A parallel chemical reactor simulation using Cactus

Karen D. Camarda, Yuan He, and Kenneth A. Bishop
Department of Chemical and Petroleum Engineering
University of Kansas

Abstract

Efficient numerical simulation of chemical reactors is an active field of research, both in industry and academia. As members of the Alliance Chemical Engineering Application Technologies (AT) Team, we have developed a reactor simulation for the partial oxidation of ortho-xylene to phthalic anhydride, an important industrial compound. This problem is of interest to chemical engineers for both theoretical and practical reasons. Theoretically, one is trying to understand the kinetics of the reactions involved. Currently, none of the several kinetic models that have been proposed is able to reproduce both the temperature profile inside the reactor and the spectrum of products observed in industry. Practically, an accurate simulation of this reaction would allow industrial chemical engineers to optimize plant operations, while ensuring a safe working environment.

We perform our simulation using a pseudo-homogeneous, non-isothermal, packed bed catalytic reactor model. The equations solved are a set of diffusion-convection partial differential equations, each with a generation term introduced by the chemical reaction model used. In general, these generation terms couple the equations in a nonlinear way, as is the case with the model we implement.

The discrete equations to be solved are derived by finite-differencing the partial differential equations to obtain the Crank-Nicholson representation, which is then linearized. Because solving the resulting system of equations is computationally expensive, an efficient parallel implementation is desired.

In order to achieve parallelism and portability with a minimum of development effort, we have ported our existing reactor model to the Cactus computational environment. To solve the linear systems that result, we have also ported a solver that uses a conjugate gradient method.

In this paper, we discuss the nature of the equations solved and the algorithm that will be used to solve them. We then present a surrogate problem with similar structure and show that solving this problem in Cactus shows good parallel scaling on the Linux cluster at the National Center for Supercomputing Applica-

tions (NCSA). We argue that the similarity between the two problems is sufficient to expect that the numerical solution of the reactor simulation will also scale well.

1 Introduction

The National Computational Science Alliance is developing a series of community specific portals [1]. The Kansas group within the Alliance's Chemical Engineering Application Technology team is working to develop topical examples that utilize and permit evaluation of the portal and the concomitant infrastructure. Current efforts are focused on demonstration of the ability to execute a chemical reactor simulation code, that is implemented in the Cactus environment, remotely on the POSIC Linux cluster at NCSA. The chemical reactor application will be managed via the Chemical Engineering Portal [2] and will use Globus technology for detailed task management. As such, the project will provide a test vehicle for the portal code being developed by the Enabling Team group at the University of Indiana. It also serves as a prototype for future projects in chemical reactor performance analysis and for other chemical engineering projects that require remote, parallel implementation of large simulation codes. Finally, it will contribute to the growing set of such applications that are/will be available via the portal.

The reactor simulator that is under development is a parallel version of a code that was developed several years ago at the University of Kansas and which was used as the topic vehicle for the collaborative Kansas Engineering and Science Interface project which was based on NCSA Habanero and Kerberos. The authentication infrastructure and the collaborative application management tasks for the new application will be handled with generic elements of the Chemical Engineering Portal; the specific computation engine is an application of Cactus technology (one of several possible engines).

The specific problem used in this paper is a surrogate for the chemical reactor code. Both codes implement sets of two-space dimension plus time instances of the diffusion convection equation that are driven by internal sources (chemical specie production and energy release). The surrogate problem used in this study does not contain the rich detail in terms of parameter setting capability and input (forcing) variable control that is part of the actual simulator. It does, however, capture the characteristics necessary for the scaling studies described here.

