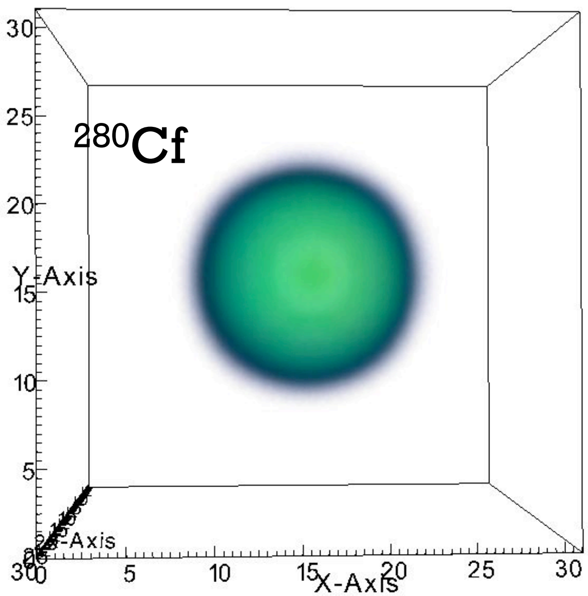
# SLDA Solver: the Pain and Joy of Growing Up

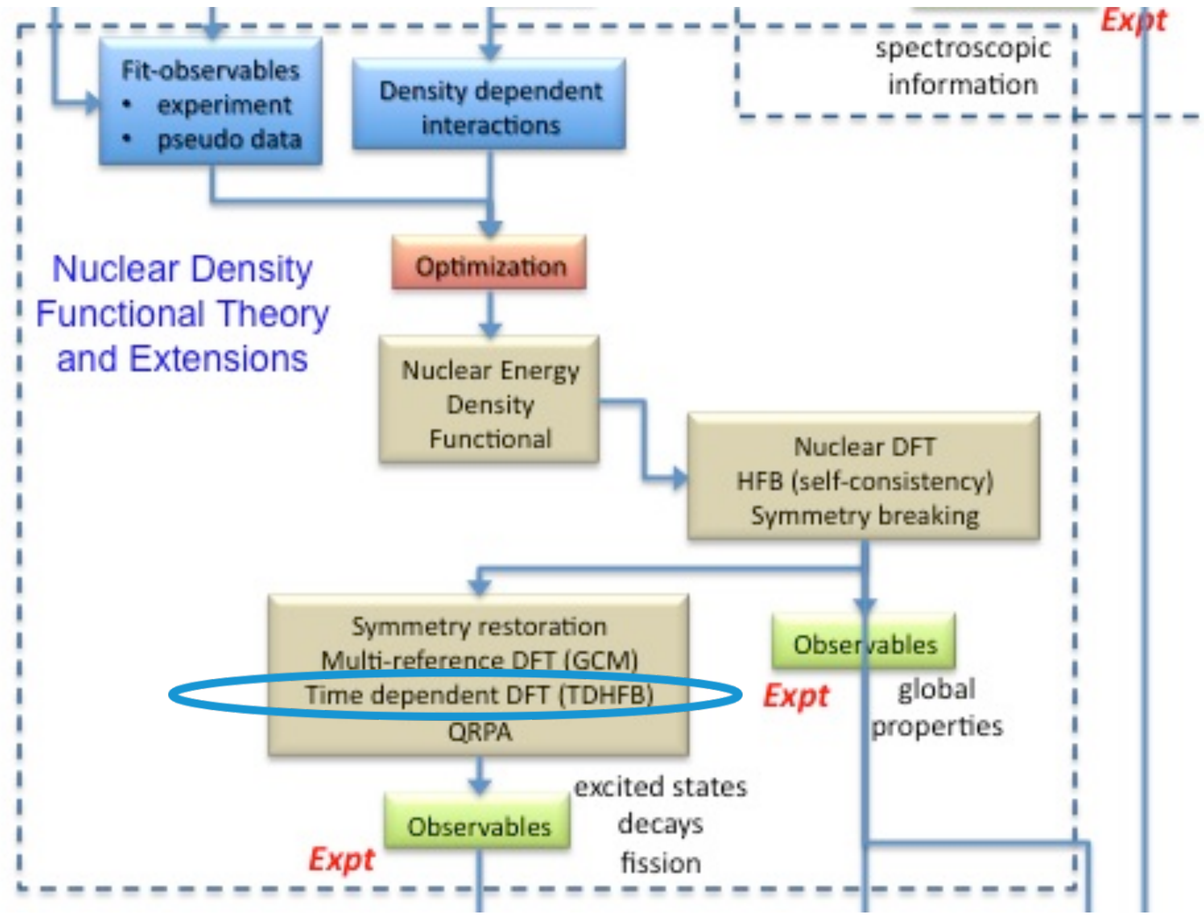
ASLDA software, a history




Outline:
- Motivation, generalities
- Solver: past and present
  - cold atoms
  - nuclear
- Conclusions

