

The SCF Workshop Notes 3

Lucas Aebersold

May 17, 2018

3. Direct Inversion of the Iterative Subspace

When solving systems of linear (or nonlinear) equations, iterative methods are often employed. Unfortunately, such methods often suffer from convergence issues such as numerical instability, slow convergence, and significant computational expense when applied to difficult problems. In these cases, convergence acceleration methods may be applied to both speed up, stabilize and/or reduce the cost for the convergence patterns of these methods, so that solving such problems become computationally tractable. One such method is known as the direct inversion of the iterative subspace (DIIS) method, which is commonly applied to address convergence issues within self consistent field computations in Hartree-Fock theory (and other iterative electronic structure methods). In this tutorial, we'll introduce the theory of DIIS for a general iterative procedure, before integrating DIIS into our previous implementation of RHF.

3.1. Theory

DIIS is a convergence acceleration method, which is applicable to numerous problems in linear algebra and the computational sciences, in particular quantum chemistry. Therefore, we will introduce the theory of this method in the general sense, before seeking to apply it to SCF.

Suppose that for a given problem, there exist a set of trial vectors $\{|\mathbf{g}_i\rangle\}$ which have been generated iteratively, converging toward the true solution, $|\bar{\mathbf{g}}\rangle$. Then the true solution can be approximately constructed as a linear combination of the trial vectors,

$$|\mathbf{g}\rangle = \sum_i c_i |\mathbf{g}_i\rangle$$

where we require that the residual vector

$$|\mathbf{r}\rangle = \sum_i c_i |\mathbf{r}_i\rangle \quad |\mathbf{r}_i\rangle = |\mathbf{g}_{i+1}\rangle - |\mathbf{g}_i\rangle$$

is a least-squares approximate to the zero vector, according to the constraint

$$\sum_i c_i = 1.$$

This constraint on the expansion coefficients can be seen by noting that each trial function \mathbf{g}_i may be represented as an error vector applied to the true solution, $|\bar{\mathbf{g}}\rangle + |\mathbf{e}_i\rangle$. Then

$$|\mathbf{g}\rangle = \sum_i c_i |\mathbf{g}_i\rangle \tag{3.1}$$

$$= \sum_i c_i (|\bar{\mathbf{g}}\rangle + |\mathbf{e}_i\rangle) \tag{3.2}$$

$$= |\bar{\mathbf{g}}\rangle \sum_i c_i + \sum_i c_i |\mathbf{e}_i\rangle \tag{3.3}$$

Convergence results in a minimization of the error (causing the second term to vanish). In order for the DIIS solution vector $|\mathbf{g}\rangle$ and the true solution vector $|\bar{\mathbf{g}}\rangle$ to be equal, it must be true that $\sum_i c_i = 1$. We satisfy our

condition for the residual vector by minimizing its norm,

$$\langle \mathbf{r} | \mathbf{r} \rangle = \sum_{ij} c_i^* c_j \langle \mathbf{r}_i | \mathbf{r}_j \rangle$$

using Lagrange's method of undetermined coefficients subject to the constraint on $\{c_i\}$:

$$\mathcal{L} = \mathbf{c}^\dagger \mathbf{B} \mathbf{c} - \lambda \left(1 - \sum_i c_i \right)$$

where $B_{ij} = \langle \mathbf{r}_i | \mathbf{r}_j \rangle$ is the matrix of residual vector overlaps. Minimization of the Lagrangian with respect to the coefficient c_k yields (for real values)

$$\frac{\partial \mathcal{L}}{\partial c_k} = 0 = \sum_j c_j B_{jk} + \sum_i c_i B_{ik} - \lambda \quad (3.4)$$

$$= 2 \sum_i c_i B_{ik} - \lambda \quad (3.5)$$

which has matrix representation

$$\begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1m} & -1 \\ B_{21} & B_{22} & \cdots & B_{2m} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ B_{n1} & B_{n2} & \cdots & B_{nm} & -1 \\ -1 & -1 & \cdots & -1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix} \quad (3.6)$$

which is referred to as the Pulay equation, named after the inventor of DIIS. It is worth noting that at this point our trial vectors, residual vectors, and solution vector may in fact be tensors of arbitrary rank. For this reason that we have used the generic notation of Dirac in the above discussion to denote the inner product between such objects.

3.2. Algorithms for DIIS

The general DIIS procedure, as described above, has the following structure during each iteration:

3.2.1 A Generic DIIS procedure

1. Compute new trial vector, $|\mathbf{g}_{i+1}\rangle$, append to list of trial vectors
2. Compute new residual vector, $|\mathbf{r}_{i+1}\rangle$, append to list of trial vectors
3. Check convergence criteria
 - If RMSD of $|\mathbf{r}_{i+1}\rangle$ sufficiently small, and
 - If change in DIIS solution vector $|\mathbf{g}\rangle$ is sufficiently small, break
4. Build \mathbf{B} matrix from previous residual vectors
5. Solve Pulay equation for coefficients $\{c_i\}$
6. Compute DIIS solution vector $|\mathbf{g}\rangle$

