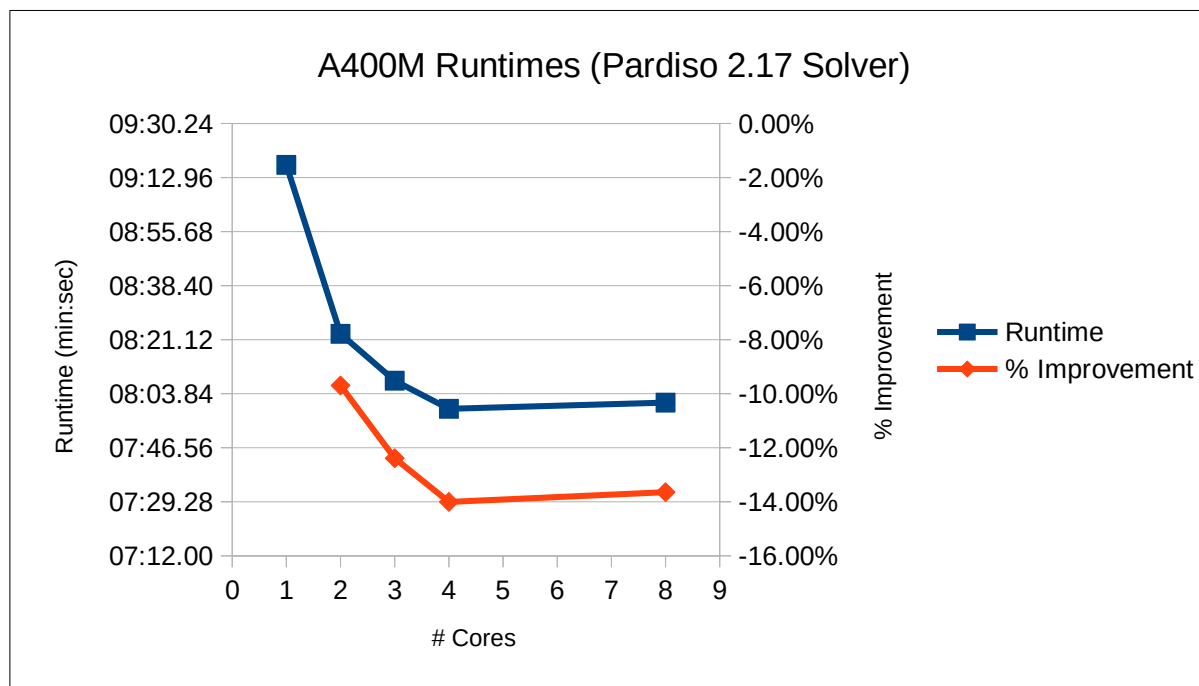


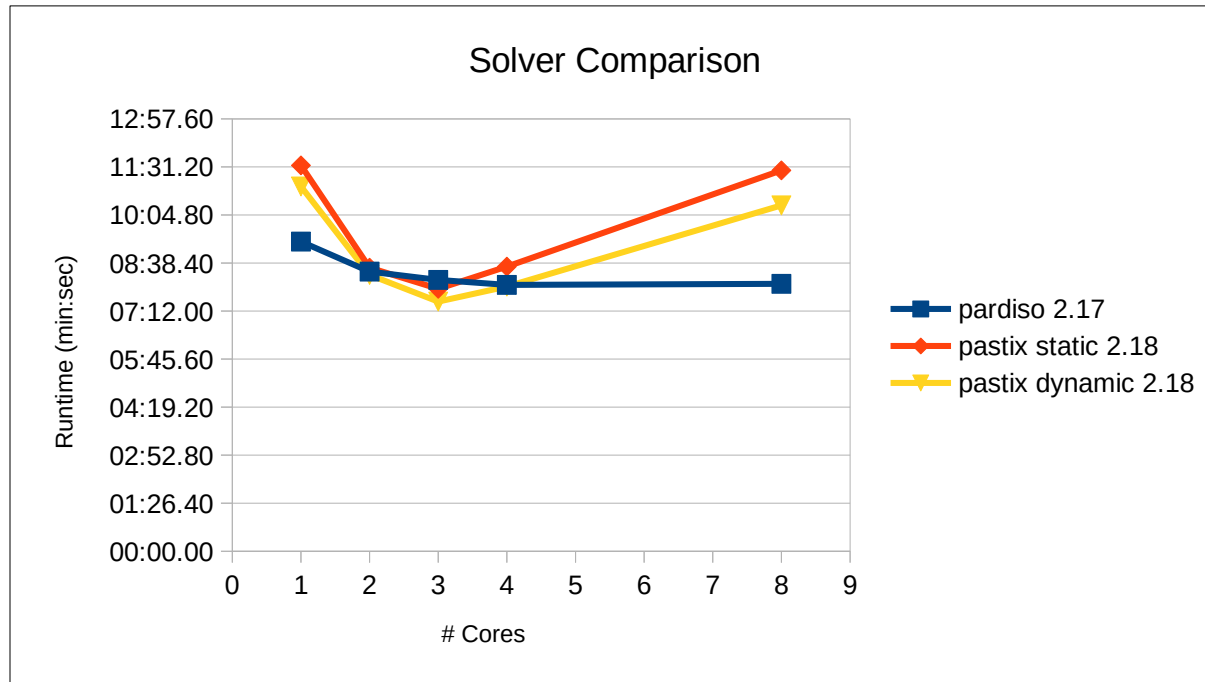
cores	Runtime	% Improvement
1	09:17.00	
2	08:23.00	-9.69%
3	08:08.00	-12.39%
4	07:59.00	-14.00%
8	08:01.00	-13.64%

