale averaging of processes in two operation limits. Either the process contacts phases by virtue of a dispersive mixing or by means of a continuous phase interaction. A hybrid between the two contact modes is possible, but in any case, consideration of the two extreme modes suffice to arrive at a unified network theory at the plant-level scale.*

The phase mixing mode is the most challenging to treat since it will require an underlying mixture theory before a network transport balance can be developed. A theory for non-mixing development is less intricate but still challenging since other complications typically are involved. Also, consideration needs to be given to hybrid contacting operations that involve partial phase mixing and partial non-mixing.

To make the foregoing clear, consider the contrast between a solvent extraction and a distillation contacting operation. In the former, vigorous mixing is used with two immiscible liquids to create a large interfacial area in the resulting dispersion. In distillation, creation of a dispersion must be avoided, therefore the contact operation employs a packed bed medium to generate the needed interfacial area by wetting the solid support. A successful network transport theory must accommodate these extremes.

The governing equations described below do reproduce and extend existing models of specific plant-level processes, but much more remains to be done before robust mathematical models can be fully developed. This development represents an initial conceptual test for the feasibility of a modern network theory. On the practical side, preliminary model prototypes, relevant to aqueous-based systems, have been worked out and will be presented in future reports (*e.g.* de Almeida, 2011).

There is considerable variability and lack of a solid theoretical reference on past models derived for physicochemical processes at the plant-level of nuclear reprocessing applications. The majority of models (Long, 1967; Benedict et al., 1981) consider processes in some form of equilibrium wherein simplified mass balances and thermodynamic relations suffice, a practice traced to elementary principles of chemical processing (Felder and Rousseau, 1986). As a result there exists uncertainty in extending the models to include additional physics and chemistry with the intention of improving the fidelity of the results. For instance, for a given classical stage-wise, time-dependent, solvent extraction model with the assumption of thermodynamic equilibrium in every stage (Gronier, 1991), what would be the path to incorporate measurements of mass transfer, viscosity, interfacial tension, rates of specie complexation, and mixing power in the model? What additional equations are needed? What modifications of the existing equations are needed? What is the basis for deriving such equations and where do they come from?

Basic continuum mass, momentum and energy balances, and entropy imbalance are seldom used in plant-level models; a much needed framework for the careful consideration of physicochemical phenomena on a scientific basis. For instance, many separation processes utilize vigorous mixing in contact devices but their plant-level models do not consider phasic kinetic energy as a state variable, making it difficult to include non-equilibrium mechanical effects in the model. This state of affairs is an indication of the work ahead needed to derive consistent models using basic universal building blocks of material behavior.

Governing equations provide the means to mathematically evolve the values of the state vector $\boldsymbol{x}(t)$ (sec. 5) of a module in a time interval

$$I := \{t \in \mathbb{P} \mid 0 \leq t \leq T\}$$

where T is a pre-specified final time of evolution. The set of variables that define the state of a generic plant-level model should be rooted in the underlying continuum chemical thermomechanical model of the system. While there are various ways of deriving reduced versions of a continuum model, this work attempts to consider what is of practical interest and value while leaving room for elaboration of the model with inclusion of finer details of the system *a posteriori*; a needed feature for later implementation of a multi-tiered structure (sec. 1, fig. 1.1).

This work considers the structural framework of the plant-level models in the form of a network of stages (sec. 5.1) where material phases interact. An important characterization of the model is the absence of spatial gradients, or any gradients for that matter, associated to a stage. Hence, a stage is essentially a reference point for the time-space average of the underlying continuum fields over the associated spatial domain, that is, effectively a control volume. Therefore, the approach followed here applies volume averaging of basic balance equations over a control volume associated to a stage. No unique model will result from this approach, which depends on the averaging method. In addition, the complexity of the sub-scale phenomena can be incorporated in many different ways which makes the resulting governing equations to vary substantially. Furthermore, the boundary conditions of the sub-scale continuum need to be properly averaged and included into the main averaged network equations; this only multiplies the various ways a reduced model can be derived.

Notwithstanding this variability, the averaged theory employed here establishes an unambiguously consistent way to derive models for any plant-level process with clarity and exposure of the assumptions used to create a reduced-order model (or a class of models). For example, one class could consider microstructure explicitly in the primary dependent variables, say a multiscale model, while another class could consider microstructure in an indirect form through parametric coupling with another model. Particular description of the averaging methods will be described in future communications.

From the network governing equations listed in the next sections, the principal, time-dependent, state variables for the kth stage are:

- phase intrinsic specie mass concentration, $\rho^{(k)}_{\mathit{a},\alpha}$, ($\mathit{a}$th specie, α phase),
- phase volume, $V^{(k)}_{\alpha}$,
- phase volumetric outflow rate, $f^{(k)}_{\alpha}$,
- phase specie dispersion flow rate, $g^{(k)}_{\mathit{a},\alpha}$,

- phase intrinsic mass density, $\rho^{(k)}_\alpha := \sum_a \rho^{(k)}_{a,\alpha}$,
- phase intrinsic specific kinetic energy, $\frac{1}{2}v^{2(k)}_\alpha$,
- phase intrinsic external traction power, $t^{(k)}_\alpha$,
- phase intrinsic specific internal energy, $\varepsilon^{(k)}_\alpha$,
- phase intrinsic external heat power, $q^{(k)}_\alpha$,
- phase intrinsic specific entropy, $s^{(k)}_\alpha$,
- absolute phase temperature, $\theta^{(k)}_\alpha$.

These are to be augmented by electromagnetic variables if the physicochemical process being modeled calls for them, such as in electrorefining operations.