For SCF iteration, the most common choice of trial vector is the Fock matrix \mathbf{F} ; this choice has the advantage over other potential choices (e.g., the density matrix \mathbf{P}) of \mathbf{F} not being idempotent, so that it may benefit from

extrapolation. The residual vector is commonly chosen to be the orbital gradient in the AO basis,

$$g_{\mu\nu} = (\mathbf{F}\mathbf{P}\mathbf{S} - \mathbf{S}\mathbf{P}\mathbf{F})_{\mu\nu}$$

however the better choice (which we will make in our implementation!) is to orthonormalize the basis of the gradient with the inverse overlap metric $\mathbf{A} = \mathbf{S}^{-1/2}$

$$r_{\mu\nu} = (\mathbf{A}^T(\mathbf{F}\mathbf{P}\mathbf{S} - \mathbf{S}\mathbf{P}\mathbf{F})\mathbf{A})_{\mu\nu}$$

Therefore, the SCF-specific DIIS procedure (integrated into the SCF iteration algorithm) will be

3.2.2 DIIS within an SCF Iteration

1. Compute \mathbf{F} , append to list of previous trial vectors
2. Compute AO orbital gradient \mathbf{r} , append to list of previous residual vectors
3. Compute RHF energy
4. Check convergence criteria
 - If RMSD of \mathbf{r} sufficiently small, and
 - If change in SCF energy sufficiently small, break
5. Build \mathbf{B} matrix from previous AO gradient vectors
6. Solve Pulay equation for coefficients $\{c_i\}$
7. Compute DIIS solution vector $\mathbf{F_diis}$ from $\{c_i\}$ and previous trial vectors
8. Compute new orbital guess with $\mathbf{F_diis}$

Now we're ready to write our SCF iterations in accordance to the above procedure.

3.3. Starting DIIS

Since DIIS builds the approximate solution vector $|\mathbf{g}\rangle$ as a linear combination of the previous trial vectors $\{|\mathbf{g}_i\rangle\}$, DIIS must start in the second SCF iteration.

3.3.1 Building \mathbf{B}

1. The \mathbf{B} matrix in the Lagrange equation is really $\tilde{\mathbf{B}} = \begin{pmatrix} \mathbf{B} & -1 \\ -1 & 0 \end{pmatrix}$.
2. Since \mathbf{B} is the matrix of residual overlaps, it will be a square matrix of dimension equal to the number of residual vectors. If \mathbf{B} is an $N \times N$ matrix, how big is $\tilde{\mathbf{B}}$?
3. Since our residuals are real, \mathbf{B} will be a symmetric matrix.
4. To build $\tilde{\mathbf{B}}$, make an empty array of the appropriate dimension, then use array indexing to set the values of the elements.

3.3.2 Solving the Pulay equation

1. Use built-in NumPy functionality to make your life easier.
2. The solution vector for the Pulay equation is $\tilde{\mathbf{c}} = \begin{pmatrix} \mathbf{c} \\ \lambda \end{pmatrix}$, where λ is the Lagrange multiplier, and the right hand side is $\begin{pmatrix} \mathbf{0} \\ -1 \end{pmatrix}$.

4. Unrestricted Hartree Fock

Here is the briefest of overviews for the unrestricted form of the SCF equations. Essentially we need to compute all previous matrix elements but for both the alpha and beta electrons, the coupled equations look like

$$\mathbf{F}^\alpha \mathbf{C}^\alpha = \mathbf{S} \mathbf{C}^\alpha \boldsymbol{\epsilon}^\alpha \quad (4.1)$$

$$\mathbf{F}^\beta \mathbf{C}^\beta = \mathbf{S} \mathbf{C}^\beta \boldsymbol{\epsilon}^\beta \quad (4.2)$$

which are the unrestricted generalizations of the restricted Roothan equations, called the Pople-Nesbitt equations. Here, the one-electron Fock matrices are given by

$$F_{\mu\nu}^\alpha = H_{\mu\nu} + (\mu\nu|\lambda\sigma) [P_{\lambda\sigma}^\alpha + P_{\lambda\sigma}^\beta] - (\mu\lambda|\nu\sigma) P_{\lambda\sigma}^\beta \quad (4.3)$$

$$F_{\mu\nu}^\beta = H_{\mu\nu} + (\mu\nu|\lambda\sigma) [P_{\lambda\sigma}^\alpha + P_{\lambda\sigma}^\beta] - (\mu\lambda|\nu\sigma) P_{\lambda\sigma}^\alpha \quad (4.4)$$

where the density matrices $P_{\lambda\sigma}^\alpha$ and $P_{\lambda\sigma}^\beta$ are given by

$$P_{\lambda\sigma}^\alpha = C_{\sigma i}^\alpha C_{\lambda i}^\alpha \quad (4.5)$$

$$P_{\lambda\sigma}^\beta = C_{\sigma i}^\beta C_{\lambda i}^\beta \quad (4.6)$$

Unlike for RHF, the orbital coefficient matrices \mathbf{C}^α and \mathbf{C}^β are of dimension $M \times N^\alpha$ and $M \times N^\beta$, where M is the number of AO basis functions and N^α (N^β) is the number of α (β) electrons. The total UHF energy is given by

$$E_{\text{total}}^{\text{UHF}} = E_e^{\text{UHF}} + E_n^{\text{BO}} \quad \text{with} \quad (4.7)$$

$$E_e^{\text{UHF}} = \frac{1}{2} [(\mathbf{P}^\alpha + \mathbf{P}^\beta) \mathbf{H} + \mathbf{P}^\alpha \mathbf{F}^\alpha + \mathbf{P}^\beta \mathbf{F}^\beta]. \quad (4.8)$$