4 cores  
 8 threads  
 11<sup>th</sup> gen Intel Core i5-1135G7  
 2.0-4.2 GHz  
 speed decreases with core usage  
 speed highest using one core  
 avx2 and avx-512 supported  
 10 iterations



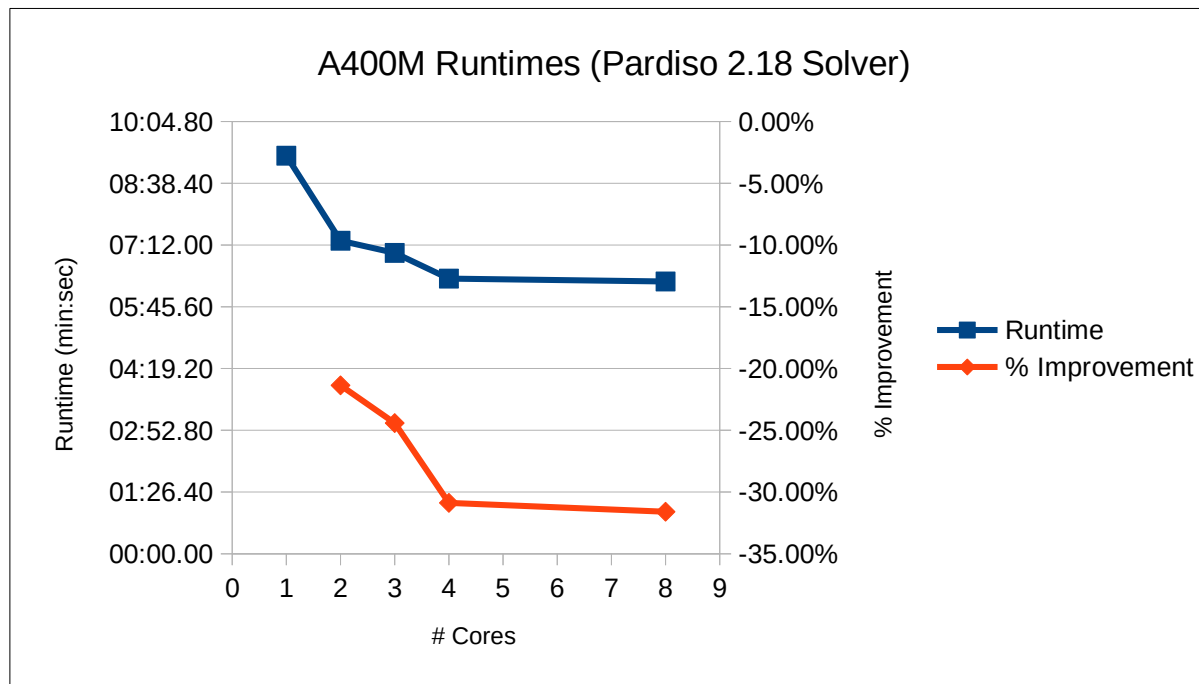
wait for results to transfer back to mecway, before reading solution time  
 spooles won't run this model  
 add copy of mkl\_intel\_thread.1.dll renamed to mkl\_intel\_thread.dll  
 add mkl\_sequential.1.dll

cores	pardiso 2.17	pastix static 2.18	pastix dynamic 2.18
1	09:17.00	11:34.00	10:56.00
2	08:23.00	08:30.00	08:17.00
3	08:08.00	07:52.00	07:29.00
4	07:59.00	08:32.00	07:56.00
8	08:01.00	11:25.00	10:22.00



cores	Runtime	% Improvement
1	09:17.00	
2	07:18.00	-21.36%
3	07:01.00	-24.42%
4	06:25.00	-30.88%
8	06:21.00	-31.60%

4 cores  
 8 threads  
 11<sup>th</sup> gen Intel Core i5-1135G7  
 2.0-4.2 GHz  
 speed decreases with core usage  
 speed highest using one core  
 avx2 and avx-512 supported  
 10 iterations