2 Governing equations

We dynamically simulate the partial oxidation of ortho-xylene to phthalic anhydride in the presence of a vanadium pentoxide (anatase) catalyst. The reaction takes place in a cylindrical packed-bed reactor immersed in an isothermal coolant bath. We employ a pseudo-homogeneous, non-adiabatic, non-isothermal model subject to the following assumptions and simplifications, which are reasonable given the operating conditions:

- The system is axisymmetric
- The gas obeys the ideal gas law
- Spatial and temporal variations in the physical properties, such as heat capacity, can be neglected

2.1 The interior equations

We apply the concepts of mass and energy balance, subject to the specifications of our model, to obtain the describing equations for our simulation:

$$\frac{\partial \theta}{\partial t} = \frac{1}{\overline{C}_p \overline{\rho}} \left[\frac{4k_{er}}{D_t^2 r} \frac{\partial}{\partial r} \left(r \frac{\partial \theta}{\partial r} \right) + \frac{k_{ez}}{L_t^2} \frac{\partial^2 \theta}{\partial z^2} - \frac{C_{pg} G}{L_t} \frac{\partial \theta}{\partial z} + H^* \right]$$

$$\frac{\partial f_l}{\partial t} = \frac{1}{\varepsilon} \left[\frac{4D_{er}}{D_t^2 r} \frac{\partial}{\partial r} \left(r \frac{\partial f_l}{\partial r} \right) + \frac{D_{ez}}{L_t^2} \frac{\partial^2 f_l}{\partial z^2} - \frac{1}{L_t} \frac{\partial (f_l u_z)}{\partial z} + R_l^* \right]$$

The dependent variable θ is the temperature of the gas T scaled by the temperature of the gas at the reactor inlet, T_0. The dependent variable f_l is the molar concentration of reactant l, c_l, scaled by the inlet concentration of ortho-xylene, c_{A0} [1]. The coordinates r and z are dimensionless variables scaled by the reactor dimensions so that the spatial domain is given by $0 \le r \le 1$ and $0 \le z \le 1$. u_z is the velocity of the gas in the axial direction. The terms H^* and R_l^* are energy and mass generation terms, the form of which are determined by the kinetic model implemented. The other quantities that appear are physical constants, which are defined in the Appendix.

2.2 Boundary conditions

At $r = 0$, we require that the dependent functions be symmetric, as is required by axisymmetry.

$$\left. \frac{\partial \theta}{\partial r} \right|_{r=0} = 0 \quad \left. \frac{\partial f_l}{\partial r} \right|_{r=0} = 0$$

At the wall of the reactor, $r = 1$, we have

$$\left. \frac{\partial \theta}{\partial r} \right|_{r=1} = \frac{-D_t h_w}{2\lambda_{reff}} (\theta - \theta_c) \quad \left. \frac{\partial f_l}{\partial r} \right|_{r=1} = 0$$

[1] Throughout this paper, A shall be used as a symbol for ortho-xylene, B for ortho-tolualdehyde, C for phthalide, and D for phthalic anhydride. A subscript 0 indicates a value at the inlet to the reactor.

where the descriptions of the constants are given in the Appendix. However, we note here that the wall heat transfer coefficient h_w was calculated from the overall heat transfer coefficient of the reactor using Beek's equation [3].

At the inlet to the reactor, $z = 0$, we have

$$\left.\frac{\partial \theta}{\partial z}\right|_{z=0} = -\beta_1(\theta - \theta_0) \quad \left.\frac{\partial f_l}{\partial z}\right|_{z=0} = -\beta_2\left(f_l - f_{l,0}\right)$$

where

$$\beta_1 = \frac{L_t G C_p}{k_{ez}} \quad \beta_2 = \frac{L_t G}{D_{ez}\rho_{g0}}$$

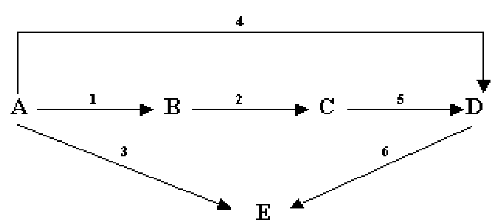
Again, please see the Appendix for the definition of the constants.

At the outlet of the reactor, $z = 1$, we require simply that the spatial variation of the evolved quantities vanishes. This is reasonable because we expect the most vigorous reaction to occur near the inlet.

$$\left.\frac{\partial \theta}{\partial z}\right|_{z=1} = 0 \quad \left.\frac{\partial f_l}{\partial z}\right|_{z=1} = 0$$

2.3 Kinetic model

The kinetic model used for calculating the mass and energy generation terms is that proposed by Calderbank [4]. The reaction network for this model is shown in Figure 1.



