Summary:
* Major rewrite of REDSTICK CI shell-model code
* Significant improvement in performance and potential
* Hosted mini-workshop on Leadership-class CI codes at SDSU
* Physics applications currently underway

-> two applications:
* study of errors arising from model space truncation
* exact vs. Fermi gas calculation of level densities
Shell-model CI codes and applications

Past year’s work on REDSTICK:

July 2009: P. Krastev joined effort
August 2009: Analysis of performance of REDSTICK and modeling parallelization algorithms
-> need to reorganize representation of “jumps”

Krastev: modeled parallel distribution of jumps
Johnson & Ormand: major rewrite of code

Feb 2009: revised 2-body version of code finished
   -- set-up is 10x faster than old version
March 2009: further improvements:
   -- Hamiltonian application is at least 2x faster than old version
      (fixed bottleneck from old version)
   -- began implementation of new parallelization scheme
April 2009: Implemented thick-restart Lanczos
Shell-model CI codes and applications

Key idea for on-the-fly algorithms for the shell model

Represent an area by its boundary

→ Factorization of problem

→ Reduces memory load

\[ |\alpha\rangle = |\alpha_p\rangle \times |\alpha_n\rangle \]

Proton SDs

Neutron SDs

Hamiltonian can be factorized in same way
Shell-model CI codes and applications

Be9, $N_{\text{manx}} = 6$

distribution of operations over 200 nodes

REDSTICK $\rightarrow$ BIGSTICK

UNEDF Pack Forest Meeting June 2009
Shell-model CI codes and applications

Why “on-the-fly“?

Factorization of Hamiltonian -> reduced memory
-> larger problem on same machine

Comparison of RAM requirements (2-body interactions only)
Does not include lanczos vector storage

<table>
<thead>
<tr>
<th>Nuclide</th>
<th>Space</th>
<th>Basis dim</th>
<th>matrix store</th>
<th>on-the-fly</th>
</tr>
</thead>
<tbody>
<tr>
<td>$^{56}$Fe</td>
<td>$pf$</td>
<td>501 M</td>
<td>290 Gb</td>
<td>0.72 Gb</td>
</tr>
<tr>
<td>$^7$Li</td>
<td>$N_{\text{max}}=12$</td>
<td>252 M</td>
<td>3600 Gb</td>
<td>96 Gb</td>
</tr>
<tr>
<td>$^7$Li</td>
<td>$N_{\text{max}}=14$</td>
<td>1200 M</td>
<td>23 Tb</td>
<td>624 Gb</td>
</tr>
<tr>
<td>$^{12}$C</td>
<td>$N_{\text{max}}=6$</td>
<td>32M</td>
<td>196 Gb</td>
<td>3.3 Gb</td>
</tr>
<tr>
<td>$^{12}$C</td>
<td>$N_{\text{max}}=8$</td>
<td>590M</td>
<td>5000 Gb</td>
<td>65 Gb</td>
</tr>
<tr>
<td>$^{12}$C</td>
<td>$N_{\text{max}}=10$</td>
<td>7800M</td>
<td>111 Tb</td>
<td>1.4 Tb</td>
</tr>
<tr>
<td>$^{16}$O</td>
<td>$N_{\text{max}}=6$</td>
<td>26 M</td>
<td>142 Gb</td>
<td>3.0 Gb</td>
</tr>
<tr>
<td>$^{16}$O</td>
<td>$N_{\text{max}}=8$</td>
<td>990 M</td>
<td>9700 Gb</td>
<td>130 Gb</td>
</tr>
</tbody>
</table>
### Shell-model CI codes and applications

#### Comparison of RAM requirements (3-body interactions) - Estimate

<table>
<thead>
<tr>
<th>Nuclide</th>
<th>Space</th>
<th>Basis dim</th>
<th>store</th>
<th>on the fly</th>
</tr>
</thead>
<tbody>
<tr>
<td>$^7\text{Li}$</td>
<td>$N_{\text{max}}=12$</td>
<td>252 M</td>
<td>100 Tb</td>
<td>2.6 Tb</td>
</tr>
<tr>
<td>$^7\text{Li}$</td>
<td>$N_{\text{max}}=14$</td>
<td>1200 M</td>
<td>760 Tb</td>
<td>20 Tb</td>
</tr>
<tr>
<td>$^{12}\text{C}$</td>
<td>$N_{\text{max}}=6$</td>
<td>32M</td>
<td>4 Tb</td>
<td>0.07 Tb</td>
</tr>
<tr>
<td>$^{12}\text{C}$</td>
<td>$N_{\text{max}}=8$</td>
<td>590M</td>
<td>180 Tb</td>
<td>3 Tb</td>
</tr>
<tr>
<td>$^{12}\text{C}$</td>
<td>$N_{\text{max}}=10$</td>
<td>7800M</td>
<td>5000 Tb</td>
<td>86 Tb</td>
</tr>
</tbody>
</table>

