Proceedings of ICCES'08, pp.677-683

A Linearly Constrained Least-Squares Problem in Electronic Structure Computations

Peng Ni[1], Homer Walker[1]

Summary

One of the fundamental problems in electronic structure calculations is to determine the electron density associated with the minimum total energy of a molecular or bulk system. The total energy minimization problem is often formulated as a nonlinear eigenvalue problem. The most widely used algorithm for solving this type of problem is the self-consistent field (SCF) iteration accelerated by Direct Inversion on the Iterative Subspace (SCF-DIIS), in which a linearly constrained least-squares problem is embedded. We will examine and compare the numerical stability of three different ways to solve this least-squares problem.

Introduction

In electronic structure computations, we are involved in solving nonlinear eigenvalue problems (see details in [5]). These have the discretized form

$$H(X)X = X\Lambda_k, \qquad X'X = I_k,$$

where the columns of $X \in \mathbb{R}^{n\times k}$ $(k < n)$ are approximate electron wave functions, $H(X) \in \mathbb{R}^{n\times n}$ is the discrete Hamiltonian, $\Lambda \in \mathbb{R}^{k\times k}$ is a diagonal matrix with the k smallest eigenvalues of $H(X)$ on the diagonal, and $I_k \in \mathbb{R}^{k\times k}$ is the identity matrix. One of the most successful approaches to numerically solving this nonlinear eigenvalue problem is the SCF-DIIS method [5].

A linearly constrained least-squares problem is embedded in the SCF-DIIS method:

$$\min_{\alpha} \|D\alpha\| \ \text{ s.t.} \sum_{i=1}^{k} \alpha_i = 1, \tag{1}$$

where $D \in \mathbb{R}^{n\times k}$, $\alpha = (\alpha_1, \ldots, \alpha_k)^T \in \mathbb{R}^{k\times 1}$ and $\|\cdot\|$ is the Euclidean norm. The matrix D is initially a single vector ($k = 1$) and varies from one SCF-DIIS iteration to the next (with k changing accordingly) by adding a column on the right to incorporate new information and, if necessary, also dropping one or more columns on the left, either to keep k from exceeding a practical bound or to keep D acceptably well-conditioned.

There are several approaches to solving the least-squares problem. In this report, we outline and compare three of these: (1) the *Lagrange-multiplier method*; (2) a new, problem-specific implementation of the *null-space method*; and (3) an

[1] Worcester Polytechnic Institute, Worcester, MA 01609 USA.

improved implementation of a *method of elimination* used previously for this problem [3]. The Lagrange-multiplier method is a commonly used technique for constrained optimization (see, e.g., [4]). The null-space method and the method of elimination are general approaches to solving linearly constrained least-squares problems (see, e.g., [1]); in the present context, each of these reduces the constrained problem on $\mathbb{R}^k$ to an unconstrained problem on $\mathbb{R}^{k-1}$.

In the following, we describe these methods and their properties, noting in particular the conditioning of the linear systems that must be solved and the number of floating-point operations (or *flops*) that are needed for implementation in the SCF-DIIS context. We conclude with an illustrative case study in a particular application, followed by a summary discussion. Throughout the following, we define the condition number of a matrix M by $\kappa(M) \equiv \max_{\|v\|=1} \|Mv\| / \min_{\|v\|=1} \|Mv\|$. This serves as a fundamental indicator of the accuracy that can be obtained in a numerical solution of a linear system or least-squares problem, with more accuracy resulting with smaller condition numbers (see, e.g., [2]).

Methods

The Lagrange-multiplier method. In this method (see general description in [1]), we set

$$\Phi(\alpha,\lambda) \equiv \frac{1}{2}|D\alpha|^2 - \lambda(\sum \alpha_i - 1) = \frac{1}{2}\alpha^T D^T D\alpha - \lambda(\sum \alpha_i - 1).$$

In order to minimize $\Phi(\alpha,\lambda)$, we set the gradient to zero, and this yields:

$$\begin{pmatrix} D^T D & -\vec{1} \\ -\vec{1}^T & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \lambda \end{pmatrix} = \begin{pmatrix} \vec{0} \\ -1 \end{pmatrix} \tag{2}$$

where $\vec{1} = (1,\ldots,1)^T \in \mathbb{R}^{k\times 1}$, $\vec{0} = (0,\ldots,0)^T \in \mathbb{R}^{k\times 1}$. The coefficient matrix of (2) may be ill-conditioned relative to D because of the term $D^T D$, which has condition number $\kappa(D^T D) = \kappa(D)^2$.

