LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-SR-473211

Progress report on Nuclear Density project with Lawrence Livermore National Lab Year 2010

C. W. Johnson, P. Krastev, W. E. Ormand

March 11, 2011

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Progress report on Nuclear Density project with Lawrence Livermore National Lab
Year 2010

The main goal for year 2010 was to improve parallelization of the configuration interaction code BIGSTICK, co-written by W. Erich Ormand (LLNL) and Calvin W. Johnson (SDSU), with the parallelization carried out primarily by Plamen Krastev, a postdoc at SDSU and funded in part by this grant. The central computational algorithm is the Lanczos algorithm, which consists of a matrix-vector multiplication (matvec), followed by a Gram-Schmidt reorthogonalization.

Since the matvec operation is the most time consuming part of the Lanczos algorithm, its efficient parallelization is essential. We have succeeded in parallelizing the matvec operation using MPI, with very good strong scaling up to 10,000 cores. BIGSTICK is designed with the ability to calculate ahead of time the number of operations. We have been working to distribute these operations equally across the cores, and we have succeeded to write an efficient algorithm to distribute the operations and tested scaling up to 10,000 cores.

We have created subroutines for calculating one-body density matrices. These are important for transitions. The code already outputs wavefunctions in a form suitable for postprocessing by a Livermore code, TRDENS, but having this capability inline will be very useful for quick calculations.

In August 2010 Dr. Ormand came down to SDSU for 4 days to discuss progress. We analyzed the efficiency of our parallel MPI distribution. We also investigated ordering our "jumps," the idea being that by improved spatial locality we could improve performance. We wrote two independent algorithms. To our disappointment, while the routines worked and sorted the jumps, no perceptible improvement in performance was achieved.

We have also investigate the distribution of the Lanczos sectors (parts of the vector) and found out that some cores would need almost the entire vector. Ultimately, one aims to have only several sectors on each core so that larger problems can be solved. With this goal in mind, we have redesigned our distribution algorithm by imposing an additional constraint on the allowed local dimension of the Lanczos vectors on each core. Our new distribution algorithm balances two constraints – the number of operations and the local dimension of the Lanczos vectors.

We have succeeded in implementing distributed (broken) Lanczos vectors. Each core only acts on a portion of the Lanczos vector; thus, to save memory, one stores only the relevant part of the Lanczos vector on each core. This means that every vector on every core must be addressed differently. Furthermore, when collecting components of Lanczos vectors, great care must be taken in communication.

We have employed an algorithm using separate MPI communication groups for each Lanczos sector, as well as specific offsets for the Lanczos sectors on each core. This allows us to orchestrate the Lanczos algebra in terms of the Lanczos sectors and reduce the dimension of Lanczos vectors on each core by, at least, a factor of three. We have also rearranged the calculation of specific observables and the Lanczos reorthogonalization in terms of distributed Lanczos vectors. For the near future, we have scheduled to perform target calculations with the latest version of BIGSTICK.