A: o-xylene B: o-tolualdehyde C: phthalide D: phthalic anhydride E: CO_x

Figure 1. Proposed Reaction Network for
the Partial Oxidation of Ortho-xylene by Calderbank et al. (1977)

Industrial experience has shown that this kinetic model predicts the location of the hot spot, as well as the ortho-xylene conversion and phthalic anhydride yield reasonably well [5]. The kinetic model also recognizes the formation of two un-desired intermediate products (o-tolualdehyde and phthalide), the presence of which influences the quality of the raw phthalic anhydride significantly, and as a result strongly affects the entire process for producing phthalic anhydride.

We note that these generation terms couple all the equations in a nonlinear way.

3 The Cactus Computational Tool Kit

The equations for the reactor simulation with the model described have already been solved numerically by our group [6]. The motivation for the current effort is the necessity for an implementation of the simulator that will run efficiently on parallel machines, in preparation for incorporating the true problem description, which is stiffer and nonlinear, into the Chemical Engineering Portal. This will be especially important when we increase the computational complexity by introducing heterogeneous chemical kinetics into the model. We wanted to be able to run it on many different architectures, from workstations to shared memory parallel machines to clusters, without needing to be experts on the details of the Message Passing Interface (MPI) or parallel I/O. To achieve these goals, we chose to implement the reactor simulation as a "thorn" in the Cactus Computational Toolkit (CCTK).

Cactus was conceived by a group of computational physicists for the purpose of solving partial differential equations numerically on regular grids. Their problems demanded large amounts of memory and computation time, and it was necessary to modify their complicated codes significantly each time a more powerful parallel machine became available. Additionally, the scientists were spending valuable resources learning the intricacies of MPI and other means of achieving parallelism. They sought to develop a framework that would minimize the amount of effort necessary to solve the same problem on different machines. From this effort came Cactus.

The Cactus Computational Toolkit [7] is an open source problem solving environment that enables parallel computation across many different architectures. Its name comes from its modular structure, in which a central core (the "flesh") provides a framework to which one can attach application or utility modules (the "thorns"). The modular structure allows collaboration among scientists working on the same project, as well as access to advanced computational technologies developed by others. It also allows machine-specific details to be largely ignored by the application developer.

We are currently implementing our reactor simulation as a thorn in Cactus. The process has been generally straightforward, with a few caveats which we mention below.

4 Numerical algorithm

We compute an approximate solution to the interior equations and boundary conditions given above by using standard finite difference techniques on a regular grid. In this section we discuss how we discretize the interior equations and boundary conditions, linearize the resulting difference equations, and solve the resulting linear system.

4.1 Finite difference approximation

Because of the diffusion terms present in the interior equations, we anticipate that any explicit numerical method we might attempt would be numerically unstable for any reasonable time step. We therefore chose to use a Crank-Nicholson type scheme for the discretization (see, e.g., [8]). It has been shown that for relatively simple equations this scheme is unconditionally stable. If we let i and j label our spatial grid points in the radial and axial directions respectively, and we let n label our time level, the discretized form of the energy balance equation becomes

$$\begin{aligned}\frac{\theta_{ij}^{n+1}-\theta_{ij}^{n}}{\Delta t} &= \frac{1}{2}\frac{4k_{er}}{\overline{C}_p\overline{\rho}D_t^2}\frac{1}{(\Delta r)^2}\left(\theta_{i+1,j}^{n+1}-2\theta_{ij}^{n+1}+\theta_{i-1,j}^{n+1}+\theta_{i+1,j}^{n}-2\theta_{ij}^{n}+\theta_{i-1,j}^{n}\right)\\ &+\frac{1}{2}\frac{4k_{er}}{\overline{C}_p\overline{\rho}D_t^2 r_i}\frac{1}{2\Delta r}\left(\theta_{i+1,j}^{n+1}-\theta_{i-1,j}^{n+1}+\theta_{i+1,j}^{n}-\theta_{i-1,j}^{n}\right)\\ &+\frac{1}{2}\frac{k_{ez}}{\overline{C}_p\overline{\rho}L_t^2}\frac{1}{(\Delta z)^2}\left(\theta_{i,j+1}^{n+1}-2\theta_{ij}^{n+1}+\theta_{i,j-1}^{n+1}+\theta_{i,j+1}^{n}-2\theta_{ij}^{n}+\theta_{i,j-1}^{n}\right)\\ &-\frac{1}{2}\frac{C_{pg}G}{\overline{C}_p\overline{\rho}L_t}\frac{1}{2\Delta z}\left(\theta_{i,j+1}^{n+1}-\theta_{i,j-1}^{n+1}+\theta_{i,j+1}^{n}-\theta_{i,j-1}^{n}\right)\\ &+\frac{1}{2}\frac{1}{\overline{C}_p\overline{\rho}}\left(H^{*n+1}+H^{*n}\right)\end{aligned}$$

