Space-time RPA method

Fundamental formulas

$$\chi^0(r,r',i\omega)=\frac{1}{N_k^2}\sum_{nm}\sum_{\mathbf{kq}}\frac{(f_{n\mathbf{k}}-f_{m\mathbf{q}})\psi^*_{n\mathbf{k}}(r)\psi_{m\mathbf{q}}(r)\psi^*_{m\mathbf{q}}(r')\psi_{n\mathbf{k}}(r')}{\epsilon_{nk}-\epsilon_{mq}-i\omega}$$

$$\chi^0(r,r',i\tau)=\frac{1}{2\pi}\int_{-\infty}^{\infty}d\omega e^{-i\omega\tau}\chi^0(r,r',i\omega)$$

$$\chi^0(r,r',i\omega)=\int_{-\infty}^{\infty}d\tau e^{i\omega\tau}\chi^0(r,r',i\tau)$$

$$\chi^0(r,r',i\tau)=-iG^0(r,r',i\tau)G^0(r',r,-i\tau)$$

$$G^0(r,r',i\tau)=\sum_{ij}\sum_{\mathbf{R},\mathbf{R}'}\varphi_i(r-\mathbf{R})G_{ij}(\mathbf{R}'-\mathbf{R},i\tau)\varphi_j(r'-\mathbf{R}')$$

$$G_{ij}(\mathbf{R},i\tau)=\begin{cases}-\frac{1}{N_k}\sum_{\mathbf{k}}\sum_n f_{n\mathbf{k}}c_{i,n}(k)c^*_{j,n}(k)e^{-i\mathbf{kR}}e^{-(\epsilon_{n,k}-\mu)\tau} & \tau<=0 \qquad (occ)\\ \frac{1}{N_k}\sum_{\mathbf{k}}\sum_n(1-f_{n\mathbf{k}})c_{i,n}(k)c^*_{j,n}(k)e^{-i\mathbf{kR}}e^{-(\epsilon_{n,k}-\mu)\tau} & \tau>0 \qquad (unocc)\end{cases}$$

$$\begin{aligned}\chi^0(r,r',i\tau)=-i\sum_{i,j,k,l}\sum_{\mathbf{R},\mathbf{R}',\mathbf{R}_1,\mathbf{R}_2}&\varphi_i(r-\mathbf{R})\varphi_k(r-\mathbf{R}_1)G_{i,j}(\mathbf{R}'-\mathbf{R},i\tau)G_{l,k}(\mathbf{R}_1-\mathbf{R}_2,-i\tau)\\ &\varphi_j(r'-\mathbf{R}')\varphi_l(r'-\mathbf{R}_2')\end{aligned}$$

$$\varphi_i(r-\mathbf{R})\varphi_k(r-\mathbf{R}_1)\approx\sum_{\mu\in\mathbf{I}}C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1-\mathbf{R})}P_\mu(r-\mathbf{R})+\sum_{\mu\in\mathbf{K}}C^{\mu(\mathbf{0})}_{i(\mathbf{R}-\mathbf{R}_1),k(\mathbf{0})}P_\mu(r-\mathbf{R}_1)$$

$$\chi^0(r,r',i\tau)=\sum_{\mu,\nu,\mathbf{R},\mathbf{R}'}P_\mu(r-\mathbf{R})\chi^0_{\mu,\nu}(\mathbf{R}'-\mathbf{R},i\tau)P_\nu(r'-\mathbf{R}')$$

