SHORT NOTE

Experiences and Challenges in Development of Sustainable Modelling and Simulation Tools

K. Juslin, VTT Technical Research Centre of Finland, *kaj.juslin@vtt.fi*

SNE Simulation Notes Europe SNE 20(1), 2010, 27-30, doi: 10.11128/sne.20.sn.09965

Modelling and simulation software tools should be sustainable even if it is not definitely required for temporary research projects with a publication as outcome. Especially, when using modelling in simulation as an integrated tool supporting both engineering and operation of industrial processes—the models should sustain for the whole life cycle of the target process—from initial design to decommission, at least 60 years. The solution relates, first, to apply such formal specifications of the physical model that easily can be carried as simple text files between generations of computer and operation system platforms. Second, reliable simulation engines are needed that are programmed in high level language enabling for transportation to the new platforms and. Third, standardised procedures for re-verification of the computational functionality are required. Fourth, easy connectivity to concurrently developing engineering databases and graphics interfaces is needed, as well. The advance of multi core processors, even on laptop computers, calls for reconsideration of the mathematical methods used so far in order to make use of the inherent computing power. Companion model graphs are supposed to have a significant mission in automated computer model generation from semantic plant specifications, as well as in even distributing to computation load on available cores.

1 Specification of simulation models with interconnects to engineering tools

My experiences relate to the Apros simulation environment developed more than 20 years ago [1]. Some models then developed are still in use! First computer platform was VAX/VMS, then several UNIX workstations, and now Windows based PCs. Now the third generation of graphics modelling and simulation interface is in use. Of course, the simulation engine has developed all the time, new functionalities and model libraries have been included, but keeping intact the formal model specifications [2]. The Apros Specification Language (ASL) supports introduction of new object types made up from predicates, subjects and required number of attributes. During application modelling these objects are instantiated by naming the subjects and assigning values to the attributes. The attribute values may also include references to other objects and attributes. The predicates include concepts for adding and deleting structures and connections in the model as well as for running the model. Apros is supplied with a Communication Library (ACL) enabling development of connections to third party software. Also, a modern OPC interface is available for standardised connections to e.g. control system software. ASL was developed before XML but the Apros specification triples are easy to wrap to XML formats. Evolving interoperability standards, such as ISO 15926, promote uniform utilisation of branch-specific specifications of equipments, industrial processes and even buildings, as well as of configuration files for automation systems.

Digital catalogues are already available over Internet for various kinds of construction materials and production plant components. New challenges are set for the simulation engines, on the other hand on increased computational speed, much faster than realtime, but also on more detailed description of the processes and on replication of more complicated physical phenomena.

2 Detailed spatial discretisation and distribution of computer power needs

Whence focusing on modelling and simulation of power plants and industrial processes, special attention has to be paid on highly parallel computation of an applicable class of non-linear but piece-vice monotonically continuous differential equations arising from spatial discretisation based on the geometry of the specific industrial processes to control volumes taking into consideration the relevant mechanistic first principles partial differential equations with regard to conservation of momentum, energy and mass substances and the related non-linear empiric material properties.

Until now it has been sufficient to model the material properties of water and steam with sufficient accuracy in conditions encountered in nuclear and conventional power plants. New plants designs cover the super-critical region of the steam tables to increase thermal efficiency. Accordingly the steam tables have been extended. In new designs of clean combustion plants also the production processes of oxygen as well as the flue gas cleaning processes need to be modelled [3]. In new designs of nuclear power plants new coolants need to be modelled, such as sodium or lead in liquid metal cooled reactors as well as helium or carbon dioxide in gas cooled reactors [4]. Especially when boiling of mixtures of several substances need to be considered the additional workload is considerable and problematic for real time simulations on single core processors. Multi-core processors and multi-node clusters may help, subject that the computing loads relevant to the spatial discretisation of the process model can be evenly distributed to the available computing nodes. The variables updated during the iterations needed for the solution at the end of each time-step should be as far as possible kept within relevant node. That means that all material properties needed on a node should be calculated on that node.

An interesting issue is, where and how to calculate the integrated pressure solution to minimize communication over main memory between the computing nodes. Specific descriptive graphs of the simulated process may provide the information needed for automated distribution of the computation tasks [5]. Both iterative and direct methods to solve the integrated pressure flow network should be considered. One possibility is communicate all readily calculated spatially distributed companion model branch values to a dedicated node which could provide for an efficient sparse matrix solution of all the pressure values to be delivered back to the spatial nodes. The other possibility is to apply parallel point Jacobi type of iterative solution of the pressure, whereas the sparse matrix not even would be necessary to solve [6].

