The No-Core Shell Model and High-Performance Computing

Pieter Maris

Dept. of Physics and Astronomy
Iowa State University
Ames, IA 50011
pmaris@iastate.edu

28th Indian Summer School
Ab Initio methods in Nuclear Physics
Aug. 29 – Sept. 2, 2016, Prague, Czech Republic

Outline

Introduction

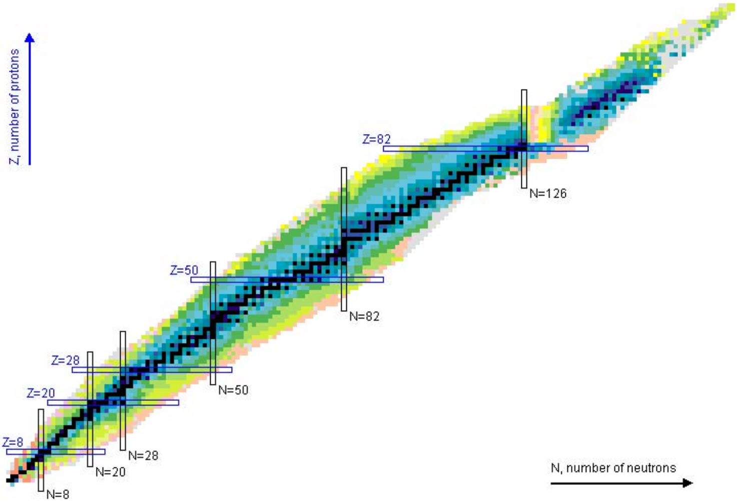
No-Core Configuration Interaction approach

High-Performance Computing

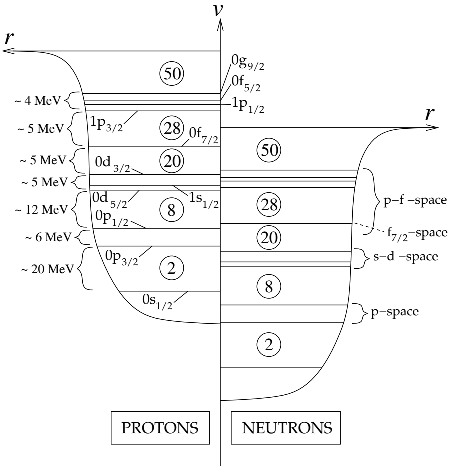
CI calculations on HPC platforms

For further reading

Nuclear Landscape: Valley of stability & Magic numbers

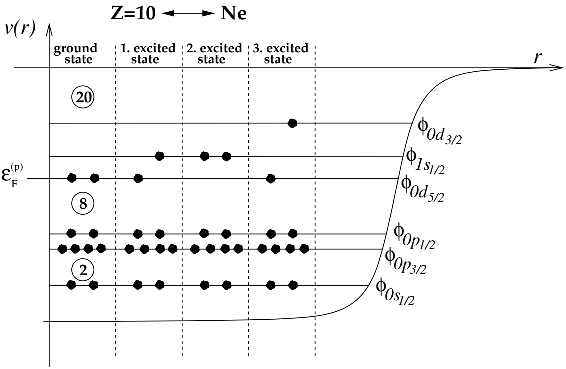


The nuclear mean field Shell Model



- Effective (phenomenological) potential for protons and neutrons

The nuclear mean field Shell Model



- Closed shell core
- Effective potential for non-interacting valence nucleons: Single-Particle energies ϵ
- Interactions between valence nucleons: Configuration mixing

The nuclear No-Core Shell Model

Given a Hamiltonian operator (in relative coordinates)

$$\hat{\mathbf{H}}_{\text{rel}} = \sum_{i<j} \frac{(\vec{p}_i - \vec{p}_j)^2}{2\, m\, A} + \sum_{i<j} V_{ij} + \sum_{i<j<k} V_{ijk} + \ldots$$

solve the eigenvalue problem for the bound state wavefunction of A nucleons

$$\hat{\mathbf{H}}_{\text{rel}}\, \Psi(r_1, \ldots, r_A) = \lambda\, \Psi(r_1, \ldots, r_A)$$

by expanding the $\Psi(r_1, \ldots, r_A)$ in products of harmonic oscillator (H.O.) single-particle states