$$\begin{aligned}\chi^0_{\mu,\nu}(\mathbf{R},i\tau)&=\chi^{0(A)}_{\mu,\nu}(\mathbf{R},i\tau)+\chi^{0(B)}_{\mu,\nu}(\mathbf{R},i\tau)+\chi^{0(C)}_{\mu,\nu}(\mathbf{R},i\tau)+\chi^{0(D)}_{\mu,\nu}(\mathbf{R},i\tau)\\
&=-i\left[\sum_{i\in\mathcal{U},j\in\mathcal{V}}\sum_{k,\mathbf{R}_1}\sum_{l,\mathbf{R}_2}C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)}G_{l,k}(\mathbf{R}_1-\mathbf{R}_2,-i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})}G_{i,j}(\mathbf{R},i\tau)\right.\\
&\quad+\sum_{i\in\mathcal{U},l\in\mathcal{V}}\sum_{k,\mathbf{R}_1}\sum_{j,\mathbf{R}_2}C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)}G_{l,k}(\mathbf{R}_1-\mathbf{R},-i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{R}_2-\mathbf{R}),l(\mathbf{0})}G_{i,j}(\mathbf{R}_2,i\tau)\\
&\quad+\sum_{k\in\mathcal{U},j\in\mathcal{V}}\sum_{i,\mathbf{R}_1}\sum_{l,\mathbf{R}_2}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_1),k(\mathbf{0})}G_{i,j}(\mathbf{R}-\mathbf{R}_1,i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})}G_{l,k}(-\mathbf{R}_2,-i\tau)\\
&\quad\left.+\sum_{k\in\mathcal{U},l\in\mathcal{V}}\sum_{i,\mathbf{R}_1}\sum_{j,\mathbf{R}_2}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_1),k(\mathbf{0})}G_{i,j}(\mathbf{R}_2-\mathbf{R}_1,i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{R}_2-\mathbf{R}),l(\mathbf{0})}G_{l,k}(-\mathbf{R},-i\tau)\right]\end{aligned}$$

For the second term, $j\leftrightarrow l$

For the third term, $i\leftrightarrow k$

For the forth term, $i\leftrightarrow k, j\leftrightarrow l.$

For Green function, $G_{i,j}(\mathbf{R},i\tau)=G^*_{j,i}(-\mathbf{R},i\tau)$

$$
\begin{aligned}
\chi^0_{\mu,\nu}(\mathbf{R}, i\tau) = -i \Bigg[& \sum_{i\in\mathcal{U},j\in\mathcal{V}} \sum_{k,\mathbf{R}_1} \sum_{l,\mathbf{R}_2} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)} G_{l,k}(\mathbf{R}_1-\mathbf{R}_2, -i\tau) C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{i,j}(\mathbf{R}, i\tau) \\
& + \sum_{i\in\mathcal{U},j\in\mathcal{V}} \sum_{k,\mathbf{R}_1} \sum_{l,\mathbf{R}_2} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)} G_{j,k}(\mathbf{R}_1-\mathbf{R}, -i\tau) C^{\nu(\mathbf{0})}_{l(\mathbf{R}_2-\mathbf{R}),j(\mathbf{0})} G_{i,l}(\mathbf{R}_2, i\tau) \\
& + \sum_{i\in\mathcal{U},j\in\mathcal{V}} \sum_{k,\mathbf{R}_1} \sum_{l,\mathbf{R}_2} C^{\mu(\mathbf{0})}_{k(\mathbf{R}_1),i(\mathbf{0})} G_{k,j}(\mathbf{R}-\mathbf{R}_1, i\tau) C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{l,i}(-\mathbf{R}_2, -i\tau) \\
& + \sum_{i\in\mathcal{U},j\in\mathcal{V}} \sum_{k,\mathbf{R}_1} \sum_{l,\mathbf{R}_2} C^{\mu(\mathbf{0})}_{k(\mathbf{R}_1),i(\mathbf{0})} G_{k,l}(\mathbf{R}_2-\mathbf{R}_1, i\tau) C^{\nu(\mathbf{0})}_{l(\mathbf{R}_2-\mathbf{R}),j(\mathbf{0})} G_{j,i}(-\mathbf{R}, -i\tau) \Bigg] \\
= -i \Bigg[& \sum_{i\in\mathcal{U},j\in\mathcal{V}} \sum_{k,\mathbf{R}_1} \sum_{l,\mathbf{R}_2} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)} \Big(G_{l,k}(\mathbf{R}_1-\mathbf{R}_2, -i\tau) C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{i,j}(\mathbf{R}, i\tau) \\
& + G_{j,k}(\mathbf{R}_1-\mathbf{R}, -i\tau) C^{\nu(\mathbf{0})}_{l(\mathbf{R}_2-\mathbf{R}),j(\mathbf{0})} G_{i,l}(\mathbf{R}_2, i\tau) \\
& + G^*_{j,k}(\mathbf{R}_1-\mathbf{R}, i\tau) C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{i,l}(\mathbf{R}_2, -i\tau) \\
& + G^*_{l,k}(\mathbf{R}_1-\mathbf{R}_2, i\tau) C^{\nu(\mathbf{0})}_{l(\mathbf{R}_2-\mathbf{R}),j(\mathbf{0})} G^*_{i,j}(\mathbf{R}, -i\tau) \Big) \Bigg]
\end{aligned}
$$

