



ELSEVIER

Contents lists available at SciVerse ScienceDirect

Linear Algebra and its applications

journal homepage: www.elsevier.com/locate/laa

Preconditioning systems arising from the KKR Green function method using block-circulant matrices

Matthias Bolten^{a,*}, Alexander Thiess^{b,c}, Irad Yavneh^d, Rudolf Zeller^b^a Bergische Universität Wuppertal, Wuppertal, Germany^b Peter Grünberg Institut and Institute for Advanced Simulation, Forschungszentrum Jülich GmbH, Jülich, Germany^c German Research School for Simulation Sciences, Jülich, Germany^d Technion—Israel Institute of Technology, Haifa, Israel

ARTICLE INFO

Article history:

Received 16 November 2010

Accepted 9 May 2011

Submitted by L. Cvetkovic

AMS classification:

15B05

65F08

Keywords:

Preconditioning

Electronic structure calculation

Block-circulant matrices

ABSTRACT

Recently, a linearly scaling method for the calculation of the electronic structure based on the Korringa–Kohn–Rostoker Green function method has been proposed. The method uses the transpose free quasi minimal residual method (TFQMR) to solve linear systems with multiple right hand sides. These linear systems depend on the energy-level under consideration and the convergence rate deteriorates for some of these energy points. While traditional preconditioners like ILU are fairly useful for the problem, the computation of the preconditioner itself is often relatively hard to parallelize. To overcome these difficulties, we develop a new preconditioner that exploits the strong structure of the underlying systems. The resulting preconditioner is block-circulant and thus easy to compute, invert and parallelize. The resulting method yields a dramatic speedup of the computation compared to the unpreconditioned solver, especially for critical energy levels.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

The time-independent many-electron Schrödinger equation is given for N electrons by

$$\hat{H}\Psi = \left[-\frac{\hbar}{2m} \sum_{i=1}^N \nabla_i^2 + \sum_{i<j}^N \sum U(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i=1}^N v_{\text{ext}}(\mathbf{r}_i) \right] \Psi = E\Psi. \quad (1)$$

* Corresponding author.

E-mail addresses: bolten@math.uni-wuppertal.de (M. Bolten), a.thiess@fz-juelich.de (A. Thiess), irad@cs.technion.ac.il (I. Yavneh), ru.zeller@fz-juelich.de (R. Zeller).

Here,

$$U(\mathbf{r}_i, \mathbf{r}_j) = \frac{e^2}{\|\mathbf{r}_i - \mathbf{r}_j\|_2}$$

is the electron–electron interaction and v_{ext} an external potential, e.g., the electrostatic potential of the nuclei. The solution of the Schrödinger equation (1) yields the electron structure that is of great importance in theoretical chemistry as well as solid-state physics.

The problem with (1)—the highly nontrivial and computationally demanding task of evaluating the many-electron wavefunction Ψ —is avoided in Hohenberg–Kohn–Sham density functional theory by using the one-electron density $n(\mathbf{r})$ as the fundamental variable. This reduces the effort dramatically, as the wavefunctions depend on $3N$ variables, whereas a density $n(\mathbf{r})$ depends on three variables, only. Application of the Hohenberg–Kohn theorem yields: (i) the ground state density uniquely determines the external potential and thus, via (1), wavefunctions and all physical observables of the system; (ii) that the ground state energy E_0 and the ground state density $n_0(\mathbf{r})$ can be obtained from a variational principle which only involves the density; and (iii) a universal functional $F[n(\mathbf{r})]$ exists (universal in the sense that it does not depend on v_{ext}) such that the ground state energy and density can be obtained by

$$E_0 = \min_n E[n(\mathbf{r})] = \min_n F[n(\mathbf{r})] + \int_{\mathbb{R}^3} n(\mathbf{r})v_{\text{ext}}(\mathbf{r})d\mathbf{r}.$$

Although $F[n(\mathbf{r})]$ can be formally defined using many-electron wavefunctions, its explicit dependence on $n(\mathbf{r})$ is not known. In the Kohn–Sham scheme it is written as

$$F[n(\mathbf{r})] = T_s[n(\mathbf{r})] + U[n(\mathbf{r})] + E_{\text{xc}}[n(\mathbf{r})],$$

where the exchange–correlation functional E_{xc} must be approximated, U is the classical electrostatic energy of the charge distribution $n(\mathbf{r})$, and T_s is the kinetic energy of the effective Kohn–Sham system, which describes a non-interacting electron system (Eq. (1) with $e^2 = 0$) in an external potential v_{eff} chosen such that the Kohn–Sham system has the same density as the real system with electron–electron interaction. The density $n(\mathbf{r})$ in the Kohn–Sham system is given by

$$n(\mathbf{r}) = \sum_{i=1}^N |\varphi_i(\mathbf{r})|^2,$$

where the φ_i are the N wavefunctions corresponding to the lowest eigenvalues ϵ_i of the single-particle Schrödinger equation

$$\hat{H}_s \varphi(\mathbf{r}) = \left[-\frac{\hbar^2}{2m} \nabla_{\mathbf{r}}^2 + v_{\text{eff}}(\mathbf{r}) \right] \varphi_i(\mathbf{r}) = \epsilon_i \varphi_i(\mathbf{r}).$$

Thus, for calculating the ground state density the solution of this eigenvalue problem is required.