The No-Core Shell Model: Plan for four lectures

1. NCSM and HPC
   - Remainder of this lecture
2. NN and 3N (effective) interactions
   - This afternoon
3. Emergence of collective motion
   (plus additional topics, depending on interests?)
   - Tomorrow
4. Beyond the H.O. basis
   (plus additional topics, depending on interests?)
   - Wednesday

Reminder: 3D quantum Harmonic Oscillator

- Hamiltonian $\hat{\mathbf{H}}^{\text{H.O.}} = \frac{p^2}{2m} + \frac{1}{2} m \omega^2 r^2$
- Solutions of Schrödinger equation $\hat{\mathbf{H}}^{\text{H.O.}} \phi(r) = E_N \, \phi(r)$

$$\phi(r) = \mathcal{N}_{nl} \left(\frac{r}{b}\right)^l \mathrm{e}^{-r^2/(2b^2)} \, L_n^{(l+\frac{1}{2})}(r^2/b^2) \; Y_{lm}(\theta, \phi)$$

  with H.O. parameter $b^2 = \hbar/(m\,\omega)$
- Eigenvalues $E_N = (\frac{3}{2} + N)\hbar\omega$ with $N = 2n + l$
- Nucleons: spin-$\frac{1}{2}$ particles, spin up or down
- Single-particle quantum number in LS-scheme: $|nlmss_z\rangle$
- Couple orbital motion and intrinsic spin (and suppress $s = \frac{1}{2}$) to J-scheme: $\hat{\mathbf{J}} = \hat{\mathbf{L}} + \hat{\mathbf{S}}$: $|nljm_j\rangle$
- Degeneracy of major (H.O. energy) shells $(N + 1)(N + 2)$
  - magic numbers 2, 8, 20, 40, ...
  - *s*-shell, *p*-shell, *sd*-shell, *pf*-shell, ...

No-Core Configuration Interaction approach

- Expand wavefunction in basis states $|\Psi\rangle = \sum a_i|\Phi_i\rangle$
- Express Hamiltonian in basis $\langle\Phi_j|\hat{\mathbf{H}}|\Phi_i\rangle = H_{ij}$
- Diagonalize Hamiltonian matrix H_{ij}
- No-Core: all A nucleons are treated the same
- Complete basis $\longrightarrow$ exact result
  - caveat: complete basis is infinite dimensional
- In practice
  - truncate basis
  - study behavior of observables as function of truncation
- Computational challenge
  - construct large $(10^{10} \times 10^{10})$ sparse symmetric matrix H_{ij}
  - obtain lowest eigenvalues & -vectors corresponding to low-lying spectrum and eigenstates

NCCI approach – Basis expansion

- Expand A-body wave function in basis functions

$$\Psi(r_1, \ldots, r_A) = \sum a_i \Phi_i(r_1, \ldots, r_A)$$

- Use basis of single Slater Determinants of single-particle states

$$\Phi_i(r_1, \ldots, r_A) \;=\; \frac{1}{\sqrt{(A!)}} \begin{vmatrix} \phi_{i1}(r_1) & \phi_{i2}(r_1) & \ldots & \phi_{iA}(r_1) \\ \phi_{i1}(r_2) & \phi_{i2}(r_2) & \ldots & \phi_{iA}(r_2) \\ \vdots & \vdots & & \vdots \\ \phi_{i1}(r_A) & \phi_{i2}(r_A) & \ldots & \phi_{iA}(r_A) \end{vmatrix}$$

  which takes care of anti-symmetrization (Fermion statistics)

- Each many-body basis state $\Phi(r_1, \ldots, r_A)$ eigenstate of $\hat{\mathbf{J}}_\mathbf{z}$, with fixed eigenvalue M, but not of $\hat{\mathbf{J}}^\mathbf{2}$ (M-scheme)
  - almost trivial to implement

$$\hat{\mathbf{J}}_\mathbf{z}|\Phi\rangle \;=\; M|\Phi\rangle \;=\; \sum_{k=1}^{A} m_{j_k}|\Phi\rangle$$

