A Complex Mix-Shifted Parallel QR Algorithm for the C-Method

Cihui Pan, Richard Dusséaux, Nahid Emad

A Complex Mix-Shifted Parallel QR Algorithm for the C-Method

Cihui Pan[1, 2, 3], Richard Dusséaux[1, *], and Nahid Emad[2, 3]

Abstract—The C-method is an exact method for analyzing gratings and rough surfaces. This method leads to large-size dense complex non-Hermitian eigenvalue. In this paper, we introduce a parallel QR algorithm that is specifically designed for the C-method. We define the "early shift" for the matrix according to the observed properties. We propose a combination of the "early shift", Wilkinson's shift and exceptional shift together to accelerate convergence. First, we use the "early shift" in order to have quick deflation of some eigenvalues. The multi-window bulge chain chasing and parallel aggressive early deflation are used. This approach ensures that most computations are performed in level 3 BLAS operations. The aggressive early deflation approach can detect deflation much quicker and accelerate convergence. Mixed MPI-Open MP techniques are used for performing the codes to hybrid shared and distributed memory platforms. We validate our approach by comparison with experimental data for scattering patterns of two-dimensional rough surfaces.

1. INTRODUCTION

The C-method is an efficient and versatile theoretical tool for analyzing gratings or rough surfaces illuminated by an electromagnetic wave [1–6]. It is based on Maxwell's equations solved in a non-orthogonal coordinate system fitted to the surface geometry. Discretizing the Maxwell's equations under a non-orthogonal coordinate system and separating variables lead to solving the eigenvalue problem of the high-dimensional, dense, complex and non-symmetric matrix. The scattered field is expanded as a linear combination of eigensolutions satisfying the outgoing wave condition. The boundary conditions allow the scattering amplitudes to be determined. The strength of the C-method is that it leads to the eigensolutions of the scattering problem. It is an accurate method and it can be used as a reference for the analytical methods [7]. The dominant computational cost for the C-method is the eigenvalue problem solution that is of the order of $O(N^3)$ where N is the size of matrix. The C-method must be improved with regard to this point, in particular, for analyzing random rough surfaces insofar as the average scattered intensity is estimated over results of several surface realizations.

In this paper, we focus on the numerical aspects of the C-method in order to reduce the computational time. Iterative eigensolvers, such as Bi-Conjugate Gradient methods, Krylov subspace methods or Jacobi-Davidson methods have been developed to deal with large-scale eigenvalue problems [8]. However, they have the possibility of missing some eigenvalues. Therefore, these iterative methods are inefficient for the C-method because all the eigenvalues and eigenvectors are needed. In contrast, the QR algorithm is based on similarity transformations and calculates all the eigenvalues and eigenvectors without any danger of missing some eigensolutions [9, 10]. We present the parallel QR algorithm that is specifically designed for the C-method. We define a technique, called "early shift" for the matrix according to the properties that we have observed. We mix the "early shift", Wilkinson's

shift and exceptional shift together to accelerate the convergence. Especially, we use the "early shift" first to have quick deflation of some eigenvalues [11, 12]. The multi-window bulge chain chasing and parallel aggressive early deflation (AED) are used [12, 13]. A Mixed Message Passing Interface and Open Multi-Processing techniques (Mixed MPI-Open MP) are utilized for parallel implementation of our application for hybrid shared and distributed memory platforms [14, 15].

The paper is structured as follows. In Section 2, we present the C-method. In Section 3, we introduce the parallel QR algorithm specifically designed for the C-method and we present the implementation details, architectures and platforms. In Section 4, we present numerical experiments and we validate numerical results by comparison with experimental data.

