The Self-Consistent Loop

\[ D = N_r \]

\[ D = N_p N_r \]
Current Implementation

EV8

c * nxmu
  damping factor for the mean-field potential (evolve)
  xmu=nxmu/100.; w(n+1)=xmu*w(n+1)+(1.-xmu)*w(n)
  if nxmu is read 0, xmu is set to 0.25

xmu = float(nxmu)/t100
ymu = one-xmu

do i=1,2*mv
   rho(i) = xmu* rho(i)+ymu*rvst(i,1)
   vtau(i) = xmu*vtau(i)+ymu*rvst(i,2)
   vdiv(i) = xmu*vdiv(i)+ymu*rvst(i,3)
   rvst(i,1) =  rho(i)
   rvst(i,2) = vtau(i)
   rvst(i,3) = vdiv(i)
endo
call newpot

HHODD

SLOWEV=0.500
XOLDEV=SLOWEV
XNEWEV=1.00D0-XOLDEV
DO IX=1,NXHERM
   DO IY=1,NYHERM
      DO IZ=1,NZHERM
         VN_MAS(IX,IY,IZ) = VN_MAS(IX,IY,IZ)*XOLDEV+XNEWEV*VNEUTR
      END DO
   END DO
END DO
Improving self-consistent calculations of Fermion systems

Michael McNeil Forbes and Aurel Bulgac
Improving self-consistent calculations of Fermion systems

![Graph showing convergence of iterations for different methods and parameters.](image-url)
Outline

• Structure of self-consistent calculations
• Current codes
• Broyden method to accelerate convergence
  • Very easy to implement
• DVR Basis to improve representation
Self-consistent calculations

Find a “fixed-point” in a high dimensional space.

\[ X \mapsto F(X) \]

- HFB
- BdG
- DFT (LDA, Kohn)
Self-consistent calculations

Weighting Scheme

\[ (1 - w)X + wF(X) \]

\[ \hat{H}[n, \Delta] \psi_n = E_n \psi_n \]

\[ n(\vec{r}) = \sum_n |\psi_n(\vec{r})|^2, \ldots \]
Self-consistent calculations

Broyden Scheme

\[ X = [n(\vec{r}), \Delta(\vec{r}), \ldots] \]

\[ \hat{H}[n, \Delta] \psi_n = E_n \psi_n \]

\[ n(\vec{r}) = \sum_n |\psi_n(\vec{r})|^2, \ldots \]
Broyden Method

\[ G(X) = X - F(X) = 0 \]

\[
\frac{(|X_n\rangle, |G_n\rangle, J_n^{-1})}{|dX\rangle = -J_n^{-1} \cdot |G_n\rangle}
\]

\[
|X_{n+1}\rangle = |X_n\rangle + |dX\rangle
\]

\[
|G_{n+1}\rangle = G(|X_{n+1}\rangle)
\]

\[
|dG\rangle = |G_{n+1}\rangle - |G_n\rangle
\]

\[
J_{n+1}^{-1} = J_n^{-1} + \frac{(|dX\rangle - J_n^{-1}|dG\rangle) \langle dX|J_{n+1}^{-1}|dG\rangle}{\langle dX|J_n^{-1}|dG\rangle}
\]

\[
(|X_{n+1}\rangle, |G_{n+1}\rangle, J_{n+1}^{-1})
\]

- Multidimensional Secant method
- Start with \( J_0^{-1} = w \):
  - \( X_1 = (1-w)X_0 + wF(X_0) \)
- May keep track of dyadics if space is large
- Hold \( J_n^{-1} = w \) for old method

Broyden Improves Convergence

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Method</th>
<th>Iteration</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>(30, 15)</td>
<td>Iterative (w=1.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(30, 15)</td>
<td>Iterative (w=0.5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(30, 15)</td>
<td>Broyden (w=1.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(30, 15)</td>
<td>Broyden (w=0.5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(30, 30)</td>
<td>Iterative (w=1.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(30, 30)</td>
<td>Iterative (w=0.5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(30, 30)</td>
<td>Broyden (w=1.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(30, 30)</td>
<td>Broyden (w=0.5)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Current Implementation