$$
\begin{aligned}
\chi^0_{\mu,\nu}(\mathbf{R}, i\tau) = -i \Bigg[& \sum_{i\in\mathcal{U}} \sum_{k,\mathbf{R}_1} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)} \Bigg(\sum_{j\in\mathcal{V}} G_{i,j}(\mathbf{R}, i\tau) \sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{l,k}(\mathbf{R}_1-\mathbf{R}_2, -i\tau) \\
& + \sum_{j\in\mathcal{V}} G^*_{i,j}(\mathbf{R}, -i\tau) \sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G^*_{l,k}(\mathbf{R}_1-\mathbf{R}_2, i\tau) \\
& + \sum_{j\in\mathcal{V}} G_{j,k}(\mathbf{R}_1-\mathbf{R}, -i\tau) \sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{i,l}(\mathbf{R}_2, i\tau) \\
& + \sum_{j\in\mathcal{V}} G^*_{j,k}(\mathbf{R}_1-\mathbf{R}, i\tau) \sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G^*_{i,l}(\mathbf{R}_2, -i\tau) \Bigg) \Bigg] \\
= -i \Bigg[& \sum_{i\in\mathcal{U}} \sum_{k,\mathbf{R}_1} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)} \Big(M^{\nu}_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau) + M^{\nu*}_{i,k}(\mathbf{R}_1, \mathbf{R}, -i\tau) \\
& + Z^{\nu}_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau) + Z^{\nu*}_{i,k}(\mathbf{R}_1, \mathbf{R}, -i\tau) \Big) \Bigg]
\end{aligned}
$$

where

$$M^{\nu}_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau) = \sum_{j\in\mathcal{V}} G_{i,j}(\mathbf{R}, i\tau) N^{\nu}_{j,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$$

$$N^{\nu}_{j,k}(\mathbf{R}_1, \mathbf{R}, i\tau) = \sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{l,k}(\mathbf{R}_1-\mathbf{R}_2, -i\tau)$$

$$Z^{\nu}_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau) = \sum_{j\in\mathcal{V}} G_{j,k}(\mathbf{R}_1-\mathbf{R}, -i\tau) X^{\nu}_{i,j}(textbfR, i\tau)$$

$$X^{\nu}_{i,j}(\mathbf{R}, i\tau) = \sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{i,l}(\mathbf{R}_2, i\tau)$$