2. THE C-METHOD AS AN EIGENVALUE PROBLEM

2.1. Formulation of the Problem

Consider a periodic surface with a height profile $z = s(x, y)$ and a period D with respect to Ox and Oy axis (See Figure 1). The surface separates the vacuum ($\nu^{(1)} = 1$) from a material with a reflective index $\nu^{(2)}$. It is illuminated by a monochromatic plane wave with wavelength $\lambda^{(1)}$. The time-dependence factor varies as $\exp(j\omega t)$ where ω is the angular frequency. Each vector function is represented by its associated complex vector function and the time factor is omitted. Superscripts (1) and (2) designate quantities relative to the upper medium and the lower medium, respectively. The incident wave vector $\mathbf{k}_0$ is defined from the zenith angle θ_0 and the azimuth angle φ_0 [4]. For an incident wave in (a) polarization, the problem consists in working out the co- and cross-polarized components of the field scattered within the two media.
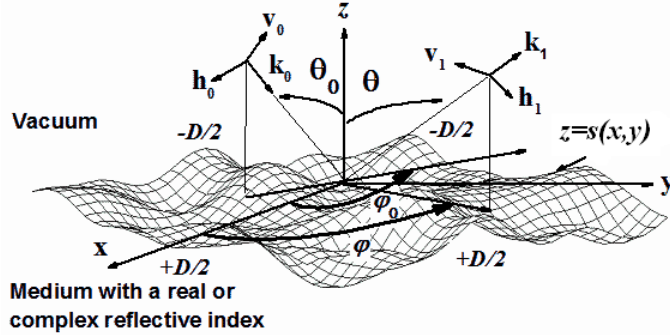

Figure 1. An elementary cell of a periodic surface illuminated by a plane wave under (θ_0, φ_0). $\mathbf{h}_0$ and $\mathbf{v}_0$ are the polarization vectors of the incident plane wave. In the vacuum, the diffracted elementary wave is characterized by the wave vector $\mathbf{k}_1$, the polarization vectors $\mathbf{h}_1$ and $\mathbf{v}_1$, the zenith angle θ and the azimuth angle φ [3, 4].

The C-method uses the translation coordinate system (u, v, ω) where $(u, v) = (x, y)$ and $w = z - s(x, y)$. So, the height function $z = s(x, y)$ coincides with the coordinate surface $w = 0$ and the change from Cartesian components $(a_x; a_y; a_z)$ of a vector $\vec{a}$ to covariant components (a_u, a_v, a_w) is given by [1, 3].

$$\begin{cases} a_u(x, y, w) = a_x(x, y, z) + \dfrac{\partial s(x, y)}{\partial x} a_z(x, y, z) \\ a_v(x, y, w) = a_y(x; , y, z) + \dfrac{\partial s(x, y)}{\partial y} a_z(x, y, z) \\ a_w(x, y, w) = a_z(x, y, z) \end{cases} \tag{1}$$

The covariant components E_u, E_v, H_u and H_v are parallel to the rough surface and satisfy the boundary value problem. Because the boundary surface coincides with a coordinate surface, the boundary value problem is simplified. In a source-free medium, it can be shown from the time harmonic Maxwell

equations and the constitutive relations expressed in the translation system that the component $\psi = E_w$ or $\psi = ZH_w$ obeys to the propagation equation:

$$\frac{\partial^2 \psi}{\partial u^2} + \frac{\partial^2 \psi}{\partial v^2} + g^{ww}\frac{\partial^2 \psi}{\partial w^2} + k^2 \psi + \frac{\partial}{\partial w}\left[g^{uw}\frac{\partial \psi}{\partial u} + \frac{\partial g^{uw}\psi}{\partial u} + g^{vw}\frac{\partial \psi}{\partial v} + \frac{\partial g^{vw}\psi}{\partial v}\right] = 0 \tag{2}$$

k is the wave number of the medium under consideration and Z, this impedance. Terms g^{uw}, g^{vw} and g^{ww} are elements of metric tensor which depend on the derivatives of function $s(u,v)$ with respect to u and v [3].

$$g^{uw} = -\frac{\partial s}{\partial u}$$
$$g^{vw} = -\frac{\partial s}{\partial v} \tag{3}$$
$$g^{ww} = 1 + \left(\frac{\partial s}{\partial u}\right)^2 + \left(\frac{\partial s}{\partial v}\right)^2$$