Ionel Stetcu (UW)
A. Bulgac, M.M. Forbes, P. Magerski, Y.-L. (Alan) Luo, K.J. Roche

# Where are we?


Expt
spectroscopic
information
Fit-observables
• experiment
• pseudo data
Density dependent
interactions
Optimization
Nuclear Density
Functional Theory
and Extensions
Nuclear Energy
Density
Functional
Nuclear DFT
HFB (self-consistency)
Symmetry breaking
Symmetry restoration
Multi-reference DFT (GCM)
Time dependent DFT (TDHFB)
QRPA
Observables
Expt
global
properties
excited states
decays
fission
Observables
Expt

# Summary Available Codes

- last year code (.f90)
- first nuclear DFT solver (.f90)
- $k_z$ solver and generalization (.c)
- the penultimate DFT solver (.f90)
- the ultimate DFT solver (.f90)

# Mathematical formulation

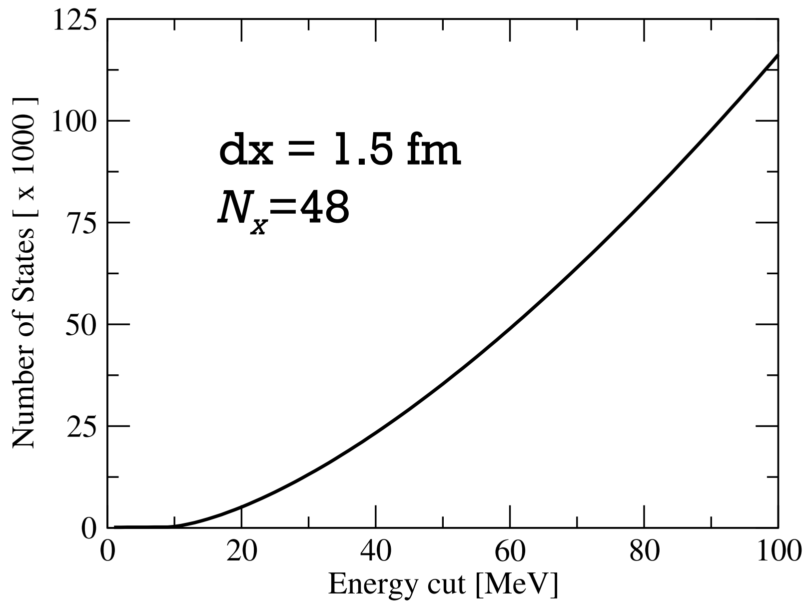
$$E_{g.s.} = \int d^3r \left( \frac{\hbar^2}{2m}\tau(r) + \mathcal{E}[\rho(\vec{r}), \tau(\vec{r}), \nu(\vec{r})] + V_{ext}(\vec{r})\rho(\vec{r}) \right)$$

$$\mathcal{E}[\rho(\vec{r}), \tau(\vec{r}), \nu(\vec{r})] = \mathcal{E}_N[\rho(\vec{r}), \tau(\vec{r})] + \mathcal{E}_S[\rho(\vec{r}), \nu(\vec{r})]$$

$$\begin{pmatrix} h(\vec{r}) - \mu & \Delta(\vec{r}) \\ \Delta^*(\vec{r}) & -(h^*(\vec{r}) - \mu) \end{pmatrix} \begin{pmatrix} u_k(\vec{r}) \\ v_k(\vec{r}) \end{pmatrix} = E_k \begin{pmatrix} u_k(\vec{r}) \\ v_k(\vec{r}) \end{pmatrix}$$

- ❑ Hermitian eigenvalue problem
- ❑ (Almost) all eigenvalues required

$$h(\vec{r}) = -\vec{\nabla}\frac{\hbar^2}{2m^*(\vec{r})}\vec{\nabla} + U(\vec{r})$$