NCCI approach – Hamiltonian

- Expand Hamiltonian in basis

$$\hat{\mathbf{H}}_{\mathrm{rel}} = \sum_{i<j} \frac{(\vec{p}_i - \vec{p}_j)^2}{2\, m\, A} + \sum_{i<j} V_{ij} + \sum_{i<j<k} V_{ijk} + \ldots$$

- *A*-body problem with *a*-body interaction: nonzero matrix elements iff at least $(A - a)$ particles are in identical single-particle states
  - many-body basis states are single Slater Determinants
  - *A*-body problem with 2-body interactions

$$H^{(A)}_{ij} = (-1)^{\text{permutations}}\ \delta_{i_1, j_1} \ldots \delta_{i_{(A-2)}, j_{(A-2)}}\ \langle a\, b | \hat{\mathbf{H}} | c\, d \rangle$$

- Sparse symmetric eigenvalue problem

NCCI approach – Truncations

Complete space – exact (for bound states), but infinite-dimensional

- N_{max} truncation:
  Each many-body basis state $\Phi(r_1, \ldots, r_A)$ satisfies

$$\sum_{k=1}^{A} \left(2\, n_k + l_k\right) \;\leq\; N_0 + N_{\text{max}}$$

  - exact factorization of Center-of-Mass motion
- Alternatives:
  - FCI (commonly used in nuclear shell model, quantum chemistry, ...) truncation on single-particle basis states only
  - Importance Truncation — Roth, PRC79, 064324 (2009)
  - No-Core Monte-Carlo Shell Model — Abe *et al*, PRC86, 054301 (2012)
  - SU(3) Truncation — Dytrych *et al*, PRL111, 252501 (2013)
  - . . .

Intermezzo: Center-of-Mass motion

- Use single-particle coordinates, not relative (Jacobi) coordinates
  - straightforward to extend to many particles
  - have to separate Center-of-Mass motion from relative motion
- Center-of-Mass wavefunction factorizes for H.O. basis functions in combination with N_{max} truncation

$$\begin{aligned} |\Psi_{\text{total}}\rangle &= |\phi_1\rangle \otimes \ldots \otimes |\phi_A\rangle \\ &= |\Phi_{\text{Center-of-Mass}}\rangle \otimes |\Psi_{\text{rel}}\rangle \end{aligned}$$

where

$$\hat{\mathbf{H}}_{\text{rel}}|\Psi_{\text{i, rel}}\rangle = E_{\text{i}}|\Psi_{\text{i, rel}}\rangle$$

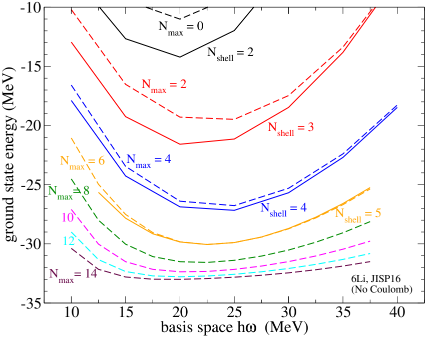
- Add Lagrange multiplier to Hamiltonian (Lawson term)

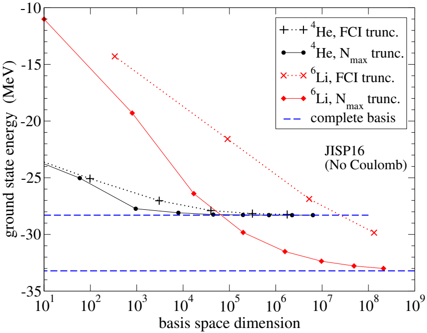
$$\hat{\mathbf{H}}_{\text{rel}} \longrightarrow \hat{\mathbf{H}}_{\text{rel}} + \Lambda_{\text{CM}}\left(\hat{\mathbf{H}}_{\text{CM}}^{\text{H.O.}} - \frac{3}{2}\hbar\omega\right)$$

with $\hat{\mathbf{H}}_{\text{rel}} = \hat{\mathbf{T}}_{\text{rel}} + \hat{\mathbf{V}}_{\text{rel}}$ the relative Hamiltonian

