thornquist

Comparison of Nonlinear Conjugate-Gradient Methods for Computing the Electronic Properties of Nanostructure Architectures ⋆.

Stanimire Tomov[1], Julien Langou[1], Andrew Canning[2], Lin-Wang Wang[2], and Jack Dongarra[1]

[1] Innovative Computing Laboratory
The University of Tennessee
Knoxville, TN 37996-3450
[2] Lawrence Berkeley National Laboratory
Computational Research Division
Berkeley, CA 94720

February 8, 2005

Abstract. In this article we report on our efforts to test and expand the current state-of-the-art in eigenvalue solvers applied to the field of nanotechnology. We singled out the nonlinear conjugate gradients method as the blackbone of our efforts for their previous success in predicting the electronic properties of large nanostructures and made a library of three different solvers (two recent and one new) that we integrated into the parallel PESCAN (Parallel Energy SCAN) code [3] to perform a comparison.

1 Introduction

First-principles electronic structure calculations are typically carried out by minimizing the quantum-mechanical total energy with respect to its electronic and atomic degrees of freedom. Subject to various assumptions and simplifications [5], the electronic part of this minimization problem is equivalent to solving the single particle Schrödinger-type equations (called Kohn-Sham equations) of the form

$$\hat{H}\psi_i(r) = \epsilon_i\psi_i(r), \tag{1}$$
$$\hat{H} = -\frac{1}{2}\nabla^2 + V$$

where $\psi_i(r)$ are the single particle wave functions (of electronic state i) that minimize the total energy, and V is the total potential of the system. The wave functions are most commonly expanded in plane-waves (Fourier components) up to some cut-off energy which discretizes the Equation (1). In this approach

⋆ This work was supported by the US Department of Energy, Office of Science, Office of Advanced Scientific Computing (MICS) and Basic Energy Science under LAB03-17 initiative, contract Nos. DE-FG02-03ER25584 and DE-AC03-76SF00098

the lowest eigen-pairs are calculated for $\hat{H}$ and the Kohn-Sham equations are solved self-consistently. For a review of this approach see reference [5] and the references therein. The computational cost of this approach scales as the cube of the number of atoms and the the maximum size of system that can be studied is of the order of hundreds of atoms. In the approach used in PESCAN developed by L-W. Wang and A. Zunger [8] a semi-empirical potential is used such that $V(\mathbf{r})$ and only the eigenstates of interest around a given energy are calculated, allowing the study of large nanosystems (up to a million atoms). The problem then becomes: find ψ and E close to a given E_{ref} such that

$$H\psi = E\psi, \tag{2}$$

where H represents the Hamiltonian matrix, which is Hermitian with dimension equal to the number of Fourier components used to expand ψ. The dimension of H may be of the order of a million for large nanosystems. The eigenvalues E (energy of state ψ) are real, and the eigenvectors ψ are orthonormal and typically complex.

The spectrum of H has energy gaps and of particular interest to physicists is to find a few, approximately 4 to 10, of the interior eigenvalues on either side of the gap which determines many of the electronic properties of the system. Due to its large size the matrix H in the eigenvalue problem of Equation 2 is never explicitly computed. We calculate the kinetic energy part in Fourier space, where it is diagonal, and the potential energy part in real space so that the number of calculations used to construct the matrix-vector product scales as $n \log n$ rather than n^2 where n is the dimension of H. Three dimensional FFTs are used to move between Fourier and real space. H is therefore available as a procedure for computing Hx for a given vector x. Thus one more requirement is that the solver is matrix free. Finally, repeated eigenvalues (degeneracy) of approximately 4 maximum are possible for the problems discussed and we need to be able to resolve such cases to fully understand the electronic properties of our systems.

In this paper, three different eigensolvers, based on the nonlinear Conjugate-Gradient algorithm, are compared on this particular problem. In Section 2 we describe the three eigensolvers investigated in the paper. Section 3 presents our numerical results. Implementation details, testing issues and optimizations performed are given in Subsection 3.1. Subsection 3.2 is for the numerical results on our real problem. Finally, in Section 4, we give some concluding remarks.

The focus of this paper is on nonlinear conjugate gradient methods with folded spectrum. Alternative for the spectral transformation are shift-and-invert or fixed-polynomial [6]. Alternative for the iterative solver are Lanczos or Jacobi Davidson. Our choice of method is not arbitary, these methods has proven to be more effective for the computation around the energy gap of quantum dots than others.

2 Nonlinear Conjugate Gradient Method for Eigenvalue Problem

The conventional approach for problems of very large matrix size is to use iterative projection methods where at every step one extracts eigenvalue approximations from a given subspace S of small dimension (see e.g. [2]). Nonlinear conjugate gradient methods belong to this class of method. Let us assume for now that we are looking for the smallest eigenvalue of the Hermitian operator A.