2. KKR method and resulting linear system

In the Korringa–Kohn–Rostoker (KKR) Green function method the density is calculated by a contour integral in the complex E plane as

$$n(\mathbf{r}) = -\frac{2}{\pi} \text{Im} \left(\int_{-\infty}^{E_F} G(\mathbf{r}, \mathbf{r}; E) dE \right), \quad (2)$$

where the energy dependent independent-particle Kohn–Sham Green function $G(\mathbf{r}, \mathbf{r}; E)$ is defined as the solution of the equation

$$\left[-\frac{\hbar^2}{2m} \Delta_{\mathbf{r}}^2 + v_{\text{eff}}(\mathbf{r}) - E \right] G(\mathbf{r}, \mathbf{r}'; E) = -\delta(\mathbf{r} - \mathbf{r}'),$$

$$G(\mathbf{r}, \mathbf{r}'; E) \xrightarrow{\|\mathbf{r}\|_2 \rightarrow \infty} 0.$$

This can be equivalently written as

$$G(\mathbf{r}, \mathbf{r}'; E) = G^r(\mathbf{r}, \mathbf{r}'; E) \hat{E} + \int_{\mathbb{R}^3} G^r(\mathbf{r}, \mathbf{r}''; E) [v_{\text{eff}}(\mathbf{r}'') - v^r(\mathbf{r}'')] G(\mathbf{r}'', \mathbf{r}'; E) d\mathbf{r}''. \tag{3}$$

Here, v^r is a reference system for which the Green function G^r is known. The integral is discretized by dividing the domain into non-overlapping space-filling cells around atomic positions $\mathbf{R}_{\mathbf{n}}$, where \mathbf{n} is the multi-index of the cell. Within a cell the so-called multiple-scattering representation is used, i.e.,

$$G(\mathbf{r} + \mathbf{R}_{\mathbf{n}}, \mathbf{r}' + \mathbf{R}_{\mathbf{n}'}; E) = \delta_{\mathbf{n}, \mathbf{n}'} G_s^{\mathbf{n}}(\mathbf{r}, \mathbf{r}'; E) + \sum_{L, L'} R_L^{\mathbf{n}}(\mathbf{r}; E) G_{L, L'}^{\mathbf{n}, \mathbf{n}'}(E) R_{L'}^{\mathbf{n}'}(\mathbf{r}'; E), \tag{4}$$

where the L 's represent the so-called *angular-momentum indices* l and m , $G_s^{\mathbf{n}}(\mathbf{r}, \mathbf{r}'; E)$ are the single-scattering Green functions, and $R_L^{\mathbf{n}}(\mathbf{r}; E)$ are wavefunctions. The latter two only depend on $v_{\text{eff}}(\mathbf{r})$ inside the cell \mathbf{n} . Now, the discretization of (3) can be written as the matrix equation

$$G_{L, L'}^{\mathbf{n}, \mathbf{n}'}(E) = (G^r)_{L, L'}^{\mathbf{n}, \mathbf{n}'}(E) + \sum_{\mathbf{n}'', L'', L'''} (G^r)_{L, L''}^{\mathbf{n}, \mathbf{n}''}(E) \Delta t_{L'', L'''}^{\mathbf{n}'', \mathbf{n}'''}(E) G_{L''', L'}^{\mathbf{n}'', \mathbf{n}'}(E).$$

The matrix is of dimension $n = N_{\text{atoms}}(l_{\text{max}} + 1)^2$, where N_{atoms} is the number of atoms and l_{max} is the highest angular-momentum. Usually $l_{\text{max}} = 3$ is sufficient. The matrix $\Delta t_{L'', L'''}^{\mathbf{n}'', \mathbf{n}'''}(E)$ depends on the difference $v_{\text{eff}}(\mathbf{r}) - v^r(\mathbf{r})$ inside the cell \mathbf{n}'' , only. In the following, we denote the different matrices as

$$\underbrace{G_{L, L'}^{\mathbf{n}, \mathbf{n}'}(E)}_{=:G} = \underbrace{(G^r)_{L, L'}^{\mathbf{n}, \mathbf{n}'}(E)}_{=:G^r} + \sum_{\mathbf{n}'', L'', L'''} (G^r)_{L, L''}^{\mathbf{n}, \mathbf{n}''}(E) \underbrace{\Delta t_{L'', L'''}^{\mathbf{n}'', \mathbf{n}'''}(E) G_{L''', L'}^{\mathbf{n}'', \mathbf{n}'}(E)}_{=:T},$$

