NIMROD Scaling Information for Nonlinear Tokamak Simulations Using SLU

Carl Sovinec, U-WI, August 13, 2004

The following tests are based on a nonlinear tokamak computation that starts from an EFIT equilibrium. They have been run on NERSC's IBM SP3 cluster (Seaborg), which has 16 processors per node. Each processor has a peak speed of 1.5 GFlops, and the efficiencies reported below are the 'ERCAP GFLOPS' from poe+ output divided by 1.5 times the number of processors. The timings are over 20 time-steps; though, the only single-processor timing is a projection for each part of the computation, based on 5- and 10-step computations—this case did not run 20 steps within the interactive limit. All runs were forced to re-compute and re-factor the vmhd matrix precisely 3 times after the first time-step to average one vmhd matrix create per 5 steps, which is representative of typical nonlinear calculations. They achieved 500 to 1500 loads/Translation Look-aside Buffer miss (grid blocks have mx × my = 12 × 6 to manage cache).

The net result regarding parallel scaling is that for our typical problems, the finite element computations (which use point-to-point communication for seaming along grid-block edges) scale well, but direct solves and FFT routines with mpi_alltoallv communication do not. Fortunately, neither of these dominates CPU time until you use a large number of processors. Thus, the scaling of the finite element computations masks poor scaling elsewhere. We may be able to do something about the FFTs by going to point-to-point communication, and using larger poly_degree values reduces the work for the direct solve, since much of it then gets done in the static condensation step.

A somewhat surprising result (in a good way) is that all of the GFlop counts are above 5% of the theoretical peak and as high as 16% for the single-processor computation. This may be due to using larger values of poly_degree than in previous tests (4 is now pretty typical), or the new GFLOP diagnostic from poe+ may be on the generous side. Our previous estimates were always below 5% and as bad as 2% in some cases. The largest computation among all of the tests is one with a 48×48 mesh of biquartic elements with 43 Fourier components. It achieves 5.2% of peak efficiency for 352 processors for an average of 32 seconds per time-step.

The first sets of computations do not have any 3D matrix-free linear system solves. All linear systems reach the specified tolerance of 10^{-11} or better in a single pass of the SLU solver.

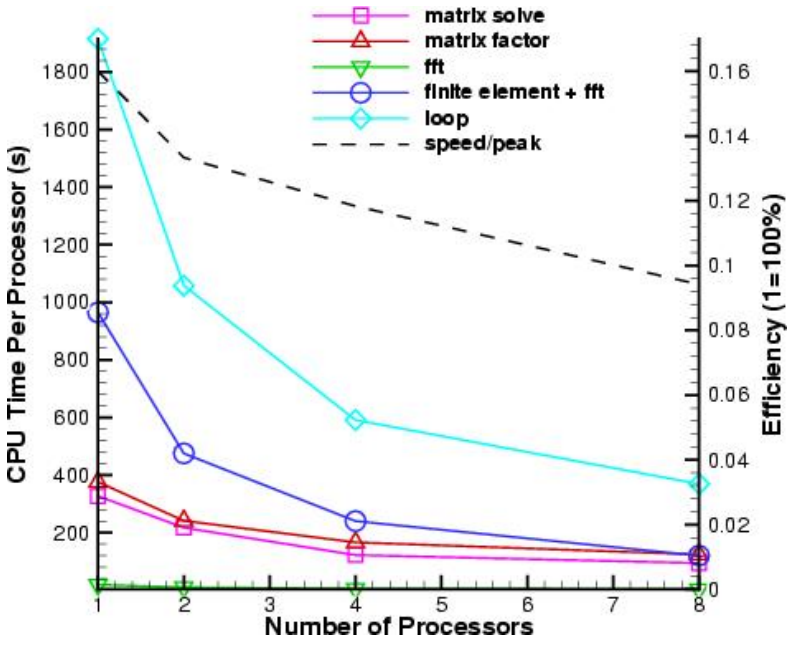

Figure 1. Strong scaling of a 24×24 poly_degree=4 tokamak computation with 6 Fourier components. Note that most of the scaling is due to the finite element computations; the direct factoring and solve times decrease at a slow rate.