# Normal Energy Functionals

Cold atoms:

$$\mathcal{E}(\vec{r}) = \frac{1}{2}\tau(\vec{r}) + \gamma\frac{|\nu(\vec{r})|^2}{\rho^{1/3}(\vec{r})} + \beta\frac{3(3\pi^2)^{2/3}\rho^{5/3}(\vec{r})}{10} + V_{ext}(\vec{r})\rho(\vec{r})$$

$$h(\vec{r}) = \frac{1}{2}\vec{\nabla}^2 + \beta\frac{3\pi^2\rho^{1/3}(\vec{r}))^{2/3}}{2} - \frac{|\Delta(\vec{r})|^2}{3\gamma\rho^{2/3}(\vec{r})} + V_{ext}(\vec{r})$$

Nuclear systems:

$$\mathcal{E}(\vec{r}) = \frac{1}{2M_n}\tau_n(\vec{r}) + \frac{1}{2M_p}\tau_p(\vec{r}) - \Delta(\vec{r})\nu_c(\vec{r})$$
$$+ \sum_{T=0,1}\left(C_T^\rho\rho_T^2 + C_T^\Delta\rho_T\nabla^2\rho_T + C_\gamma\rho_0^\gamma\rho_T^2\right.$$
$$\left. + C_T^\tau(\rho_T\tau_T - \vec{j}_T^2) + C_T^{\nabla J}(\rho_T\vec{\nabla}\cdot\vec{J} + \vec{s}_T\times\vec{j}_T)\right)$$

Galilean invariance

$$h(\vec{r}) = U(\vec{r}) + \vec{V}(\vec{r})\cdot\vec{\sigma} - i\vec{V}_1(\vec{r})\cdot\vec{\nabla} - i\vec{W}(\vec{r})\cdot(\vec{\sigma}\times\vec{\nabla})$$

# Pairing Renormalization

$$\mathcal{E}_S \stackrel{def}{=} -\Delta(\vec{r})\nu_c(\vec{r}) = g_{eff}(\vec{r})|\nu_c(\vec{r})|^2$$

$$\frac{1}{g_{eff}(\vec{r})} = \frac{1}{g[\rho(\vec{r})]} - \frac{m(\vec{r})k_c(\vec{r})}{2\pi^2\hbar^2}\left\{1 - \frac{k_F(\vec{r})}{2k_c(\vec{r})}\ln\frac{k_c(\vec{r})+k_F(\vec{r})}{k_c(\vec{r})-k_F(\vec{r})}\right\}$$

$$E_c + \mu = \frac{\hbar^2 k_c^2(\vec{r})}{2m(\vec{r})} + U(\vec{r}) \qquad \mu = \frac{\hbar^2 k_F^2(\vec{r})}{2m(\vec{r})} + U(\vec{r})$$