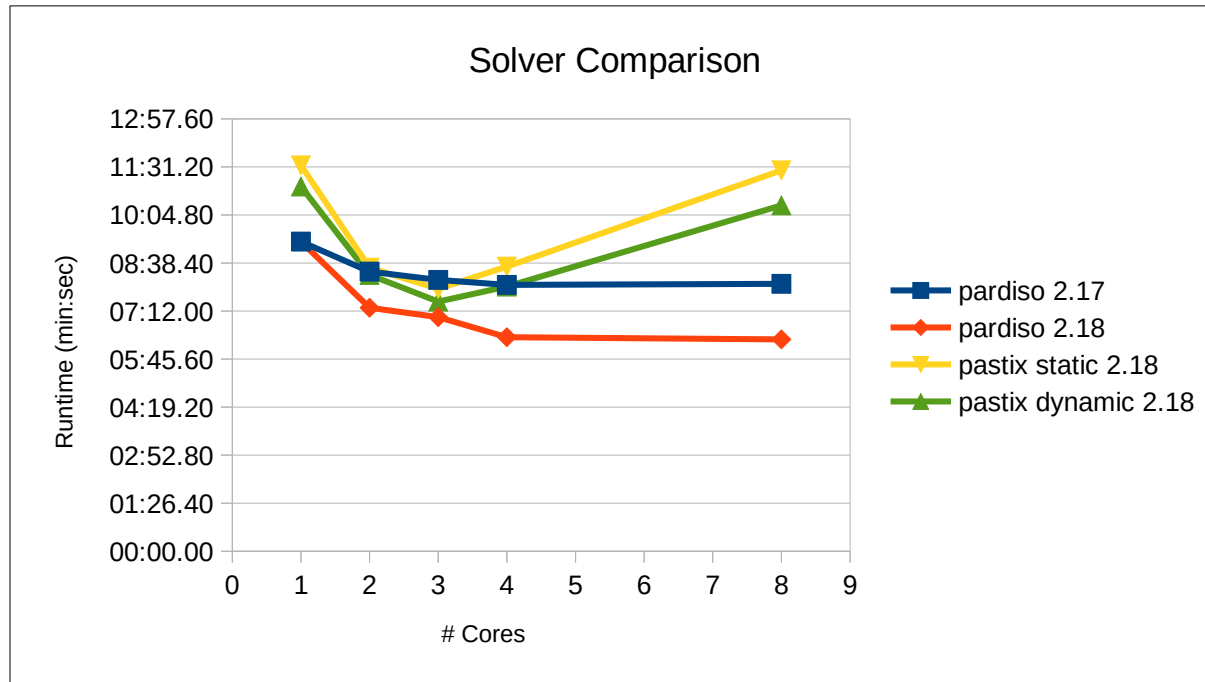
wait for results to transfer back to mecway, before reading solution time  
 spooler won't run this model

**This sheet has results using the following env vars:**

MKL\_INTERFACE\_LAYER=LP64  
 MKL\_THREADING\_LAYER=INTEL

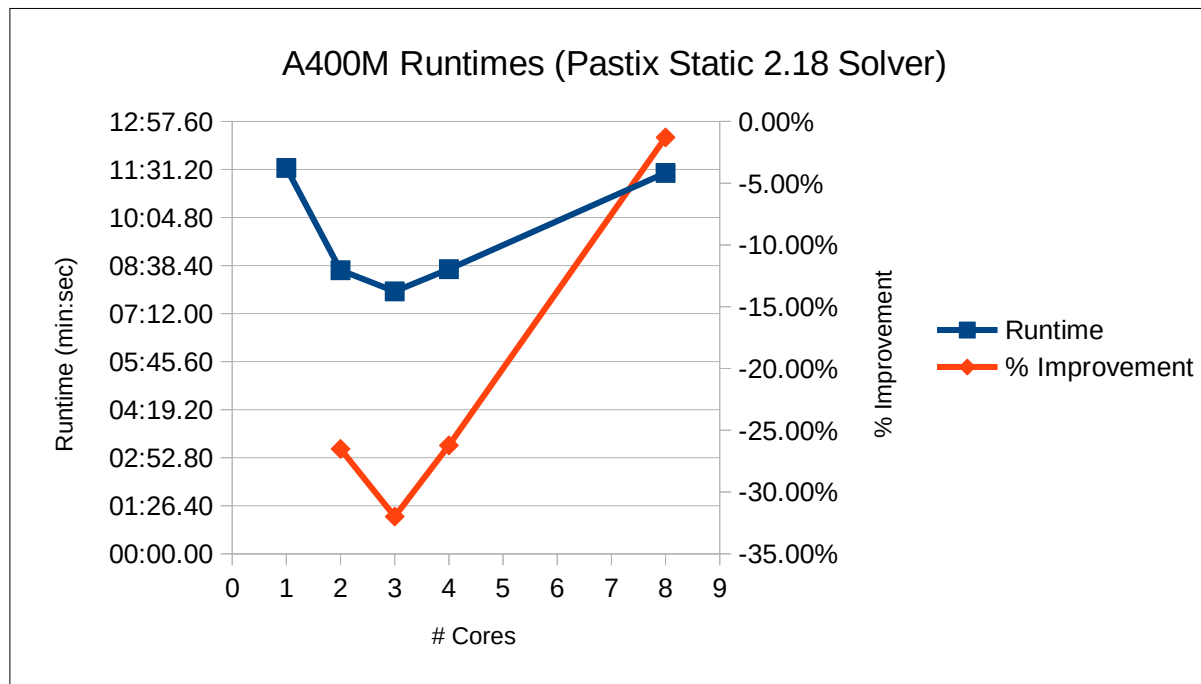
confirmed adding mkl\_avx512.1.dll provides a big speedup

cores	pardiso 2.17	pardiso 2.18	pastix static 2.18	pastix dynamic 2.18
1	09:17.00	09:17.00	11:34.00	10:56.00
2	08:23.00	07:18.00	08:30.00	08:17.00
3	08:08.00	07:01.00	07:52.00	07:29.00
4	07:59.00	06:25.00	08:32.00	07:56.00
8	08:01.00	06:21.00	11:25.00	10:22.00