  - separates states with CM excitations from states with 0s CM motion $|\Phi_{\text{CM}}\rangle = |\Phi_{0s}\rangle$

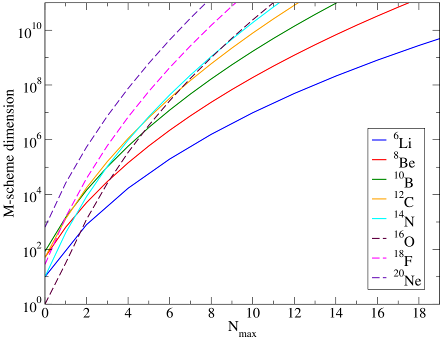
Intermezzo: FCI vs. Nmax truncation

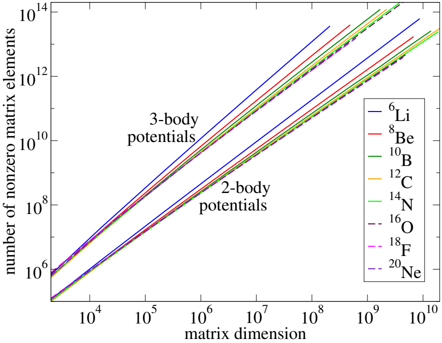




- N_{max} truncation
  - exact factorization of Center-of-Mass motion
- Infinite basis space limit
  - both N_{max} truncation and FCI converge to the same results
  - N_{max} truncation does so much more rapidly

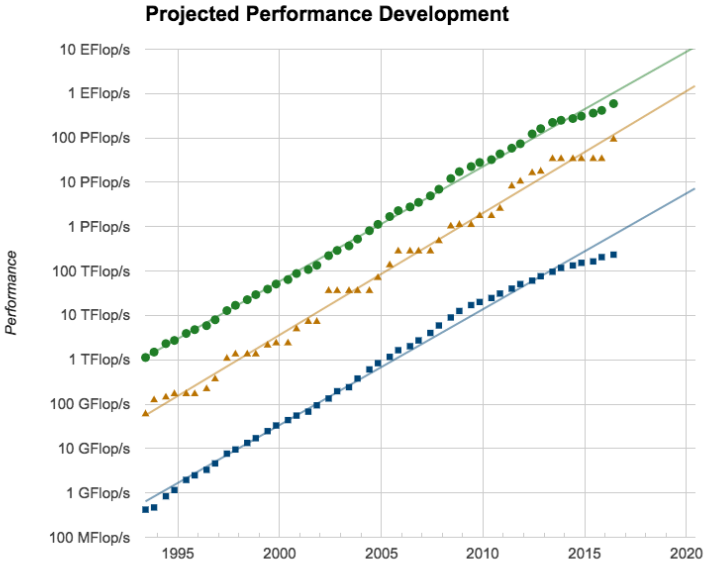
NCCI approach – Main Challenge





- Increase of basis space dimension with increasing A and N_{max}
  - need calculations up to at least $N_{\text{max}} = 8$ for meaningful extrapolation and numerical error estimates
- More relevant measure for computational needs
  - number of nonzero matrix elements
  - current limit 10^{13} to 10^{14} (Edison, Mira, Titan)

High-Performance Computing: Moore's law



High-Performance Computing

- Parallel computing
  - Initially: Shared memory or distributed memory parallel systems
  - Currently: Systems have shared and distributed memory
    - use OpenMP within a node and MPI between nodes
- Accelerators
  - GPU's (NVIDIA), Xeon Phi (Intel), . . .
  - Initially: as co-processor
  - Soon: self-hosted
- Vectorization
  - Xeon Phi (KNL) has 512-bit vector units (8 double precision floats)
- Increasing performance gap between processor and memory
  - Available memory and memory bandwidth per PU decreases
  - Data locality and data placement is crucial