Observables are cutoff independent

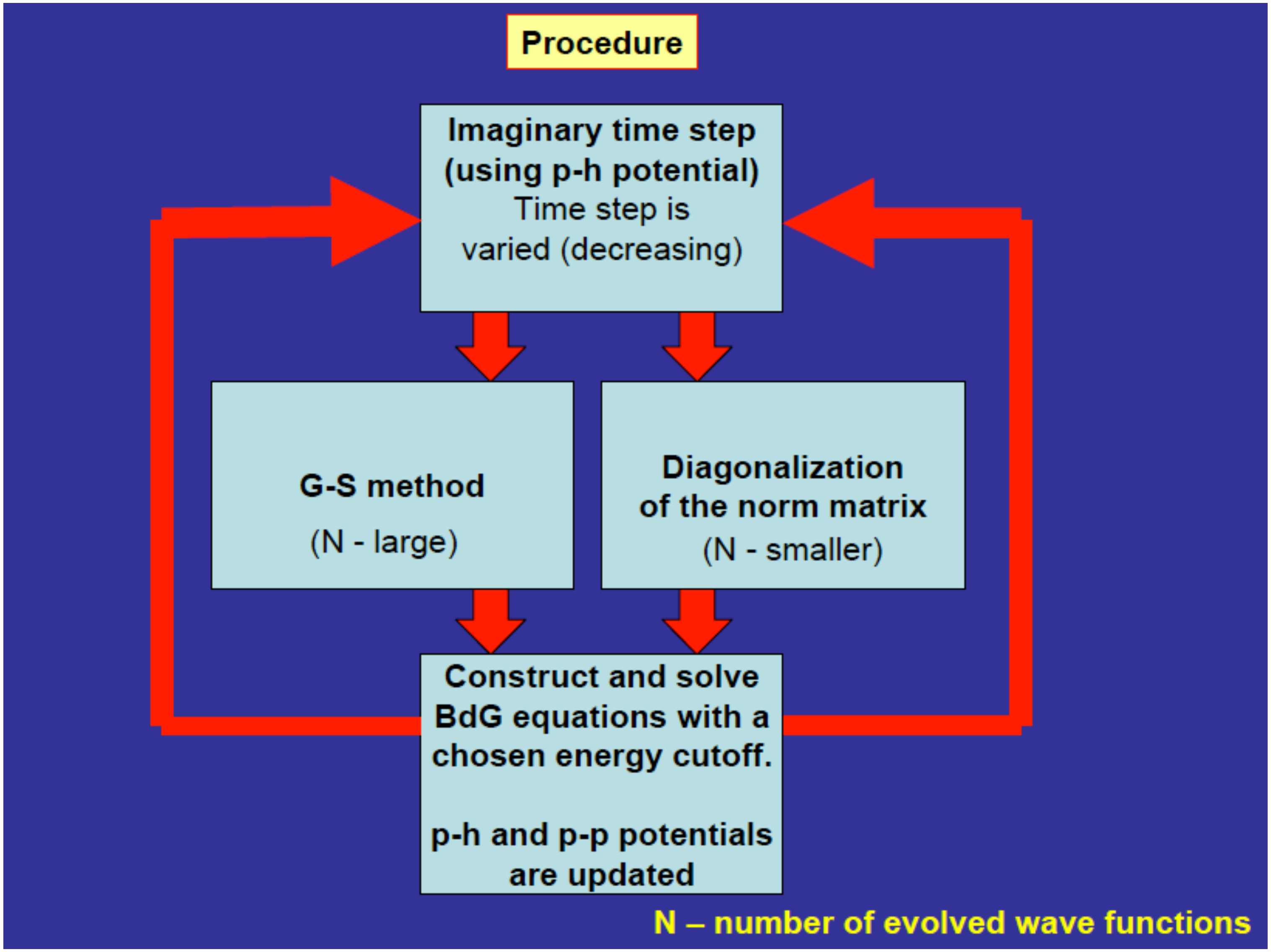
$$\frac{1}{g_{eff}(\vec{r})} = \frac{1}{g[\rho(\vec{r})]} - \frac{m(\vec{r})k_c(\vec{r})}{2\pi^2\hbar^2}\left\{1 - \frac{k_F(\vec{r})}{2k_c(\vec{r})}\ln\frac{k_c(\vec{r})+k_F(\vec{r})}{k_c(\vec{r})-k_F(\vec{r})}\right.$$

$$|\vec{V}(\vec{r})|^2 = \frac{\hbar^2 k_1^2(\vec{r})}{2m(\vec{r})} \qquad \left. + \frac{k_1^2(\vec{r})}{24k_c^2(\vec{r})}\left(1 - \frac{k_c(\vec{r})}{k_F(\vec{r})}\ln\frac{k_F(\vec{r})+k_c(\vec{r})}{\kappa(\vec{r})}\right)\right\}$$

# Lattice representation

- quasiparticle wavefunctions represented on a lattice
- periodic boundary conditions
- $N_x, N_y, N_z$ spatial points
- derivatives computed with FFT
- good description of the relevant DOF for E>0
- (almost) unique ability to describe correctly all components of the quasiparticle wavefunctions

# One year ago


Procedure
Imaginary time step (using p-h potential) Time step is varied (decreasing)
G-S method
(N - large)
Diagonalization of the norm matrix
(N - smaller)
Construct and solve BdG equations with a chosen energy cutoff.
p-h and p-p potentials are updated
N – number of evolved wave functions

# One year ago

$$e^{-\lambda\hat{T}}\psi(\vec{p}) = e^{-\lambda\frac{p^2}{2m}}\psi(\vec{p})$$

$$e^{-\lambda\hat{V}}\psi(\vec{r}) = e^{-\lambda V(\vec{r})}\psi(\vec{r})$$

**REAL SPACE** ⟺ *FFT* ⟺ **MOMENTUM SPACE**

**Advantages:**

- Much faster convergence (**order of magnitude** difference between the first order and the second order method).
- The methods **do not diverge** even for large time steps.
- The low cost of **FFT** instead of matrix multiplication.