As shown in [3], E_u, E_v, H_u and H_v can be expressed in terms of E_w and H_w only:

$$\frac{\partial^2 E_u}{\partial w^2} + k^2 E_u = \frac{\partial^2 E_w}{\partial u \partial w} - k^2 g^{uw} E_w - jk\left(g^{vw}\frac{\partial ZH_w}{\partial w} + \frac{\partial ZH_w}{\partial v}\right) \tag{4}$$

$$\frac{\partial^2 E_v}{\partial w^2} + k^2 E_v = \frac{\partial^2 E_w}{\partial v \partial w} - k^2 g^{vw} E_w + jk\left(g^{uw}\frac{\partial ZH_w}{\partial w} + \frac{\partial ZH_w}{\partial v}\right) \tag{5}$$

$$\frac{\partial^2 H_u}{\partial w^2} + k^2 H_u = \frac{\partial^2 H_w}{\partial u \partial w} - k^2 g^{uw} H_w + j\frac{k}{Z}\left(g^{vw}\frac{\partial E_w}{\partial w} + \frac{\partial E_w}{\partial v}\right) \tag{6}$$

$$\frac{\partial^2 H_v}{\partial w^2} + k^2 H_v = \frac{\partial^2 H_w}{\partial v \partial w} - k^2 g^{vw} H_w - j\frac{k}{Z}\left(g^{uw}\frac{\partial E_w}{\partial w} + \frac{\partial E_w}{\partial v}\right) \tag{7}$$

2.2. Eigenvalue System

In [3], a procedure is proposed for solving the propagation Equation (2). $\psi(x, y, w)$ is represented in terms of the quasi-periodic functions $\exp(-j\alpha_p x)\exp(-j\beta_q y)$:

$$\psi(x, y, w) = \sum_p \sum_q \psi_{pq}(w)\exp(-j\alpha_p x)\exp(-j\beta_q y) \tag{8}$$

where

$$\alpha_p = k^{(1)}\sin\theta_0\cos\varphi_0 + p\frac{2\pi}{D}, \quad \beta_q = k^{(1)}\sin\theta_0\sin\varphi_0 + q\frac{2\pi}{D} \tag{9}$$

Substituting (8) and (9) into (2) and projecting on the quasi-periodic functions gives:

$$\frac{j}{k^{(1)}}\frac{\partial}{\partial w}\left[\sum_{p,q}\left(\frac{\alpha_s}{k^{(1)}}g^{uw}_{s-p,t-q} + g^{uw}_{s-p,t-q}\frac{\alpha_p}{k^{(1)}} + \frac{\beta_t}{k^{(1)}}g^{vw}_{s-p,t-q} + g^{vw}_{s-p,t-q}\frac{\beta_q}{k^{(1)}}\right)\psi_{pq}(w)\right]$$

$$+\frac{j}{k^{(1)}}\frac{\partial}{\partial w}\left[\sum_{p,q}g^{ww}_{s-p,t-q}\psi'_{pq}(w)\right] = \frac{\gamma^2_{st}}{k^{(1)2}}\psi_{st}(w) \tag{10}$$

where

$$\frac{j}{k^{(1)}}\frac{\partial \psi_{pq}(w)}{\partial w} = \psi'_{pq}(w) \tag{11}$$

and $\gamma^2_{st} = k^2 - \alpha^2_s - \beta^2_t$. It is interesting to note that γ_{st} is the propagation coefficient of an elementary plane wave with respect to the Oz axis [1]. Terms $g^{uw}_{p,q}$, $g^{vw}_{p,q}$ and $g^{ww}_{p,q}$ are the Fourier coefficients of the periodic functions g^{uw}, g^{vw} and g^{ww}. Equations (10) and (11) can be written in matrix form:

$$\frac{j}{k^{(1)}}L_l\frac{\partial}{\partial w}\begin{pmatrix}\vec{\psi}\\\vec{\psi'}\end{pmatrix} = L_r\begin{pmatrix}\vec{\psi}\\\vec{\psi'}\end{pmatrix} \tag{12}$$

ψ_{pq} and ψ'_{pq} are the components of vectors $\vec{\psi}$ and $\vec{\psi}'$. For a truncation order M, $-M \le p, q \le +M$. As a result, L_l and L_r are $2M_s$-square matrices with $M_s = (2M+1)^2$. The elementary solution of Equation (12) is defined as follows:

$$\begin{pmatrix} \vec{\psi}_n(w) \\ \vec{\psi}'_n(w) \end{pmatrix} = \begin{pmatrix} \vec{\phi}_n \\ \vec{\phi}'_n \end{pmatrix} \exp(-jk^{(1)}r_n\,w) \tag{13}$$