EV8

c   * nxmu
  c   * damping factor for the mean-field potential (evolve)
  c   *    xmu=nxmu/100.; w(n+1)=xmu*w(n)+1.-xmu)*w(n)
  c   *        if nxmu is read 0, xmu is set to 0.25
  
xmu = float(nxmu)/100  
ymu = one-xmu
  do i=1,2*mv
    rho(i) = xmu* rho(i)+ymu*rvst(i,1)
    vtau(i) = xmu* vtau(i)+ymu*rvst(i,2)
    vdiv(i) = xmu* vdiv(i)+ymu*rvst(i,3)
    rvst(i,1) = rho(i)
    rvst(i,2) = vtau(i)
    rvst(i,3) = vdiv(i)
  enddo
  call newpot

HHODD

SLOWEV=0.500
XOLDEV=SLOWEV
XNEWEV=1.00D0-XOLDEV
DO IX=1,NXHERM
  DO IY=1,NYHERM
    DO IZ=1,NZHERM
      VN_MAS(IX,IY,IZ) = VN_MAS(IX,IY,IZ)*XOLDEV+XNEWEV*VNEUTR
    END DO
  END DO
END DO
MATLAB Code

if iter == 1  % usual step for first iteration or if using usual procedure
    G0    =   x0 - [V_0;D_0;V_1;D_1; mu_a*N_a/N0_a;mu_b*N_b/N0_b];
    Jinv0 =   w  % use weight on initial step or if using usual procedure
    dx    =  - Jinv0*G0;
    x0    =   x0 + dx;
elseif iter > 1  % broyden step from second iteration
    G1    =    x0 - [V_0;D_0;V_1;D_1; mu_a*N_a/N0_a;mu_b*N_b/N0_b];
    dG    =    G1 - G0;
    ket   =    dx - Jinv0*dG;
    bra   =    dx'*Jinv0;
    inorm =    1.0/(bra*dG);
    Jinv0 =    Jinv0 + ket*bra*inorm;  % update inverse jacobian here
    dx    =  - Jinv0*G1;
    x0    =   x0 + dx;
    G0    =    G1;
end
Broyden Method

\[ G(X) = X - F(X) = 0 \]

\[
\begin{align*}
\langle |X_n\rangle, |G_n\rangle, J_n^{-1} \rangle \\
|dX\rangle &= -J_n^{-1} \cdot |G_n\rangle \\
|X_{n+1}\rangle &= |X_n\rangle + |dX\rangle \\
|G_{n+1}\rangle &= G(|X_{n+1}\rangle) \\
|dG\rangle &= |G_{n+1}\rangle - |G_n\rangle \\
J_{n+1}^{-1} &= J_n^{-1} + \frac{(|dX\rangle - J_n^{-1} |dG\rangle) \langle dX | J_n^{-1}}{\langle dX | J_n^{-1} |dG\rangle} \\
\langle |X_{n+1}\rangle, |G_{n+1}\rangle, J_{n+1}^{-1} \rangle
\end{align*}
\]

- Multidimensional Secant method
- Start with \( J_0^{-1} = w \):
  \[ X_1 = (1-w)X_0 + wF(X_0) \]
- May keep track of dyadics if space is large
- Hold \( J_n^{-1} = w \) for old method

Broyden Costs

- Simple: (Maintain and update Jacobian inverse)
  - $O(N^2) \times N_{iter}$

- Dyadic Representation:
  - $O(N \times N_{iter}) \times N_{iter}$

\[
J_{m+1}^{-1} = w \mathbf{1} + \sum_{n=1}^{m} |a_n\rangle \langle b_n|
\]

\[
|a_n\rangle = |dX_n\rangle - J_n^{-1} |dG_n\rangle
\]

\[
\langle b_n| = \frac{\langle dX_n| J_n^{-1}}{\langle dX_n| J_n^{-1} |dG_n\rangle}
\]
Difficulties with HO Basis