Basic transport balance equations, constitutive equations, initial data, and material properties must be specified to allow for the calculation of the history of the state variables in all vertices of a given contactor graph (sec. 5.1). The aforementioned list of variables is not final, for instance, future modification of a reduced model could call for a multi-pressure or a multi-temperature model.

From a graph view point, transport balances can be taken, in a generic programing software sense, as *algorithms* applied to the contactor graph data. In general, the basic conservation and balance equations of continuum thermomechanics are applied to the stages of a contactor, producing a network-based system of equations coupled by the material flow represented by the edges of the network graph. This approach does resemble the numerical discretizations of partial differential equations on a volume mesh, cells or control volumes.

5.2.1 Mass balance

A vertex-wise mass balance on the kth vertex with incoming material flow from any of its neighbors ℓ results in:

$$\boxed{\begin{aligned} d_t\big(\rho^{(k)}_{a,\alpha}\; V^{(k)}_\alpha\big) = -\rho^{(k)}_{a,\alpha}\big(f^{(k)}_\alpha + g^{(k)}_{a,\alpha}\big) + \sum_{\ell\neq k}\rho^{(\ell)}_{a,\alpha}\big(f^{(\ell)}_\alpha - g^{(\ell)}_{a,\alpha}\big) + \mathscr{M}^{(k)}_{a,\alpha} \\ \forall \quad k=1,\ldots,N_{\text{v}}, \quad \alpha=1,\ldots,A^{(k)}, \quad \text{and} \quad a=1,\ldots,\mathscr{A}^{(k)}_\alpha, \quad (5.3)\end{aligned}}$$

where N_{v} is the total number of vertices (stages), $A^{(k)}$ is the total number of material phases in the kth stage, and $\mathscr{A}^{(k)}_\alpha$, the number of species in the α phase. The source/sink terms, $\mathscr{M}^{(k)}_{a,\alpha}$, couple the equations within a stage, and the convective rates, $f_\alpha^{(\cdot)}$ and $g_{a,\alpha}^{(\cdot)}$, couple the equations across stages according to the connectivity of the vertices in the network. The conservation of mass requires $\sum_\alpha \sum_a \mathscr{M}_{a,\alpha}^{(k)} = 0 \;\forall\; k$.

The source/sink terms arise from reactions within each phase involving the phasic chemical species, and from interfacial interaction between phases, including mass transfer and phase change. In addition, $\mathscr{M}_{a,\alpha}^{(k)}$, absorbs covariance terms resulting from volume averaging while expressing integrals of products as the product of integrals. The convective exchange of species mass is accounted in two terms, namely, the phasic flow rate, $f_\alpha^{(\cdot)} \geq 0$, and the dispersion for each phasic specie, $g_{a,\alpha}^{(\cdot)}$, respectively; the latter takes into account the chemical transport effects of entrainment which causes carryover; also known as backmixing or frontmixing. These mass convection terms are rooted on the surface averaging of the volume-averaged sub-scale governing equations expressed as a stage volume average. This double average gives rise to additional covariance terms involving area and volume integrals; these quantities are included into the source/sink $\mathscr{M}_{a,\alpha}^{(k)}$. For cases where either diffusion or mass transfer of species takes place through a closed contact with the environment, the dispersion terms $g_{a,\alpha}^{(\cdot)}$ also accumulate this added effect. These terms typically involve the absorption of a boundary condition of mass transfer in the sub-scale averaged continuum.

The source/sink terms can be of considerable complexity depending on the system under study and the level of fidelity. Development of both mass fluxes and source terms from the underlying continuum theory allows for sub-scale models to be coupled to the network model. For instance, backmixing effects are often introduced in mass balances of plant-level modules without theoretical support; here, an underlying dispersion model for $g_{a,\alpha}^{(\cdot)}$ will better quantify the phenomenon and guide corresponding experimental measurements. In general, it can be argued that the rigorous theoretical derivation of the source/sink terms will inevitably require a review of experimental data and methods to obtain these data, including new requirements for novel experiments guided by new models.

There are $2\,A^{(k)}{+}3\,\sum_{\alpha=1}^{A^{(k)}} \mathscr{A}_\alpha^{(k)}$ time-dependent variables per stage and only $\sum_{\alpha=1}^{A^{(k)}} \mathscr{A}_\alpha^{(k)}$ equations (5.3). The closure of the system is the subject of a constitutive theory (sec. 5.3).

5.2.2 Power balance

When continuum linear and angular momentum balances are developed for the network vertices, multiple reduced working equations can be obtained. At the moment, a first-order consideration of mechanical effects can be obtained by calculating the rate of change of the kinetic energy as derived from the momentum balances. Existing plant-level models rarely employ any statement derived from a momentum balance.

Here, for each material phase α a power balance follows for each stage k

$$\boxed{\begin{aligned} d_t\big(\rho_\alpha^{(k)}\tfrac{1}{2}v_\alpha^{2(k)}\,V_\alpha^{(k)}\big) = -\big(\rho_\alpha^{(k)}\tfrac{1}{2}v_\alpha^{2(k)} + t_\alpha^{(k)}\big)f_\alpha^{(k)} + \sum_{\ell\neq k}\big(\rho_\alpha^{(\ell)}\tfrac{1}{2}v_\alpha^{2(\ell)} + t_\alpha^{(\ell)}\big)f_\alpha^{(\ell)} + \mathscr{P}_\alpha^{(k)} \\ \forall \quad k = 1,\ldots,N_{\text{v}}, \quad \text{and} \quad \alpha = 1,\ldots,A^{(k)}. \quad (5.4) \end{aligned}}$$

The source terms, $\mathscr{P}_\alpha^{(k)}$, represent sinks and sources of power and they have two parts. One represents the external power expended on the phases, and another part incorporates the internal power expended within the phases (*e.g.*, viscous dissipation and interfacial interactions through mass transfer and phase change). Also included in the source terms are: covariance variables from space averaging and, potentially, time derivatives of additional variables. The flux terms containing the phasic volumetric flow rates in (5.4) are similar in form to the mass balance equation; here kinetic and traction power are either added to or removed from the system by convection.