$\vec{\phi}_n$ and $\vec{\phi}'_n$ are the upper eigenvector and the lower eigenvector associated with the eigenvalue r_n. They are solutions of the eigenvalue problem (14).

$$r_n L_l \begin{pmatrix} \vec{\phi}_n \\ \vec{\phi}'_n \end{pmatrix} = L_r \begin{pmatrix} \vec{\phi}_n \\ \vec{\phi}'_n \end{pmatrix} \tag{14}$$

System (14) gives $2M_s$ eigensolutions $\psi_n(x,y,w)$ defined as follows:

$$\psi_n(x,y,w) = \sum_{p=-M}^{+M} \sum_{q=-M}^{+M} \varphi_{n,pq}\exp(-j\alpha_p x)\exp\left(-j\beta_q y\right)\exp\left(-jk^{(1)}r_n\,w\right) \tag{15}$$

For each medium (m), the C-method requires the solution of an eigenvalue problem. The signs of the real and imaginary parts of the eigenvalue define the nature of the wave corresponding to the elementary wave function (15). $\psi_n^{(m)}(x,y,w)$ represents an outgoing wave travelling without attenuation if $\mathrm{Re}\left[(-1)^m r_n^{(m)}\right] < 0$ and $\mathrm{Im}(r_n^{(m)}) = 0$. For an evanescent wave, $\mathrm{Im}\left[(-1)^m r_n^{(m)}\right] > 0$.

The component $\psi_s^{(m)}(x,y,w)$ of the field scattered within the medium (m) is expanded as a linear combination of eigensolutions $\psi_n^{(m)}(x,y,w)$ satisfying the outgoing wave condition. Among the $2M_s$ eigenfunctions, there are M_s of them that correspond to outgoing waves and as many of them to incoming waves.

$$\psi_s^{(m)}(x,y,w) = \sum_{n=1}^{M_s} A_n^{(m)}\,\psi_n^{(m)}(x,y,w) \tag{16}$$

For an incident wave in (a) polarization, we deduce from Eq. (4) to Eq. (7) the (ha)-polarized components $(E_{su}^{(m,ha)}, E_{sv}^{(m,ha)}, H_{su}^{(m,ha)}, H_{sv}^{(m,ha)})$ by taking $E_{sw}^{(m)} = 0$ and $H_{sw}^{(m)} = \psi_s^{(m)}$. The (va)-polarized components $(E_{su}^{(m,va)}, E_{sv}^{(m,va)}, H_{su}^{(m,va)}, H_{sv}^{(m,va)})$ by taking $E_{sw}^{(m)} = \psi_s^{(m)}$ and $H_{sw}^{(m)} = 0$. The co-amplitudes $A_n^{(aa)}$ and the cross-amplitudes $A_n^{(ba)}$ are found by solving the boundary value problem and a $2M_s$-dimensional matrix system. In this section, we have succinctly presented numerical implementation of the C-method. For more information, we refer the reader to Reference [3] for 2D-gratings and to References [4–6] for 2D random rough surfaces.

3. ALGORITHMS

The most time-consuming part of C-method is to compute the eigenvalues and eigenvectors of $A = L_r^{-1}L_l$ where A is a large size, complex, dense and non-Hermitian matrix. For determining the complete spectrum, the most powerful and stable algorithm is the QR algorithm [8–10]. We choose this algorithm to keep the stability and to save the computation time.

One of the important aspects of QR method is its capacity to conserve over iterations the Hessenberg form of a matrix. This prevents computations on the zeros and therefore has a significant impact on the cost of calculations of an iteration. Thus, a first stage of the method consists to similarly transform the original matrix to a Hessenberg matrix and then apply the QR method on this matrix. The convergence of the QR method for a Hessenberg matrix can be improved dramatically by introducing shifts (Wilkinson's shift, for example) or its variant techniques (double-shifts, multi-shift, etc.) into the algorithm. The iterations of a shifted Hessenberg QR algorithm introduce a few non-zero values under the sub-diagonal elements of the upper Hessenberg matrix [11, 12]. This group of non-zero elements is called a bulge. The algorithm allows pushing this bulge over the iterations until his vanishing (chasing

the bulge). In general, as the complement to the use of the shift, the deflation technique is used in the QR algorithm. This means that once an eigenvalue has converged, the corresponding rows and columns of the Hessenberg matrix will be removed and thus, the following QR iterations will be applied on a smaller matrix.