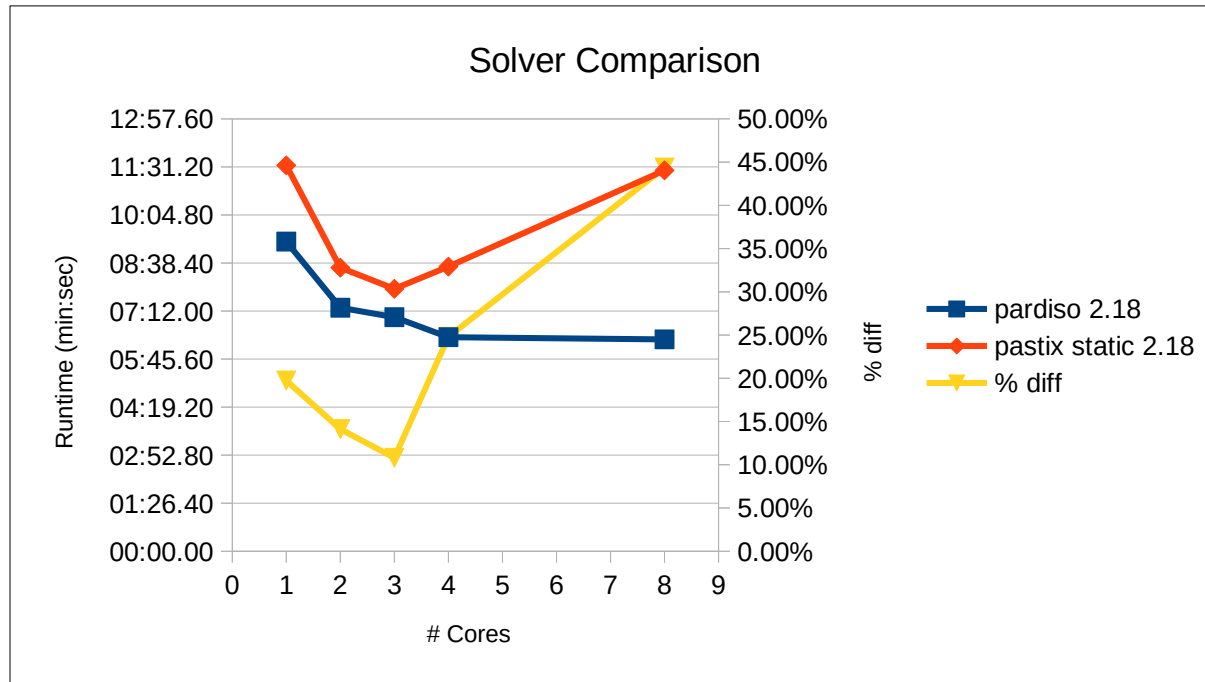
cores	Runtime	% Improvement
1	11:34.00	
2	08:30.00	-26.51%
3	07:52.00	-31.99%
4	08:32.00	-26.22%
8	11:25.00	-1.30%

4 cores  
 8 threads  
 11<sup>th</sup> gen Intel Core i5-1135G7  
 2.0-4.2 GHz  
 speed decreases with core usage  
 speed highest using one core  
 avx2 and avx-512 supported  
 10 iterations



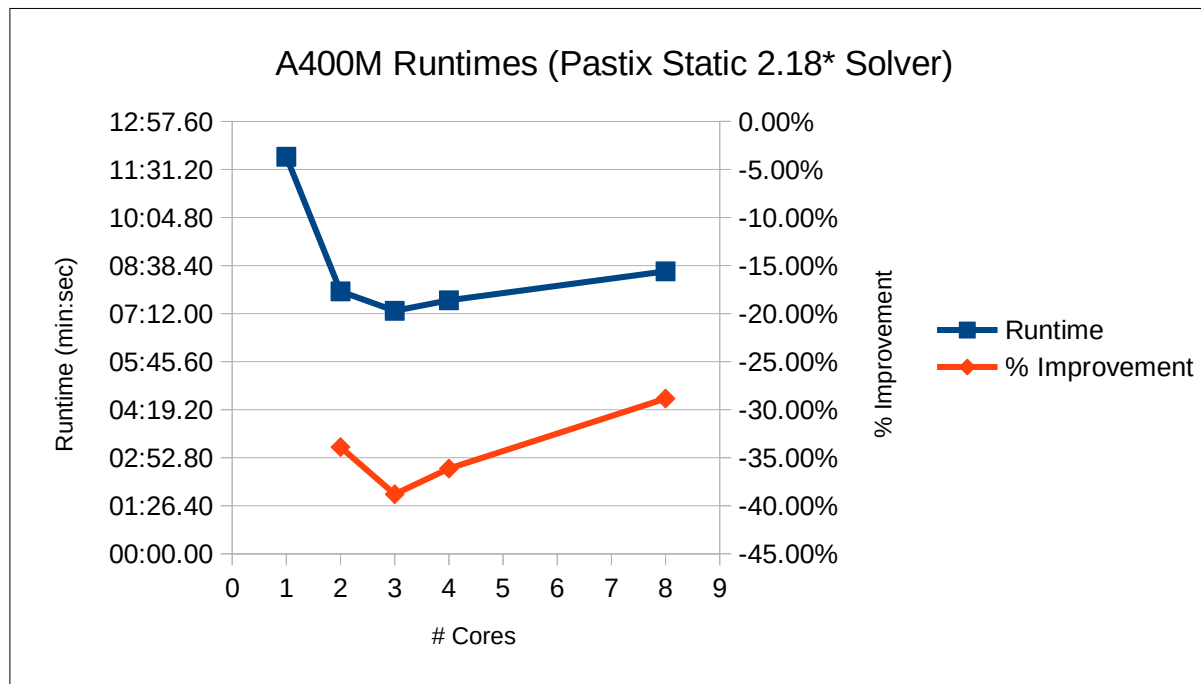
wait for results to transfer back to mecway, before reading solution time  
 spooles won't run this model  
 pastix using 1-2 more cores than allotted

cores	pardiso 2.18	pastix static 2.18	% diff
1	09:17.00	11:34.00	19.74%
2	07:18.00	08:30.00	14.12%
3	07:01.00	07:52.00	10.81%
4	06:25.00	08:32.00	24.80%
8	06:21.00	11:25.00	44.38%



cores	Runtime	% Improvement
1	11:54.00	
2	07:52.00	-33.89%
3	07:17.00	-38.80%
4	07:36.00	-36.13%
8	08:28.00	-28.85%

