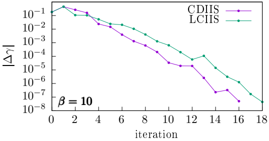
Iterative subspace algorithms for finite-temperature solution of Dyson equation

Pavel Pokhilko

University of Michigan



Matsubara one-particle Green's function

$$\hat{H} = \sum_{pq} h_{pq} p^\dagger q + \frac{1}{2} \sum_{pqrs} \langle pq|rs \rangle p^\dagger q^\dagger s r$$

$$G_{pq}(\tau - \tau') = -\langle T p(\tau) q^\dagger(\tau') \rangle$$

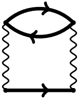
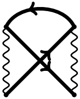
$$\omega_n = \frac{(2n+1)\pi}{\beta}$$

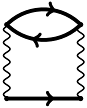
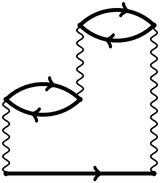
Self-consistent methods:

$$G^{-1}(i\omega_n) = G_0^{-1}(i\omega_n) - \Sigma[G](i\omega_n)$$

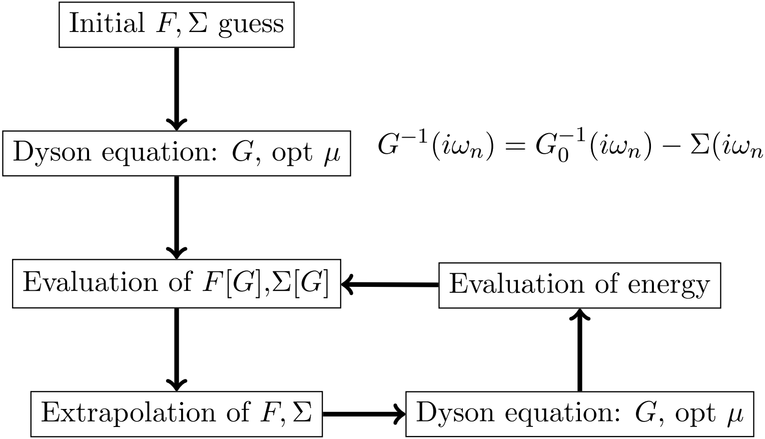
$$G_0^{-1}(i\omega_n) = i\omega_n - \mu\hat{N} - \hat{H}_0$$

Self-consistent approximations

$\Sigma^{(2)}[G] =$  $+$ 

$\tilde{\Sigma}^{GW}[G] =$  $+$  $+ \cdots$

Basic workflow


Initial F, Σ guess
Dyson equation: G, opt μ
Evaluation of $F[G]$,$\Sigma[G]$
Extrapolation of F, Σ
Dyson equation: G, opt μ
Evaluation of energy

$$G^{-1}(i\omega_n) = G_0^{-1}(i\omega_n) - \Sigma(i\omega_n)$$

Algorithms

1. Damping

$$F_i^{mix} = \alpha F_i + (1-\alpha) F_{i-1}$$
$$\Sigma_i^{mix} = \alpha \Sigma_i + (1-\alpha) \Sigma_{i-1}$$

2. Chia-Nan: DIIS + difference residuals
3. Me: DIIS, KAIN, LCIIS
4. Residuals:
   $e_i^d = \Sigma_i - \Sigma_{i-1};$
   $C_{ii}(i\omega) = \left[G_i(i\omega), G_0^{-1}(i\omega) - \Sigma_i(i\omega)\right]$

Original methods:
DIIS: Pulay. Chem. Phys. Lett. 73, 393 (1980); Pulay. J. Comput. Chem. 3, 556 (1982).
KAIN: Harrison. J. Comput. Chem. 25, 328 (2004).

LCIIS: Li, Yaron. J. Chem. Theory Comput. 12, 5322 (2016).