- Large radius behaviour of HO Basis introduces artifacts
- Need large number of states to correct
- (Requires HO Basis wavefunctions to high precision)

Grasso and Urban, BCS Code

![Graph of Grasso and Urban, BCS Code]

Our Unitary HO Basis Code

![Graph of Our Unitary HO Basis Code]
Problem with HO Basis

- Tails spoil large $r$ behaviour
DVR solves the problem

Our code in HO Basis

Our code in DVR basis
**DVR Basis in one-dimension**

(Higher dimensional generalization is straightforward)

\[ P^2 = P \quad \text{(Projection onto restricted Hilbert space)} \]

\[ \langle x | P | y \rangle = \int_{-\pi/l}^{\pi/l} \frac{dk}{2\pi} e^{ik(x-y)} = \frac{\sin \left( \frac{\pi}{l} (x-y) \right)}{\pi(x-y)}, \]

\[ \Delta_\alpha = P[\delta(x - x_\alpha)], \]

\[ \langle \Delta_\alpha | \Delta_\beta \rangle = \Delta_\alpha(x_\beta) = \Delta_\beta(x_\alpha) = K_\alpha \delta_{\alpha\beta}, \]

\[ \psi(x) = \sum_{\alpha=1}^{N} c_\alpha \Delta_\alpha + \mathcal{O}(e^{-cN}) \approx \sum_{n} \psi(nl) \frac{\sin \left[ \frac{\pi}{l} (x - nl) \right]}{\pi/l (x - nl)} \]

\[ c_\alpha = \int dx \frac{1}{K_\alpha} \Delta_\alpha(x) \psi(x) = \frac{1}{K_\alpha} \psi(x_\alpha), \quad x_\alpha = nl \]

\[ \psi(x) = \sum_{\alpha=1}^{N} d_{\alpha} F_{\alpha}(x) + O(e^{-cN}), \]

\[ F_{\alpha} = \frac{1}{\sqrt{K_{\alpha}}} \Delta_{\alpha}(x), \quad x_{\alpha} = n l, \quad \langle F_{\alpha} | F_{\beta} \rangle = \delta_{\alpha,\beta}, \]

\[ \sum_{\beta} \left[ \langle F_{\alpha} | \hat{T} | F_{\beta} \rangle + V(x_{\alpha}) \delta_{\alpha\beta} \right] d_{\beta} = E d_{\alpha} \]
DVR for Radial Equation: Bessel DVR Basis


\[ F_{vn}(r) = (-1)^{n+1} \frac{K z_{vn} \sqrt{2r}}{K^2 r^2 - z_{vn}^2} J_\nu(K r) \]

\[
P(r, r') = \int_0^K dk \langle kr | J_\nu \rangle \langle J_\nu | kr' \rangle
\]

\[
\langle F_{vn} | k_r^2 + \frac{\nu^2 - \frac{1}{4}}{r^2} | F_{vn'} \rangle
= \begin{cases} 
\frac{K^2}{3} \left[ 1 + \frac{2(\nu^2 - 1)}{z_{vn}^2} \right], & n = n', \\
(-1)^{n-n'} 8K^2 \frac{z_{vn} z_{vn'}}{(z_{vn}^2 - z_{vn'}^2)^2}, & n \neq n',
\end{cases}
\]

FIG. 2. Plots of the Bessel DVR functions \( F_{vn}(r) \) for \( K = 1 \) and for selected values of \( \nu \) and \( n \).
\[ \epsilon_F, \Delta, T \ll \frac{\hbar^2 \pi^2}{2ml^2}, \]
\[ \delta \epsilon > \frac{2\hbar^2 \pi^2}{mL^2}, \]
\[ \epsilon_F, \Delta T \gg \frac{2\hbar^2 \pi^2}{mL^2}, \]
\[ \xi \ll L = N_sl, \]
\[ \delta p > \frac{2\pi \hbar}{L}. \]
Summary

- Broyden Improves Convergence
  - Extremely easy to implement
  - Can be made inexpensive
- HO Basis has problems with large $r$ tails
- DVR Basis solves these problems
  - Near optimal phase-space coverage