# Poor-man parallelization of the existing code for (trapped) neutrons

- two-processor run: protons + neutrons worlds (communicators)
- little communication
- limited in the size it can handle

**Issues**

❖FFTW requires the entire function on one processor

❖ distribution of wfs. on different processors would make the orthogonalization & computation of the HFB matrix complicated

# Switch gears: discrete variable representation basis

$$F(x) = \frac{1}{N}\sum_{n=0}^{N-1} \exp(ik_n x) \qquad F((i-j)a) = \delta_{ij}$$

$$k_n = -\frac{\pi}{a} + \frac{2\pi}{Na}n, \qquad n = 0, \ldots, N$$

1D basis states:

$$\varphi_i(x) = \frac{1}{N} e^{-\frac{i\pi(x-x_i)}{Na}} \frac{\sin\frac{\pi(x-x_i)}{a}}{\sin\frac{\pi(x-x_i)}{Na}}$$

$$(\partial_x)_{nm} = \frac{\pi}{Na}(-1)^{n-m}\left[(1-\delta_{nm})\cot\left(\frac{\pi(n-m)}{N}\right) - \frac{i}{N}\right]$$

$$(\partial_{xx})_{nm} = \frac{\pi^2}{2N^2a^2}\frac{(-1)^{n-m}(\delta_{nm}-1)}{\sin\frac{\pi(n-m)}{N}} - \frac{\pi^2}{3a^2}\left(1 + \frac{2}{N^2}\delta_{nm}\right)$$

# Matrix generation

Local terms: $(U(\vec{r}))_{mn} = U_n \delta_{nm}$

Non-local terms require more attention:

$$-\vec{\nabla}\frac{\hbar^2}{2m^*(\vec{r})}\vec{\nabla}v(\vec{r}) = -\frac{1}{2}\left[\frac{\hbar^2}{2m^*(\vec{r})}\vec{\nabla}^2 v(\vec{r}) + \vec{\nabla}^2\left(\frac{\hbar^2}{2m^*(\vec{r})}v(\vec{r})\right) - \left(\vec{\nabla}^2\frac{\hbar^2}{2m^*(\vec{r})}\right)v(\vec{r})\right]$$

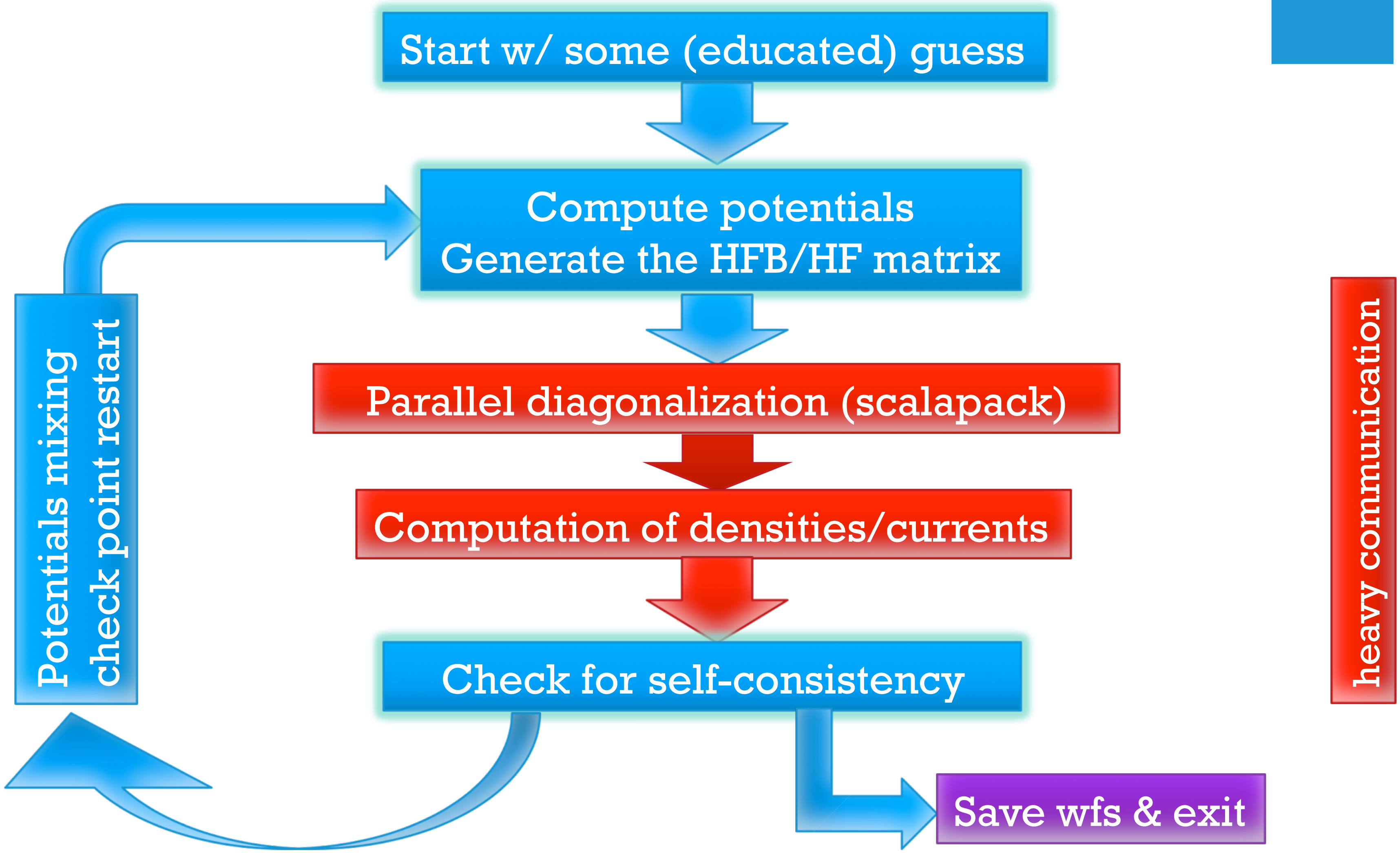
$$\left(-\vec{\nabla}\frac{\hbar^2}{2m^*(\vec{r})}\vec{\nabla}\right)_{nm} = -\frac{1}{2}(\vec{\nabla}^2)_{nm}\left[\frac{\hbar^2}{2m^*_n} + \frac{\hbar^2}{2m^*_m}\right] + \frac{1}{2}\left(\vec{\nabla}^2\frac{\hbar^2}{2m^*}\right)_n \delta_{nm}$$