This equation can be written

$$c_c\theta_{ij}^{n+1}+c_e\theta_{i+1,j}^{n+1}+c_w\theta_{i-1,j}^{n+1}+c_n\theta_{i,j+1}^{n+1}+c_s\theta_{i,j-1}^{n+1}+N_\theta=S_\theta$$

where N_θ contains the nonlinear terms that contain the evolved quantities at time level $n+1$, and S_θ contains all terms, linear and nonlinear, that include the evolved quantities at time level n. The mass balance equation for the quantities f_l can be written in the same form, although for that case we note that the nonlinearities come not just from the generation term R_l^*, but also from the axial velocity u_z.

We approximate the derivatives which appear in the boundary conditions with the one-sided, second-order accurate expression

$$\left.\frac{\partial f}{\partial x}\right|_{x_0}\cong\frac{-1}{2\Delta x}\left[3f(x_0)-4f(x_0+\Delta x)+f(x_0+2\Delta x)\right]$$

4.2 Linearization

In order to solve the above difference equation, we first linearize it about some initial guess for $\{\theta_{ij}^{n+1}, f_{l,ij}^{n+1}\}$. We define Q_θ as

$$Q_\theta = c_c\theta_{ij}^{n+1} + c_e\theta_{i+1,j}^{n+1} + c_w\theta_{i-1,j}^{n+1} + c_n\theta_{i,j+1}^{n+1} + c_s\theta_{i,j-1}^{n+1} + N_\theta$$

The first order Taylor series expansion of Q_θ about the initial guess is

$$\begin{aligned}
&Q_\theta(\theta_{ij}^{n+1} + \delta\theta_{ij}^{n+1}, \theta_{i+1,j}^{n+1} + \delta\theta_{i+1,j}^{n+1}, \theta_{i-1,j}^{n+1} + \delta\theta_{i-1,j}^{n+1}, \theta_{i,j+1}^{n+1} + \delta\theta_{i,j+1}^{n+1}, \theta_{i,j-1}^{n+1} + \delta\theta_{i,j-1}^{n+1}, \\
&\qquad f_{A,ij}^{n+1} + \delta f_{A,ij}^{n+1}, f_{B,ij}^{n+1} + \delta f_{B.ij}^{n+1}, f_{C,ij}^{n+1} + \delta f_{C,ij}^{n+1}, f_{D,ij}^{n+1} + \delta f_{D,ij}^{n+1}) = \\
&Q_\theta\left(\theta_{ij}^{n+1}, \theta_{i+1,j}^{n+1}, \theta_{i-1,j}^{n+1}, \theta_{i,j+1}^{n+1}, \theta_{i,j-1}^{n+1}, f_{A,ij}^{n+1}, f_{B,ij}^{n+1}, f_{C,ij}^{n+1}, f_{D,ij}^{n+1}\right) \\
&+ \frac{\partial Q_\theta}{\partial \theta_{ij}^{n+1}}\delta\theta_{ij}^{n+1} + \frac{\partial Q_\theta}{\partial \theta_{i+1,j}^{n+1}}\delta\theta_{i+1,j}^{n+1} + \frac{\partial Q_\theta}{\partial \theta_{i-1,j}^{n+1}}\delta\theta_{i-1,j}^{n+1} + \frac{\partial Q_\theta}{\partial \theta_{i,j+1}^{n+1}}\delta\theta_{i,j+1}^{n+1} + \frac{\partial Q_\theta}{\partial \theta_{i,j-1}^{n+1}}\delta\theta_{i,j-1}^{n+1} \\
&+ \frac{\partial Q_\theta}{\partial f_{A,ij}^{n+1}}\delta f_{A,ij}^{n+1} + \frac{\partial Q_\theta}{\partial f_{B,ij}^{n+1}}\delta f_{B,ij}^{n+1} + \frac{\partial Q_\theta}{\partial f_{C,ij}^{n+1}}\delta f_{C,ij}^{n+1} + \frac{\partial Q_\theta}{\partial f_{D,ij}^{n+1}}
\end{aligned}$$