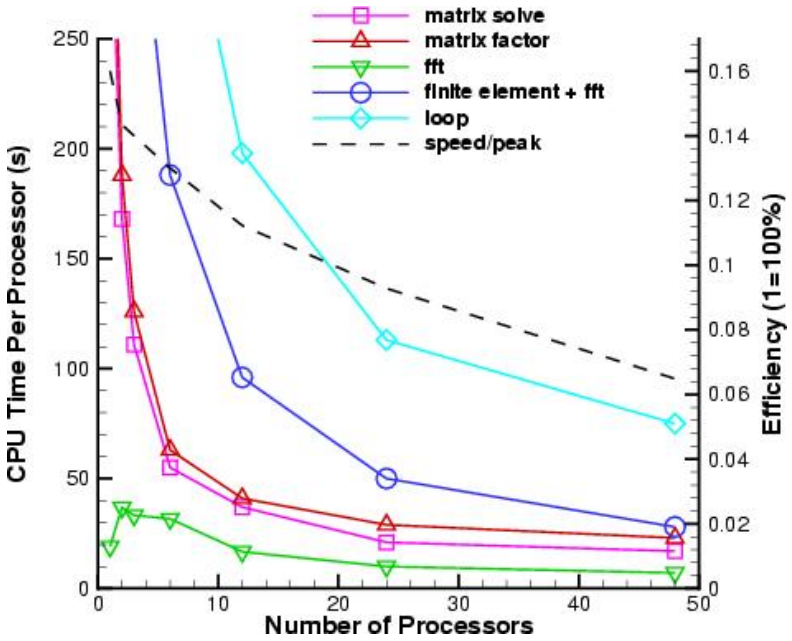

Figure 2. Same strong scaling except the domain decomposition is over Fourier components up to 6 processors, then it's over the poloidal plane. The FFT time does not scale with layers (it should and did on the T3E).

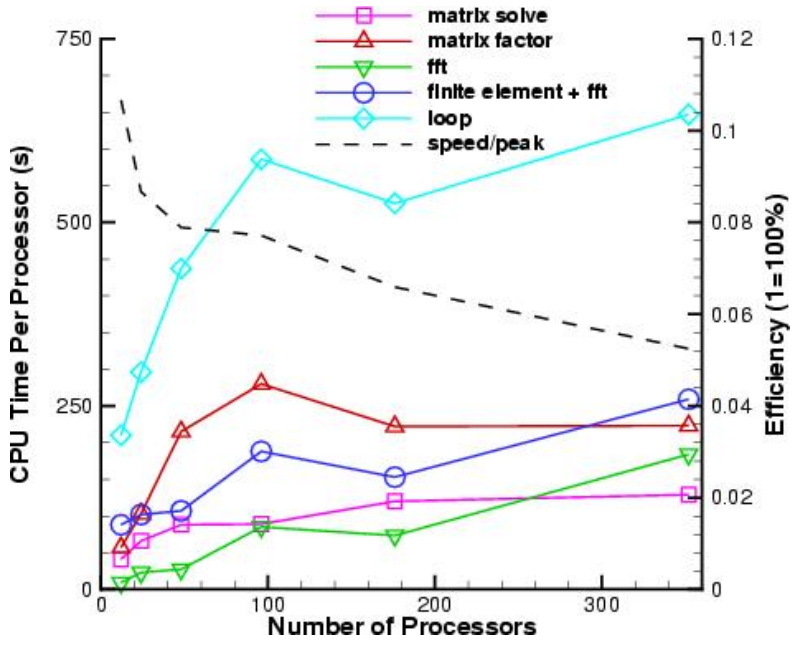

Figure 3. This weak scaling doubles mx to 48, then my to 48, then increases the number of Fourier components to 11 (6 layers), 22 (11 layers), and finally to 43 (22 layers). The computational efficiency decreases from 10.7% of Seaborg peak to 5.2% for the respective number of processors as the problem size is increased by a factor of 29.

The next two figures show computations with poly_degree increased to 6 and the mesh sizes reduced to use the same amount of data as the poly_degree=4 computations. CPU times from the last computation of the second strong-scaling plot for poly_degree=4 (Fig. 2) may be compared with the last computation in Fig. 4; however, for smooth fields, the poly_degree=6 case with the same amount of data would have much better resolution. Note that with more grid blocks, the poly_degree=6 case should continue to scale, since it is still dominated by finite-element-computation time at 48 processors.

[As a side note, the poly_degree=6 computation with the largest mesh was run with both Lagrange polynomials and with Gauss-Lobatto-Legendre polynomials. The latter are legitimate bases for “spectral elements,” like the modal representation used for SEL, since they do not become linearly dependent as poly_degree is increased.]