4 cores  
 8 threads  
 11<sup>th</sup> gen Intel Core i5-1135G7  
 2.0-4.2 GHz  
 speed decreases with core usage  
 speed highest using one core  
 avx2 and avx-512 supported  
 10 iterations



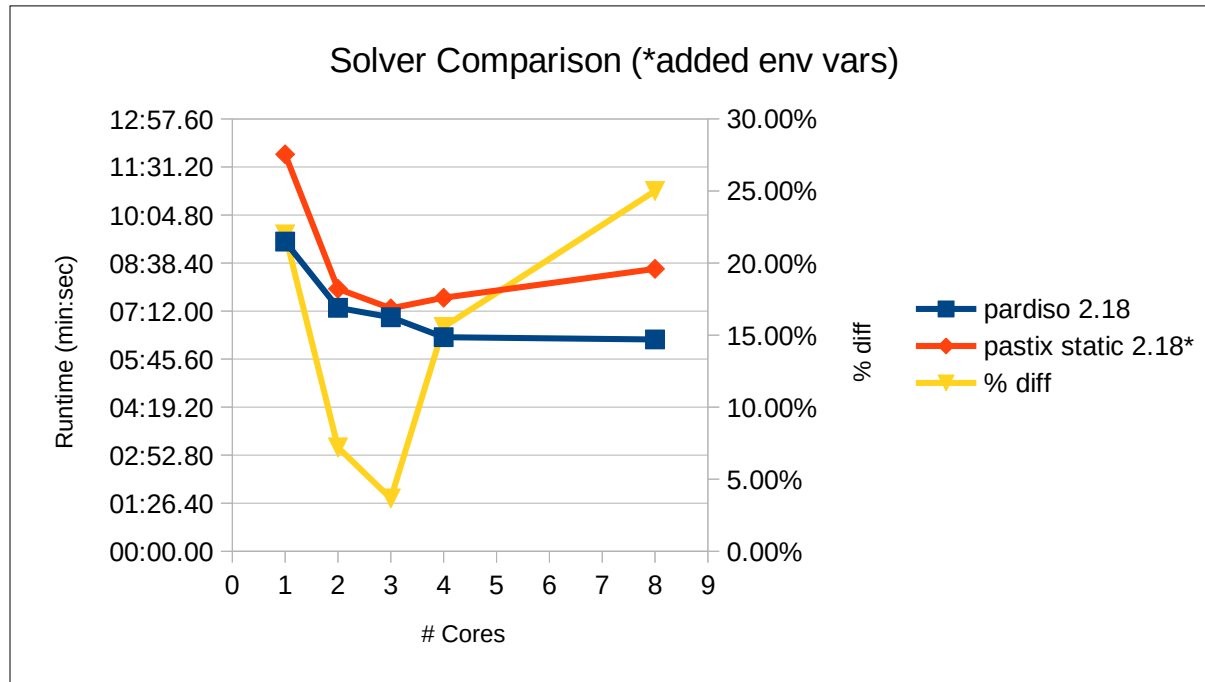
wait for results to transfer back to mecway, before reading solution time  
 spooles won't run this model