The source terms in the power balance provide a necessary coupling quantity in the model to consider sub-scale effects of mixing power on systems that involve intense rotation or other flows without moving parts but in turbulent regime. The traction power contribution allows for mechanistic effects (say drag) associated to carryover between stages be included from sub-scale modeling or experimentally measured data.

There may be a need for the consideration of a multi-pressure model (*e.g.* in absorption or adsorption applications where interfacial pressures can be of significance). This can be accommodated in the internal power part of the power sources or/and in the traction power; all these quantities are phase dependent. Finally, power balance and mass balance equations on a given network vertex (stage) are coupled not only through the phase volumes and volumetric flow rates but through both bulk mass sources terms and interfacial mass transfer terms. The kinetic energy of chemical species within each phase is not necessarily neglected in this presentation (note absence of sub-index $\boldsymbol{a}$ in (5.4)). This can be introduced as a sub-model within the kinetic energy balance on a needed basis.

In summary, the form of $\mathscr{P}_\alpha^{(k)}$ will be complex for any given realistic system but it provides a consistent means for building and improving scientific models systematically.

There are three new time-dependent variables added per phase in any stage equation (5.4). This brings the total number of variables to $5\,A^{(k)} + 3\,\sum_{\alpha=1}^{A^{(k)}}\,\mathscr{A}_\alpha^{(k)}$ per stage and only $A^{(k)} + \sum_{\alpha=1}^{A^{(k)}}\,\mathscr{A}_\alpha^{(k)}$ equations.

5.2.3 Energy balance

Energy balance for multiphase, multicomponent continuum systems is often controversial, starting with the possibility of multi-phasic, multi-component temperatures and ending with the many possible alternative versions of the same energy balance statement. This is a preliminary discussion aimed at developing a generic mathematical framework. In a way similar to the previous balance equations, an energy balance can be stated for each phase in each stage

$$\boxed{\begin{aligned} d_t\big(\rho_\alpha^{(k)}\varepsilon_\alpha^{(k)}\,V_\alpha^{(k)}\big) = -\rho_\alpha^{(k)}\varepsilon_\alpha^{(k)}\,f_\alpha^{(k)} + \sum_{\ell\neq k}\rho_\alpha^{(\ell)}\varepsilon_\alpha^{(\ell)}\,f_\alpha^{(\ell)} + \sum_{\ell} q_\alpha^{(\ell)} + \mathscr{E}_\alpha^{(k)} \\ \forall \quad k = 1,\ldots,N_\text{v}, \quad \text{and} \quad \alpha = 1,\ldots,A^{(k)}, \quad (5.5) \end{aligned}}$$

where the internal source/sink terms, $\mathscr{E}_\alpha^{(k)}$, include surface interactions by virtue of interfacial energy flux on each phase, mass transfer, and phase change. Also, bulk heat generation on each phase and mechanical dissipation by phase motion are included in the source terms. They are coupled, in particular, with terms present in the power balance sources. It holds true the same comment made for the power balance (5.4) regarding the account of the internal energy of chemical species. Finally, the phasic heat power provided for, or removed from, the stage by closed contact with the exterior is represented in the sum of $q_\alpha^{(\ell)}$ with the convention that external net heat power added to the stage is positive, otherwise, negative. The contact heat rates are typical products from averaging of boundary conditions present in the sub-scale averaged continuum, where radiative, convective cooling, diffusive heating, etc., are incorporated by means of a transport coefficient.

With the introduction of the energy balance, three additional time-dependent variables were created for each phasic equation, therefore $8\,A^{(k)} + 3\,\sum_{\alpha=1}^{A^{(k)}} \mathscr{A}_\alpha^{(k)}$ variables are present in any given stage and only $2\,A^{(k)} + \sum_{\alpha=1}^{A^{(k)}} \mathscr{A}_\alpha^{(k)}$ equations are available.

5.2.4 Entropy imbalance

A statement of entropy imbalance is used to assist the consistent derivation of constitutive material relations needed for the development of explicit functions for the time-dependent variables. Here, an assumption that an entropy imbalance statement holds for each phase leads to

$$\boxed{\begin{aligned} d_t\big(\rho_\alpha^{(k)}s_\alpha^{(k)}\,V_\alpha^{(k)}\big) \geq -\rho_\alpha^{(k)}s_\alpha^{(k)}\,f_\alpha^{(k)} + \sum_{\ell\neq k}\rho_\alpha^{(\ell)}s_\alpha^{(\ell)}\,f_\alpha^{(\ell)} + \sum_{\ell}\frac{q_\alpha^{(\ell)}}{\theta_\alpha^{(\ell)}} + \mathscr{S}_\alpha^{(k)} \\ \forall \quad k = 1,\ldots,N_\text{v}, \quad \text{and} \quad \alpha = 1,\ldots,A^{(k)}, \quad (5.6) \end{aligned}}$$

where the source/sink terms $\mathscr{S}_\alpha^{(k)}$ consist of volume contributions: bulk heat and mass sources; and surface contributions: interfacial entropy sources from mass transfer, interfacial interactions, and interface heat fluxes. Similarly to the energy balance

(5.5), $\mathscr{S}_\alpha^{(k)}$ can be positive or negative, but in contrast with (5.5), the entropy balance entails an additional hypothesis which makes it an imbalance. To clarify, consider the case of an isolated stage, thus (5.6) sets the rate of change of the stage entropy as greater than or equal to the internal source/sink of entropy. In contrast, the energy balance for the same isolated stage, sets the rate of internal energy change exactly equal to the internal source/sink of energy.