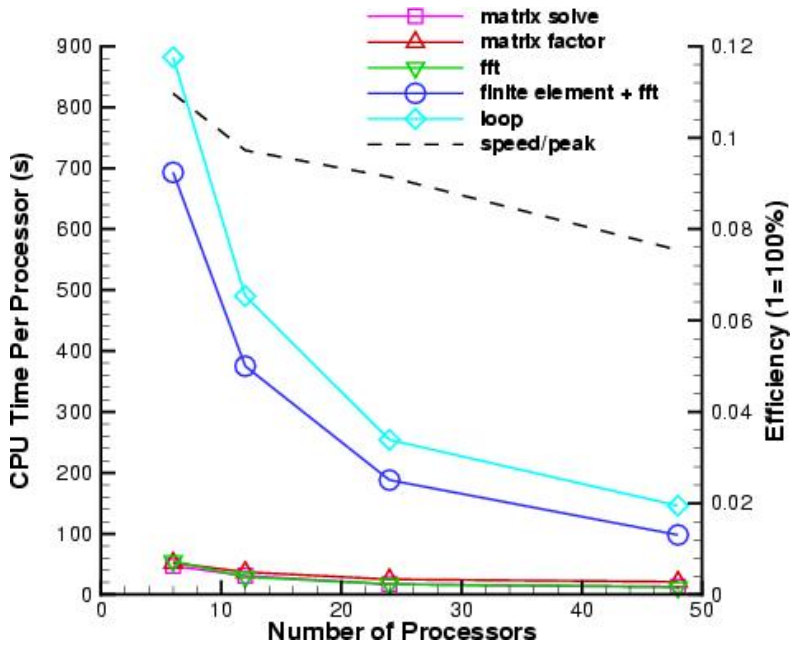

Figure 4. This is a poly_degree=6 strong scaling using a 16×16 mesh. The domain decomposition is over the poloidal plane only. This problem scales well, because most of the time is spent in the finite element computations, and the point-to-point block seaming scales well.

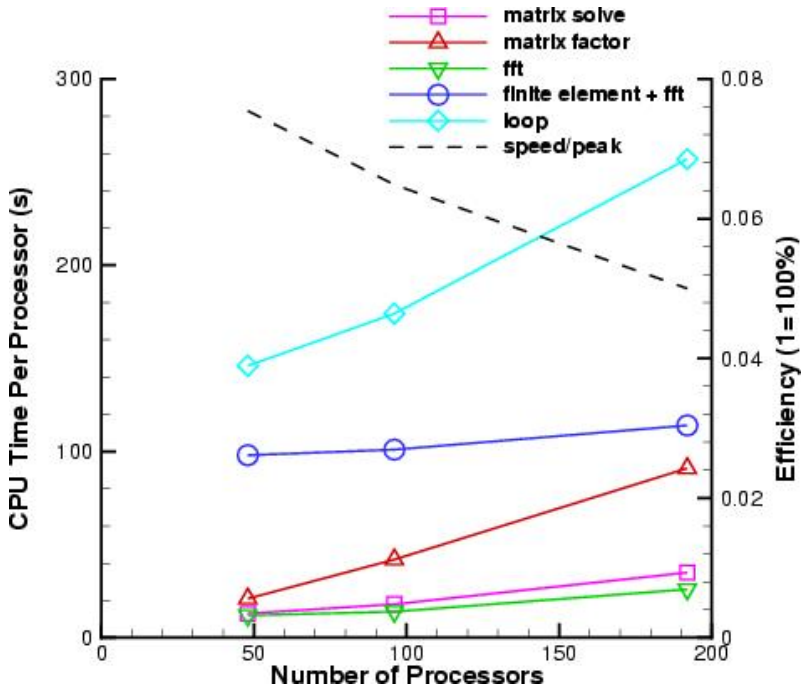

Figure 5. This is a poly_degree=6 weak scaling increasing mx to 32 then my to 32. The number of Fourier components is fixed at 6, and nlayers=6.

The last sets of runs have 3D linear-system solves for anisotropic thermal conduction and full continuity evolution, and poly_degree is back to 4. The strong scaling case performs 7 to 8 ‘matrix-free’ vmhd iterations per step, and the temperature advance takes less than 20 iterations. In the weak scaling, the number of temperature iterations increases to about 30 in the 11 Fourier-component computation, while the vmhd iterations increase to 10 or so. They increase just a little more in going to 22 Fourier components.

Note that unlike the computations without the 3D linear-system solves, the iteration time includes part of the finite element time. CPU times are 2-3 times greater than without the 3D solves, but the efficiencies are comparable.