obtaining

$$G = G^r + G^r T G.$$

We have

$$\begin{aligned} G &= G^r + G^r T G \\ \Leftrightarrow (I - G^r T) G &= G^r = -(I - G^r T) T^{-1} + T^{-1} \\ \Leftrightarrow G &= -T^{-1} + (I - G^r T)^{-1} T^{-1} \\ &= -T^{-1} + T^{-1} \underbrace{(T^{-1} - G^r)^{-1}}_{=:M} T^{-1}. \end{aligned}$$

To compute G we are interested in the inverse of M . Setting $X = M^{-1}$ and using the identity

$$T M X = T,$$

together with the definition of M , gives

$$T(T^{-1} - G^r)X = T \Leftrightarrow (I - T G^r)X = T. \tag{5}$$

Thus, the computation of the ground state density can be carried out by numerical integration over the solutions of the matrix equation for different energies E . The solution of (5) requires solving n linear systems. In [1] the authors propose the use of the transpose free quasi minimal residual method (TFQMR) [2] for that purpose. However, it turns out that the number of iterations necessary to solve

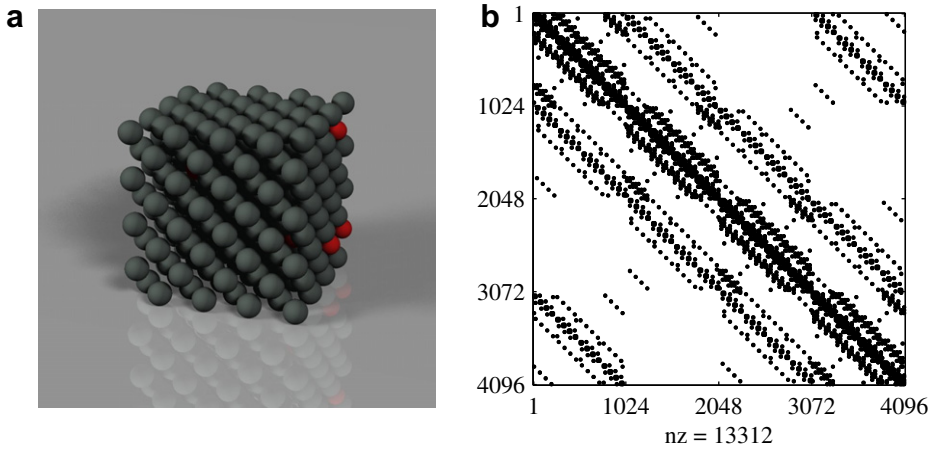


Fig. 1. Structure and spplot of the associated matrix of a Ni–Pd alloy. (a) Structure of the system. (b) Spplot of the matrix.

the system becomes larger as the energy appearing in the numerical integration of (2) approaches the critical energy E_F .

In fact, the matrices in (5) typically possess a great deal of structure. In many applications in solid state physics periodic systems are studied. In contrast to perfect crystals, where the whole system is described by one reference box of a certain geometry, systems are studied where the crystal system is perturbed by introducing different atoms in the crystal, by slightly moving the atoms away from their optimal positions, or by a combination of both. In these cases, the systems consist of $n_x \times n_y \times n_z$ cells, where, depending on the geometry in each cell a certain number of atoms is located. In Fig. 1(a) the geometry of a nickel–palladium alloy with slightly perturbed atomic positions is depicted. This system has a face-centered cubic (FCC) geometry and is thus broken up into $4 \times 4 \times 4$ cells, each cell contains 4 atoms, and, as $l_{\max} = 3$ in the sums in (4), 16 terms have to be taken into account. Hence, the blocks inside the matrix are of size 64×64 . For the actual computation, periodic boundary conditions are imposed and matrix elements $(G^r)_{L,L'}^{\mathbf{n},\mathbf{n}'}$ between atoms inside of cells that are beyond a specified cut-off parameter are neglected, which introduces a negligible approximation error due to the exponential decay of the entries of G^r with distance $\|R_{\mathbf{n}} - R_{\mathbf{n}'}\|_2$. The result is a block system that has a sparsity pattern typical of stencil-based codes on structured 3D grids, c.f., the matrix depicted in Fig. 1(b), further on, the cost of a matrix-multiplication is proportional to the system size, i.e., the number of atoms.

