
Density Mixing Notes

Nick Woods

March 2017

Contents

Contents	i
1 Introduction	1
2 Theory	2
2.1 Marks and Luke Implementation	2
2.1.1 Y_n and S_n	2
2.1.2 Ψ_n	3
2.1.3 The Initial Matrix A_0	4
2.1.4 σ_n	4
2.1.5 A_n	5
2.1.6 A Note on Preconditioning	5
3 Application	6
3.1 CASTEP	6
References	7

Introduction

Theory

2.1 Marks and Luke Implementation

The ML update equation is,

$$\rho_{n+1}^{\text{in}} = \rho_n^{\text{in}} - A_{n,0}(I - Y_{n-1}A_n)g_n - S_{n-1}A_ng_n. \quad (2.1)$$

How is each vector/matrix now constructed from information in CASTEP?

2.1.1 Y_n and S_n

These are extensions of a set of more fundamental objects, y_n and s_n . In conventional ‘single’ secant Broyden methods, y_n and s_n are,

$$s_n = \rho_n^{\text{in}} - \rho_{n-1}^{\text{in}}, \quad (2.2)$$

$$y_n = R_n - R_{n-1}, \quad (2.3)$$

with R_n the conventionally defined residual. The *most recent* secant condition at iteration n used to build a Jacobian B_n then takes the form $B_ns_{n-1} = y_{n-1}$. The idea of ML is to implement a *multisecant* Broyden method. This means one enforces B_n to satisfy the secant condition from a history of iterations, rather than only the most recent secant equation. The logic is, there is no *a priori* reason to suggest that using ρ from the previous iteration (compared to ρ_n) provides a ‘better’ secant condition than ρ_n compared to any of the previous iterations of ρ_j in the history. Therefore, when iterating through a multisecant technique, each iteration adds to a history of iterations, and the update is determined by using (equally) all information in the history, rather than just the most recent. I.e. multisecant techniques are more of a ‘sample’ of a high dimensional space, rather than a step-by-step series (the previous step uniquely determining the next). To implement this process, the vectors above need to be extended to matrices. Currently, s_n and y_n are k -dimensional vectors ($k = N_{\text{gridpoints}}$). We now create a new matrix

containing the history,

$$S_n = [s_{n-1-m,n}, s_{n-m,n}, \dots, s_{n-1,n}] \quad (2.4)$$

$$Y_n = [y_{n-1-m,n}, y_{n-m,n}, \dots, y_{n-1,n}] \quad (2.5)$$

where m is a parameter called the *history length*. $s_{j,n}$ (and the same for y hereafter) is defined as $s_{j,n} = \rho_j - \rho_n$. This has been assuming we are on an iteration n larger than the history length m (the process for when we are on iterations smaller than the specified m is as expected). The matrices S_n and Y_n are now of dimension $k \times m$ and satisfy $B_n S_{n-1} = Y_{n-1}$. To give an example, imagine we are on iteration 11, and our history length is 5, the vectors S and Y are,

$$S_{n-1} = [\rho_4 - \rho_{10}, \rho_5 - \rho_{10}, \dots, \rho_9 - \rho_{10}] \quad (2.6)$$

$$Y_{n-1} = [R_4 - R_{10}, R_5 - R_{10}, \dots, R_9 - R_{10}] \quad (2.7)$$

and the Jacobian update satisfies the following ‘history’ of equations,

$$B_{11}(\rho_4 - \rho_{10}) = R_4 - R_{10} \quad (2.8)$$

$$B_{11}(\rho_5 - \rho_{10}) = R_5 - R_{10}$$

$$B_{11}(\rho_6 - \rho_{10}) = R_6 - R_{10}$$

$$\vdots$$

$$B_{11}(\rho_9 - \rho_{10}) = R_9 - R_{10}.$$

2.1.2 Ψ_n

Another object required in constructing A_n is Ψ_n . This is applied as it was suggested that it is numerically advantageous to normalise Y_n and S_n . This has no effect on the behaviour of the scheme, but affects things such as the regularisation parameters. In the interest of following the ML paper (and using their regularisation parameters), Ψ is introduced to normalise Y and S , and it is defined as such,

$$\Psi_n = \text{Diag}\left(\frac{1}{\|y_{(n-2-m),n-1}\|}, \frac{1}{\|y_{(n-1-m),n-1}\|}, \dots, \frac{1}{\|y_{n-2,n-1}\|}\right). \quad (2.9)$$