DIIS

$$\mathbf{v}_i = \mathbf{v}_* + \mathbf{e}_i$$

$$\mathbf{v}_{\text{extr}} = \sum_i c_i \mathbf{v}_i = \mathbf{v}_* \sum_i c_i + \sum_i c_i \mathbf{e}_i$$

$$\mathbf{e}_{\text{extr}} = \sum_i c_i \mathbf{e}_i \qquad \sum_i c_i = 1 \qquad ||\mathbf{e}_{\text{extr}}||^2 \to min$$

$$L^{\text{DIIS}}(c_i, \lambda) = \frac{1}{2} \sum_{ij} c_i B_{ij} c_j - \lambda(1 - \sum_i c_i)$$

$$B_{ij} = \langle \mathbf{e}_i, \mathbf{e}_j \rangle$$

$$\begin{pmatrix} \mathrm{Re}(B_{11}) & \cdots & \mathrm{Re}(B_{1n}) & 1 \\ \cdots & \cdots & \cdots & \\ \mathrm{Re}(B_{1n}) & \cdots & \mathrm{Re}(B_{nn}) & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \cdots \\ c_n \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \cdots \\ 0 \\ 1 \end{pmatrix}$$

Pulay. Chem. Phys. Lett. 73, 393 (1980); Pulay. J. Comput. Chem. 3, 556 (1982).

LCIIS

$$\mathbf{\Sigma}_{\text{extr}} = \sum_i c_i \mathbf{\Sigma}_i \qquad \mathbf{G}_{\text{extr}} = \sum_i c_i \mathbf{G}_i$$

$$C(i\omega) = \left[G_{\text{extr}}(i\omega), G_0^{-1}(i\omega) - \Sigma_{\text{extr}}(i\omega)\right]$$

$$\sum_i c_i = 1 \qquad f(c) = ||C||^2 \rightarrow min$$

$$L^{\text{LCIIS}}(c_i, \lambda) = f(c) - \lambda(1 - \sum_i c_i)$$

Optimization: Newton method (with backtracking line search) + tangent gradient projection

Li, Yaron. J. Chem. Theory Comput. 12, 5322 (2016)

KAIN

Goal: $\mathbf{f}(\mathbf{v}_*) = \mathbf{0}$

$$\mathbf{v}_{\text{new}} = \mathbf{v} + \Delta\mathbf{v}$$
$$\mathbf{F}\Delta\mathbf{v} = -\mathbf{f}$$
$$\mathbf{F} = \nabla\mathbf{f}$$

$$\mathbf{f}(\mathbf{v}_i) = \mathbf{f}(\mathbf{v}_n + \mathbf{v}_i - \mathbf{v}_n) \approx \mathbf{f}(\mathbf{v}_n) + (\nabla\mathbf{f}(\mathbf{v}_n))(\mathbf{v}_i - \mathbf{v}_n)$$
$$\mathbf{F}_n(\mathbf{v}_i - \mathbf{v}_n) \approx \mathbf{f}_i - \mathbf{f}_n$$

$\mathbf{P}$ projects onto $\mathbf{v}_i - \mathbf{v}_n$

$$\mathbf{F}_n\mathbf{P}\Delta\mathbf{v} + (1 - \mathbf{P})\Delta\mathbf{v} = -\mathbf{f}_n$$

KAIN

$$\mathbf{F}_n\mathbf{P}\Delta\mathbf{v} + (1 - \mathbf{P})\Delta\mathbf{v} = -\mathbf{f}_n$$

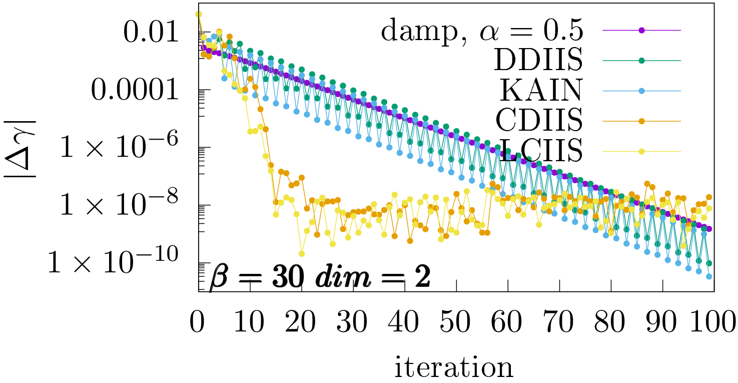
$$\mathbf{P}\Delta\mathbf{v}_n = \sum_{i}^{n-1} c_i(\mathbf{v}_i - \mathbf{v}_n)$$