Loop structure

Loop τ

Loop R

Loop $\mathcal{U}(0), \mathcal{V}(0)$

Loop $\mathcal{L}(R_2) \in Near[\ \mathcal{V}(0)\ ]$

Calculate $X_{i,j}^{\nu}(\mathbf{R}, i\tau)$

End loop $\mathcal{L}(R_2) \in Near[\ \mathcal{V}(0)\ ]$

Loop $\mathcal{K}(R_1) \in Near[\ \mathcal{U}(0)\ ]$

Loop $\mathcal{L}(R_2) \in Near[\ \mathcal{V}(0)\ ]$

Calculate $N_{j,k}^{\nu}(\mathbf{R}_1, \mathbf{R}, i\tau)$

End loop $\mathcal{L}(R_2) \in Near[\ \mathcal{V}(0)\ ]$

Calculate $M_{i,k}^{\nu}(\mathbf{R}_1, \mathbf{R}, i\tau)$, $Z_{i,k}^{\nu}(\mathbf{R}_1, \mathbf{R}, i\tau)$

End loop $\mathcal{K}(R_1) \in Near[\ \mathcal{U}(0)\ ]$

Calculate temporary variable $O_{\mu,\nu}[\ \mathcal{U}(0), \mathcal{V}(0)\ ]$

End loop $\mathcal{U}(0), \mathcal{V}(0)$

Calculate $\chi_{\mu,\nu}^{0}(\mathbf{R}, i\tau)$

End loop R

End loop τ

```
//*** Calculate  chi0[tau][R](mu,nu) ***

//1.Every process pre-calculate Green function
for tau:
    for R:
        for ik:
            for n_band:
            Green[tau][R](i,j)+=Green[tau][R][ik][n_band](i,j)

//2.According to {i,j}-blocks, search adjacent {K,L}-atoms
set<int> I,J // i->I(U) atoms, j->J(V) atoms
set<int> I_J  // I+J all related atoms set
for U(0):
    search adjcent atoms: map<int,vector<R1>> K;
for V(0):
    search adjcent atoms: map<int,vector<R2>> L;

//3.Pre-calculate Cs
Abfs::cal_Cs( I_J ) -> get Cs[I(0)][K(0)][R](ik,mu) //contain search adjcent atoms in the function

//4.Do all-to-all communication to get and save Green[tau][R](k,l)
for tau:
    for K(0):
        for L(0):
            cal (R1-R2) set
            cal related k,l global_index -> get k,l corresponding processes and local_index
            call and save Green[tau][R1-R2](k,l)

//5.Cal chi0
```

```
for tau:
    for R:
        for [U(0),V(0)]:
            for K(0) in N[U(0)]:
                for R1[K(0)]:
                    for L(0) in N[V(0)]:
                        for R2[L(0)]:
                            Cs[V(0)][L(0)][R2-R](jl,nu) -> reshape Cs[V(0)][L(0)][R2-R](lj,nu)
                            N[J][K](kj,nu)+=Green[tau][R1-R2][K][L](k,l)*Cs[V(0)][L(0)][R2-R](lj,nu) //blas,sum R2, sum L
                            X[I][J](ij,nu)+=Green[tau][R2][I][L](i,l)*Cs[V(0)][L(0)][R2-R](lj,nu)
                        reshape: N(jk,nu),X(ji,nu)
                    M[I][J](i,knu)=Green[tau][R][I][J](i,j)*N[J](jk,nu) //blas,sum j
                    Z[I][J](k,inu)=Green[-tau][R1-R][K][J](k,j)*X(ji,nu) ->reshape Z(i,kmu)
                    O[I][J](ik,nu)=M+M*+Z+Z*
                    Cs[U(0)][K(0)][R1](ik,mu) transpose: Cs[U(0)][K(0)][R1](mu,ik)
                    O_sum[I][J](mu,nu)+=Cs[U(0)][K(0)][R1](mu,ik)*O[I][J][K](ik,nu) //blas, sum R1, sum K
            chi0[tau][R][U(0)][V(0)](mu,nu)=O_sum(mu,nu)
            //chi0[tau][R](U_mu,V_nu)+=chi0[tau][R][U(0)][V(0)](mu,nu)//joint to a big matrix
```