We set the right hand side of the above to S_θ. Using the fact that N_θ contains no derivatives of the dependent variables, we can write the linear system to be solved as

$$\begin{aligned}
&\left(c_c + \frac{\partial N_\theta}{\partial \theta_{ij}^{n+1}}\right)\delta\theta_{ij}^{n+1} + c_e\delta\theta_{i+1,j}^{n+1} + c_w\delta\theta_{i-1,j}^{n+1} + c_n\delta\theta_{i,j+1}^{n+1} + c_s\delta\theta_{i,j-1}^{n+1} + \\
&\qquad \frac{\partial N_\theta}{\partial f_{A,ij}^{n+1}}\delta f_{A,ij}^{n+1} + \frac{\partial N_\theta}{\partial f_{B,ij}^{n+1}}\delta f_{B,ij}^{n+1} + \frac{\partial N_\theta}{\partial f_{C,ij}^{n+1}}\delta f_{C,ij}^{n+1} + \frac{\partial N_\theta}{\partial f_{D,ij}^{n+1}}\delta f_{D,ij}^{n+1} = R_\theta
\end{aligned}$$

where

$$R_\theta = S_\theta - Q_\theta\left(\theta_{ij}^{n+1}, \theta_{i+1,j}^{n+1}, \theta_{i-1,j}^{n+1}, \theta_{i,j+1}^{n+1}, \theta_{i,j-1}^{n+1}, f_{A,ij}^{n+1}, f_{B,ij}^{n+1}, f_{C,ij}^{n+1}, f_{D,ij}^{n+1}\right)$$

There are similar expressions for the mass balance equations, although those have many more terms because of the derivatives introduced by the term with u_z.

4.3 Solving the linear system

After discretizing and linearizing, we are left with a linear system of $(n_r - 2)\times(n_z - 2)\times 5$ equations in the same number of unknowns. This system can be pictured

$$\begin{pmatrix} c^{(\theta\theta)} & c^{(\theta A)} & c^{(\theta B)} & c^{(\theta C)} & c^{(\theta D)} \\ c^{(A\theta)} & c^{(AA)} & c^{(AB)} & c^{(AC)} & c^{(AD)} \\ c^{(B\theta)} & c^{(BA)} & c^{(BB)} & c^{(BC)} & c^{(BD)} \\ c^{(C\theta)} & c^{(CA)} & c^{(CB)} & c^{(CC)} & c^{(CD)} \\ c^{(D\theta)} & c^{(DA)} & c^{(DB)} & c^{(DC)} & c^{(DD)} \end{pmatrix} \begin{pmatrix} \delta\theta^{n+1} \\ \delta f_A^{n+1} \\ \delta f_B^{n+1} \\ \delta f_C^{n+1} \\ \delta f_D^{n+1} \end{pmatrix} = \begin{pmatrix} R_\theta \\ R_A \\ R_B \\ R_C \\ R_D \end{pmatrix}$$

where the $c^{(lm)}$ are coefficient matrices, with l labeling the particular balance equation and m labeling the dependent variable. The matrices off the diagonal of the main matrix represent contributions from the coupling terms.