This eigenvalue problem can be expressed in terms of function minimization as: find the variational minimum of $F(\mu) =< \mu, A\mu >$, on which a nonlinear conjugate gradient method is performed to find the minima.

In this section, we give a description of the algorithm we have implemented in our library, namely: PCG, PCG-XR and LOBPCG plus the folded spectrum spectral transformation.

2.1 PCG Method

In Table 1, we give is a pseudo-code of the PCG algorithm for eigen-problems. This is the algorithm originally implemented in the PESCAN code (see also [5, 7]).

```
1   do i = 1,niter
2      do m = 1,blockSize
3         orthonormalize X(m) to X(1 : m − 1)
4         ax = A X(m)
5         do j = 1,nline
6            λ(m) = X(m) · ax
7            if (||ax − λ(m) X(m)||_2 < tol .or. j == nline) exit
8            r_{j+1} = (I − X X^T) ax
9            β = ((r_{j+1} − r_j)·Pr_{j+1}) / (r_j·Pr_j)
10           d_{j+1} = −P r_{j+1} + β d_j
11           d_{j+1} = (I − XX^T) d_{j+1}
12           γ = ||d_{j+1}||_2^{−1}
13           θ = 0.5 |atan( 2 γ d_{j+1}·ax / (λ(m) − γ^2 d_{j+1}·A d_{j+1}) )|
14           X(m) = cos(θ) X(m) + sin(θ) γ d_{j+1}
15           ax = cos(θ) ax + sin(θ) γ A d_{j+1}
16        enddo
17     enddo
18     [X, λ] = Rayleigh − Ritz on span{X}
19  enddo
```

Table 1. PCG algorithm

Here P is a preconditioner for the operator A, X is the block (of size blockSize) of column eigenvectors sought, λ is the corresponding block of eigenvalues.

2.2 LOBPCG Method

Briefly, the LOBPCG method can be described with the following pseudo-code

1	do i = 1,niter
2	R = P (A X_i − λ X_i)
3	check convergence criteria
4	[X_i, λ] = Rayleigh − Ritz on span{X_i, X_{i-1}, R}
5	enddo

Table 2. LOBPCG algorithm

i.e. the m and j loops from Table 1 are replaced with just the blocked computation of the preconditioned residual, and the Rayleigh-Ritz on span{X_i} with Rayleigh-Ritz on span{X_{i-1}, X_i, R}. The direct implementation of this algorithm becomes unstable as X_{i-1} and X_i become closer and closer, and therefore special care and modifications have to be taken (see [4]).

2.3 PCG-XR Method

PCG-XR is PCG algorithm except that line 18 in Table 1 is replaced by

$$18 \qquad [X,\ \lambda] \ = \ \texttt{Rayleigh} - \texttt{Ritz on span}\{X, R\}.$$

The idea is to store the vectors R to perform a more efficient Rayleigh-Ritz step.

2.4 Folded Spectrum

Projection methods are good at finding well separated (non-clustered) extremal eigenvalues. In our case, we are seeking for interior eigenvalues and thus we have to use a spectral transformation, the goal being to map the sought eigenvalue of our operator to extremal eigenvalues of another one.

To do so we use the folded spectrum method. The interior eigenvalue problem $Hx = \lambda x$ is transformed to find the smallest eigenvalues of $(H - E_{ref}I)^2 x = \mu x$. The eigenvalues of the original problem are given back by $\mu = (\lambda - s)^2$.

The PCG algorithm in its folded form (FS-PCG) is described in [7]. To adapt the folded spectrum to LOBPCG (FS-LOBPCG), we have added three others block vectors that stores the matrix-vector product of the blocks X, R and P with the matrix H. This enables us to control the magnitude of the residuals for a few axpys more (otherwise we just have access to the magnitude of the

residuals of the squared operator). Also the deflation strategy of LOBPCG is adapted in FS-LOBPCG, the vectors are deflated when the residual relative to H have converged (not H^2).

3 Numerical Results

3.1 Software Package

We implemented LOBPCG algorithm [3], PCG and PCG-XR in a software library.

Currently, our library has single and double, real and complex arithmetic with both parallel (using MPI) and sequential versions, it is written in Fortran 90. The folded spectrum spectral transformation is optional. The implementation is stand alone and meant to be easily integrated in various physics codes (like PESCAN).