Implementation details

1. calculate $G_{ij}(\mathbf{R}, i\tau)$

$$G_{ij}(\mathbf{R}, i\tau) = \begin{cases} -\frac{1}{N_k}\sum_{\mathbf{k}}\left(\sum_n f_{n\mathbf{k}} c_{i,n}(k) c^*_{j,n}(k) e^{-(\epsilon_{n,k}-\mu)\tau}\right) e^{-i\mathbf{k}\mathbf{R}} & \tau <= 0 \\ \frac{1}{N_k}\sum_{\mathbf{k}}\left(\sum_n (1-f_{n\mathbf{k}}) c_{i,n}(k) c^*_{j,n}(k) e^{-(\epsilon_{n,k}-\mu)\tau}\right) e^{-i\mathbf{k}\mathbf{R}} & \tau > 0 \end{cases}$$

Use 2D density matrix(Actually I rewrite 2D_DM program):
Need

> N_k
> k-point-index
> 2D density matrix
> Energy level, Fermi energy
> τ grid (Minimax grid)
> R grid (generated by k-point-grid, periodic)

1. construct $f_{n\mathbf{k}} e^{-(\epsilon_{n,k}-\mu)\tau}$ matrix as wg_G(ik,ib) [ik=kv.nks, ib<=NLOCAL(=NBNDS)]

> Use wf.wg(ik,ib) to construct wg_G, also need wf.ekb[ik][ib], en.ef
> R grid use

2. use 2D_DM interface:

> Wfc_Dm_2D g;
> g.init();
> g.wfc_k=LOC.wfc_dm_2d.wfc_k;
> g.cal_dm(wg_G) => get dm_K [ik] (i,j);

3. sum k

> $1/N_k * \sum_k dm\_k[ik](i,j) * e^{-kR}$
> need care about k and R grid (Direct or Cartesian coordinates, 2PI or not)

2. Parallel scheme

1. Taking advantage of 2D-block parallel scheme, every process contains part Green function according to initial 2D_DM.

2.The data structure of χ^0 is adjacent atoms depended, and related Green function stored in others processes need to be transported.

3. Use MPI_Alltoall or MPI_Alltollv to communicate.

3. Reshape matrices

In order to efficiently take use of Blas to realize matrix multiplication and reduce calculation, we adjusted matrix index, meaning reshape matrix. We also take memory-saving into consideration when optimize code.

4. calculate $\chi^0_{\mu,\nu}(\mathbf{R}, i\tau)$

$$\chi^0_{\mu,\nu}(\mathbf{R}, i\tau) = -i \left[\sum_{i\in\mathcal{U}} \sum_{k,\mathbf{R}_1} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)} \left(M^{\nu}_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau) + M^{\nu*}_{i,k}(\mathbf{R}_1, \mathbf{R}, -i\tau) \right.\right. \\ \left.\left. + Z^{\nu}_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau) + Z^{\nu*}_{i,k}(\mathbf{R}_1, \mathbf{R}, -i\tau) \right) \right]$$

1. Step by step.

2. At least twice communication to get Green function.

3. Careful memory cost. Free intermediate variable.

5. Fourier transformation (from imaginary time to imaginary frequency)

$$\chi^0(r, r', i\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega e^{-i\omega\tau} \chi^0(r, r', i\omega)$$

$$\chi^0(r, r', i\omega) = \int_{-\infty}^{\infty} d\tau e^{i\omega\tau} \chi^0(r, r', i\tau)$$

note the symmetry:
$\chi^0(r, r', i\omega) = \chi^0(r, r', -i\omega), \quad \chi^0(r, r', i\tau) = \chi^0(r, r', -i\tau)$
and:

$$\chi^0(r,r',i\tau) = \frac{1}{\pi}\int_0^{\infty} d\omega\chi^0(r,r',i\omega)cos(\omega\tau)$$

$$\chi^0(r,r',i\omega) = 2\int_0^{\infty} d\tau\chi^0(r,r',i\tau)cos(\omega\tau)$$