1 Tb requires approximately 1000 cores (depending on architecture)
Shell-model CI codes and applications

In March 2009 we hosted a mini-workshop on Leadership Class CI codes at SDSU

Attendees:
Johnson, Krastev, Ormand
Vary, Maris
Navratil
Horoi
Ng, Yang

An excellent chance to “look under the hood” of our algorithms and share ideas for next-generation calculations; these discussions have led to mutual improvements in our CI codes cooperation not competition!
Shell-model CI codes and applications

Preliminary physics applications (serial so far):

* Comparison of exact results with pairing approximations in pf shell (with N. Sandulescu; ran full $^{52}$Fe in 30 hrs)
* Calculation of sd shell nuclei in 2hw space to study effects of model space truncation (with G. Bertsch; ran many cases with 20-50M states, approx 1-2 days each)
* EFT in atomic gases (with I. Stetcu et al; 4 particles up to 16hw)

Applications this fall:
* Electric polarizability in A = 6 (with I. Stetcu et al)
* $p$-shell nuclides (P. Navratil)
* Isospin breaking and CKM unitarity in pf-shell nuclei (EO+CJ)
Shell-model CI codes and applications

UNEDF Pack Forest Meeting June 2009
New application: effects of shell model space truncation (with George Bertsch)

**Motivation:** Shell-model calculations work in finite many-body space, using either phenomenological fits (a la Alex Brown) or renormalization scheme. Such truncations induce many-body interactions which may not be part of the fit—thus the rms error has an intrinsic limitation.

**Methodology:** Generate “exact” spectrum in large model space \((sd + 2p-2h pf + 1p-1h sdg)\); adjust USD interaction in \(sd\) shell in least-squares fit of “exact” spectrum. **How much can we reduce the rms error?**
Shell-model CI codes and applications

New application: effects of shell model space truncation

\[ \hat{H} \rightarrow \hat{H} + \Delta \hat{H} \]

Do least-squares fit using SVD

rms error in g.s. energies

rms error in excitation energies

monopole vs contact

rms residual (MeV)

SV fit dimension

SV fit dimension
Lessons learned:

As expected, one cannot exactly reproduce the “full space” calculation using only two-body interactions in the smaller space (many-body forces induced by truncation)

Least-squared fit dominated by a handful of interactions

Two “simple” forces do this most of the renormalization:
either
* isoscalar contact interaction (best); or
* $N(N+1)$ (“monopole”)
Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density

Motivation: Many approaches to nuclear level density begin with Bethe’s Fermi gas model, usually with strong, phenomenological corrections. How good/bad is it?

Methodology: Compute exact level (state) density with CI code. Using same model space & input Hamiltonian, compute HF s.p.e.s and compute Fermi gas partition function:

\[ \ln Z(\alpha, \beta) = \sum_i \ln(1 + \exp(\alpha - \beta \varepsilon_i)) \]

Invert Laplace transform to get density of states
Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density

\[ {^32}_S \]

![Graph showing state density vs. excitation energy for \(^{32}\)S](image)

UNEDF Pack Forest Meeting June 2009
Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density
Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density

What if nuclei aren’t spherical?
Add rotational partition function with parameters from cranked HF

$$Z_{rot} = \sum_{j} (2J + 1) a_j^2 \exp(-\beta E_j) \approx \frac{\sqrt{\pi J}}{(1 + \beta E_{rot})^{3/2}} , \bar{J} = \sqrt{\langle J(J + 1) \rangle} , \bar{E}_{rot} = \frac{\bar{J}^2}{2I}$$
Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density

$^{24}\text{Mg}$

![Graph showing state density versus excitation energy](image)

- exact (CI shell model)
- deformed s.p.e.s
- deformed s.p.e.s + rotation

UNEDF Pack Forest Meeting June 2009
Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density

![Graph showing state density vs. excitation energy for 45 Ti]
Shell-model CI codes and applications

New application: tests of a cheap method to compute the level density

Method requires further investigation / validation:

Are other corrections (pairing/vibration) needed?
What about strong shape coexistence?
What about spin-cutoff factor?
What about spurious c.m. motion?

Nonetheless, a promising, cheap method to get level densities