Obtaining α directly from (2) requires $O(nk^2)$ flops to form $D^T D$ and $O(k^3)$ flops to solve (2). However, this cost can be reduced in the SCF-DIIS context, as follows: One easily obtains from (2) that $\alpha = (D^T D)^{-1}\vec{1} / \vec{1}^T (D^T D)^{-1}\vec{1}$; thus one needs only to solve a system with coefficient matrix $D^T D$ to obtain α. Suppose we have the QR decomposition $D = QR$, where $Q \in \mathbb{R}^{n\times k}$ is orthogonal, i.e., $Q^T Q = I_k$, and $R \in \mathbb{R}^{k\times k}$ is upper-triangular. Then $D^T D = R^T Q^T QR = R^T R$, and α can be obtained by solving triangular systems with R^T and R. Since D is obtained from its predecessor at the previous iteration by adding a new final column and, if necessary, deleting one or more initial columns, one can update the QR decomposition from the previous iteration in $O(nk)$ flops (see below). Thus at each iteration (other than the first one), one can update the QR decomposition at a cost of $O(nk)$ flops and obtain α by solving triangular systems with R^T and R at an additional cost of $O(k^2)$ flops.

We sketch the steps of the updating, referring the reader to [2] for full details. Suppose that, at some iteration, we have a predecessor matrix D and decomposition $D = QR$ from the previous iteration. Then the updating proceeds as follows:

- When adding a new final column to D, we apply the Gram–Schmidt process to orthogonalize the new final column against the columns of Q. The resulting vector and the orthogonalization coefficients then become new last columns of Q and R, respectively.
- When deleting the first column of D, we also delete the first column of R, so that we still have $D = QR$. Now R is upper-Hessenberg, and we left-multiply R by Givens rotations (see details in §5.1.8 of [2]) to restore R to triangular form. We then right-multiply Q by the transposes of the rotations in reverse order to obtain the final Q.

The null-space method. The basic idea of the general method, as described in [1], is to decompose the vector we want into the sum of a vector that satisfies the constraint and another vector in the null space of the constraint matrix. Thus the constrained problem becomes one of solving for the vector in the null space. By choosing a basis of the null space, one can then reduce the problem to an unconstrained, lower-dimensional problem of finding a minimizing linear combination of basis vectors.

For the problem (1), we introduce a particular implementation of the null-space method that avoids the ill-conditioning of the previous approach and has other numerical advantages. Denote $v = (0,\ldots,0,1)^T \in \mathbb{R}^{k\times 1}$ and set $\vec{1} = (1,\ldots,1)^T \in \mathbb{R}^{k\times 1}$ as before. Write $\alpha = v+\beta$, where β is in the null space of $\vec{1}^T$, i.e., $\vec{1}^T\beta = 0$. If $V \in \mathbb{R}^{k\times(k-1)}$ is full-rank and such that $\vec{1}^T V = 0$, then the columns of V constitute a basis of the null-space of $\vec{1}^T$, and we can write $\beta = V\gamma$, where $\gamma \in \mathbb{R}^{(k-1)\times 1}$. Then the minimization problem becomes

$$\min_{\vec{1}^T\alpha=1} \|D\alpha\| = \min_{\gamma\in\mathbb{R}^{k-1}} \|D(v+V\gamma)\| = \min_{\gamma\in\mathbb{R}^{k-1}} \|d_k + DV\gamma\|, \tag{3}$$

where $d_k = Dv$. Note that, with of our choice of v, d_k is just the last column of D and thus is available at no cost.

We choose V so that $V = (v_1, \dots, v_{k-1})$, where

$$v_j = \begin{pmatrix} \left.\begin{matrix} -1/\sqrt{j(j+1)} \\ \vdots \\ -1/\sqrt{j(j+1)} \end{matrix}\right\} j \text{ components} \\ \\ \sqrt{j/(j+1)} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad j = 1, \dots, k-1. \tag{4}$$

It is easily verified that $\vec{1}^T V = 0$ and V is full-rank; moreover, $V^T V = I_{k-1}$.