We make use of three techniques in the implementation of our parallel QR algorithm: early shift, parallel bulge chasing and parallel aggressive early deflation (AED). We introduce in parallel QR algorithm, the early shifts to give approximation of a part of the eigenvalues of the matrix. For the bulge chasing, instead of only a single bulge, containing two shifts, a chain of several tightly coupled bulges, each containing two shifts, is chased in the course of one multi-shift QR iteration [11, 12]. This idea allows performing most of the computational work in terms of matrix-matrix multiplications to benefit from highly efficient level 3 basic linear algebra subprograms (BLAS, the level 3 contains matrix-matrix operations) [16]. The idea of AED allows to detect converged eigenvalues much earlier than conventional deflation strategies. We will first present QR sequential algorithm and the shift strategy to accelerate the convergence, then we present all the parallel techniques for this specifically designed QR algorithm.

3.1. Double Shifted QR Algorithm

The QR algorithm is an iterative method for reducing a matrix to almost upper triangular form by unitary similarity transformations. The idea is to compute the Schur decomposition of a N-square complex matrix A (here, $N = 2M_s$ with $M_s = (2M + 1)^2$): $T = W^H A W$ [9] where T is an upper triangular matrix and W^H is the conjugate transpose of W which is the Schur matrix of A. Formally, let $A_0 := A$. $A_0 := A$. At the k-th step (starting with $k = 0$), we compute the QR decomposition: $A_k = Q_k R_k$ where Q_k is an unitary matrix and R_k is an upper triangular matrix. We then form $A_{k+1} = R_k Q_k$ and:

$$A_{k+1} = R_k Q_k = Q_k^H A_k Q_k \tag{17}$$

So all the matrices A_k are similar to each other and hence they have the same eigenvalues. By this iteration, the matrices A_k converge to an upper triangular matrix, the Schur form of A. Usually, the QR algorithm starts with the Hessenberg decomposition [10]:

$$H = Q^H A Q \tag{18}$$

H is a Hessenberg matrix and Q is a unitary matrix. This is because the Hessenberg form is preserved by the QR algorithm and this form can decrease the cost of the QR iterations [10]. There are several ways of Hessenberg reduction such as Gram-Schmidt transformation, Householder reduction and Givens rotation [10]. Efficient parallel algorithm for this Hessenberg reduction is implemented in the ScaLAPACK (Scalable Linear Algebra PACKage) routine PZGEHRD [17]. So we will focus on the iterative part that comes after this Hessenberg reduction.

The convergence of the Hessenberg QR algorithm can be improved dramatically by introducing spectral shifts into the algorithm. Nevertheless, in some cases, the shifted Hessenberg QR algorithm does not work well. For the single shift, when the element $h_{N-1,N-1}$ of the Hessenberg matrix H is negligible with respect to $h_{N,N-1}$, the convergence could be slow even if the Rayleigh quotient shift gives a very good approximation (e.g., $h_{N,N-1}$ is very small). For this reason, we make use of double shifted Hessenberg QR algorithm. This technique is also used for real matrices with complex conjugate eigenvalues because it allows avoiding the introduction of complex numbers in calculations during QR iterations. Indeed, the so-called implicit Q theorem allows us to use the implicit double shift QR algorithm without explicitly computing the QR decomposition [10]. The algorithm is characterized by a "bulge chasing" procedure.