3 Direct and iterative sparse solution of companion model based equations

Tearing and lumping procedures require knowledge of the interaction intensity between the variables of heterogeneous real world systems. The dependences f_{ij} between two local variables x_i and x_j and the transition variable w_{ij} are described by the Mechanistic Transition Equation (MTE),

$$f_{ij}\left(x_i, x_j, w_{ij}, \frac{d}{dt} w_{ij}, \int w_{ij} dt, t\right) = 0$$

After suitable spatial discretisation, temporal discretisation, and instant linearization the dependencies can be written as

$$x_i g_{ij} - x_j g_{ji} + s_{ij} - w_{ij} = 0$$

In this Versatile Companion Model (VCM) equation g_{ij} and g_{ji} represent the forward and backward coefficients and s_{ij} the source term. Directed graphs have been considered well suited to describe physical system interactions. The tearing and lumping may allow for simulation of different islands of a homogeneous with different time-steps.

3.1 Graphs may be used to illustrate mechanistic dependencies

Isomorphisms between oriented linear graphs and lumped physical systems were already discussed more than 50 years ago [7]. In Figure 1, a continuously stirred tank with in flows and out flows is shown. The tank is considered as a lumped control volume with the pressure p, specific entalphy of the mixture h and the mass fractions of three included substances c. The horizontal edges denote accumulation of mass, energy and respective concentrations. The edges between the concentration vertices denote chemical reactions. Graphs may be the solution whence developing intelligent modelling tools making use of semantic plant structural specifications and design databases.

3.2 The mechanistic edge and relevant companion model graph

We may consider a specific nonlinear mechanistic dependency f_{ij} between the two nodes x_i and x_j with a simple edge as set forth in Figure 2. The versatile companion model has been developed provide for depicting the instantly, at the end of each iteration step, linearized dependencies. The internal structure

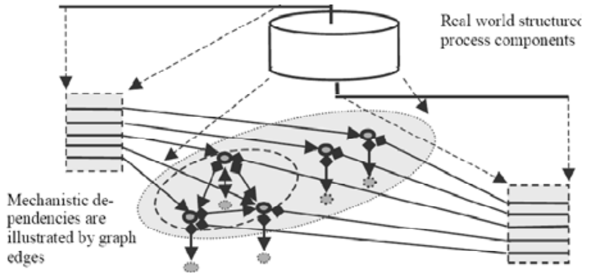

Figure 1. Illustration of real world dependencies by a mechanistic graph.

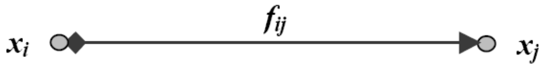

Figure 2. Mechanistic dependency edge.

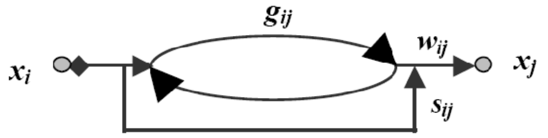

Figure 3. Companion model graph.

of the versatile companion model can be illustrated by three parallel sub-edges representing separate dependencies whereas g_{ij} denotes the forward influence edge, g_{ji} the backward influence edge and s_{ij} the forced source influence edge (Figure 3). The causality of the companion model graph is specified by the relevant coefficients. For instance, the forward and backward coefficients may be different.

3.3 Direct sparse matrix solution

Separate implicit islands are identified from the graph network for the solution of each related set of equations. Whilst no accumulation is considered in a simple vertex we may write for all simple edges that are implicitly connected to node j as follows:

$$\sum_i w_{ij} = 0$$

Applying this scheme for all implicitly interconnected simple vertices i the relevant through variables w_{ij} are eliminated and we get the linear matrix equation for solving the internal local state vector $\underline{x}_e$ for the implicit island:

$$A\,\underline{x}_e = \underline{b}$$

It is to be noted that also parallelised sparse matrix solution schemes are available. Fine grain parallelisation on available cores on the solver node may be used in first case. Ehen the state vector is solved we may proceed by solving the through variables w_e from the relevant companion model equations.

3.4 Iterative sparse matrix solution

If the matrix is less sparse and strongly diagonally dominant, it may be suited for iterative solution schemes such as Point Jacobi that are much easier to parallel than sparse direct methods. Temporary elements b_i and d_i are built by looping through the relevant edges connected to the internal node i, whereas j denotes both internal and external nodes, except for node i. The new iteration step value is denoted by k and the old by $k-1$.

$$\mathrm{d_i} = \textstyle\sum_{\mathrm{j,out}} \mathrm{g_{ij}} + \sum_{\mathrm{j,in}} \mathrm{g_{ji}}$$

$$b_i = \textstyle\sum_{j,out}(-s_{ij} + g_{ij}x_{j,k-1} + \sum_{j,in}(s_{ji} + g_{ji}x_{j,k-1})$$

The new internal node i values x_i, k are then solved from

$$x_{i,k} = b_i/d_i$$

Note that in this iterative case the matrix equation is not needed to be constructed at all. Neither is there any need for solution order optimisation to avoid fill in of new matrix elements. The success of the iteration loop may be checked by solving the values v_{ij} of each edge from companion model equation and subsequently calculating the mass errors in each node.