The normal equation of the least-squares problem (3) is

$$(DV)^T (DV)\gamma = -(DV)^T d_k. \tag{5}$$

This has condition number $\kappa((DV)^T(DV)) = \kappa(DV)^2$. One can obtain a better-conditioned system with a QR decomposition $DV = QR$, where as above $Q \in \mathbb{R}^{n\times(k-1)}$ is orthogonal and $R \in \mathbb{R}^{(k-1)\times(k-1)}$ is upper-triangular. Then (5) becomes

$$(QR)^T (QR)\gamma = -(QR)^T d_k \iff R\gamma = -Q^T d_k.$$

Thus, we can obtain γ and subsequently $\alpha = v + V\gamma$ by solving a linear system with condition number $\kappa(R) = \kappa(QR) = \kappa(DV) = \sqrt{\kappa(DV)^2}$, which is typically much smaller than that of (5). Since $V^T V = 1$, we also have the bound $\kappa(DV) \leq \kappa(D)\kappa(V) = \kappa(D)$.

As in the previous method, one can at each SCF-DIIS iteration obtain the QR decomposition of DV in $O(nk)$ flops by updating a QR decomposition from the previous iteration. In this case, we store Q_0 and R_0 such that $D = Q_0 R_0$ in the previous iteration. When D is modified at the current iteration by adding or dropping columns, we update Q_0 and R_0 in $O(nk)$ flops as in the previous method to obtain $D = Q_0 R_0$ for the modified D. Noting that $R_0 V$ is upper-Hessenberg since V is upper-Hessenberg, we then apply Givens rotations to $R_0 V$ and to Q_0 as in the previous method to obtain $DV = QR$ in $O(nk)$ flops.

The method of elimination. The general approach of the method, as described in [1], is to use the constraint to express some of the variables in terms of others in order to reduce the constrained least-squares problem to an unconstrained problem in fewer variables. The specific method considered here comes from [3]. Writing $D = (d_1, \dots, d_k)$, we introduce new variables $\bar{\alpha} = (\bar{\alpha}_1, \dots, \bar{\alpha}_{k-1})^T$ such that

$$D\alpha = \sum_{i=1}^{k} \alpha_i d_i = \bar{\alpha}_1(d_2 - d_1) + \bar{\alpha}_2(d_3 - d_2) + \cdots + \bar{\alpha}_{k-1}(d_k - d_{k-1}) + d_k = \bar{D}\bar{\alpha} + d_k,$$

where $\bar{D} = DW$ and

$$W = \begin{pmatrix} -1 & & & \\ 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & 1 & -1 \\ & & & 1 \end{pmatrix}. \tag{6}$$

Then the minimization problem becomes an unconstrained one:

$$\min_{\bar{\alpha} \in \mathbb{R}^{k-1}} \|\bar{D}\bar{\alpha} + d_k\| \tag{7}$$

Once this has been solved for $\bar{\alpha}$, one can calculate α by $\alpha = W\bar{\alpha} + (0,\ldots,0,1)^T$.

One possibility for solving (7) is to solve the normal equation $\bar{D}^T\bar{D}\bar{\alpha} = -\bar{D}^T d_k$ for $\bar{\alpha}$. This approach is suggested in §4.2 of [3]. However, this normal equation involves $\kappa(\bar{D}^T\bar{D}) = \kappa(\bar{D})^2$. As in the null-space method, we can improve the condition number with QR decomposition, this time of $\bar{D}$, i.e., $\bar{D} = QR$, where $Q \in \mathbb{R}^{n\times(k-1)}$ is orthogonal and $R \in \mathbb{R}^{(k-1)\times(k-1)}$ is upper-triangular. Then

$$(QR)^T(QR)\bar{\alpha} = -(QR)^T d_k \iff R\bar{\alpha} = -Q^T d_k.$$

Thus one can obtain $\bar{\alpha}$ by solving a linear system with R, which has condition number $\kappa(R) = \kappa(QR) = \kappa(\bar{D}) = \sqrt{\kappa(\bar{D})^2}$. We also have the bound $\kappa(\bar{D}) = \kappa(DW) \leq \kappa(D)\kappa(W)$, and one can show numerically that $\kappa(W) \approx 2k/\pi$ for all but the smallest values of k.