It is tacitly assumed here that the entropy balance can be applied to each individual material phase. This is not the case for individual chemical species within each phase unless a very specific ideal material behavior is desired; the entropy imbalance applies to the whole mixture of chemical species to be consistent with classical thermochemistry (Bowen, 1976). Therefore, this is in contrast with how chemical species are considered in the energy balance.

The positive phasic temperature of the stage is an additional hypothesis of the entropy imbalance for the specification of the external entropy flux in the stage. Multiple phase temperatures are allowed for flexibility of the development of the sub-scale models.

The entropy imbalance introduces two new time-dependent phasic variables and one inequality per phase. The final number of variables is $11\,A^{(k)} + 3\,\sum_{\alpha=1}^{A^{(k)}} \mathscr{A}_\alpha^{(k)}$ for the same number of equations as before, *i.e.*, $2\,A^{(k)} + \sum_{\alpha=1}^{A^{(k)}} \mathscr{A}_\alpha^{(k)}$.

5.2.5 Generalized balance

The basic balances and imbalance share a common form worth exploring for economy of presentation and for implementing solution procedures based on numerical methods. In addition, as mentioned before, additional variables may be added to this mathematical framework, say, by the introduction of electromagnetic phenomena. Therefore, a generalized form can help expedite the derivation of a new balance postulate. If Γ_α is an intrinsic phasic scalar quantity then a general balance can be stated as

$$\boxed{\begin{aligned} d_t\big(\rho_\alpha^{(k)}\Gamma_\alpha^{(k)}\;V_\alpha^{(k)}\big) = -\rho_\alpha^{(k)}\Gamma_\alpha^{(k)}\,f_\alpha^{(k)} + \sum_{\ell\neq k}\rho_\alpha^{(\ell)}\Gamma_\alpha^{(\ell)}\,f_\alpha^{(\ell)} + \sum_{\ell}\gamma_\alpha^{(\ell)} + \mathscr{G}_\alpha^{(k)}& \\ \forall\quad k=1,\ldots,N_\mathrm{v},\quad \text{and}\quad \alpha=1,\ldots,A_\alpha^{(k)},\quad (5.7)& \end{aligned}}$$

where $\mathscr{G}_\alpha^{(k)}$ is the associated internal source/sink rate of the phasic Γ_α quantity. The $\gamma_\alpha^{(\ell)}$ term is the net rate of gain/loss of Γ_α by means of closed contact with the external environment. Last, the inflow/outflow terms quantify the open interaction with the environment.

Therefore the balance (5.7) states that the rate of change of the intrinsic phase quantity Γ_α in the kth stage is equal to the rate of inflow minus the outflow, plus the gain/loss of Γ_α by external contact, plus the source/sink rate of internal production.

There exists great complexity in $\mathscr{G}_\alpha^{(k)}$ associated with the discontinuity of the α phase and its interaction with the other phases. The complexity is augmented when specific behavior of matter is taken into account within a constitutive theory. A mathematical ramification of such difficulties is the fact the $\mathscr{G}_\alpha^{(k)}$ terms can include higher-order time derivatives of other coupled variables.

5.3 Constitutive restrictions

From the previous sections, it is observed that an additional $9\,A^{(k)} + 2\,\sum_{\alpha=1}^{A^{(k)}} \mathscr{A}_\alpha^{(k)}$ equations are needed to balance the number of variables and equations, that is, to bring the system of equations and variables into closure. Therefore the system of basic equations is underdetermined, hence not solvable. Solvability can be obtained by proposing new functional relations or constitutive equations among some of the dependent variables until closure is obtained. This is the practical purpose of a constitutive theory. That is, to guide the selection of closure equations so the final system is compatible with all governing equations (5.3)–(5.6).

A more fundamental role of constitutive equations is stated as follows (Truesdell and Noll, 1965; Bowen, 1976; Dobran, 1991; Gurtin et al., 2010). The balance equations (5.3)–(5.6) originate from postulates of basic principles for any network system. The behavior of particular materials or phases within the system is not accounted by the theory until constitutive equations are established; this is the fundamental role of the closure mechanism. A constitutive theory specializes and augments the set of basic governing equations to apply to particular classes of materials.

The kth stage thermomechanical variables involved in the transport balances (5.3)–(5.6), define a *chemical thermomechanical process* represented by $A^{(k)}$ lists of length $11 + 3\,\mathscr{A}_\alpha^{(k)}$ of functions $(\cdot)\colon I \to \mathbb{R}$

$$\begin{aligned}\mathcal{P}_\alpha^{(k)} := \big(&\rho_{a,\alpha}^{(k)},\ V_\alpha^{(k)},\ f_\alpha^{(k)},\ g_{a,\alpha}^{(k)},\ \tfrac{1}{2}v_\alpha^{2(k)},\ t_\alpha^{(k)},\ \varepsilon_\alpha^{(k)},\ q_\alpha^{(k)},\ s_\alpha^{(k)},\ \theta_\alpha^{(k)},\\ &\mathscr{M}_{a,\alpha}^{(k)},\ \mathscr{P}_\alpha^{(k)},\ \mathscr{E}_\alpha^{(k)},\ \mathscr{S}_\alpha^{(k)}\big)\quad \forall\ \alpha = 1,\dots,A^{(k)},\ a = 1,\dots,\mathscr{A}_\alpha^{(k)}.\end{aligned} \tag{5.8}$$

The inter-stage coupling terms associated to the governing equations are omitted from the above lists since they are considered known quantities from a constitutive viewpoint. Their role is similar to a body force term which is typically provided from external information. Henceforth, the kth superscript will be dropped for the sake of notational simplicity. There should be no confusion with the resulting notation since constitutive equations are defined on a stage basis.

