OPENMP and MPI Parallelisation in FRESCO

Ian Thompson, LLNL

June 25, 2020

1 Solving the Coupled Equations

WIth only local couplings, the coupled equations for a given J^π describe a *vector* of M channel wave functions $\psi_{\alpha\alpha_i}$, and a *matrix* of coupling potentials $\mathsf{V}(R)$. We integrate a trial solutions, grouped together into a matrix Y, as the solution of

$$\mathsf{Y}''(R) = \mathsf{F}(R)\mathsf{Y}(R) \tag{1}$$

$$\text{for } \mathsf{F}(R) = \frac{2\mu_\alpha}{\hbar^2}[\mathsf{V}(R) - E_\alpha] + \frac{L_\alpha(L_\alpha{+}1)}{R^2}, \tag{2}$$

where the energies E_α and centrifugal barriers are diagonal matrices. In component form, Eq. (**??**) is

$$\mathsf{Y}''_{\alpha\beta}(R) = \sum_{\alpha'=1}^{M} \mathsf{F}_{\alpha\alpha'}(R)\mathsf{Y}_{\alpha'\beta}(R) \ . \tag{3}$$

These coupled differential equations can be solved by forming the linearly independent solution sets $\{\mathsf{Y}_{\alpha\beta}(R)\}$, where the β'th solution consists of a set of all channels ($\alpha = 1 \ldots M$) that is made linearly independent of the other sets by having a distinctive starting value at $R = R_{\rm min}$:

$$\mathsf{Y}_{\alpha\beta}(R_{\rm min} - h) = 0, \qquad \mathsf{Y}_{\alpha\beta}(R_{\rm min}) = \delta_{\alpha\beta}\frac{(k_\alpha R_{\rm min})^{L_\alpha+1}}{(2L_\alpha{+}1)!!} \tag{4}$$

for the initial conditions in the radial integration of equations (**??**). The factor of $\delta_{\alpha\beta}$ may be varied. These solutions together form a matrix, where the row index of Y refers to the channel within the coupled channels, and the column index refers to a linearly independent solution. The scattering boundary conditions of are to be satisfied by the $\psi_{\alpha\alpha_i}$ solution, for incoming plane wave in channel α_i.

The solutions $\psi_{\alpha\alpha_i}$ are linear combinations of the $\mathsf{Y}_{\alpha\beta}(R)$

$$\psi_{\alpha\alpha_i} = \sum_{\beta=1}^{M} c_{\beta\alpha_i}\mathsf{Y}_{\alpha\beta}(R) \ , \tag{5}$$

again by satisfying the boundary conditions at $R = a$ and say $R = a - 5h$. The column of S matrix elements $S_{\alpha\alpha_i}$ are a product of the matching procedure.

2 Parallelisation

2.1 MPI

The coupled channels sets for each J^π set are logically independent, and so may be calculated in parallel. This is done using MPI parallelisation, is organised so the different calculations may be on different mpi nodes, without any shared fast memory.

Information specifying the couplings has to be passed from the primary node to the slave nodes. All the nodes calculate the coupling matrix $\mathsf{V}(R)$, first generically, and then later for each J^π set. The resulting groups of columns of the S matrix are sent back in turn to the primary node, along with the segments of stdout file for inclusion in the primary job output.

This works well, especially with reactions of heavier particles when the coupled channels sets are all approximately of the same size. For nucleon incident particles, by constrast, there are only 20 to 30 sets, and the sets with small J are smaller in size.

Performance

Works efficiently if number of mpi nodes is less than half of the number of J^π sets. Also need all the J^π set calculations for a given cluster-node to fit in the memory available. If there are 8 mpi-nodes per cluster-node for example, then only 1/8 of the cluster-node memory is available for each J^π set.

2.2 OPENMP

We use OPENMP parallelisation primarily to divide up in Eq. (**??**) the calculations for different β values. These separate columns of the $\mathsf{Y}_{\alpha\beta}$ are distinct solutions of the coupled equations that are linearly independence. This independence implies that they may be calculated in parallel.

OPENMP parallelisation works with shared-memory computers, such as several CPUs do on a single node. Many large computers now have 8 cpus (or at least 8 parallel threads) on each node. The shared memory suits the parallelisation over β, since the coupling matrix coefficients $\mathsf{F}_{\alpha\alpha'}(R)$ may be shared between the OPENMP threads.

We can also use parallel OpenMP to calculate the Racah algebra coefficients for the couplings between all pairs of channels. A parallelisation of the outer channel loop is all that is needed.

Intel Core Duo / Macbook	Total CPU	Elapsed Time
Serial	140.0	
1 openmp	157.1	160.4
2	185.6	115.0
3	180.7	111.9
AMD Opteron / Yana		
serial	309.4	
1 openmp	341.5	344.6
2 openmp	273.7	188.7
4 openmp	286.1	143.2
6 openmp	305.4	121.4
8 openmp	346.5	134.0

Table 1: OPENMP Performance Tests

Performance

Consider an example of neutron+Zr scattering with up to $M = 385$ channels (ph-e20-cc/e20-s). The performances for OPENMP are given in Table **??** for first a two-threaded Intel Core Duo at 2.16 GHz in a Macbook Pro, and secondly on a Yana node with up to 8 threads with AMD Opterons at 2.4 GHz. Both machines used the Intel ifort compiler: the macbook using version 10.1 and Yana using version 9.1.

Conclusions

On a dedicated machine such as the Macbook, the elapsed times are reduced by 30% (theoretical maximum is 50%).

On the Yana 8-way cluster, efficiencies are gained for 2 – 6 openmp threads, but with diminishing returns. Using all 8 cpus on the node appears to increase the elapsed time. (We will have to recheck that the cluster was correctly dedicated to this job).