4 Multicore processors are now available in ordinary laptop computers

Apros simulation engine has been optimised for running in single core processors with vector processing extensions. A complete power plant model can already be run on a modern PC with 100 ms time-step connected to a DCS system for testing of its functionality before installation at real plant. Of course, such a system may be used for optimising operational procedures, checking of emergency procedures and for training of plant operators, as well. New promising applications include predictive simulators for enhanced control systems or to provide 'what if' evaluation possibilities for the operators. Moore's law has been supposed to continue at least for 10 years with regard to number of transistors that can be placed on an integrated circuit. The obviously slower advances in increasing the single processor's speed will be compensated with multiple cores on the chip. Now we have quad core chips in our laptop computers. In a few years we will have hundreds or even thousands of cores at our disposal. We will face embedded highly parallel processing devices at our ubiquitous service, not only residing in supercomputer centres.

The new challenge is how to benefit from highly parallel computing. There are several bottlenecks. The speed of the main memory has not been increasing. Previously a floating point calculation on a single core computer consumed tens of memory cycles. The on-chip cash memories of modern multicore processors are, however, very fast. Now tens of floating point calculations is to be made on the chip to keep the processor node busy awaiting for next data from common main memory. Affordable compute cluster servers can be equipped with tens of such nodes to gain more virtual computing power. Compiler and operating system developers have implemented tools to support the take-up of multi-threaded processing: Concurrent threads on each core, threads cores in parallel on the processor nodes, and nodes in parallel on cluster installations. Separate programs run already nicely in parallel, but only using one core each, and with no own speed up. The programs need to be recompiled with parallelisation options to get any benefit from the new hardware at all. Specific comment lines can be included in existing source code to instruct the compiler on such portions of the code that can be computed in parallel. OpenMP standard has been implemented on many platforms [8].

However, to grab the possibilities now soon available, the source codes need to be revised. The program data organisation needs to be re-developed to minimise required cash operations to main memory. The computing algorithms need to be re-developed to promote on chip operations and minimize communication between computer nodes. At the same time the mathematical solver algorithms need to be developed to support multi scale modelling of heterogeneous processes requiring that separate parts of the integrated model are simulated with different time-steps [9]. It seems that Niclas Wirths law: “Software gets slower faster than hardware gets faster”, will come true even with regard to scientific computing subject that the software architectures and mathematical methods not are strongly adapted to the hardware developments. New laptops on the shelves of the computer stores are already supplied with multicore processor chips, 64 bit memory access, and 64 bit operating systems. Do not panic, referring to the millennium syndrome, your 32 bit single threaded applications will most likely still run, because traditional 32 bit single thread instruction sets are still directly supported, or at least emulated with some performance degradation. However, if you would take advance of the new platform developments you should act. The scientific problem is, that inherently serial algorithms are not very easy to parallelise. This seems to be a generic challenge. Orchestrated research efforts are definitely required, also at European level.

References

[1] E. Silvennoinen, I. Raiskinen, K. Juslin. *Introduction to APROS Simulation Environment*. ESM’88, European Simulation Multiconference, Nice, France, June 1–3, 1988, 4 pp.

[2] K. Juslin, *A companion Model Approach to Modelling and Simulation of Industrial Processes*. Espoo 2005, VTT Publications 574, ISBN 951-38-6660-2, URL: *http://www.vtt.fi/inf/pdf/*, 155+15 pp.

[3] K. Juslin, R. Lilja, A. Tourunen. *Challenges in Dynamic Simulation of Clean Combustion Power Plants*. PowerPlantSim 2008, Austin, Texas, SCS, March 25-27, 2 pp.

[4] K. Juslin, E. Puska, K. Porkholm. *Are You Ready for New Generation of Nuclear Power Plants? APROS is!* PowerPlantSim 2008 congress, Austin, Texas, SCS, March 25-27, 2 pp.

[5] K. Juslin, *Advanced Graph Based Industrial Process Modelling*. 7th Int. Conference on System Simulation and Scientific Computing, ICSC.2008, October 10-12, Beijing, China, 8p.

[6] K. Juslin, *Simulated Highly Parallel Solution of Versatile Companion Model Graph Equations*. PARA 2008, May 13-16, NTNU, Trondheim, Norway, 4 pp.

[7] H.M. Trent. *Isomorphisms between Oriented linear graphs and Lumped Physical Systems*, Journal Acoustic Society of America, Vol. 27, 1955, pp. 500 – 527.

[8] OpenMP Architecture Review Board. *OpenMP Application Program Interface*. Version 3.0 May 2008. Online: http://www.openmp.org/mp-documents/spec30.pdf , January, 7, 2009.

[9] Intel Corporation. *From a Few Cores to Many: A Tera-scale Computing Research Overview*. Online: http://download.intel.com/research/platform/terascale/terascale_overview_paper.pdf , January 7, 2008.

Corresponding author: Kaj Juslin
VTT Technical Research Centre of Finland
Information and Communication Technologies
P.O.Box 1000, Vuorimiehentie 3, Espoo
FI-02044 VTT, Finland, *kaj.juslin@vtt.fi*

Accepted: MATHMOD 2009
Revised: June 15, 2009
Accepted: July 20, 2009