Here we adopt Minimax quadrature which transforms the interval [0,inf) into [0,1]:

$$\chi^0(r,r',i\omega) = 2\sum_{n=1}^{Npoint} d\tau\chi^0(r,r',i\tau(n))cos(\omega\tau)weight(n)$$

6.Π matrix

Option 1:

$$\Pi_{I\mu,J\nu}(R,i\omega) = \sum_{Q\xi}\chi^0_{I\mu,Q\xi}(R,i\omega)V_{Q\xi,J\nu}(R)$$

Option 2:

$$\chi^0_{I\mu,Q\xi}(k,i\omega) = \sum_R e^{-2\pi ikR}\chi^0_{I\mu,Q\xi}(R,i\omega)$$

$$V_{Q\xi,J\nu}(k) = \sum_R e^{-2\pi ikR}V_{Q\xi,J\nu}(R)$$

$$\begin{aligned}\Pi_{I\mu,J\nu}(k,i\omega) &= \sum_{Q\xi}\chi^0_{I\mu,Q\xi}(k,i\omega)V_{Q\xi,J\nu}(k)\\ &= \sum_{R,R'} e^{-2\pi ik(R+R')}\sum_{Q\xi}\chi^0_{I\mu,Q\xi}(R,i\omega)V_{Q\xi,J\nu}(R')\end{aligned}$$

7. MP2 energy

$$E_c^{(2)} = -\frac{1}{8\pi}\int_{-\infty}^{\infty} d\omega\, Tr\{(\chi^0(i\omega)V)^2\}$$

$$E_c^{(2)} = -\frac{1}{4\pi}\sum_{n=1}^{Npoint} Tr(\Pi(i\omega(n)) * \Pi(i\omega(n))^*) * weight(n)$$

$$\Pi^2_{I,J} = \sum_{R,Q}\Pi_{I,QR}\Pi_{QR,J}$$

$$Tr(\Pi^2) = \sum_{R,I} Tr(\Pi_{I,QR} * \Pi_{QR,I})$$

8. RPA correlation energy

$$E_c^{RPA} = \frac{1}{2\pi}\int_0^{\infty} d\omega Tr[ln(1-\chi^0(i\omega)V) + \chi^0(i\omega)V]$$

Define: $\chi^0(i\omega)V = V^{1/2}\chi^0(i\omega)V^{1/2} = \Pi(i\omega)$

$$\begin{aligned}E_c^{RPA} &= \frac{1}{2\pi}\int_0^{\infty} d\omega Tr[ln(1-\Pi(i\omega)) + \Pi(i\omega)]\\ &= \frac{1}{2\pi}\int_0^{\infty} d\omega\{ln[det(1-\Pi(i\omega)] + Tr[\Pi(i\omega)]\}\end{aligned}$$

In abfs:

$$\Pi_{\mu\nu}(i\omega) = \sum_{\nu'} \chi^0_{\mu\nu'}(i\omega) V_{\nu'\nu} = \sum_{\mu',\nu'} V^{1/2}_{\mu\nu'} \chi^0_{\nu'\mu'}(i\omega) V^{1/2}_{\mu'\nu}$$

Again, we adopt Minimax quadrature to discrete the integral:

$$E_c^{RPA} = \frac{1}{2\pi} \sum_{n=1}^{Npoint} \{ln[det(1 - \Pi(i\omega(n))] + Tr[\Pi(i\omega(n))]\} * weight(n)$$

First, calculate $\Pi_{\mu\nu}(i\omega)$ matrix using χ^0 and V:

$$\Pi_{I\mu,J\nu}(i\omega) = \sum_{Q\xi} \chi^0_{I\mu,Q\xi}(i\omega) V_{Q\xi,J\nu}$$

```
for R:
    for I:
        for J:
            for Q:
                Pi[omega][R][I][J](mu,nu)+=chi0[omega][R][I][Q](mu,xi)*V[Q]
[J][R](xi,nu)

```

Another option is diagonalize Coulomb matrix.