A vast number of processes $\mathcal{P}_\alpha$ result from the corresponding thermomechanical functions compatible with the governing balance equations; this is a consequence of an underdetermined mathematical problem. Many of these processes are not physically meaningful, that is, are not realizable for realistic materials. Additional restrictions are needed for the lists (5.8) before they can represent an acceptable material behavior. To this end, a set of independent variables is assumed,

$$\big\{\rho_{\mathscr{a},\alpha}, V_\alpha, \theta_\alpha\big\} \quad \forall\ \alpha = 1,\ldots,A,\ \mathscr{a} = 1,\ldots,\mathscr{A}_\alpha, \tag{5.9}$$

which defines a thermokinetic process in the time interval I. To determine $\mathcal{P}_\alpha$ it is sufficient to provide constitutive relations (or restrictions, or equations) for $8\,A + 2\,\sum_{\alpha=1}^{A}\,\mathscr{A}_\alpha$ constitutive variables

$$\Upsilon_\alpha := \big(f_\alpha,\ g_{\mathscr{a},\alpha},\ t_\alpha,\ \varepsilon_\alpha,\ q_\alpha,\ s_\alpha,\ \mathscr{M}_{\mathscr{a},\alpha},\ \mathscr{P}_\alpha,\ \mathscr{E}_\alpha,\ \mathscr{S}_\alpha\big) \\ \forall\ \alpha = 1,\ldots,A,\ \mathscr{a} = 1,\ldots,\mathscr{A}_\alpha, \tag{5.10}$$

and determine the remaining $3\,A + \sum_{\alpha=1}^{A}\,\mathscr{A}_\alpha$ variables, $\rho_{\mathscr{a},\alpha}, V_\alpha, \theta_\alpha, \frac{1}{2}v_\alpha^{2(k)}$, from (5.3)–(5.5) (the entropy inequality does not participate in pairing equations and variables).

The assumption that the history of the thermokinetic process (5.9) determines $\mathcal{P}_\alpha$ is a statement of the principle of determinism which applied to (5.10) may be expressed as follows:

$$\Upsilon_\alpha(t) = \mathscr{F}_\alpha\big[\rho^t_{\mathscr{b},1}(s),\ldots,\rho^t_{\mathscr{b},A}(s),\, V^t_1(s),\ldots,\mathrm{V}^t_A(s),\, \theta^t_1(s),\ldots,\theta^t_A(s),\, t\big] \\ \forall\ \alpha = 1,\ldots,A,\ \mathscr{b} = 1,\ldots,\mathscr{A}_\alpha, \tag{5.11}$$

where $\mathscr{F}_\alpha[\cdot]$ is a response functional depending on the history functions

$$\rho^t_{\mathscr{b},\beta}(s) := \rho^t_{\mathscr{b},\beta}(t-s), \qquad \mathrm{V}^t_\beta(s) := \mathrm{V}^t_\beta(t-s), \quad \text{and} \qquad \theta^t_\beta(s) := \theta^t_\beta(t-s) \tag{5.12}$$

for all $s \geq 0$ such that these functions represent any possible thermokinetic history of each phase in the kth stage. The value of any of these functions at time s is the past value relative to the present time t. The notation in (5.11) indicates that the constitutive variables for one phase are dependent on the thermokinetic variables of all phases. The initial form of $\mathscr{F}_\alpha[\cdot]$ effectively prevents future events from determining the past. However, (5.11) does not determine the past from the present state because of the need to incorporate the irreversibility of macroscopic processes. In this network theory, irreversibility is inherent in the entropy imbalance (5.6). An *admissible* chemical thermomechanical process, is determined by the balance equations (5.3)–(5.5), the constitutive restrictions (5.11), and the entropy imbalance (5.6). The study of the functional $\mathscr{F}_\alpha[\cdot]$ is the subject of a constitutive theory, and the derivation of explicit forms of this functional is the subject of a constitutive modeling effort for various classes of materials.

A constitutive theory rests on various concepts and principles. The principle of determinism has been used to arrive at (5.11). Another useful principle is the local, smooth memory principle which asserts that the thermokinetic processes at a distant past do not significantly affect the present value of the response functional $\mathscr{F}_\alpha[\cdot]$. Mathematically, this entails to approximating (5.12) by their Taylor series expansions in the time interval $|t-s| \leq \delta$, *i.e.*,

$$\begin{aligned} \rho^t_{\mathscr{b},\beta}(s) &\approx \rho_{\mathscr{b},\beta}(t) + (t-s)\, d_t\rho_{\mathscr{b},\beta}, \\ V^t_\beta(s) &\approx V_\beta(t) + (t-s)\, d_t V_\beta, \\ \theta^t_\beta(s) &\approx \theta_\beta(t) + (t-s)\, d_t\theta_\beta. \end{aligned}$$