$$\vec{W}\cdot\left(\vec{\sigma}\times\vec{\nabla}v(\vec{r})\right) = \frac{1}{2}\left[\vec{W}\cdot\left(\vec{\sigma}\times\vec{\nabla}v(\vec{r})\right) + \vec{\sigma}\cdot\left(\vec{\nabla}\times\left(\vec{W}v(\vec{r})\right)\right) - \vec{\sigma}\cdot\left(\vec{\nabla}\times\vec{W}\right)\right]$$

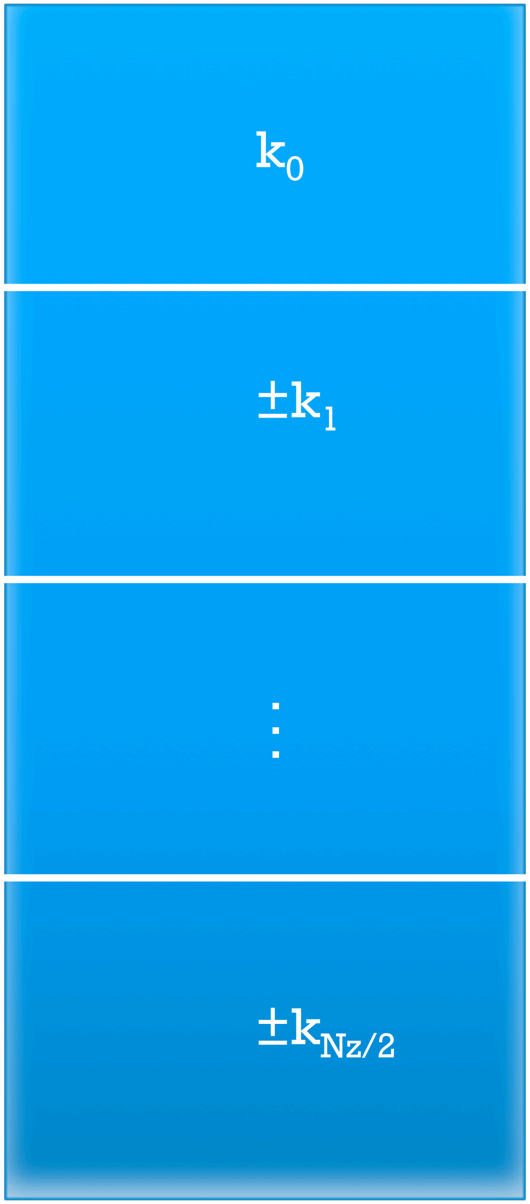
**additional important overhead for TD**

$$\left(2\vec{j}(\vec{r}) + \vec{\nabla}\cdot\vec{j}(\vec{r})\right)v = \vec{j}(\vec{r})\cdot\vec{\nabla}v + \vec{\nabla}\cdot\left(\vec{j}(\vec{r})v\right)$$