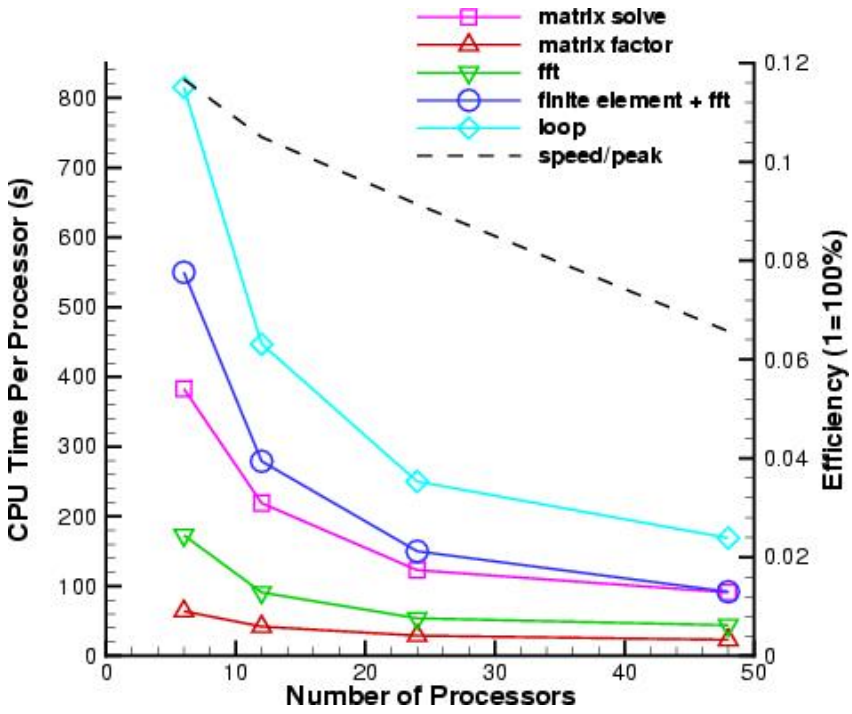

Figure 6. Stong scaling for poly_degree=4 with a 24×24 mesh using 3D linear-system solves. There are 6 Fourier components and 6 layers in each of these computations.

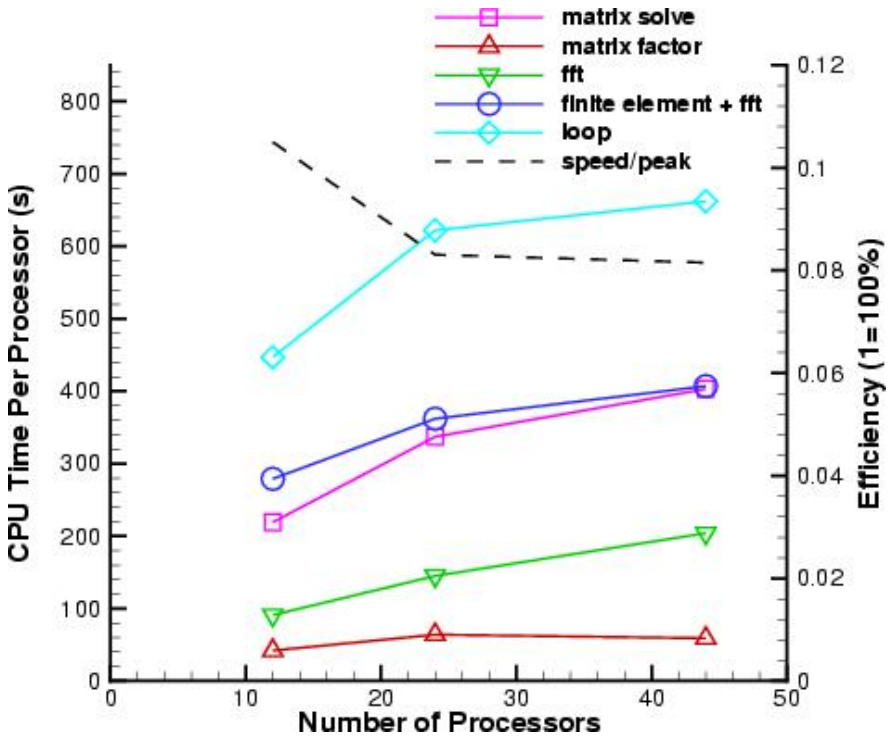

Figure 7. Weak scaling for poly_degree=4 with 3D linear-system solves. The first computation has a 24×24 mesh, 6 Fourier components and 6 layers. The second has the same mesh, 11 Fourier components and 6 layers (more poloidal domain decomposition). The third has the same mesh, 22 Fourier components, and 11 layers (more toroidal decomposition).