Upon substitution in (5.11), a response function for smooth, short (local) memory material is obtained

$$\Upsilon_\alpha(t) = \mathscr{F}_\alpha\big(\rho_{\mathscr{b},\beta}(t),\, d_t\rho_{\mathscr{b},\beta},\, V_\beta(t),\, d_tV_\beta,\, \theta_\beta(t),\, d_t\theta_\beta,\, t\big) \qquad \forall\ \alpha = 1,\ldots,A,\ \ \mathscr{b} = 1,\ldots,\mathscr{A}_\alpha, \tag{5.13}$$

where in this notation, β represents the dependency of any phasic response function upon all phases. Another constitutive principle related to the material frame-indiference principle asserts that under a shift of time (5.13) does not depend explicitly on time, hence

$$\boxed{\Upsilon_\alpha(t) = \mathscr{F}_\alpha\big(\rho_{\mathscr{b},\beta}(t),\, d_t\rho_{\mathscr{b},\beta},\, V_\beta(t),\, d_tV_\beta,\, \theta_\beta(t),\, d_t\theta_\beta\big) \qquad \forall\ \alpha = 1,\ldots,A,\ \ \mathscr{b} = 1,\ldots,\mathscr{A}_\alpha. \quad (5.14)}$$

As stated earlier, $9\,A^{(k)} + 2\,\sum_{\alpha=1}^{A^{(k)}} \mathscr{A}^{(k)}_\alpha$ response functions represented by (5.14) are needed to specify the material behavior of the system. The particular form of $\mathscr{F}_\alpha(\cdot)$ is the subject of further modeling of the particular material behavior in the system in conjunction with the entropy imbalance. The restrictions that follow from (5.6) on (5.14) are not obvious and require a detailed study for particular classes of materials. This undertaking is only practical on a case-by-case basis for each plant-level component.

5.3.1 Internal constraints

The inclusion of internal constraints in multiphase, multicomponent mixture theory is an unsolved problem (Dobran, 1991). Therefore, this difficulty is inherited at the plant-level modeling scale. Individual internal constraints can be treated following the principle that internal constraints should produce no work and no entropy. This task adds complexity to the derivation of a consistent transport theory, and requires explicit models for the source/sink terms $\mathscr{M}_{a,\alpha}$, $\mathscr{P}_{\alpha}$, $\mathscr{E}_{\alpha}$, and $\mathscr{S}_{\alpha}$.

Internal constraints are *a priori* constitutive relations with origins in the underlying fluid/solid mechanics, thermochemistry, and chemical kinetics. For instance, material incompressibility, solution (non)ideality, and constrained reaction mechanisms are a few common examples of internal constraints. Internal constraints require a revision of previously used constitutive principles for consistency of the theory.

5.4 Differential-algebraic equations

Previous sections have outlined a system of governing equations for a theory of network material transport. Transport balances (sec. 5.2) and constitutive response functions (sec. 5.3) are postulated for the vertices of a network representing a contact operation (sec. 5.1). The assembly of all stage-wise equations in the network can be made in a similar fashion to the discretization of continuous systems (fig. 5.5). In general, this assembly leads naturally to a system of nonlinear, differential-algebraic equations (DAE), summarized next.

The state vector for a contactor network $\boldsymbol{x}(t)$ collects the values of a subset of the thermomechanical variables (5.8) at time t for all stages in the vertex set V (5.1), say $\boldsymbol{x}(t) \in \mathbb{R}^m$. The general form of the vector-valued function describing the nonlinear set of implicit, algebraically constrained, ordinary differential equations can be expressed as

$$\boldsymbol{F}\big(\boldsymbol{x}(t), \ldots, d_t^n \boldsymbol{x}, t; \boldsymbol{p}(t), \boldsymbol{q}(t)\big) = \boldsymbol{0}, \tag{5.15a}$$

$$\boldsymbol{G}\big(\boldsymbol{x}(t), t; \boldsymbol{p}(t), \boldsymbol{q}(t)\big) = \boldsymbol{0}. \tag{5.15b}$$

These equations emerge from the basic network transport balances (5.3)–(5.5), the constitutive response functions (5.14), and any existing internal constraints (sec. 5.3.1). The parameter vector $\boldsymbol{p}(t) \in \mathbb{R}^k$ and the coupling vector $\boldsymbol{q}(t) \in \mathbb{R}^l$ have been discussed at length (sec. 4.2 and 4.3), they represent additional time-dependent input data for the model. For a module M_i at time t_i, $\boldsymbol{p}(t)$ stands for the parameter vector of the module, $\boldsymbol{p}_i(t_i)$, and $\boldsymbol{q}(t)$ stands for the coupling data, $\boldsymbol{q}_{\overrightarrow{j,i}}(t_i)$, requested by M_i from all connected neighbors M_j modules ($j \neq i$). The initial conditions at the initial time t_0, that is, $\boldsymbol{x}(t_0) = \boldsymbol{x}_0, \ldots, d_{t_0}^n \boldsymbol{x} = \boldsymbol{x}_0^{(n)}$, $\boldsymbol{p}(t_0) = \boldsymbol{p}_0$, and $\boldsymbol{q}(t_0) = \boldsymbol{q}_0$ complete the problem statement.

The mathematical expression of algebraic constraints is denoted by the function $\boldsymbol{G}\colon \big(\boldsymbol{x}(t), t; \boldsymbol{p}(t), \boldsymbol{q}(t)\big) \in \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^k \times \mathbb{R}^l \to \mathbb{R}^c$ with c constraint equations. Equilibrium thermodynamic constraints involve no rates of any of the dependent variables $\boldsymbol{x}$

therefore these constraints belong in $\boldsymbol{G}(\cdot) = \boldsymbol{0}$ which is typically a nonlinear (possibly linear) algebraic relation between the unknowns.