Highly nontrivial to achieve good performance

- Need to collaborate with applied mathematicians and computer scientists

Many-Fermion Dynamics for nuclear structure

Configuration Interaction code for nuclear structure calculations

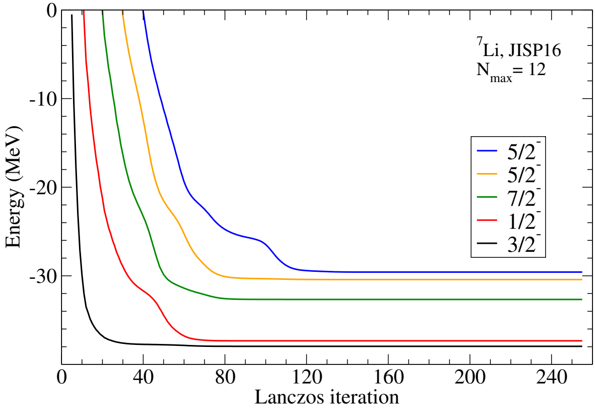
- Platform-independent, hybrid OpenMP/MPI, Fortran 90
- Generate many-body basis space
  subject to user-defined single-particle and many-body truncation
- Construct of many-body matrix H_{ij}
  - determine which matrix elements can be nonzero
    based on quantum numbers of underlying single-particle states
  - evaluate and store nonzero matrix elements
    in compressed sparse block format
- Obtain lowest eigenpairs using Lanczos algorithm (or LOBPCG)
  - typical use: 8 to 12 lowest eigenvalues and eigenvectors
  - typically need $\sim$ 300 to $\sim$ 500 Lanczos iterations
  - some applications need hundreds of eigenvalues
- Write eigenvectors (wavefunctions) to disk
- Calculate selected set of observables

Lanczos algorithm as implemented in MFDn

Let H be a symmetric matrix. Then H can be reduced to a symetric tri-diagonal matrix T via orthogonal unitary transformations, $H = Q_n T_n Q_n^T$

- For $i = 1$, set $\beta_1 = 0$ and initial vector q_1 with $||q_1|| = 1$
- While (not converged) do
  1. compute $p = H\,q_i$ i.e. perform Sparse Matrix-Vector Multiplication
  2. compute $\alpha_i = q_i^T \cdot (H\,q_i)$ i.e. perform dot-product
  3. compute $k = p - \alpha_i q_i - \beta_i q_{i-1}$
  4. (orthogonalize k w.r.t. q_i for numerical stability) more dot-products
  5. compute $\beta_{i+1} = ||k||$ and one more dot-products
  6. set $q_{i+1} = k/||k||$
  7. increment $i = i + 1$
  8. check (convergence) diagonalize small tridiagonal matrix
     - obtain eigen-values λ and -vectors v of T_n LAPACK
     - compute $\beta_i|\frac{v_i}{\lambda}|$ for each desired eigenvalue
- Compute approximate eigenvectors of H from T_n and Q_n

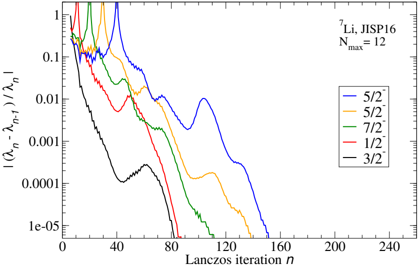
Lanczos algorithm

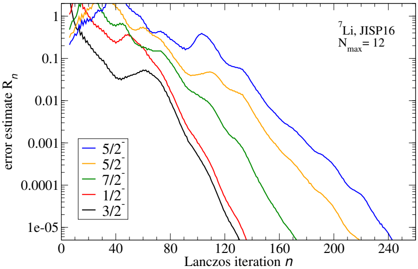


- dimension 252 million, with 400 billion nonzero matrix elements
- runs on 124 nodes Edison at NERSC using 496 MPI ranks with 6 OpenMP threads/MPI
- total runtime less than 10 minutes