Suppose that two shifts σ_1 and σ_2 of the Hessenberg matrix H are given. The implicit double shift QR algorithm proceeds as follows:

a) Calculate the first column of the shift polynomial

$$v = (H - \sigma_1 I)(H - \sigma_2 I)e_1 = \begin{pmatrix} * \\ * \\ * \\ * \\ 0 \\ \updownarrow \\ 0 \end{pmatrix} \tag{19}$$

b) Construct a 3×3 Householder transformation Q_1 such that the second and third entries of v are transformed to zero (i.e., $Q_1 v$). The similarity transformation gives the updated matrix H_1:

$$H_1 = Q_1^H H Q_1 = \begin{pmatrix} * & * & * & * & * & * & \leftrightarrow \\ X & X & X & * & * & * & \leftrightarrow \\ X & X & X & * & * & * & \leftrightarrow \\ X & X & X & * & * & * & \leftrightarrow \\ 0 & 0 & 0 & * & * & * & \leftrightarrow \\ 0 & 0 & 0 & 0 & * & * & \leftrightarrow \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow \end{pmatrix} \tag{20}$$

The Hessenberg structure is damaged by the bulge that we denote with symbol "X".

c) Construct a 3×3 Householder transformation Q_2 such that the third and fourth entries of the first column of H_1 reduce to zero. The similarity transformation gives the updated matrix H_2:

$$H_2 = Q_2^H H_1 Q_2 = \begin{pmatrix} * & * & * & * & * & * & \leftrightarrow \\ * & * & * & * & * & * & \leftrightarrow \\ 0 & X & X & X & * & * & \leftrightarrow \\ 0 & X & X & X & * & * & \leftrightarrow \\ 0 & X & X & X & * & * & \leftrightarrow \\ 0 & 0 & 0 & 0 & * & * & \leftrightarrow \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow \end{pmatrix} \tag{21}$$

d) Continue similar operations to chase the bulge. In general, construct a 3×3 Householder transformation Q_k such that the $(k+1)$th and $(k+2)$th entries of the $(k-1)$th column of H_k are mapped to zero. Applying the corresponding similarity transformation to H_k results the updated matrix H_{k+1} where $k = 2, 3, \ldots, N-1$. The bulge will be chased to vanish at the bottom right corner and leads to zeros, thus deflations. For example, H_3 will be look like as follows:

$$H_3 = Q_3^H H_2 Q_3 = \begin{pmatrix} * & * & * & * & * & * & \leftrightarrow \\ * & * & * & * & * & * & \leftrightarrow \\ 0 & * & * & * & * & * & \leftrightarrow \\ 0 & 0 & X & X & X & * & \leftrightarrow \\ 0 & 0 & X & X & X & * & \leftrightarrow \\ 0 & 0 & X & X & X & * & \leftrightarrow \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow \end{pmatrix} \tag{22}$$

3.2. Early Shift

Some eigenvalues of the scattering matrix can be approximated as follows [1]:

$$r_n^{(m)} \approx \pm \tilde{\gamma}_{pq}^{(m)} = \pm \sqrt{\nu^{(m)2} - \tilde{\alpha}_p^2 - \tilde{\beta}_q^2} \tag{23}$$

where $\tilde{\alpha}_p = \alpha_p/k^{(1)}$, $\tilde{\beta}_q = \beta_q/k^{(1)}$ and $\tilde{\gamma}_{pq}^{(m)} = \gamma_{pq}^{(m)}/k^{(1)}$. As shown in [1], the values $\tilde{\gamma}_{pq}^{(m)}$ constitute very good approximations when the indexes p and q are small relative to the matrix size. In general, if $n^{(m)}$ is real or has a very small imaginary part, for the pair (p, q) that satisfies $\tilde{\alpha}_p^2 + \tilde{\beta}_q^2 < \mathrm{Re}(\nu^{(m)})$, the approximation can be quite accurate. We recall that for a lossless medium (m), a real value $\tilde{\gamma}_{pq}^{(m)}$ can be associated with two angles defining a propagation direction with $\tilde{\alpha}_p = \sin\theta_n^{(m)} \cos\varphi_n^{(m)}$,

$\tilde{\beta}_q = \sin\theta_n^{(m)} \sin\varphi_n^{(m)}$ and $\tilde{\gamma}_{pq}^{(m)} = \pm\cos\theta_n^{(m)}$ (see Figure 1). The real eigenvalue $r_n^{(m)}$ defines the propagation direction of the eigenfunction $\psi_n^{(m)}(x, y, w)$. In [1], this approximation is used in order to describe the asymptotic scattered field in the lossless medium by a sum of elementary plane waves propagating without any attenuation. As shown in [1], this procedure is numerically efficient.

