The sinc-based DFT solver

Piotr Magierski (Warsaw), Aurel Bulgac (Seattle), Kenneth Roche (Oak Ridge)
Goal: Provide an accurate and symmetry free DFT solver.

Status in 2007:

The method

\[ \psi^{(n+1)}_j = e^{-\lambda \hat{H}} \psi^{(n)}_j, \quad j = 1, \ldots, N \]

\[ \Psi^{(n+1)} \xrightarrow{\text{orthogonalization}} \psi^{(n+1)}_j \xrightarrow{\text{normalization}} \]

\[ \lambda - \text{imaginary time step} \]

\[ \hat{H} \psi^{(n)}_j \xrightarrow{n \to \infty} E_j \psi^{(n)}_j \]
\[ e^{-\lambda \hat{H}} = 1 + \sum_{i=1}^{n} \frac{(-\lambda)^i}{i!} \hat{H}^i + O(\lambda^{n+1}) \]

**Drawbacks:**
- slow convergence.
- divergent for too large imaginary time step.
- numerically costly for \( n>1 \).

**Alternative:** we look for the expansion of \( \exp(-\lambda \hat{H}) = \exp(-\lambda (\hat{T} + \hat{V})) \) in the form of product of exponents.

\[
e^{-\lambda \hat{H}} = e^{\frac{\lambda \hat{T}}{2}} e^{-\hat{\nabla}^2} e^{\frac{\lambda \hat{T}}{2}} + O(\lambda^3), \quad \text{- second order method}
\]

\[
e^{-\lambda \hat{H}} = e^{\frac{\lambda \hat{V}}{6}} e^{\frac{\lambda \hat{T}}{2}} e^{\frac{2\lambda \hat{V}}{3}} e^{\frac{\lambda \hat{T}}{2}} e^{\frac{\lambda \hat{V}}{6}} + O(\lambda^5), \quad \text{- fourth order method}
\]

\[ \tilde{V} = \hat{V} + \frac{1}{48} \lambda^2 [\hat{V}, [\hat{T}, \hat{V}]] \quad \text{- dominant term is local in space if} \ \hat{V} \ \text{is local.} \]
The above expansions are particularly useful if the Hamiltonian can be expressed as a sum of terms either local in the coordinate space (potential), or in the momentum space (kinetic energy).
In such a case one is advised to use Fast Fourier Transform algorithm during the evolution of the wave function.

\[ e^{-\lambda \hat{T}} \psi(\vec{p}) = e^{-\lambda \frac{p^2}{2m}} \psi(\vec{p}) \]
\[ e^{-\lambda \hat{V}} \psi(\vec{r}) = e^{-\lambda V(\vec{r})} \psi(\vec{r}) \]

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.
Procedure

Imaginary time step
Time step is varied (decreasing)

Preorthogonalization
G-S method is used to decouple space of already converged states

G-S method
(N - large)

Diagonalization of the norm matrix
(N - smaller)

N – number of evolved wave functions
What has been done during the last year:

- Inclusion of spin-orbit term:

\[
e^{-\lambda \hat{H}} \approx e^{-\frac{\lambda \hat{t}}{2}} e^{-\frac{\lambda \hat{\nu}}{2}} \left(1 - \lambda \nabla V \cdot (\hat{\sigma} \times \hat{p}) + \frac{\lambda^2}{2} \left(\nabla V \cdot (\hat{\sigma} \times \hat{p})\right)^2\right) e^{-\frac{\lambda \hat{\nu}}{2}} e^{-\frac{\lambda \hat{t}}{2}}
\]