- Lowest 5 eigenvalues of T_n after n Lanczos iterations
- Note: in MFDn we use single-precision for H and $Q = \{q_i\}$ but double-precision for dot-products and T_n for numerical stability

Lanczos algorithm – convergence criterium



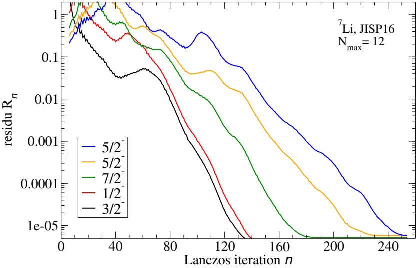


- Error estimate (cheap)

$$R_n \quad \approx \quad \beta_n \left| \frac{v_n}{\lambda} \right|$$

- Residu (expensive)

$$R^2 \quad = \quad \frac{\sum_i \left(\sum_j H_{ij} a_j - \lambda a_i\right)^2}{\lambda^2}$$



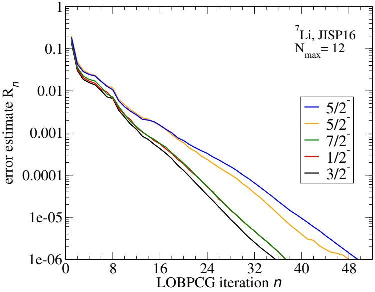
LOBPCG

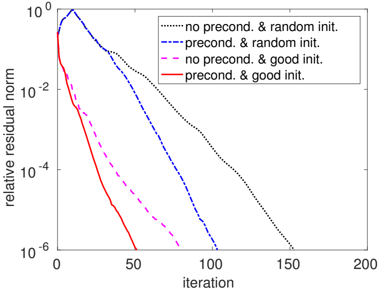
in collaboration with applied mathematicians from Berkeley

- Set initial guess for $X^{(1)}$ consisting of k orthonormal vectors
  - ideally, consisting of approximate eigenvectors
  - e.g. smaller basis space, different H.O. parameter $\hbar\omega$, . . .
- While (not converged) do
  1. apply preconditioner T
     - preconditioning is an art . . .
     - kinetic energy is likely to be efficient, but too expensive
     - diagonal matrix element is cheap, but not efficient
     - compromise:
       diagonal tiles of H, based on many-body (n, l, j) orbitals
  2. orthonormalize using Cholesky QR
  3. compute $H\,X^{(i)}$ Sparse Matrix-Matrix Multiplication
  4. do LOBPCG magic . . .
  5. check convergence
- $X^{(n)}$ consists of k orthonormal eigenvectors

LOBPCG

Shao, Aktulga, Yang, Ng, PM, Vary, to be submitted

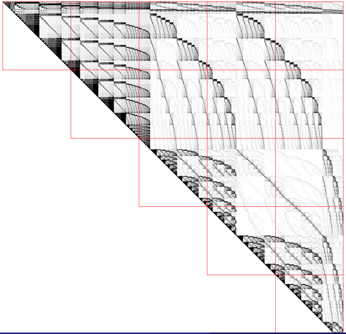


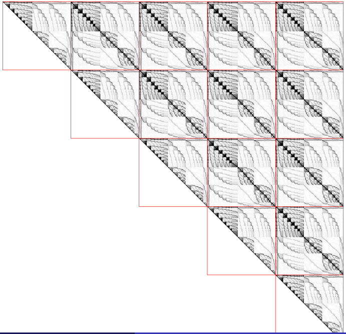


- Blocks of 8 (12) vectors, targeting lowest 5 (8) eigenstates
- $N_{\text{max}} = 8$: 114 iterations in 6.5 seconds (using random initial vectors)
- $N_{\text{max}} = 10$: 67 iterations in 19.8 seconds
- $N_{\text{max}} = 12$: 50 iterations in 109.4 seconds
- Despite doing approximately 1.6 times more work in SpMV/SpMM, LOBPCG factor of 2 faster than Lanczos