$$Ac = b$$

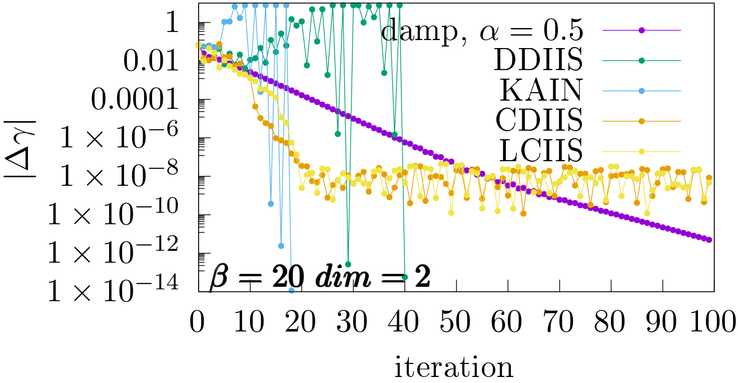
$$A_{ij} = \langle \mathbf{v}_i - \mathbf{v}_n | \mathbf{f}_j - \mathbf{f}_n \rangle$$

$$b_i = -\langle \mathbf{v}_i - \mathbf{v}_n | \mathbf{f}_n \rangle$$

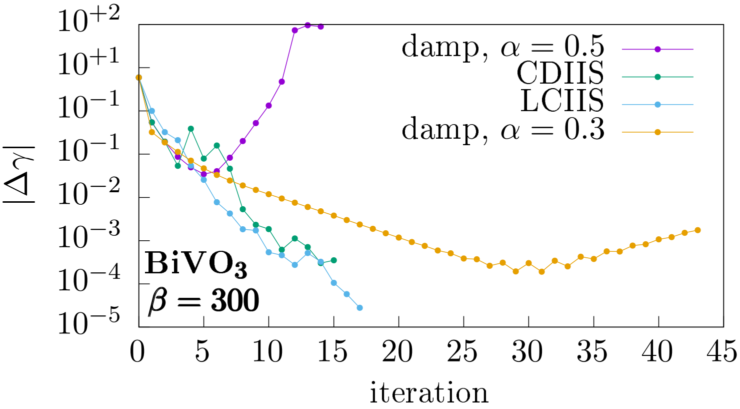
$$(1 - \mathbf{P})\Delta\mathbf{v} = -\left(\sum_{i=1}^{n-1} (\mathbf{f}_i - \mathbf{f}_n) c_i + \mathbf{f}_n \right)$$



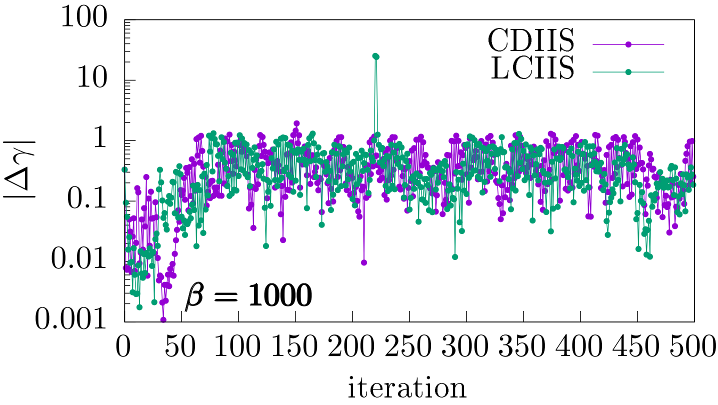
Be atom, GW/cc-pVDZ



Be atom, GW/cc-pVDZ

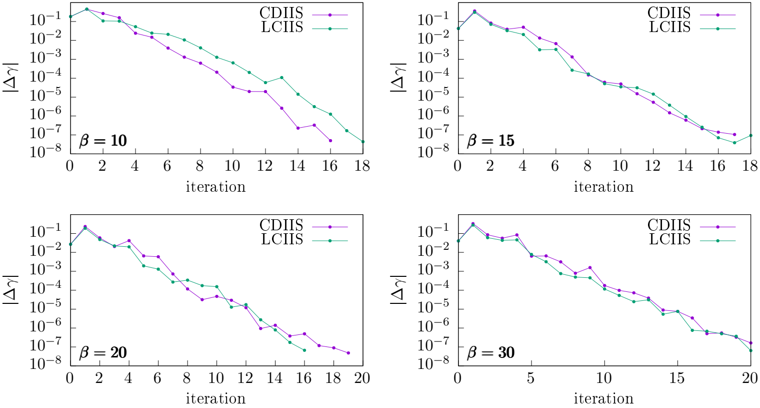


BiVO3, GW, start from PBE0.