A test case is provided with the software. It represents a 5-point operator where the coefficents a (diagonal) and b (for the connections with the 4 closest neighbors on a regular 2D mesh) can be changed. In Table 3.1 the output of the test is presented, it is performed on a Linux Intel Pentium IV with Intel Fortran compiler and parameters ($a = 8, b = 1 - i$), we are looking for the 10 smallest eigenstates, the matrix size is $20,000$, the iterations are stopped when all the eigencouples (x, λ) satisfies $\|Hx - x\lambda\| \leq \texttt{tol}\|x\|$, with $\texttt{tol} = 10^{-8}$.

	Method		
	PCG	LOBPCG	PCG-XR
time (s)	37.1	61.7	20.2
matvecs	3,555	1,679	1,760
dotprds	68,245	137,400	37,248
axpys	66,340	158,261	36,608
copys	6,190	9,976	3,560

Table 3. Comparison of the PCG, PCG-XR and LOBPCG methods in finding 10 eigenstates on a problem of size $20,000 \times 20,000$.

In a general manner, LOBPCG always performs less iterations (i.e. less matrix-vector products) than PCG. This advantage comes to the cost of more vector operations (axpys and dot products) and more memory requirements. In this case, LOBPCG performs approximately 2 times more dot products for 2 times less matrix vector products than the PCG method, since the 5-point matrix-vector product takes approximately the time of 7 dot products, PCG gives the best time.

[3] http://www-math.cudenver.edu/ aknyazev/software/CG/latest/lobpcg.m (revision 4.10 written in Matlab, with some slight modifications)

The CG-XR method represents for this test case an interesting alternative for thos two methods: it inherits the low number of matrix vector products from the LOBPCG and the low number of dot products from the PCG method.

3.2 Numerical Results on some Quantum Dots

In this section we present numerical results on quantum dots up to thousand of atoms. The experiments are performed on the IBM-SP `seaborg` at NERSC in Lawrence Berkeley National Laboratory.

For all the experiments we are looking for $\texttt{mx} = 10$ interior eigenvalues around $E_{ref} = -4.8$eV. The stopping criterion is $\|Hx - x\lambda\| \leq \texttt{tol}\|x\|$ where $\texttt{tol} = 10^{-6}$. All the runs are performed on one node with 16 processors except for the smallest case (20Cd,19Se) which is run on 8 processors. All the solvers are started with the same initial guess.

The preconditioner is the one given in [7], it is diagonal with diagonal elements p_i

$$p_i = \frac{E_k^2}{(\frac{1}{2}q_i^2 + V_0 - E_{ref})^2 + E_k^2},$$

where q_i is the diagonal term of the Laplacian, V_0 is the average potential and E_k is the average kinetic energy of the wavefunction ψ. It is meant to be an approximation of the inverse of $(H - E_{ref})^2$.

A notable fact is that all the solvers find the same 10 eigenvalues with the correct accuracy for all the runs in less than 30 minutes. Therefore they are all robust.

The timing results are given in Table 4. For each test case the number of atoms of the quantum dot and the order n of the corresponding matrix is given The parameter for the number of iterations in the inner loop (*nline*) for FS-PCG and FS-PCG-XR is chosen to be the optimal one among the values 20, 50, 100, 200, and 500 and is given in bracket after the solver.

From Table 4, we observe that the three methods FS-PCG, FS-PCG-XR and FS-LOBPCG behaves almost the same. The best method (in term of time) being either FS-PCG-XR or FS-LOBPCG.

FS-LOBPCG should also benefit in speed over FS-PCG and FS-PCG-XR from the fact that the matrix-vector products are performed by block. This is not the case in the version of the code used for this paper where the experiments are performed on a single node. The blocked implementation of FS-LOBPCG in PESCAN should run faster and also scale to larger processor counts as latency is less of an issue in the communications part of the code.

Another feature of FS-LOBPCG that is not stressed in Table 4 is its overwhelming superiority over FS-PCG when no preconditioner is available. In Table 5, we illustrate this later feature. For the quantum dot (83Cd,81Se), FS-LOBPCG runs 4 times faster than FS-PCG without preconditioner whereas it runs only 1.4 times faster with.

For the four experiments presented in Table 4, the number of inner iteration that gives the mimimum total time is always attained for a small number of outer

(20Cd, 19Se) n = 11,331

	# matvec	# outer it	Time
FS-PCG(50)	4898	(8)	50.4s
FS-PCG-XR(50)	4740	(6)	49.1s
FS-LOBPCG	4576		52.0s

(83Cd, 81Se) n = 34,143

	# matvec	# outer it	Time
FS-PCG(200)	15096	(11)	264 s
FS-PCG-XR(200)	12174	(5)	209 s
FS-LOBPCG	10688		210 s

(232Cd, 235Se) n = 75,645

	# matvec	# outer it	Time
FS-PCG(200)	15754	(8)	513 s
FS-PCG-XR(200)	15716	(6)	508 s
FS-LOBPCG	11864		458 s

(534Cd, 527Se) n = 141,625

	# matvec	# outer it	Time
FS-PCG(500)	22400	(6)	1406 s
FS-PCG-RX(500)	21928	(4)	1374 s
FS-LOBPCG	17554		1399 s

Table 4. Comparison of FS-PCG, FS-PCG-XR and FS-LOBPCG methods in finding 10 eigenstates around the gap of quantum dots of increasing size.