Efficient distributed SpMV

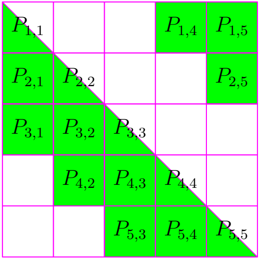
- Symmetric matrix
  - store only half the matrix (upper or lower triangle)
  - have to do SpMV and SpMV$^{\mathrm{T}}$ with same data structure
- Load-balancing
  - load determined by number of nonzero matrix elements
  - 2-dimensional distribution of matrix
  - round-robin distribution of (groups of) many-body states



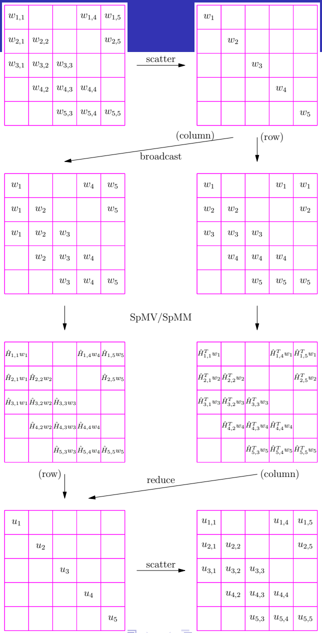


Efficient distributed SpMV

- Communication needs to be load-balanced as well
- Vectors distributed over all processors for orthogonalization



$P_{1,1}$			$P_{1,4}$	$P_{1,5}$
$P_{2,1}$	$P_{2,2}$			$P_{2,5}$
$P_{3,1}$	$P_{3,2}$	$P_{3,3}$		
	$P_{4,2}$	$P_{4,3}$	$P_{4,4}$	
		$P_{5,3}$	$P_{5,4}$	$P_{5,5}$



$w_{1,1}$			$w_{1,4}$	$w_{1,5}$
$w_{2,1}$	$w_{2,2}$			$w_{2,5}$
$w_{3,1}$	$w_{3,2}$	$w_{3,3}$		
	$w_{4,2}$	$w_{4,3}$	$w_{4,4}$	
		$w_{5,3}$	$w_{5,4}$	$w_{5,5}$

scatter

w_1				
	w_2			
		w_3		
			w_4	
				w_5

(column) broadcast (row)

w_1			w_4	w_5
w_1	w_2			w_5
w_1	w_2	w_3		
	w_2	w_3	w_4	
		w_3	w_4	w_5

w_1			w_1	w_1
w_2	w_2			w_2
w_3	w_3	w_3		
	w_4	w_4	w_4	
		w_5	w_5	w_5

SpMV/SpMM

$\hat{H}_{1,1}w_1$			$\hat{H}_{1,4}w_4$	$\hat{H}_{1,5}w_5$
$\hat{H}_{2,1}w_1$	$\hat{H}_{2,2}w_2$			$\hat{H}_{2,5}w_5$
$\hat{H}_{3,1}w_1$	$\hat{H}_{3,2}w_2$	$\hat{H}_{3,3}w_3$		
	$\hat{H}_{4,2}w_2$	$\hat{H}_{4,3}w_3$	$\hat{H}_{4,4}w_4$	
		$\hat{H}_{5,3}w_3$	$\hat{H}_{5,4}w_4$	$\hat{H}_{5,5}w_5$

$\hat{H}^T_{1,1}w_1$			$\hat{H}^T_{1,4}w_1$	$\hat{H}^T_{1,5}w_1$
$\hat{H}^T_{2,1}w_2$	$\hat{H}^T_{2,2}w_2$			$\hat{H}^T_{2,5}w_2$
$\hat{H}^T_{3,1}w_3$	$\hat{H}^T_{3,2}w_3$	$\hat{H}^T_{3,3}w_3$		
	$\hat{H}^T_{4,2}w_4$	$\hat{H}^T_{4,3}w_4$	$\hat{H}^T_{4,4}w_4$	
		$\hat{H}^T_{5,3}w_5$	$\hat{H}^T_{5,4}w_5$	$\hat{H}^T_{5,5}w_5$

(row) reduce (column)