- Inclusion of pairing:

\[
\begin{pmatrix}
\hat{h} & \hat{\Delta} \\
\hat{\Delta}^\dagger & -\hat{h}^*
\end{pmatrix}
\begin{pmatrix}
U_n(\vec{r}) \\
V_n(\vec{r})
\end{pmatrix} = E_n
\begin{pmatrix}
U_n(\vec{r}) \\
V_n(\vec{r})
\end{pmatrix}
\]

\[
\hat{h} = \begin{pmatrix}
-\frac{\hbar^2}{2m_+} \nabla^2 - \mu_+ + V(\vec{r}) & 0 \\
0 & -\frac{\hbar^2}{2m_-} \nabla^2 - \mu_- - V(\vec{r})
\end{pmatrix} + \lambda \nabla V \cdot (\hat{\sigma} \times \hat{p})
\]

\[
\hat{\Delta} = \begin{pmatrix}
0 & \Delta(\vec{r}) \\
-\Delta(\vec{r}) & 0
\end{pmatrix};
\]

\[
U_n(\vec{r}) = \begin{pmatrix} U_{n+}(\vec{r}) \\ U_{n-}(\vec{r}) \end{pmatrix};
\]

\[
V_n(\vec{r}) = \begin{pmatrix} V_{n+}(\vec{r}) \\ V_{n-}(\vec{r}) \end{pmatrix}
\]
Procedure

Imaginary time step
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.

N – number of evolved wave functions
Potential in p-h channel

\[ V(\vec{r}) = \frac{V_1}{1 + \cosh\left(\frac{r_1}{a}\right)} + \frac{V_2}{\left(1 + \cosh\left(\frac{r_{xy}}{b}\right)\right)\left(1 + \cosh\left(\frac{z + \xi}{b}\right)\right)} \]

\[ r_1 = \sqrt{x^2 + 0.4y^2 + (z - \xi)^2}; r_{xy} = \sqrt{0.2x^2 + y^2} \]

\[ V_1 = -120\,MeV, V_2 = -250\,MeV, a = 1.5\,fm, b = 1.2\,fm, \xi = 5\,fm \]
Potential in p-p channel

\[
\Delta(\vec{r}) = \frac{\Delta_1}{1 + \cosh\left(\frac{r_1}{a}\right)} + \frac{\Delta_2}{(1 + \cosh\left(\frac{r_{xy}}{b}\right))(1 + \cosh\left(\frac{z + \xi}{b}\right))}
\]

\[r_1 = \sqrt{x^2 + 0.4y^2 + (z - \xi)^2};\quad r_{xy} = \sqrt{0.2x^2 + y^2}\]

\[
\Delta_1 = 2\text{MeV}, \quad \Delta_2 = 3\text{MeV}, \quad a = 1.5\text{fm}, \quad b = 1.2\text{fm}, \quad \xi = 5\text{fm}
\]
Accuracy (numerical tests)

Number of wave functions: 70
Chemical potential (both spins): -3 MeV (about 40 particles)
Energy cutoff: 1 MeV
Box size: 30 fm

\[ \rho_i (\vec{r}) = \sum_n V^*_{n I} (\vec{r}) V^T_{n I} (\vec{r}) ; i = \pm \]

\[ \chi_{+-} (\vec{r}) = \sum_n V^*_{n+} (\vec{r}) U^T_{n-} (\vec{r}) \]

\[ \int \left( \rho (\vec{r}) \right)^3 d^3 r \]

\[ \int \left| \chi_{+-} (\vec{r}) \right|^3 d^3 r \]
\[ \int (\rho(\vec{r}))^3 \, d^3r \]
\[ \int |\chi_{+-}(\vec{r})|^3 \, d^3r \]

Box: 30 fm
Lattice constant: 0.6 fm
Summary

At the moment ‘the non-selfconsistent DFT solver’ exist. It can handle arbitrary shape of potentials in p-h and p-p channels. The accuracy of the results is stable with respect to variations of the potentials, chemical potentials, energy cutoff, effective masses, etc.

1. Is your Year-2 plan well on track?
   Absolutely.

2. What are the aspects of your science that require high-performance computing?
   Data and computational complexity.

3. Plan for Year-2 and Year-3:
   - profile current code.
   - parallelization of the code.
   - self-consistent version.
   - interface between TDSLDA code.