Although the standard Cactus distribution does include linear system solvers, they were not immediately useable, due to the fact that they were dependent on thorns used to solve the Einstein equations. Instead we ported a code that implemented a conjugate gradient method originally written by Towns [9], based on an algorithm developed by Van der Vorst [10]. This code was written with parallel and vector computers in mind, and was straightforward to modify to incorporate it into Cactus.

4.4 Algorithm for advancing simulation one time step

For a given timestep, once the linear system has been solved, the solution is used to update the dependent variables, and the procedure is repeated until the system of equations has converged. Below is the algorithm used to advance the whole system one timestep.

1. Compute the source terms S_θ and S_l.
2. Set an initial guess for the new values of θ^{n+1} and f^{n+1}.
3. Compute the right hand side vectors R_θ and R_l.
4. Compute the stencil coefficients.
5. Incorporate the boundary conditions into the stencil coefficients.
6. Solve the linear system.
7. Apply boundary conditions to $\delta\theta^{n+1}$ and δf^{n+1}.
8. Compute some norm of δ.
9. While the norm of δ exceeds some tolerance…
    a. Set $\theta^{n+1} = \theta^{n+1} + \delta\theta^{n+1}$ and $f_l^{n+1} = f_l^{n+1} + \delta f_l^{n+1}$.
    b. Compute R_θ and R_l.
    c. Compute the stencil coefficients.
    d. Incorporate the boundary conditions into the stencil coefficients.
    e. Solve the linear system.
    f. Apply boundary conditions to $\delta\theta^{n+1}$ and δf^{n+1}.
    g. Compute some norm of δ.
10. Set $\theta^{n+1} = \theta^{n+1} + \delta\theta^{n+1}$ and $f_l^{n+1} = f_l^{n+1} + \delta f_l^{n+1}$.

5 A surrogate problem

While the reactor thorn is under development, we constructed another thorn to solve a simpler problem with similar structure. Solving this problem allows us to evaluate whether our reactor thorn will scale on parallel architectures.

The system of equations being solved is

$$\frac{\partial u_0}{\partial t} = \frac{\partial^2 u_0}{\partial x^2} - \frac{\partial u_0}{\partial x} + \frac{\partial^2 u_0}{\partial y^2} - \frac{\partial u_0}{\partial y} + \left(2\pi^2 - 1\right)u_0 + \pi u_1 + \pi u_2$$

$$\frac{\partial u_1}{\partial t} = \frac{\partial^2 u_1}{\partial x^2} - \frac{\partial u_1}{\partial x} + \frac{\partial^2 u_1}{\partial y^2} - \frac{\partial u_1}{\partial y} + \left(2\pi^2 - 1\right)u_1 - \pi u_0 + \pi u_3$$

$$\frac{\partial u_2}{\partial t} = \frac{\partial^2 u_2}{\partial x^2} - \frac{\partial u_2}{\partial x} + \frac{\partial^2 u_2}{\partial y^2} - \frac{\partial u_2}{\partial y} + \left(2\pi^2 - 1\right)u_2 - \pi u_0 + \pi u_3$$

$$\frac{\partial u_3}{\partial t} = \frac{\partial^2 u_3}{\partial x^2} - \frac{\partial u_3}{\partial x} + \frac{\partial^2 u_3}{\partial y^2} - \frac{\partial u_3}{\partial y} + \left(2\pi^2 - 1\right)u_3 - \pi u_1 - \pi u_2$$

We solve this system for $0 \le x \le 1$, $0 \le y \le 1$ subject to the boundary conditions

$$u_0(0, y, t) = u_0(1, y, t) = 0 \quad u_0(x, 0, t) = u_0(x, 1, t) = 0$$

$$u_1(0, y, t) = u_1(1, y, t) = 0 \quad \left.\frac{\partial u_1}{\partial y}\right|_{y=0} = \left.\frac{\partial u_1}{\partial y}\right|_{y=1} = 0$$

$$\left.\frac{\partial u_2}{\partial x}\right|_{x=0} = \left.\frac{\partial u_2}{\partial x}\right|_{x=1} = 0 \quad u_2(x, 0, t) = u_2(x, 1, t) = 0$$