**This sheet has results using the following env vars:**

OPENBLAS\_NUM\_THREADS = 1  
 PASTIX\_MIXED\_PRECISION = 1

using the above two env vars stopped pastix from exceeding allotted cores

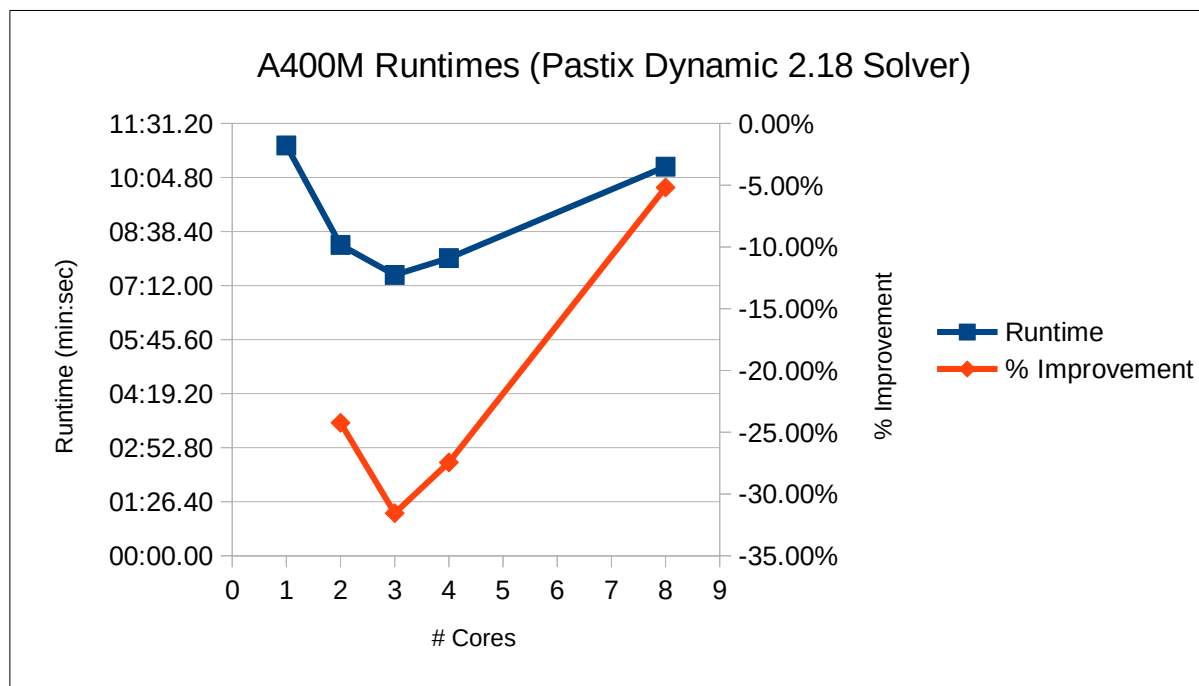
cores	pardiso 2.18	pastix static 2.18*	% diff
1	09:17.00	11:54.00	21.99%
2	07:18.00	07:52.00	7.20%
3	07:01.00	07:17.00	3.66%
4	06:25.00	07:36.00	15.57%
8	06:21.00	08:28.00	25.00%





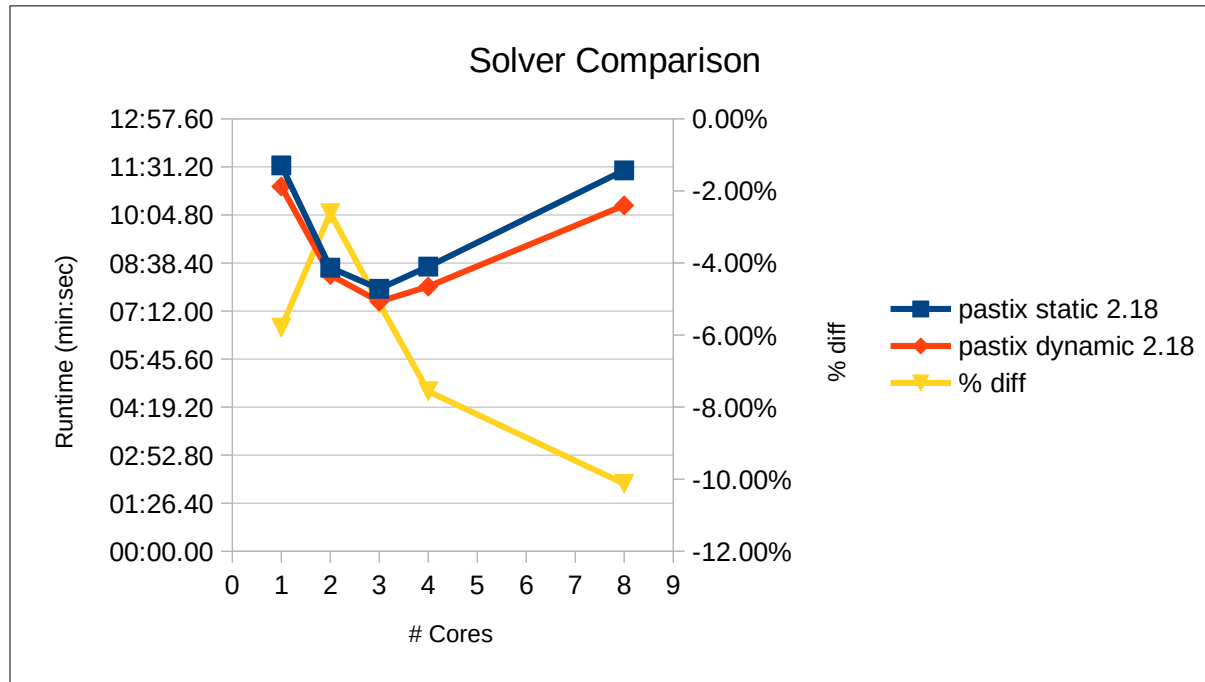
cores	Runtime	% Improvement
1	10:56.00	
2	08:17.00	-24.24%
3	07:29.00	-31.55%
4	07:56.00	-27.44%
8	10:22.00	-5.18%

4 cores  
 8 threads  
 11<sup>th</sup> gen Intel Core i5-1135G7  
 2.0-4.2 GHz  
 speed decreases with core usage  
 speed highest using one core  
 avx2 and avx-512 supported  
 10 iterations