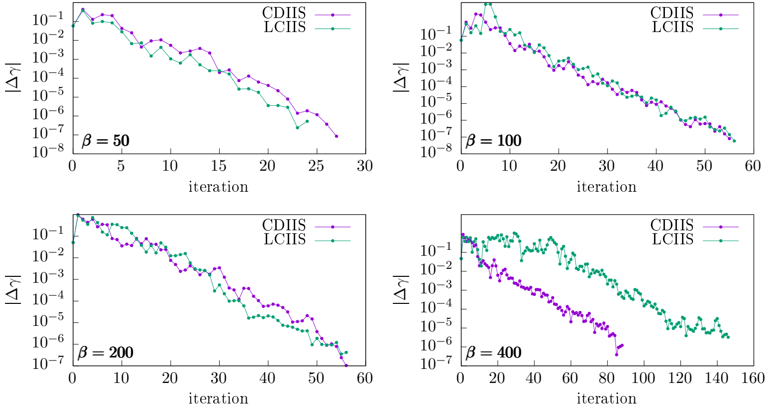


H_2, RGF2/cc-pVDZ, r(HH) = 3.15 Å

What can we do?



N2, RGF2/cc-pVDZ, r(NN) = 3.15 Å



N2, RGF2/cc-pVDZ, r(NN) = 3.15 Å

More:

Pokhilko, Yeh, Zgid. J. Chem. Phys., 2022, 156, 094101

Practice

Deployed within UGF2 suite for HF, GW, GF2 for solids and molecules.

Limitations:

- No MPI parallelization (does not seems to be a big deal)
- No full restart (at restart the subspace will be re-built)
- Microiterations are not very well tested

Practice: DIIS

Chia-Nan's code (DIIS with difference residuals):

```
DIIS_size - subspace size
DIIS_start - when to switch to DIIS
DIIS_interval - perform DIIS extrapolation only at certain iterations
```

My code:

```
new_DIIS = true - activate my DIIS implementation
com_DIIS = true - use commutator residuals
max_trust_rad - trust region for large coefficients for rescaling
DIIS_size
DIIS_start
```

Recommendation: use

```
new_DIIS = true
com_DIIS = true
DIIS_size = 2--8
DIIS_start = 3 (early start is useful from UHF starting point)
max_trust_rad - usually not useful at all
```

Practice: DIIS, NiO example

```
itermax=100
rst=true
scf_type=0
IR=true
CONST_DENSITY=true
E_thr=1e-7
tol=1e-10
damp=0.3
max_trust_rad=100000
new_DIIS=true
com_DIIS=true
DIIS_size=4
DIIS_start=5
nel_cell=48
nao=78
nk=27
ink=14
ns=2
mu=0.0
NQ=382
```

Practice: LCIIS

```
LCIIS = true - use LCIIS
LCIIS_thr_dir - DM diff value switching back to damping
DIIS_size = 2--4
DIIS_start
mod_LCIIS - do not use (to be removed)
```

Recommendation: use smaller subspaces, since the number of commutators scales as dim^4;
do not mix LCIIS = true and new_DIIS = true or com_DIIS = true

Practice: KAIN

```
KAIN = true - activates KAIN
```

Since the formulation permits only difference residuals, usage of the current implementation of KAIN is not recommended.

Practice: how to restart?

Remove the temporary files: new_diis_vectors.h5, new_diis_residuals.h5, new_diis_g.h5, *_micro.h5, commutators.h5

Make sure not to delete sim.h5!

Use restart=true

Use early DIIS_start = 2

estimate damping from previous extrapolation coefficients $\alpha \approx c_n$ (DIIS, LCIIS, if $c_n > 0$)

Microiterations

Within each iteration, do the full update, extrapolation; then
Freeze the dynamical self-energy, run only updates for the Fock matrix.

Status: experimental

Sometimes it works, but sometimes it does not!

```
micro_itermax=10 -- the number of microiterations for each normal iteration
micro_DIIS_size=5 -- the subspace for CDIIS for microiterations
```