The following is the text plot data input for Tecplot. Raw GFflop and memory per task are recorded here:

```
VARIABLES="nproc" "nl" "mx" "my" "pd" "nf" "it" "fac" "fft" "fe" "loop" "GFs"

# All tests are based on a nonlinear tokamak computation from an EFIT
equilibrium.
# The timings are over 20 time-steps; though, the very first result reports
# projections for each part of the computation it it were able to run 20
steps
# within the interactive limit.  All computations achieved 500-1500 loads/TLB
# miss (blocks have mx x my = 12 x 6).

# Computations without 3D solves were forced to recompute the vmhd matrix 3
times
# after the first time-step.

# This series is a strong scaling (fixed problem size) with poloidal domain
# decomposition only.

ZONE I=4 F=POINT

1  1  24  24  4  6  327  376  19.1  965  1915  0.24
2  1  24  24  4  6  217  241  9.66  475  1058  0.40
4  1  24  24  4  6  122  166  4.61  240   591  0.71
8  1  24  24  4  6   94  124  2.48  120   369  1.13

#ERCAP MB/task are, respectively, 1519, 874, 533, and 369.


# This series is a strong scaling over Fourier components first then over
blocks.

ZONE I=7 F=POINT

1  1  24  24  4  6  327  376  19.1  965  1915  0.24
2  2  24  24  4  6  168  188  36.8  503   990  0.43
3  3  24  24  4  6  111  126  33.3  341   666  0.63
6  6  24  24  4  6   55   63  31.7  188   353  1.17
12 6  24  24  4  6   37   41  16.7   96   198  2.02
24 6  24  24  4  6   21   29  10.0   50   113  3.34
48 6  24  24  4  6   17   23   7.1   28    75  4.67

#ERCAP MB/task are, respectively, 1519, 808, 561, 322, 254, 189, 157.


# This is a weak scaling increasing the poloidal resolution first then
toroidal.
# The last computation has 29 times as much data as the first.

ZONE I=6 F=POINT

12   3  24  24  4  6  41   57   9.1   88   210  1.92
24   3  48  24  4  6  66  103  23.0  102   296  3.12
48   3  48  48  4  6  88  215  27.4  107   437  5.68
96   6  48  48  4 11  89  280  85.1  188   586  11.11
176 11  48  48  4 22 120  222  73.4  153   526  17.4
```

```
352 22  48  48  4 43 129  223  184   259   647  27.7

#ERCAP MB/task are, respectively, 247, 395, 586, 570, 587, 584


# This is a strong scaling with a 16x16 mesh of pd=6 and 6 layers.

ZONE I=4 F=POINT

6   6  16  16  6  6  47  52  55  693  882  0.987
12  6  16  16  6  6  31  37  29  375  490  1.75
24  6  16  16  6  6  17  25  17  188  254  3.29
48  6  16  16  6  6  13  21  12   98  146  5.43

#ERCAP MB/task are, respectively, 358, 254, 188, 157


# This is a weak scaling with pd=6, increasing poloidal resolution only.
# Even with a reduced time-step, the last computation required 10 bmhd
its/step.

ZONE I=3 F=POINT

48  6  16  16  6  6  13  21  12   98  146  5.43
96  6  32  16  6  6  18  42  14  101  174  9.34
192 6  32  32  6  6  35  91  26  114  257  14.4

#ERCAP MB/task are, respectively, 157, 262, 472


# Poly_degree=4 but anisotropic conduction and continuity=full.  This takes
about
# 8 vmhd its and 17 T its; though, the iterations increase on the larger
problems.
# Strong scaling:

ZONE I=4 F=POINT

6   6  24  24  4  6  383  64  173  550  815  1.05
12  6  24  24  4  6  219  42   91  279  447  1.89
24  6  24  24  4  6  123  29   54  150  250  3.29
48  6  24  24  4  6   91  23   44   92  169  4.73

#ERCAP MB/task are, respectively, 374, 258, 185, 153


# Weak scaling increasing toroidal resolution only.  [but poloidal decomp
increased then nl]
# The iterations increase on the larger problems.

ZONE I=3 F=POINT

12  6  24  24  4  6  219  42   91  279  447  1.89
24  6  24  24  4 11  337  64  145  362  622  2.99
44 11  24  24  4 22  403  59  204  407  662  5.38

#ERCAP MB/task are, respectively, 258, 251, 265
```