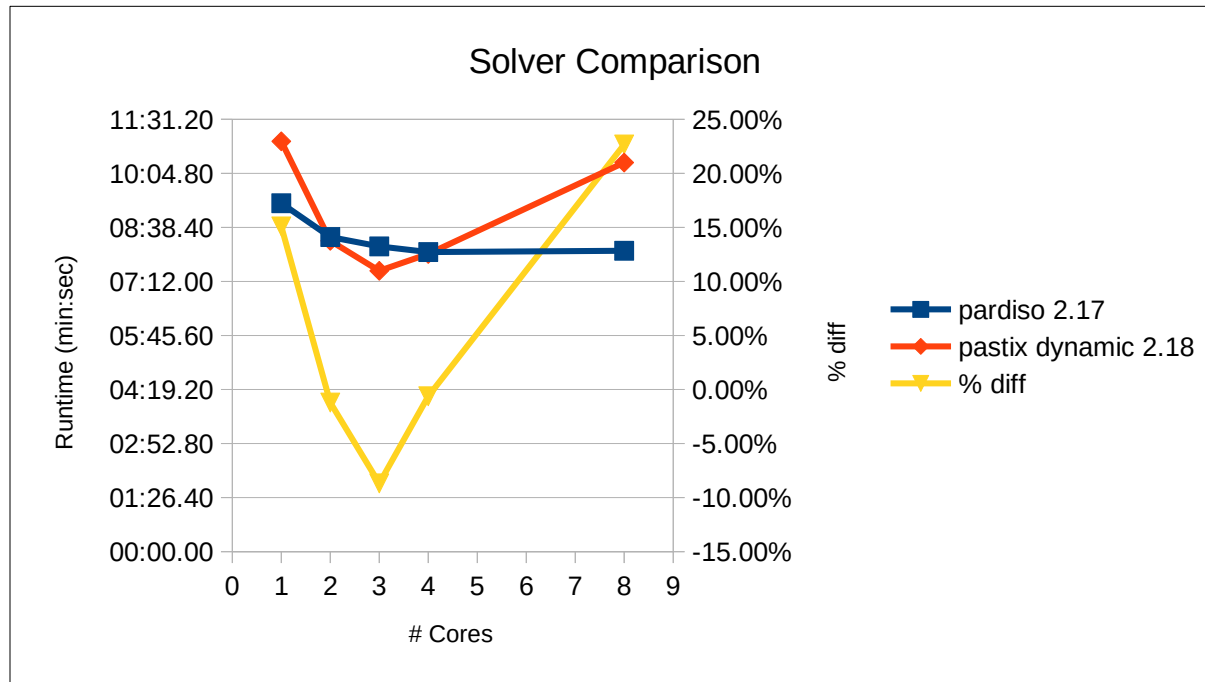


wait for results to transfer back to mecway, before reading solution time  
 spooler won't run this model  
 pastix using 1-2 more cores than allotted

cores	pastix static 2.18	pastix dynamic 2.18	% diff
1	11:34.00	10:56.00	-5.79%
2	08:30.00	08:17.00	-2.62%
3	07:52.00	07:29.00	-5.12%
4	08:32.00	07:56.00	-7.56%
8	11:25.00	10:22.00	-10.13%



cores	pardiso 2.17	pastix dynamic 2.18	% diff
1	09:17.00	10:56.00	15.09%
2	08:23.00	08:17.00	-1.21%
3	08:08.00	07:29.00	-8.69%
4	07:59.00	07:56.00	-0.63%
8	08:01.00	10:22.00	22.67%

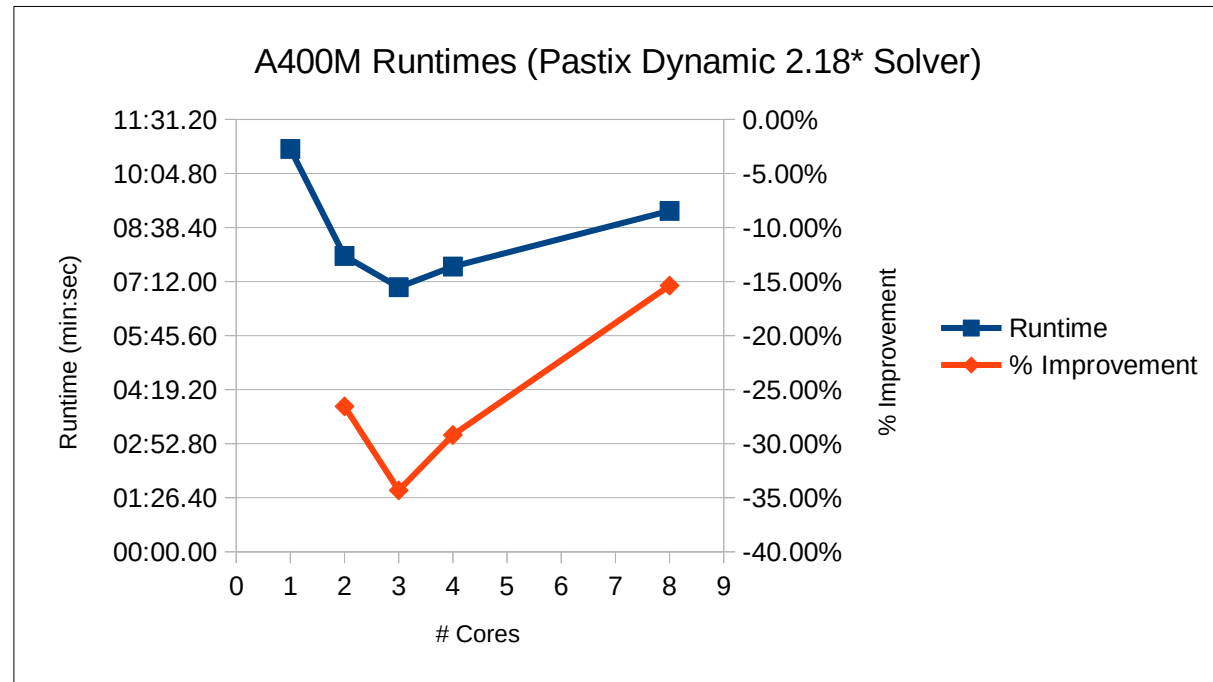


cores	Runtime	% Improvement
1	10:44.00	
2	07:53.00	-26.55%
3	07:03.00	-34.32%
4	07:36.00	-29.19%
8	09:05.00	-15.37%