(83Cd, 81Se) n = 34,143

		# matvec	Time
FS-PCG(200)	precond	15096	264 s
FS-LOBPCG	precond	10688	210 s
FS-PCG(200)	no precond	71768	1274 s
FS-LOBPCG	no precond	17810	341 s

Table 5. Comparison of FS-PCG and FS-LOBCG with and without preconditioner to find `mx` = 10 eigenvalues of the quantum dots (83Cd,81Se)

iteration, this is illustrated in Table 6 for (232Cd, 235Se) where the minimum time is obtained for 6 outer iterations. Another and more practical way of stopping the inner iteration is in fixing the requested tolerance reached at the end of the inner loop. We call FS-PCG(k) FS-PCG where the inner loop is stopped when the accuracy is less than $k^{n_{outer}}$, where n_{outer} is number of the corresponding outer iteration. In Table 6, we give the results for FS-PCG(10^{-1}) and (223Cd,235Se). It comes without a surprise that this solver converge in 6 outer steps: first inner loop guarantees an accuracy of 10^{-1}, second inner loop guarantees an accuracy of 10^{-2}, and so on until 10^{-6}. This schemes looks promising. It also allows a synchronized convergence of the block vectors.

(232Cd, 235Se) n = 75,645			
	# matvec	# outer it	Time
FS-PCG(100)	17062	(15)	577s
FS-PCG(200)	15716	(6)	508s
FS-PCG(300)	15990	(4)	517s
FS-PCG(10^{-1})	15076	(6)	497s

Table 6. The problem of finding the best inner length for FSM-PCG can be avoided by fixing a tolerance as stopping criterion in the inner loop

4 Conclusions

In this paper, we have compared three nonlinear conjugate gradient methods with folded spectrum to find a small amount of interior eigenvalues around a given point: FS-PCG, FS-PCG-XR, and FS-LOBPCG; the application being the computation the prediction of the electronic properties of quantum nanostructures.

All three methods are similar and thus often the results are close; a general ranking being: FS-LOBPCG is the fastest, then come FS-PCG-XR and finally FS-PCG. In term of memory requirement the three methods are ranked in the same way: FS-LOBPCG/FS-PCG-XR requires four/two times as much memory as FS-PCG. As our problem scales up the memory has not shown up as a bottleneck yet, i.e. using FS-LOBPCG is affordable.

The main drawback of FS-PCG and FS-PCG-XR is their sensitivity to the parameter `nline` (the number of iterations in the inner loop). In order to get rid of this parameter one can rather have a fixed residual tolerance to be achieved on each step of the outer loop.

On other applications, the performance of FS-LOBPCG shall be better provided a fast block matrix-vector product and an accomodating preconditioner.

Acknowledgements

This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy.

References

1. Arbenz, P., Hetmaniuk, U.L., Lehoucq, R.B., Tuminaro, R.S.: A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods. Int. J. Numer. Meth. Engng (to appear)
2. Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H. (Editors): *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia (2000)
3. Canning, A., Wang, L.., Williamson, A., Zunger, A.: Parallel empirical pseudopotential electronic structure calculations for million atom systems. J. Comp. Phys. **160** (2000) 29–41
4. Knyazev, A.: Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method. SIAM J. on Scientific Computing **23(2)** (2001) 517–541
5. Payne, M.C., Teter, M.P., Allan, D.C., Arias, T.A., Joannopoulos, J.D.: Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients. Rev. Mod. Phys. **64** (1992) 1045–1097
6. Thornquist, H.: Fixed-Polynomial Approximate Spectral Transformations for Preconditioning the Eigenvalue Problem. Masters thesis, Rice University, Department of Computational and Applied Mathematics (2003)
7. Wang, L.W., Zunger, A.: *Pseudopotential Theory of Nanometer Silicon Quantum Dots application to silicon quantum dots.* In Kamat, P.V., Meisel, D.(Editors): *Semiconductor Nanoclusters* (1996) 161–207
8. Wang, L.W., Zunger, A.: Solving Schrodinger's equation around a desired energy: application to silicon quantum dots. J. Chem. Phys. **100(3)** (1994) 2394–2397