$$\left.\frac{\partial u_3}{\partial x}\right|_{x=0} = \left.\frac{\partial u_3}{\partial x}\right|_{x=1} = 0 \quad \left.\frac{\partial u_3}{\partial y}\right|_{y=0} = \left.\frac{\partial u_3}{\partial y}\right|_{y=1} = 0$$

and the initial conditions

$$u_0(x, y, 0) = \sin(\pi x)\sin(\pi y)$$
$$u_1(x, y, 0) = \sin(\pi x)\cos(\pi y)$$
$$u_2(x, y, 0) = \cos(\pi x)\sin(\pi y)$$
$$u_3(x, y, 0) = \cos(\pi x)\cos(\pi y)$$

The analytic solution to this problem is

$$u_0(x, y, t) = e^{-t}\sin(\pi x)\sin(\pi y)$$
$$u_1(x, y, t) = e^{-t}\sin(\pi x)\cos(\pi y)$$
$$u_2(x, y, t) = e^{-t}\cos(\pi x)\sin(\pi y)$$
$$u_3(x, y, t) = e^{-t}\cos(\pi x)\cos(\pi y)$$

We discretize these equations using a Crank-Nicholson-type method in the same way as we do the reactor equations, and solve them via the same algorithm. Timing results are present in the next section.

6 Performance on Linux clusters

We computed solutions to our surrogate problem on the POSIC cluster of Linux workstations at NCSA. In the queue in which our computations were run, there are 64 Hewlett Packard Kayak Visualize workstations. Each workstation contains two PIII Xeon processors running at 550 MHz, 512 kb of cache per processor, and 1 Gb of memory. The machines are connected by a Myrinet Lanai 7 high speed interface.

Run times were measured for three different grid sizes for different numbers of processors. The times were obtained using the C function gettimeofday, called by the Cactus flesh. We only report the time spent in the evolution routine (and the routines called therein). Time spent in other routines was negligible.

In Figure 2 we show speedup data for each of the three grid sizes up to 32 processors, where we define speedup as the time spent with one processor divided by the time spent for N processors for a given problem size.

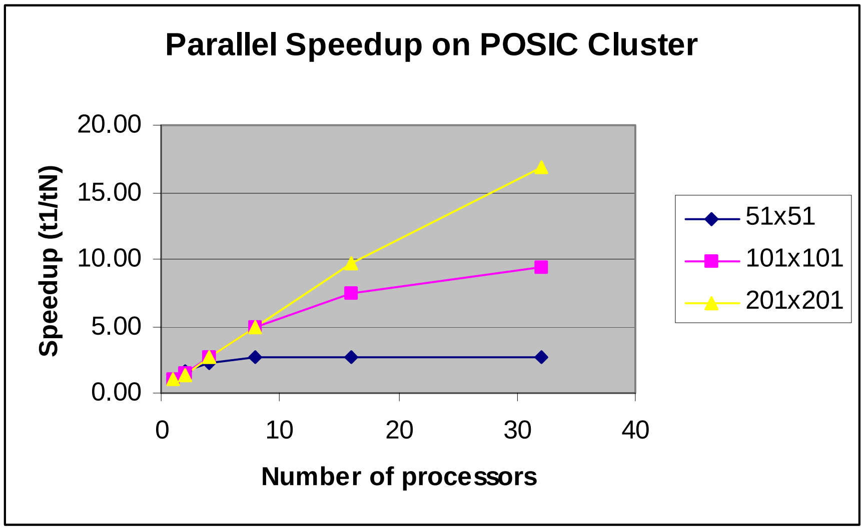


Figure 2. Parallel speedup vs. number of processors for different problem sizes.

Because we did not perform these timings in dedicated queues, we repeated all measurements, and found in all cases that the times varied by less than 10%, usually much less. As expected, the runs on smaller grids for which the surface area to volume ratio for each processor's sub-grid is relatively large, did not scale as well as the larger problems. In fact, for the largest problem run, the speedup curve had not yet turned over at 32 processors.