4 cores  
8 threads  
11<sup>th</sup> gen Intel Core i5-1135G7  
2.0-4.2 GHz  
speed decreases with core usage  
speed highest using one core  
avx2 and avx-512 supported

10 iterations

3	06:42.00
3	06:57.00



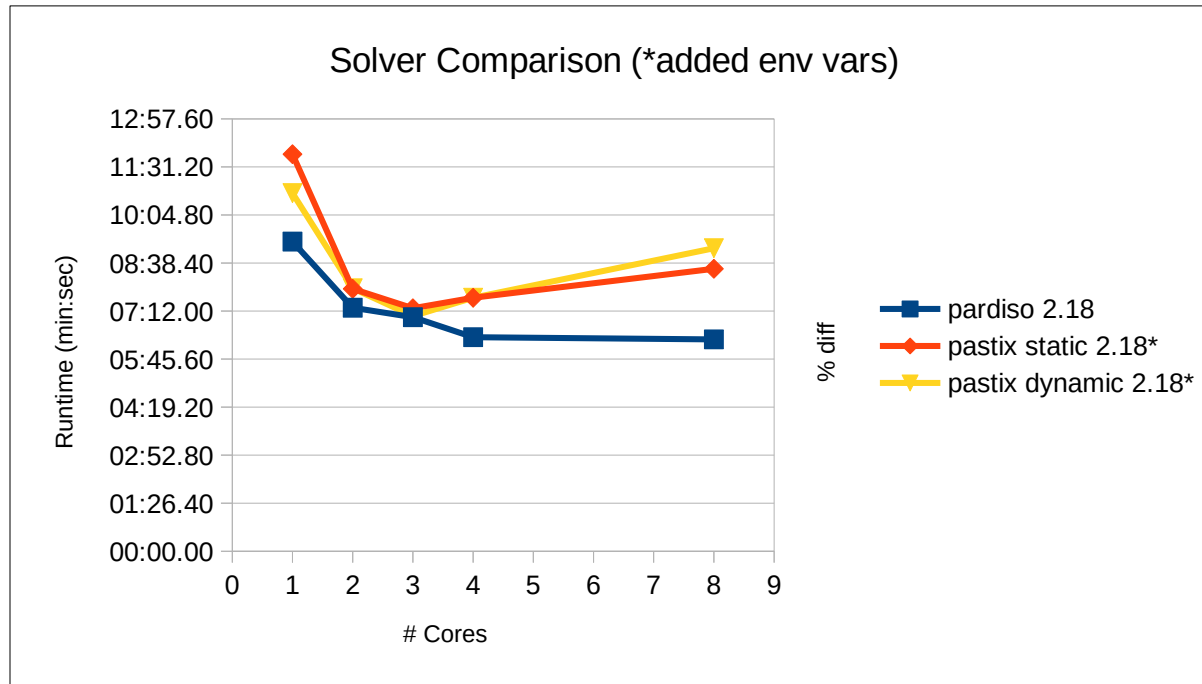
wait for results to transfer back to mecway, before reading solution time  
spooles won't run this model

**This sheet has results using the following env vars:**

OPENBLAS\_NUM\_THREADS = 1  
PASTIX\_MIXED\_PRECISION = 1

using the above two env vars stopped pastix from exceeding allotted cores

cores	pardiso 2.18	pastix static 2.18*	pastix dynamic 2.18*
1	09:17.00	11:54.00	10:44.00
2	07:18.00	07:52.00	07:53.00
3	07:01.00	07:17.00	07:03.00
4	06:25.00	07:36.00	07:36.00
8	06:21.00	08:28.00	09:05.00



The added env vars fix the problem of the ccx exe file using more than the allotted cores. However, there is still mystery core usage at times. For instance, when running 4 cores (50% usage) there will be 100% usage. The additional 50% is not attributed to any running app. So when you specify 8 cores (100% usage), pastix isn't able to run at 100%. This is most likely due to the mystery usage. When you run pastix at 4 cores it can run at 50% usage. So the mystery usage is probably why it runs so poorly when 8 cores is specified. On the other hand, pardiso only uses the specified cores. Pardiso also does a better job with memory. It uses less memory in general and the memory scales with core usage. Pastix takes all the free memory and holds it throughout the solve. This happens regardless of core usage. Pardiso isn't doing as much multi-core work. It is often running single core. The iparm settings may have something to do with this. It might also explain why pardiso is currently faster. The cpu clock speed is inversly proportional to core usage.

cores	pardiso 2.17	pastix dynamic 2.18	% diff
1	09:17.00	10:44.00	13.51%
2	08:23.00	07:53.00	-6.34%
3	08:08.00	07:03.00	-15.37%
4	07:59.00	07:36.00	-5.04%
8	08:01.00	09:05.00	11.74%