Our aim is to construct a preconditioner that can be evaluated efficiently and that exploits the structure of the matrix. One option would be to use a physics-based preconditioner, i.e., the solution of a similar perfect crystal. As in this case the similar system had to be supplied as additional user input, we choose an algebraic approach that automatically embeds information from the system into the preconditioner matrix. A block-circulant matrix is well-suited for that purpose and the resulting preconditioner is similar to the physics-based approach mentioned before and could be interpreted as the sought-for physical system.

3. Block-circulant preconditioner

The preconditioner we use for this system is a multilevel block-circulant matrix. Therefore, we first briefly review the definitions and properties of circulant matrices. A more detailed introduction can be found, e.g., in [3] or in [4].

3.1. Circulant matrices

Definition 3.1 (circulant matrix). A circulant matrix $C \in \mathbb{C}^{N \times N}$ is of the form

$$C = \begin{pmatrix} c_0 & c_1 & \cdots & c_{N-1} \\ c_{N-1} & c_0 & \cdots & c_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{pmatrix}.$$

The generating symbol f of C is the 2π -periodic function given by

$$f : \mathbb{R} \rightarrow \mathbb{C},$$

$$x \mapsto f(x) = \sum_{j=-\lceil N/2 \rceil}^{\lceil N/2 \rceil} c_{j \bmod N} e^{ijx}.$$

Circulant matrices are a special class of Toeplitz matrices forming a matrix algebra. They are tightly connected to their generating symbol and to the Fourier matrix F_N given by

$$(F_N)_{j,k} = e^{2\pi i \frac{jk}{N}},$$

as stated by the following theorem:

Theorem 3.2. Let $C \in \mathbb{C}^{N \times N}$ be a circulant matrix and let f be its generating symbol, then the following holds true:

- (a) The eigenvalue $\lambda_j, j = 0, \dots, N - 1$, of C is given by $\lambda = f(j \frac{2\pi}{N})$.
- (b) The eigenvector v_j of C is given by the column $F(:, j)$ of the Fourier matrix.

Theorem 3.2 states that for any circulant matrix $C \in \mathbb{C}^{N \times N}$ we have

$$C F_N = F_N \text{diag}(\lambda_1, \dots, \lambda_N),$$

i.e., F_N diagonalizes C . Due to this fact, circulant matrices can easily be inverted using the FFT. Let us now turn to the case where the “elements” of the circulant matrices can themselves be matrix-valued. The result is a block-matrix with circulant structure as given by the following definition.

Definition 3.3 (Block-circulant matrix). A block-circulant matrix $C_B \in \mathbb{C}^{(bN) \times (bN)}$ is a matrix of the form

$$C = \begin{pmatrix} C_0 & C_1 & \cdots & C_{N-1} \\ C_{N-1} & C_0 & \cdots & C_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ C_1 & C_2 & \cdots & C_0 \end{pmatrix},$$

where the C_j are in $\mathbb{C}^{b \times b}$. The generating symbol of a block-circulant matrix is a 2π -periodic function f given by

$$f_B : \mathbb{R} \rightarrow \mathbb{C}^{b \times b},$$

$$x \mapsto f_B(x) = \sum_{j=-\lceil N/2 \rceil}^{\lceil N/2 \rceil} C_{j \bmod N} e^{ijx}.$$

A block-circulant matrix is block-diagonalized by $F_N \otimes I_b$, where I_b is the $b \times b$ -identity.

The concept of circulant matrices can be extended to higher dimensions. If a block-circulant matrix has circulant blocks, the matrix is called a *BCCB-matrix* or a *2-level circulant matrix*. This motivates the recursive definition of *d-level circulant matrices*.

Definition 3.4 (*d-level circulant matrices*). Let $C_2 \in \mathbb{C}^{(N_1 \cdot N_2) \times (N_1 \cdot N_2)}$ be a block-circulant matrix with blocks $C_{2,j}, j = 0, \dots, N_1 - 1$, on the diagonals. If the $C_{2,j}$ blocks are circulant with generating symbol $f_{2,j} : \mathbb{R} \rightarrow \mathbb{C}$, then C_2 is a *2-level circulant matrix*. Its generating symbol f_2 is given by

$$f_2 : \mathbb{R}^2 \rightarrow \mathbb{C},$$

$$(x, y) \mapsto f_2(x, y) = \sum_{j=-\lceil N_1/2 \rceil}^{\lceil N_1/2 \rceil} f_{2,(j \bmod N)}(y) e^{ijx}.$$

For $d > 2$ we call a block-circulant matrix $C_d \in \mathbb{C}^{(\prod_{k=1}^d N_k) \times (\prod_{k=1}^d N_k)}$ with blocks $C_{j,j}, j = 0, \dots, N_1 - 1$, *d-level circulant*, if the blocks are $d - 1$ -level circulant. In all cases, a block-matrix with *d-level circulant* structure is called a *d-level block-circulant matrix*.