# Current implementations



# $k_z$ solver (cold-atom gas)




$$u(x,y,z) = u(x,y)\exp(ik_z z)$$

$$v(x,y,z) = v(x,y)\exp(ik_z z)$$

$N_z/2+1$ eigenvalue problems of dimension 2 $N_x N_y$

Static+TD application (see A.B.)

generalization to axially sysmetric systems (cylindrical coordinates)

heavy communication / BIG computer (see KJR)

# Two Large-Scale Nuclear Solvers

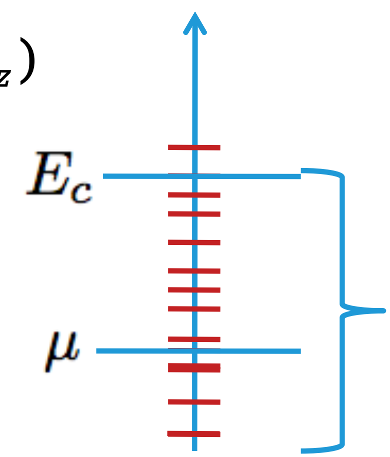
1. One diagonalization ($4xN_xN_yN_z$)

$$\begin{pmatrix} h_{++}-\mu & h_{+-} & 0 & \Delta \\ h_{-+} & h_{--}-\mu & -\Delta & 0 \\ 0 & -\Delta^* & \mu-h^*_{++} & -h^*_{+-} \\ \Delta^* & 0 & -h^*_{-+} & \mu-h^*_{--} \end{pmatrix} \begin{pmatrix} u_+ \\ u_- \\ v_+ \\ v_- \end{pmatrix} = E \begin{pmatrix} u_+ \\ u_- \\ v_+ \\ v_- \end{pmatrix}$$

2. Two consecutive diagonalizations ($2xN_xN_yN_z$ +~$N_xN_yN_z$)

$$\begin{pmatrix} h_{++} & h_{+-} \\ h_{-+} & h_{--} \end{pmatrix} \begin{pmatrix} \phi_+ \\ \phi_- \end{pmatrix} = \varepsilon \begin{pmatrix} \phi_+ \\ \phi_- \end{pmatrix}$$

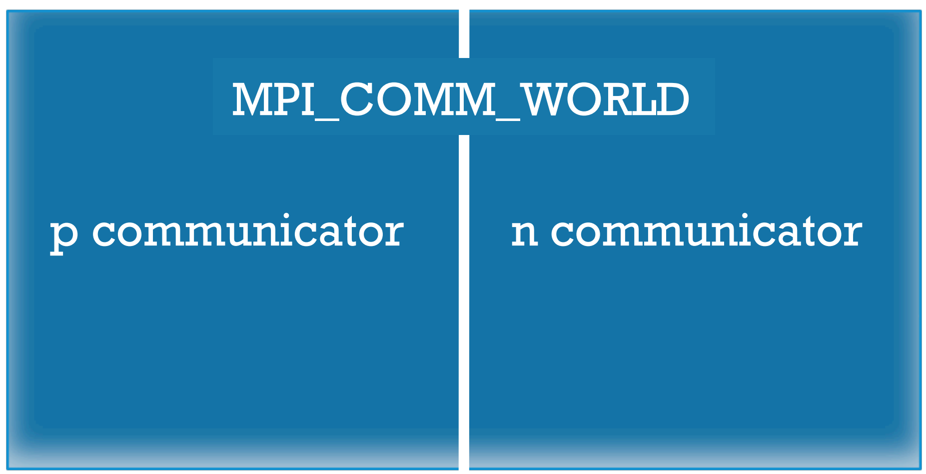
$$\begin{pmatrix} \varepsilon-\mu & 0 & 0 & \Delta \\ 0 & \varepsilon-\mu & -\Delta & 0 \\ 0 & -\Delta^* & -(\varepsilon-\mu) & 0 \\ \Delta^* & 0 & 0 & -(\varepsilon-\mu) \end{pmatrix} \begin{pmatrix} u_+ \\ u_- \\ v_+ \\ v_- \end{pmatrix} = E \begin{pmatrix} u_+ \\ u_- \\ v_+ \\ v_- \end{pmatrix}$$



