DIIS SCF Methods and BFGS Optimization

Accelerating Convergence to Finding Fixed Points

Trevor Oldham

Three SCF Algorithms

- `void DIIS_FP(molecule & mol, double tolerance, int history, bool verbose);`
  - A mixture of DIIS and FP methods. Start with FP iterations and collect past Fock matrices until max_history is reached, then switch to DIIS
- `void DIIS(molecule & mol, double tolerance, int max_history, bool verbose);`
  - A pure DIIS implementation. Start immediately using the DIIS algorithm to build the Fock matrices
- `void fixed_point_iteration(molecule & mol, double tolerance, bool verbose);`
  - The method we used in the problems sets this semester.

DIIS: Direct Inversion of the Iterative Subspace

Motivations

- Extrapolate the solution to a set of linear equations by minimizing the error residual
- The newest residual vector is a linear combination of the previous error vectors on the condition that the coefficients must sum to one
- Using Lagrange multipliers construct the DIIS matrix and solve for coefficients
- Construct new Fock matrix as a linear combination of m most recent Fock matrices

$$B_{ij} = e_i \cdot e_j \qquad F^{\star} = \sum_{i=0}^{m} c_i F_i$$

$$\begin{bmatrix} B_{11} & B_{12} & B_{13} & \dots & B_{1m} & 1 \\ B_{21} & B_{22} & B_{23} & \dots & B_{2m} & 1 \\ B_{31} & B_{32} & B_{33} & \dots & B_{3m} & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & B_{m3} & \dots & B_{mm} & 1 \\ 1 & 1 & 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_m \\ -\lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

DIIS: Direct Inversion of the Iterative Subspace

The Algorithm

1. Perform FP iterations until **m** past Fock matrices are saved. Save $\mathbf{e_i} = F_\alpha P_\alpha - P_\alpha F_\alpha$ at each iteration

2. Save P_α and P_β

3. Build DIIS matrix $\mathbf{A}$ using B_{ij} values computed from past $\mathbf{e_i}$

4.Solve $\mathbf{y} = \mathbf{Ax}$ for $\mathbf{x} = [c_0, c_1, c_2, \ldots, c_m, -\lambda]$

5. Compute $F^\star = \sum_{i=0}^{m} c_i F_i$

6.Solve symmetric eigenvalue problem $F^\star C^{i+1} = SC^i E$ for new coefficient matrix $C^{(i+1)}$

7.Compute P_α, P_β and save $F^\star, e_i, P_\alpha, P_\beta$

8. Continue 3 - 7 until $P_\alpha^i - P_\alpha^{i+1} < tol$

DIIS: Direct Inversion of the Iterative Subspace

Results

- The DIIS/FP method does accelerate convergence
- The DIIS method alone converges too quickly
- Neither method returns the same value found from FP iteration
- The FP method gets very close to the solution, and DIIS makes smaller adjustments toward the fixed point
- The DIIS method is very dependent on the number M.

BFGS: Broyden–Fletcher–Goldfarb–Shanno

Motivations

- Minimize $f(x)$ with initial guess $x_0 \in \mathbb{R}^n$
- Choose descent direction using the negative gradient
- Perform line search in direction of the negative gradient
- Approximate the Hessian B_k using gradient and curvature information
- Complexity $O(n^2)$

BFGS: Broyden–Fletcher–Goldfarb–Shanno

Algorithm

1. Start with $B_k = I$
2. Compute search direction $\mathbf{p_k} = B_k^{-1} \cdot (-\nabla(f(\mathbf{x}))$
3. Perform line search for scalar α_k such that $\alpha_k = argmin(f(\mathbf{x_k}) + \alpha_k \mathbf{p_k})$ using Wolfe Conditions
4. Compute $\mathbf{s}_k = \alpha_k \mathbf{p_k}$ and $\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k \mathbf{p_k}$
5. Compute $\mathbf{y_k} = \nabla(f\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$
6. Approximate $B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k^T}{s_k^T B_k s_k}$
7. Continue 2 - 6 until $||\nabla f(\mathbf{x_k})|| < \epsilon$

•

BFGS: Broyden–Fletcher–Goldfarb–Shanno

Results

- Still in the debugging stage - but some interesting results!
- Slow: from performing SCF iterations after each $\mathbf{x_k}$ update
- Unstable: B_k can be singular meaning arma::solve() will not work. Must use arma::pinv()
- Results dependent on line search parameters
- Energy does descrease despite the gradient increasing
- Does not coverge before maximum number of iterations is reached
- Atoms stay in similar geometry but get very far apart
- Final Energy approaches the exact value of the electron energy term as nuclear repulsion approaches 0

References

- https://en.wikipedia.org/wiki/Broyden-Fletcher-Goldfarb-Shanno_algorithm
- https://en.wikipedia.org/wiki/Wolfe_conditions
- https://medium.com/@tru11631/l-bfgs-algorithm-ecfb832a4176
- https://en.wikipedia.org/wiki/DIIS
- https://manual.q-chem.com/5.3/sect_diis.html