A 2-level circulant matrix of dimension $(mn) \times (mn)$, where n is the size of the blocks, is diagonalized by the matrix $F_m \otimes F_n$, with an analogous extension to higher levels.

3.2. Block-circulant preconditioners

The properties of circulant matrices allow us to define a preconditioner that can be inverted efficiently using the FFT. The idea goes back to the work of Chan and Chan [5], who define point- and block-circulant preconditioners for elliptic PDEs. Our approach combines the two ideas.

Definition 3.5 (*Block-circulant preconditioner*). Let $A \in \mathbb{C}^{N \times N}$ be an arbitrary matrix, where $N = n \cdot b$, $n, b \in \mathbb{N}$. The *block-circulant preconditioner* $C \in \mathbb{C}^{N \times N}$ of A is the block-circulant matrix given by its blocks

$$M_j = \frac{1}{n} \sum_{k=0}^{n-1} A_{k, (k+j) \bmod n},$$

where $A_{l,m} = A(l \cdot b + 1 : (l + 1) \cdot b, m \cdot b + 1 : (m + 1) \cdot b)$ is a $b \times b$ -block of A .

Thus, the block-circulant preconditioner is an average of the diagonal blocks of A . This definition can be extended to the *d-level* case.

3.3. Application to the KKR Green function method

In a system of $n_x \times n_y \times n_z$ cells with N_c atoms per cell, the matrices T, G^r and X are of dimension $N \times N$, where $N = n_x n_y n_z b$, with $b = N_c (l_{\max} + 1)^2$. Thus, the matrices consist of $(n_x n_y n_z) \times (n_x n_y n_z)$ blocks of size $b \times b$. In the following, let

$$A = I - TG^r,$$

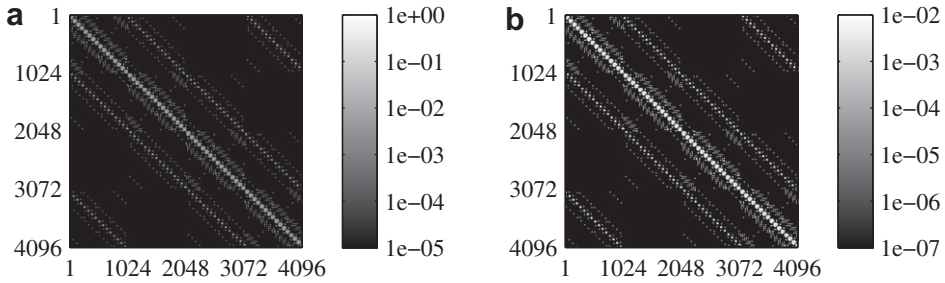


Fig. 2. Magnitude of the system matrix and of the difference of the system matrix and the preconditioner of a Ni–Pd alloy. (a) Magnitude of the system matrix A . (b) Magnitude of the difference $|M - A|$.

and denote the $b \times b$ block $(i, j, k), (i', j', k')$, $i, i' = 0, \dots, m - 1, j, j' = 0, \dots, n - 1, k, k' = 0, \dots, o - 1$ of a matrix B by $B_{(i,j,k),(i',j',k')}$.

Under the assumption that the blocks do vary, but not by very much, we next construct a 3-level block-circulant preconditioner M , with blocks

$$(M)_{(i,j,k),(i',j',k')} = M_{(i'-i) \bmod m, (j'-j) \bmod n, (k'-k) \bmod o}.$$

In analogy to Definition 3.5, the blocks are given by

$$M_{i,j,k} = \frac{1}{m \cdot n \cdot o} \sum_{i'=0}^{m-1} \sum_{j'=0}^{n-1} \sum_{k'=0}^{o-1} A_{(i',j',k'),((i'+i) \bmod m, (j'+j) \bmod n, (k'+k) \bmod o)}. \tag{6}$$

As M is block-diagonalized by $F_m \otimes F_n \otimes F_o \otimes I_b$, it can easily be inverted using 3D FFT and solved involving only $b \times b$ -blocks. The block solves can be improved further by precomputing the LU-decompositions of the blocks and using, for example, optimized BLAS2 routines for the actual application of the preconditioner. The result is a preconditioner that scales like $\mathcal{O}(b^3 m n o \log(\min\{m, n, o\}))$.

4. Numerical results

We implemented the construction of the preconditioner as outlined in the previous section in MATLAB, so the elapsed times might heavily depend on MATLAB. All matrices are stored as sparse matrices, so we expect the matrix-vector multiplication to be optimal, i.e., $\mathcal{O}(b^3 m n o)$. The inverses of the diagonal blocks are precomputed before the actual application and the application of the preconditioner is implemented as a 3D FFT, followed by a multiplication by the block diagonal matrix, and finally an inverse 3D FFT.