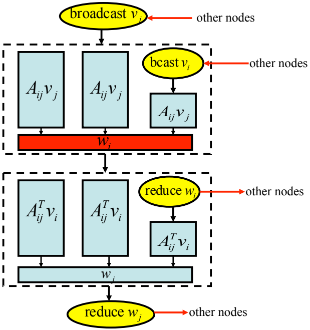
u_1				
	u_2			
		u_3		
			u_4	
				u_5

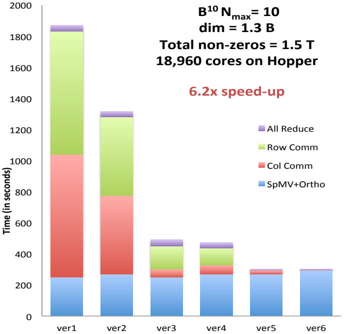
scatter

$u_{1,1}$			$u_{1,4}$	$u_{1,5}$
$u_{2,1}$	$u_{2,2}$			$u_{2,5}$
$u_{3,1}$	$u_{3,2}$	$u_{3,3}$		
	$u_{4,2}$	$u_{4,3}$	$u_{4,4}$	
		$u_{5,3}$	$u_{5,4}$	$u_{5,5}$

Efficient distributed SpMV – MPI communication

Aktulga, Yang, Ng, PM, Vary, Concurr. Comput. 26 (2014), doi:10.1002/cpe.3129

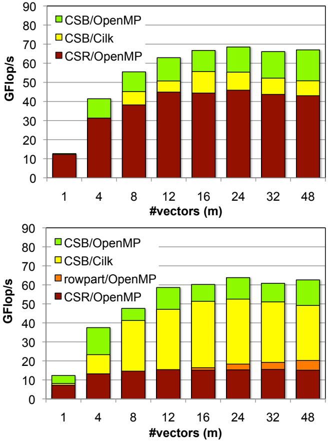




- Overlap communication with computation
- Optimize mapping onto network topology for non-overlapping communication

see also Oryspayev, PhD thesis 2016, ISU

Efficient distributed SpMV – single-node performance



Aktulga, Afibuzzaman, Williams, Buluç, Shao, Yang, Ng, PM, Vary, submitted for publication

- Compressed sparse block (CSB)
  - improves data locality and cache performance
  - allows for efficient OpenMP parallelization within nodes, avoiding race conditions for both SpMV and SpMV^{T}
- Block algorithm (LOBPCG)
  - SpMV on 'block of vectors' allows for vectorization
- Performance tuning for KNL in progress in collaboration with NERSC, Cray, and Intel

For further reading

- J. Suhonen,
  From Nucleons to Nucleus,
  Springer, 2007
- R.R. Whitehead, A. Watt, B.J. Cole and I. Morrison,
  Computational Methods for Shell Model Calculations,
  Adv. Nucl. Phys. **9**, 123 (1977).
- B.R. Barrett, P. Navrátil and J.P. Vary,
  Ab initio no core shell model,
  Prog. Part. Nucl. Phys. **69** (2013) 131.

For further reading – CI codes

- OXBASH/NuShellX
  B.A. Brown, A. Etchegoyen and W.D.M. Rae, *Computer code OXBASH: the Oxford University-Buenos Aires-MSU shell model code*, MSU Cyclotron Laboratory Report No. 524, 1985;
  B.A. Brown and W.D.M. Rae. Nucl. Data Sheets 120, 115 (2014).
- Antoine
  F. Nowacki and E. Caurier, Acta Phys. Pol. **B 30**, 705 (1999);
  E. Caurier *et al.*, Rev. Mod. Phys. **77**, 427 (2005).
- Bigstick
  C.W. Johnson, W.E. Ormand and P.G. Krastev,
  Comp. Phys. Comm. **184**, 2761 (2013) [arXiv:1303.0905 [nucl-th]];
  H. Shan, S. Williams, C.W. Johnson, K. McElvain and
  W.E. Ormand, SuperComputing 2015.
- MFDn
  should be made public this fall/winter . . .