Again, we can avoid doing a direct QR decomposition of $\bar{D}$ at every iteration by making use of the upper-Hessenberg property of W. Specifically, we store Q_0 and R_0 such that $D = Q_0R_0$ and update them at each iteration as in the null-space method, with the upper-Hessenberg W replacing the upper-Hessenberg V.

Comparison of the three methods: a case study

We performed numerical experiments with these methods in SCF-DIIS iterations using data for various test materials. Our main interest was in observing the maximum condition numbers encountered by the methods with varying bounds on the maximum allowable value of k. Table 1 shows typical results, which were obtained in the case of a water molecule. In the table, the first column indicates the maximum allowable k-value (denoted k_{max}), and second through fourth columns indicate the maximum condition numbers observed during the iterations.

Conclusion

We have outlined three methods for solving the linearly constrained least-squares problem (1) and have discussed their relative merits in the context of the SCF-DIIS method. By updating QR decompositions, each of the three can be implemented in SCF-DIIS iterations at a cost of $O(nk) + O(k^2)$ flops per iteration.

Table 1: Maximum Observed Condition Numbers

k_{max}	Lagrange Multipliers	Null-Space Method	Method of Elimination
1	1.000e+000	1.000e+000	1.000e+000
2	3.377e+002	1.000e+000	1.000e+000
3	6.096e+004	3.965e+001	3.750e+001
4	9.640e+005	1.725e+002	1.386e+002
5	4.467e+007	1.489e+003	1.015e+003
6	1.839e+010	2.028e+004	1.546e+004
7	7.703e+012	3.861e+005	2.795e+005
8	1.599e+014	8.838e+005	5.123e+005

However, the null-space method and the method of elimination are somewhat more expensive than the Lagrange-multiplier method, since each requires an additional update costing $O(nk) + O(k^2)$ flops. Whether this additional expense is significant seems likely to depend on the overall cost of implementing SCF-DIIS in a particular application.

The three methods result in different linear systems that must be solved. The condition numbers of these systems, which govern the accuracy with which they can be solved numerically, are as follows: For the Lagrange-multiplier method, the condition number is $\kappa(D)^2$, which is likely to be very large relative to $\kappa(D)$. For the null-space method, the condition number is $\kappa(DV)$, where the columns of V are defined by (4). Since V is orthogonal, we have $\kappa(DV) \leq \kappa(D)$. For the method of elimination, the condition number is $\kappa(DW)$, where W is defined by (6). With the numerically observed approximation $\kappa(W) \approx 2k/\pi$, we have the bound $\kappa(DW) \leq \kappa(D)\kappa(W) \approx (2k/\pi)\kappa(D)$.

These observations clearly indicate that the Lagrange-multiplier method is likely to encounter much worse condition numbers than the other two methods, and this is borne out in the case study included above. In that study, the method of elimination exhibits very slightly smaller condition numbers than the null-space method, but the difference seems unlikely to be significant. The condition numbers for these two methods are, however, significantly smaller than the square roots of the corresponding condition numbers for the Lagrange-multiplier method, i.e., the corresponding values of $\kappa(D)^2$. This observation suggests that the bounds $\kappa(DV) \leq \kappa(D)$ and $\kappa(DW) \leq \kappa(D)\kappa(W) \approx (2k/\pi)\kappa(D)$ are both somewhat pessimistic.

Acknowledgement

Advice and guidance from Dr. Chao Yang from Lawrence Berkeley National Laboratory are happily acknowledged.

References

1. Björck, Å (1996): *Numerical methods for least squares problems*, SIAM,

Philadelpha.

2. Golub, G. H., Van Loan, C. F. (1996): *Matrix Computations*, third edition, The Johns Hopkins University Press, Baltimore.
3. Kresse, G., Furthmuller, J. (1996): "Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set", *Computational Materials Science 6*, pp. 15-50.
4. Nocedal, J., and Wright, S. J. (1999), *Numerical Optimization*, Springer–Verlag, New York.
5. Yang, C., Meza, J. C., and Wang, L. (2007): "A trust region direct constrained minimization algorithm for the Kohn-Sham equation", SIAM J. Sci. Comput. Vol. 29, No. 5, pp. 1854-1875.