To verify the quality of our preconditioner, we first computed the magnitude of the difference $M - A$. The result is depicted in Fig. 2(b), as a reference the magnitude of A is given in Fig. 2(a).

Obviously the preconditioner matrix is very similar to the original matrix in a Frobenius-norm sense. Of course, this is not a proof that the preconditioner is efficient, but it suggests that the construction yields a good approximation to the matrix.

Next, we show the eigenvalues of A and M , as well as those of $M^{-1}A$, in Fig. 3(a) and (b). Obviously, the eigenvalues of both matrices are very close to each other, an observation that is underlined by the fact that the eigenvalues of the preconditioner applied to the original matrix nicely cluster around 1, with a few outliers larger than 1 in magnitude. This indicates that the proposed preconditioner will be very effective. Finally, the preconditioner has been tested with the MATLAB supplied GMRES and TFQMR routines. Note that the number of iterations implies different numbers of matrix-vector multiplications, as GMRES needs one matrix-vector multiplication per iteration, while TFQMR uses two.

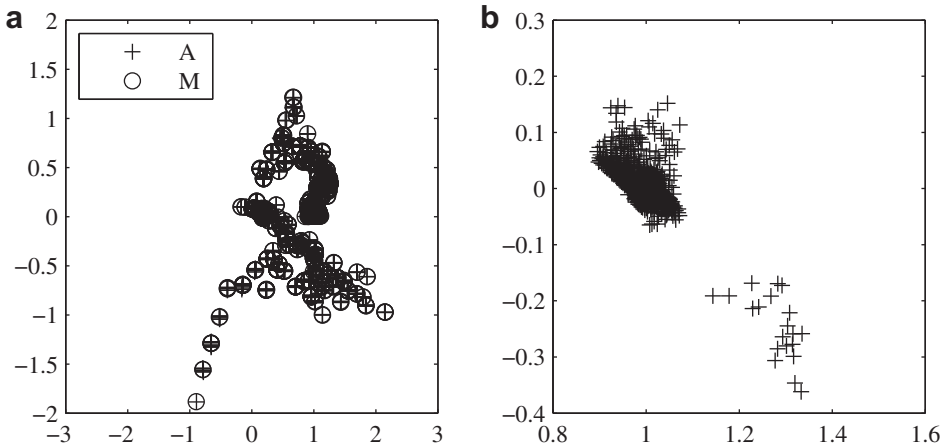


Fig. 3. Eigenvalues of the system matrix A , the preconditioner M and the preconditioned system matrix $M^{-1}A$ of a Ni-Pd alloy. (a) Eigenvalues of A and M . (b) Eigenvalues of $M^{-1}A$.

We consider three different systems here. The first is a palladium system, where the atoms are displaced slightly from the regular FCC-grid. The second system is a nickel–palladium alloy, where the atoms are located on the regular FCC-grid, but some palladium atoms are replaced by nickel atoms. Thirdly, we consider a combination of both cases, i.e., a nickel–palladium alloy, where some of the palladium atoms are replaced by nickel atoms within the structure and the atoms are slightly moved away from the regular positions. In all cases four cells are taken into account in each direction, i.e., $n_x = n_y = n_z = 4$, and, as the geometry is face-centered cubic, there are four atoms per cell, albeit the exact positions and kinds of the atoms inside of the boxes are distorted. The cut-off parameter in the multiple-scattering representation was set to $l_{\max} = 3$, resulting in a matrix size of $N = 4096$. The timings were measured on a machine with a Core 2 Quad Q9650 clocked at 3.00 GHz, running MATLAB 7.10. In all cases the setup of the preconditioner, i.e., the computation of (6), takes about 2s.

The results for the first case, using GMRES(20) and TFQMR as solvers, can be found in Fig. 4. The results are obtained for different energies E that are needed during the numerical evaluation of the integral (2). While the iteration count of the unpreconditioned solvers depends heavily on the energy, the number of iterations for the preconditioned solvers only mildly depends on the energy. Furthermore, the time needed for the solution is dramatically reduced, although one forward and one backward 3D FFT and a multiplication with the blockdiagonal matrix are needed for the application of the preconditioner. Because a matrix equation has to be solved, i.e., as many right hand sides as the number of unknowns, even the lowest energies benefit from using the preconditioner, as the setup of the preconditioner is relatively inexpensive, and even these systems are solved much faster by the preconditioned solver.

The results are similar for the case of a nickel–palladium alloy, where some of the palladium atoms are replaced by nickel atoms. In this case the atoms are still at the optimal FCC grid points, that is, the locations are not perturbed for either type of atom. The iteration counts and timings are reported in Fig. 5. The third case, combining a perturbation of both the type and location of atoms, is reported in Fig. 6.