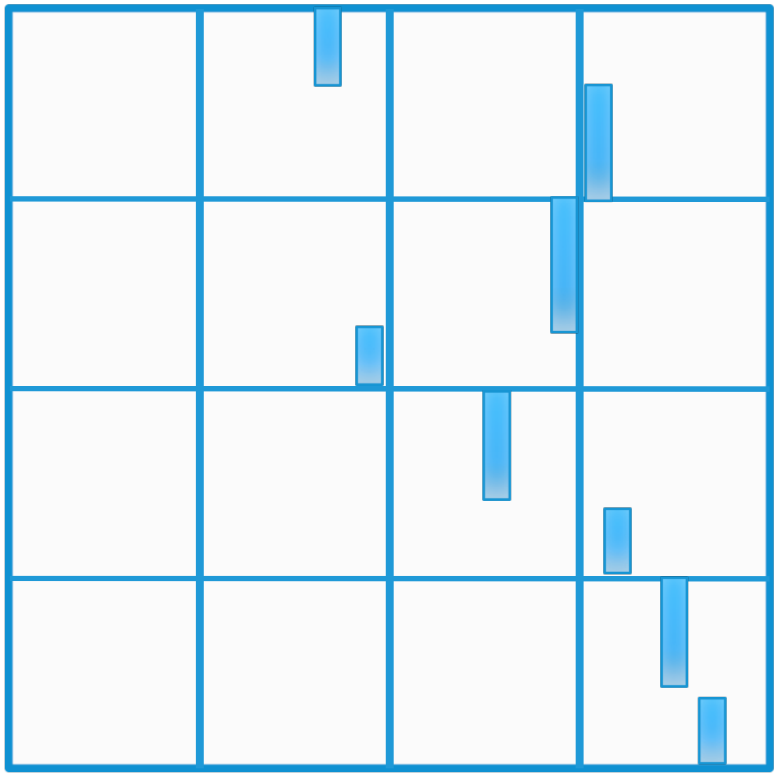
# Nuclear Solver: Parallel Implementation

Input:

- Lattice size/constant, particle numbers, etc
- # processors for grid, block size
- Potentials/Densities (hfbrad, ev4, ev8, etc.)



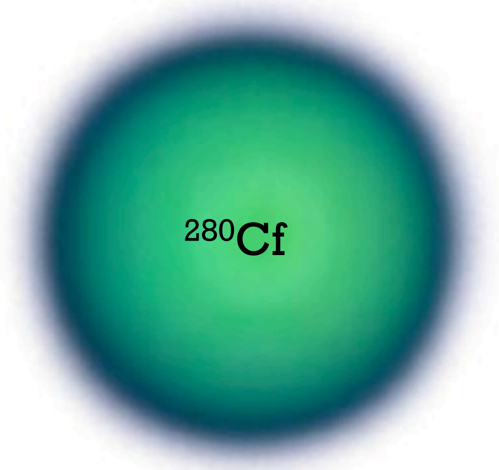
# Nuclear Solver: Selected Details



processor grid for each communicator

On each communicator (p/n):

- ✧ setup/compute the HF/HFB matrix
- ✧ diagonalize the HF/HFB matrix
- ✧ (construct and diagonalize the cyclically decomposed pairing matrix)
- ✧ reconstruct each wavefunction on all group processors and compute densities (involve heavy communication)
- ✧ communication of densities




# Performance on a $32^3$ lattice

\# of processors: 2x11664
dimension of the Hilbert space: 2x131072

| | time (s) | # instructions | fp |
|---|---|---|---|
| h: | 0. 23 | 200.01E10 | 15.96E08 |
| D: | 2336.84 | 985.47E14 | 138.54E14 |
| SC: | 1164.37 | 424.84E14 | 380.94E12 |

real 59m6.027s
user 0m1.800s
sys 0m0.244s

Deliverables:
✓ Profile ASLDA DFT solver with pairing (27-28)

$40^3$ requires all Jaguarpf (XT5) for one iteration/hr (see K.J.R)

# Benchmarks

- tested simple solutions: KE only, KE+constant pairing, etc.
- tested the solutions in the TD code: energy and number of particle conservation within expected numerical precision
- good agreement with HFBRAD for spherical systems

# Summary

- ✓ SLDA solver ready to run
- ✓ Connection with the TD-code
- ✓ Deliverables year 4 achieved
- ✓ Stay tuned for applications (tomorrow)

# For Monday

- better I/O for saving the wavefunctions
- connection with the TDSLDA code