The equations containing rates of the unknowns are collected in the function $\boldsymbol{F}\colon \big(\boldsymbol{x}(t), \ldots, d_t^n\boldsymbol{x}, t; \boldsymbol{p}(t), \boldsymbol{q}(t)\big) \in \mathbb{R}^m \times \ldots \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^k \times \mathbb{R}^l \to \mathbb{R}^{m-c}$. Although it is unlikely for high-order time derivatives to occur, provisions can be made to accommodate this model at least at the DAE setting. In practice, the most common models will use n equal to one in the time derivative $d_t^n\boldsymbol{x}$.

All source/sink terms $\mathscr{M}_{\mathscr{a},\alpha}(t), \mathscr{P}_\alpha(t), \mathscr{E}_\alpha(t)$, and $\mathscr{S}_\alpha(t)$ are basically a composition of rates of change. In a variety of separation processes chemical non-equilibrium considerations are a necessity. Therefore, rates of change of species mass concentration are specified by a chemical reaction mechanism often modeled as a rate law of varying order. This is accommodated by a constitutive relation of the form (5.14), say

$$\mathscr{M}_{\mathscr{a},\alpha}(t) = \widehat{\mathscr{M}_{\mathscr{a},\alpha}}\big(\rho_{\mathscr{b},\beta}(t),\, d_t\rho_{\mathscr{b},\beta},\, V_\beta(t),\, d_tV_\beta,\, \theta_\beta(t),\, d_t\theta_\beta\big),$$

where the rate of supply of the $\mathscr{a}$th specie in the α-phase depends, potentially, on the thermokinetic variables of all phases and species $\beta = 1, \ldots, A,\ \mathscr{b} = 1, \ldots, \mathscr{A}_\beta$.

While the DAE (5.15) can be reduced to ordinary differential equations by differentiation of the constraints and change of variables, this is not desirable since the procedure may lack generality and may defeat the search for a common approach for all (or most) plant-level separation processes; in addition, the physical interpretation of the equations may be lost. Hence, time-advancing the equations in the natural form is a preferred precondition for enabling exploration of commonality at the mathematical level. However, this is not without difficulties. Methods for solving differential-algebraic equations require consistent initial conditions for the derivatives of the vector of unknowns (Brenan et al., 1996). This information is often not available but computable from an available initial state vector. Therefore, a necessary internal feature of a module is the capability to handle initial conditions robustly without posing unreasonable external requirements. *If this is done successfully, plant-level separation modules will share the same DAE solver libraries; a savings of enormous efforts.* For instance, existing codes for separation processes may be up to 80% devoted to programing algorithms for solving the resulting set of specialized equations. These algorithms are typically not reusable outside the codes they were built in. Last, but not least, existing robust DAE solvers provide sensitivity analysis capability; therefore, an additional savings of extra work is obtained that will be shared by all separation modules and will enable initial aspects of uncertainty quantification to be readily available.

6 Final comments

This report elaborates on the common foundation needed for the development of an advanced modeling and simulation toolkit for used nuclear fuel recycling technology. Significant physicochemical processes are realized as software modules in the toolkit, while a particular composition of modules results in a software application. This composition can be made systematic by the definition of a common, external behavior of a module, and the use of an environment for time-coupling the modules. This environment is best implemented as a network of modules (sec. 4).

A module in the toolkit is the software realization of a model of a particular separation process. A theory for network transport has been advanced for exploring modeling commonality for all physicochemical processes. The basis of this theory is the up-scaling of a continuum chemical thermomechanics model in conjunction with a network model of any contacting operation (sec. 5). Therefore this modeling paradigm requires the consideration of a model at the continuum level (fig. 6.1) followed by a space-time average on a network of stages to derive the plant-level model; a reduced model. A by-product of this approach is the effective implementation of a the hierarchical, multi-tier strategy for incorporating sub-scale phenomena (sec. 1).

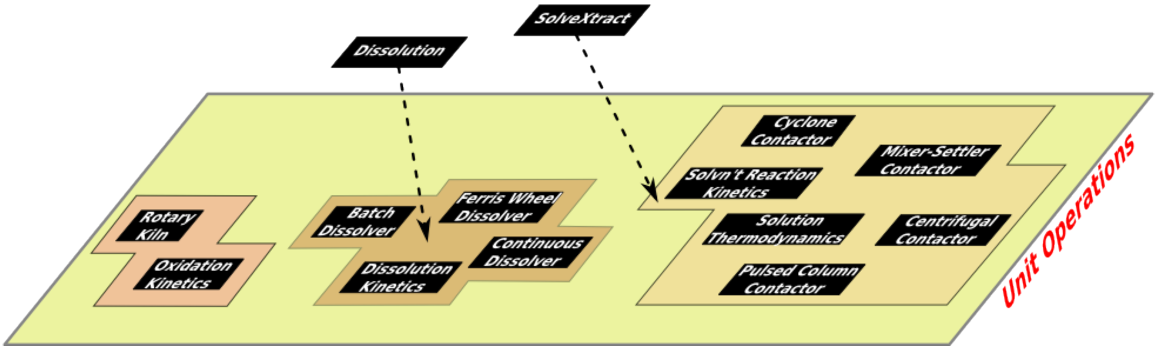

Fig. 6.1: *Plant-level modules are reduced representations of higher fidelity modules at lower tiers. This paradigm for creating lower-order models promotes extensibility and improves calibration of the module.*

The modeling commonality allows for significant reuse of the corresponding software components, as mentioned in previous sections. For instance, a network description of a contactor equipment is a reusable component, so is the DAE library for solving the resulting system of equations.