Overall, the time needed for the solution of the linear system is reduced by a factor between 2.68 and 3.33 in the case that GMRES(20) is used for energy 12. For TFQMR, the factor is between 2.23 and 2.88. For the critical energy the savings are much bigger: If GMRES(20) is used the reduction factor is between 81.72 and 115.94, and for TFQMR the factor is between 25.10 and 32.18. In all cases, TFQMR outperforms restarted GMRES, independently of whether preconditioning is used or not. As noted before, the setup time for the preconditioner is about 2s and it can be neglected, at least in

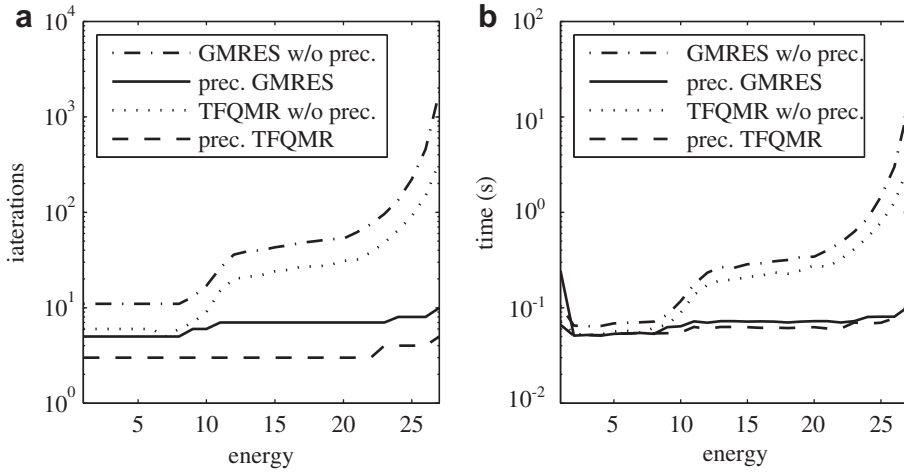


Fig. 4. Number of iterations and timings for the palladium system with slightly perturbed atomic positions. (a) Number of iterations. (b) Timings.

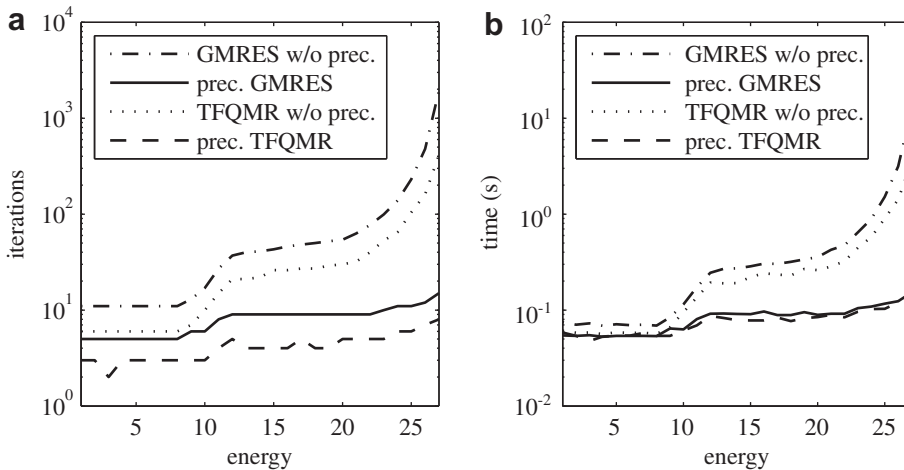


Fig. 5. Number of iterations and timings for the nickel-palladium system with regular atomic positions. (a) Number of iterations. (b) Timings.

the critical energy case. In the case of non-critical energies preconditioning still helps even for lower energies because of the many different right hand sides that have to be solved in the application.

We also did some comparisons of our preconditioner to other preconditioners, e.g., the incomplete LU factorization [6]. In Fig. 7 the results of a comparison of our preconditioner to the ILU using a drop tolerance of 10^{-2} in the third case can be found. The setup of the MATLAB-supplied ILU takes a little more than 5s on our test machine, so it is higher than the time needed to build the block-circulant preconditioner. Additionally, the number of iterations needed to solve the system preconditioned with the ILU is slightly higher, and the time needed to solve the system is much higher than in the case of the block-circulant preconditioner. Finally, the performance of ILU shows a larger dependence on the energy level, so the developed preconditioner outperforms it.