7 Conclusions

We find that our surrogate problem scaled well on the Linux cluster, and expect that our reactor simulation will also. In the case of the reactor simulation, for the same size spatial grid, there will be much more computation for the same amount of communication, so we might expect the scaling to be a bit better. (Most, but not all, of the increased computation occurs outside the linear solver, so we do not expect large gains in scaling.) However, the grid sizes used for the reactor simulation are generally smaller – a typical high resolution run uses 36 points in the radial direction and 132 points in the axial direction, so the current two-dimensional simulation may not scale well beyond a few processors. Nonetheless, any significant scaling would be useful, as these runs can be time consuming to perform.

We further conclude that the use of Cactus is a well-designed computational framework. We were able to port our problem and have it run in parallel much more quickly than we could have if we had had to learn the details of MPI and data distribution.

Acknowledgements

The Authors acknowledge the financial support of the National Science Foundation under the Cooperative Agreement with the University of Illinois # ACI-9619019, The University of Kansas #KUCR12581 and # KUCR12583, and Microsoft. We acknowledge the Alliance Chemical Engineering Applications Technology team, specifically Mr. Jay Alameda, for the continuing collaboration that we enjoy. We would also like to thank the Cactus development team for their patient help, specifically Gabrielle Allen and Tom Goodale. Computations were performed on the NCSA Linux cluster.

Appendix: Physical constants

D_t - Diameter of the reactor tube, in *m*

D_{er}, D_{ez} - Effective radial and axial diffusivities, in m^2/s

k_{er}, k_{ez} - Effective radial and axial thermal conductivities, in *J/(s m K)*

$C_p, \overline{C}_p$ - Heat capacity of the gas phase, and the average heat capacity of the gas phase and solid phase, in *J/(g K)*

L_t - Length of the reactor, in *m*

u_z - Gas phase velocity in axial direction, in *m/s*

c_l - Gas phase concentration of species *l*, in mol/m^3

T - Temperature of the gas and solid phase, in *K*

T_c - Temperature of the coolant, in *K*

t - Time, in *s*

f_l - Dimensionless concentration of species *l*

r, z - Dimensionless distance in radial and axial directions

h_w - Wall heat transfer coefficient, in $J/(s\ m^2\ K)$

ε - Porosity of the bed

$\bar{\rho}, \rho_g$ – Average density of the gas and solid phases, and density of the gas phase, g/m^3

$\theta, \theta_w, \theta_c$ - Dimensionless temperature of the reactor, the wall and the coolant

References

[1] http://www.extreme.indiana.edu/an/index.html.

[2] Krishnan, S., Bramley, R., Gannon, D., Govindaraju, M., Indurkar, R., Slominski, A., Temko, B., Alkire, R., Drews, T., Webb, E., and Alameda, J. The XCAT Science Portal. Submitted to Supercomputing 2001 (preprint available at http://www.extreme.indiana.edu/an/index.html).

[3] Beek, J., Design of packed catalytic reactors. *Advances in Chemical Engineering (Edited by Th. Drew, J. W. Hoops and Th. Vermeulen)*, ***3***, 203-277, Academic Press, New York, 1962.

[4] Calderbank, P.H., Chandrasekharan, K. & Fumagalli, C. The prediction of the performance of packed-bed catalytic reactors in the air-oxidation of o-xylene. *Chemical Engineering Science*, **32**, pp. 1435-1443, 1977.

[5] Anastasov, A.I. & Nikolov, V.A. Optimal policies of operation of a fixed reactor for oxidation of o-xylene into phthalic anhydride. *Ind. Eng. Chem. Res.*, **37**, pp. 3424-3433, 1998.

[6] He, Yuan. Studies on a 2D Pseudo-homogeneous Reactor Model. MS thesis, Department of Chemical and Petroleum Engineering, University of Kansas, 2000.

[7] http://www.CactusCode.org/.

[8] Press, W.H., Flannery, B.P., Teukolsky, S.A & Vetterling, W.T. *Numerical Recipes*, Cambridge University Press, Cambridge, England, 1986.

[9] Towns, J. & Seidel, E. A review of five linear system solvers used for elliptic partial differential equations. *NCSA data link*, July-August 1992.

[10] van der Vorst, H.A. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **13(2)**, pp. 631-644, 1992.