Application of the concepts developed here to processes representative of the PUREX (plutonium uranium extractation) reprocessing method will be the subject of future communications (de Almeida, 2011).

A Appendix: Modeling questionnaire

The following list of questions were created to assist the development of models and simulation software. At the minimum, the exercise of answering these questions help guiding the development process and communicate the modeling work among a group of collaborators. Full answers to these questions prove to be a lenthy exercise that can be used as a template for reporting on the progress of developing a module. This is a proposed format to be followed in future communications (de Almeida, 2011).

1. **Plant-Level Localization.** What plant-level module does the model relate to? If a corresponding module has not been listed in the plant-level tier, propose where it should be placed starting with the sub-system.

2. **Physicochemical Phenomena.** What physicochemical phenomena the model is trying to capture?

3. **System Description.** Describe the system including the principal length and time scales, and how it couples to the outside.

4. **Mathematical Model.** Describe the mathematical model including the simplifying assumptions. For instance, if the model is a partial differential equation set, what are the independent variables, dependent variables, domain, boundary, boundary conditions, etc. as fully as possible.

5. **Data.** What data is used and what data is provided?

6. **Solution Method.** What is the solution method of the mathematical problem?

7. **Verification Method.** What is the verification method of the solution of the mathematical problem?

8. **Uncertainty.** How is the uncertainty of the solution assessed?

9. **Fidelity Improvement.** How is the model fidelity improved? What is needed and why does the model require higher fidelity?

10. **Model Transferability.** Is the model transferable? If so, what other systems is it applicable to? Apply above questions 1 to 10 recursively.

References

Manson Benedict, Thomas H. Pigford, and Hans Wolfgang Levi. *Nuclear Chemical Engineering*. McGraw-Hill, Inc., New York, New York, 1981. first priting in 1957.

J. A. Bondy and U. S. R. Murty. *Graph Theory*, volume 244 of *Graduate Texts in Mathematics*. Springer, New York, New York, 2008.

Ray M. Bowen. Theory of mixtures. In A. Cemal Eringen, editor, *Continuum Physics*, part 1, pages 1–127. Academic Press, New York, New York, 1976.

K. E. Brenan, S. L. Campbel, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, volume 14 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1996.

Valmor F. de Almeida. Progress on plant-level components for nuclear fuel recycling: Dissolver. Letter Report ORNL/LTR-2011-279, Oak Ridge National Laboratory, Tennessee, TN 37831-6181, U.S.A., August 2011. In preparation.

Valmor F. de Almeida, Benjamin P. Hay, and David W. DePaoli. Status of safeguards and separations model develpment at plant and molecular levels. Letter Report ORNL/TM-2009-253, Oak Ridge National Laboratory, Tennessee, TN 37831-6181, U.S.A., October 2009. Also available by request to `dealmeidav@ornl.gov`.

Valmor F. de Almeida, Jay J. Billings, and David W. DePaoli. Safeguards and separations modeling and simulation—plant-level architecture development. Letter Report ORNL/TM-2010-229, Oak Ridge National Laboratory, Tennessee, TN 37831-6181, U.S.A., September 2010a. Also available by request to `dealmeidav@ornl.gov`.

Valmor F. de Almeida, Benjamin P. Hay, and David W. DePaoli. Safeguards and separations modeling and simulation—paralelized version of hostdesigner. Letter Report ORNL/TM-2010-94, Oak Ridge National Laboratory, Tennessee, TN 37831-6181, U.S.A., April 2010b. Also available by request to `dealmeidav@ornl.gov`.

Valmor F. de Almeida, Alex McCaskey, and Jay J. Billings. Safeguards and separations modeling and simulation: Reprocessing toolkit development. Letter Report ORNL/LTR-2011-115, Oak Ridge National Laboratory, Tennessee, TN 37831-6181, U.S.A., April 2011. Also available by request to `dealmeidav@ornl.gov`.

Flavio Dobran. *Theory of Structured Multiphase Mixtures*, volume 372 of *Lecture Notes in Physics*. Springer-Verlag, New York, New York, first edition, 1991.

Richard M. Felder and Ronald W. Rousseau. *Elementary Principles of Chemical Processes*. John Wiley & Sons, New York, New York, second edition, 1986.

W. S. Gronier. Technical manual for SEPHIS mod4, version 2.11. Technical Manuscript ORNL/TM-11589, Oak Ridge National Laboratory, Tennessee, TN 37831-6181, U.S.A., February 1991.

Morton E. Gurtin, Eliot Fried, and Lallit Anand. *The Mechanics and Thermodynamics of Continua*. Cambridge University Press, New York, New York, first edition, 2010.

Dieter Jungnickel. *Graphs, Networks and Algorithms*, volume 5 of *Algorithms and Computation in Mathematics*. Springer, New York, New York, third edition, 2008.

Justin T. Long. *Engineering for Nuclear Fuel Reprocessing*. Gordon and Breach Science Publishers Inc., New York, New York, 1967.

Walter Noll. *Finite-Dimensional Spaces: Algebra, Geometry and Analysis*, volume 1. Martinus Nijhoff Publishers, AD Dordrecht, The Netherlands, 1987.

C. Truesdell and W. Noll. The non-linear field theories of mechanics. In S. Flügge, editor, *The Non-Linear Field Theories of Mechanics*, volume III/3 of *Encyclopedia of Physics*, pages 1–602. Springer-Verlag, New York, New York, 1965.

C. Truesdell and R. Toupin. The classical field theories. In S. Flügge, editor, *Principles of Classical Mechanics and Field Theory*, volume III/1 of *Encyclopedia of Physics*, pages 226–858. Springer-Verlag, New York, New York, 1960.