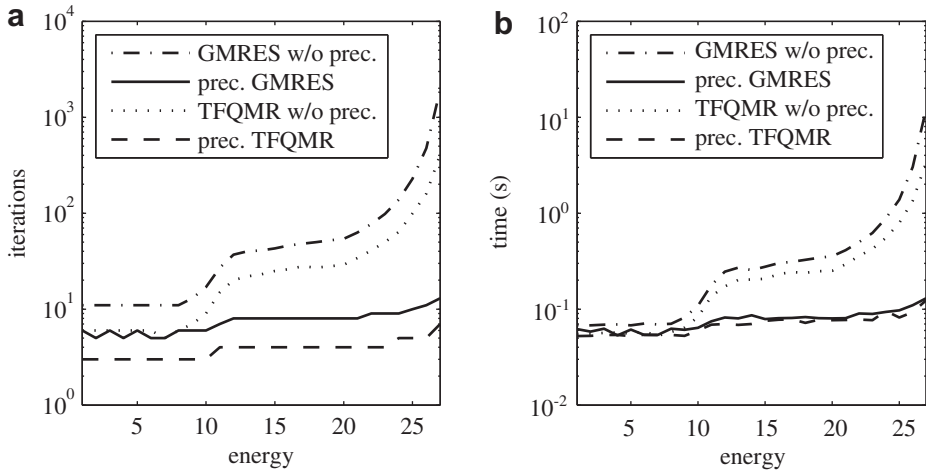


Fig. 6. Number of iterations and timings for the nickel-palladium system with slightly perturbed atomic positions. (a) Number of iterations. (b) Timings.

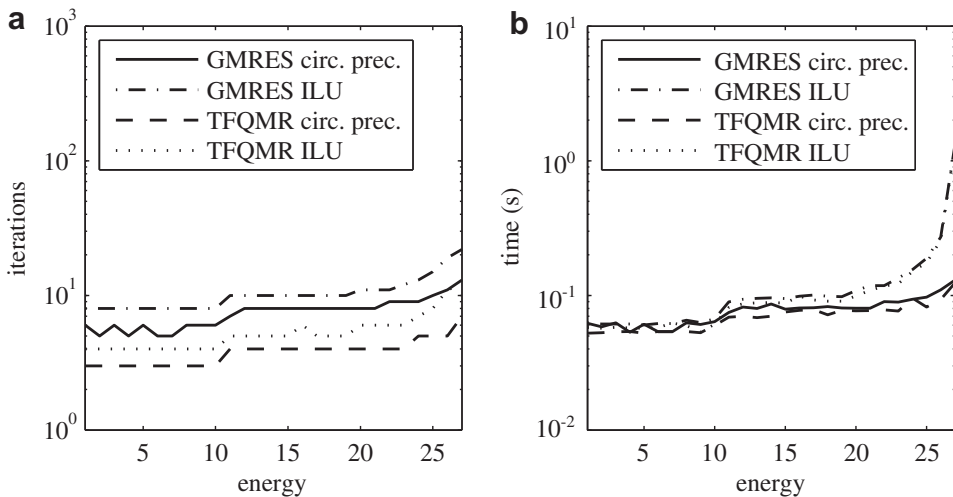


Fig. 7. Number of iterations and timings for the developed preconditioner and the ILU with a drop tolerance of 10^{-2} applied to the nickel-palladium system with slightly perturbed atomic positions. (a) Number of iterations. (b) Timings.

5. Conclusion

For the application studied in this paper, a constant-block preconditioner is very well suited. The systems only vary slightly from the ideal case, so the approximate inverse is close enough to the original inverse. As the boundary conditions in the application are periodic, the averaged version can be efficiently solved using the FFT, for which optimized versions exist that yield high performance on different computer architectures. The numerical results show that the approach yields a huge performance gain. Our preconditioner has been integrated into the production code `KKRnano` of the Peter Grünberg Institut at the Forschungszentrum Jülich. Currently, we are working on substantiating the numerical results presented here by theoretical analysis.

Acknowledgements

We like to thank Karsten Kahl who is working at the University of Wuppertal for discussions about different permutations of the system matrices.

Parts of this work are funded by the Umbrella cooperation of RWTH, Research Centre Jülich, and Technion-Israel.

References

- [1] R. Zeller, Towards a linear-scaling algorithm for electronic structure calculations with the tight-binding Korringa–Kohn–Rostoker Green function method, *J. Phys.: Condens. Matter* 20 (2008) 294215.
- [2] R.W. Freund, A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems, *SIAM J. Sci. Comput.* 14 (2) (1993) 470–482.
- [3] R. Gray, Toeplitz and circulant matrices: a review, Tech. Rep. 6504-1, Stanford University, Stanford, CA, 1977.
- [4] M.K. Ng, *Iterative Methods for Toeplitz Systems*, Oxford University Press, Oxford, 2004.
- [5] R.H. Chan, T.F. Chan, Circulant preconditioners for elliptic problems, *J. Numer. Linear Algebra Appl.* 1 (1) (1992) 77–101.
- [6] Y. Saad, *Iterative Methods for Sparse Linear Systems*, second ed., SIAM, Philadelphia, 2003.