This is an $m \times m$ matrix. In the example above ($n = 11$ and $m = 5$),

$$\Psi_n = \begin{bmatrix} \frac{1}{\|y_{5,10}\|} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\|y_{6,10}\|} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\|y_{7,10}\|} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\|y_{8,10}\|} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\|y_{9,10}\|} \end{bmatrix}$$

2.1.3 The Initial Matrix A_0

A_0 can be thought of as similar to the initial guess of the Jacobian. Depending on the problem, it might be best to choose A_0 as a finite difference approximation. To populate A_0 in this fashion using DFT would be difficult, due to the complexity of the KS DFT scheme. A_0 here is chosen to be an appropriately chosen scaling of the identity. A_0 however serves as more than just an initial guess. The MS update initially stated can be split up into an update in the direction of R_n (the residual at the current step), and an update in all directions orthogonal to R_n . The ML method (and even standard Broyden) gives a natural prediction for how best to update in the direction of R_n when producing the next iteration ρ_{n+1} – ML calls this the ‘predicted’ direction. All directions orthogonal to this are therefore in the ‘unpredicted’ direction. Controlling the step length in the unpredicted directions is crucial (a source of instabilities), and iteration dependent scalings of A_0 can do this. One defines $A_{n,0}$ as,

$$A_{n,0} = \sigma_n I. \quad (2.11)$$

σ_n is therefore an iteration (n) dependent ‘step length’ parameter in the unpredicted direction. σ_n is a real number, and I is of dimension $k \times k$.

2.1.4 σ_n

This is defined the real number,

$$\sigma_n = \min\{\tilde{\sigma}_n, \frac{R|p_n|}{|R_n|}, \bar{\sigma}\} \quad (2.12)$$

First, it is important to note that this scaling is bound by the quantity $\frac{R|p_n|}{|R_n|}$. R is a parameter of the ML model (0.05 – 0.15), and the meaning of this fraction is that it is the ‘magnitude’ of the step length in the predicted direction. p_n is the full predicted direction vector, and R_n is the direction of the predicted vector (although it possesses

a magnitude). So $\frac{R|p_n|}{|R_n|}$ is a normalised step size in the predicted direction scaled by R . In words, we want the step size in the unpredicted direction to be bounded by a scaled (R) version of the normalised magnitude of the step direction in the predicted direction. $\bar{\sigma}$ is simply another parameter of the system ($0.1 - 0.2$). Finally,

$$\tilde{\sigma}_n = \sigma_{n-1} \times \max\{0.5, \min\{2, \frac{|R_{n-1}|}{|R_n|}\}\} \quad (2.13)$$

In words, this is a statment of ‘if the residual on the next iteration is larger, we make the step in the unpredicted direction smaller’ (or is it??). The three elements in 2.12 are three levels of control to keep the step in the unpredicted direction low, but sensible (‘sensible’ in this context comes solely from numerical experience of ML in wein2k). The only thing left is how to decide σ_0 . A generic ‘small’ step is made,

2.1.5 A_n

A_n is most of the ML scheme (the solution to the multisecant optimisation problem as they present it) and is defined as,

$$A_n = \dots \quad (2.14)$$

MSB1 and MSB2 as defined in paper. These are $m \times k$ matrices, where the only unknown now is the parameter α , which is a further regularisation parameter to avoid divergence (from negative e’vals??). $\alpha > 10^{-6}$ and in ML test cases $\alpha = 10^{-4}$.

2.1.6 A Note on Preconditioning

In ML they describe a scheme for preconditioning specific to APW methods. This scheme rescales Y , S , A etc appropriately from the muffin tin approximation. Their preconditioner relies on ratios of residuals in the muffin tin region, and interstitial region. There is no such regions in CASTEP, thus treat the precondition as I for now. How do the dielectric preconditioners map onto where they have placed their preconditioners? investigation..

Application

3.1 CASTEP

References

- [1] P. H. Dederichs and R. Zeller. Self-consistency iterations in electronic-structure calculations. *Phys. Rev. B* 28, 5462, (1983).
- [2] X. Gonze. Towards a potential-based conjugate gradient algorithm for order-n self-consistent total energy calculations. *Phys. Rev. B* 54, 4383, (1996).
- [3] P.-M. Anglade and X. Gonze. Preconditioning of self-consistent-field cycles in density-functional theory: The extrapolar method. *Phys. Rev. B* 78, 045126, (2008).
- [4] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.* 19, 577-593, (1965).