Here, the expression of $\tilde{\gamma}_{pq}^{(m)}$ can be used as shifts to approximate the real eigenvalues or eigenvalues with very small imaginary part. The convergence is very quick due to the good approximation. We call this the "early shift". This specific shift is used in pairs corresponding to above double shift QR algorithm. One pair of the "early shift" will create a bulge to be chased. Each pair will be used only once during one iteration of double shifted Hessenberg QR which corresponds to $(n-1)$ right and left multiplications of A by Householder reflectors. Wilkinson shift can be used after the "early shift" [10]. The "early shift" represents a further step forward in relation to previous works on the C method.

3.3. Parallel QR with Tightly Coupled Bulge Chasing

The parallel bulge chasing algorithm was first proposed by Braman et al. [11]. In order to benefit from the level 3 BLAS, they parallelize the bulge chasing procedure by performing the chasing of multiple chains of tightly coupled bulges. With the delay and accumulate technique, the main computation work becomes the matrix-matrix multiplications. The procedure of intra-block chasing and inter-block chasing are described below.

To describe the parallel algorithm, we first introduce the data layout mapping in a distributed memory environment as follows:

- The processors are arranged into a $p_r \times p_c$ rectangular mesh labeled from $(0, 0)$ to $(p_r - 1, p_c - 1)$ according to their specific positions indices in the mesh. Usually the values of p_r and p_c are set to be as close as possible.
- The $N \times N$ matrix is partitioned in 2D block cyclic scheme [17] and is mapped on our $p_r \times p_c$ grid. The block size is $M_b \times N_b$ and we require the block to be square $M_b = N_b$. Generally, a processor will store a collection of non-contiguous blocks. Figure 2 shows an example with a fourth-order grid where $p_r = p_c = 2$. The four processors are represented by different colors. If the matrix size is $N = 16$, then the block size is 4x4 and the processor $(0, 0)$ will store the elements $A(1:4, 1:4)$, $A(1:4, 9:12)$, $A(9:12, 1:4)$ and $A(9:12, 9:12)$. An array descriptor stores the details of data layout. The mapping between entries of the global matrix and their corresponding locations in the memory can be established from the array descriptor.
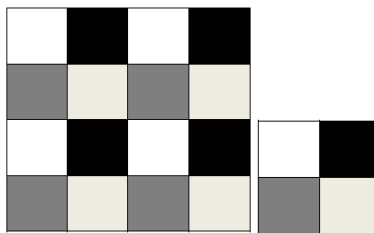

Figure 2. 2D block cyclic scheme with four processors and $p_r = p_c = 2$.

Locally, each processor in the mesh may also utilize multithreading often associated with shared memory architectures. This can be seen as adding another level of explicit parallelization by organizing all the $p_r \times p_c$ processors into a three-dimensional mesh [12].

We use the shifts that are mentioned earlier to introduce the chain of bulges into diagonal blocks. Each of the chains resides on a different diagonal block. We choose the number of shifts in such a way that each chain covers at most half of the data layout block. The "early shift" is distributed from left-upper diagonal blocks to right-lower diagonal blocks. Each "early shift" is used once. When there is no "early shift" to distribute, we use Wilkinson shift [10].

For the intra-block chasing where the chain is chased from the top left corner to the lower right corner within a contiguous diagonal block, we may use a sequence of 3×3 Householder transformations
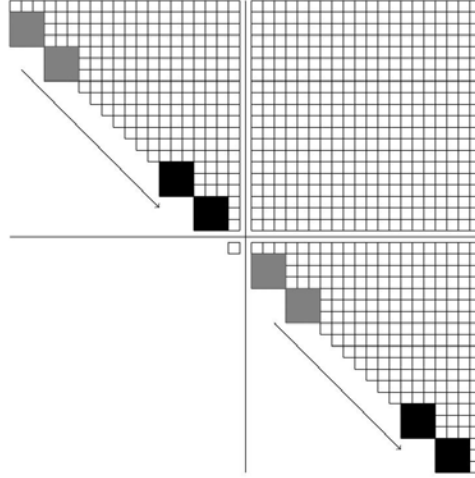
Figure 3. Intra-block parallel bulge chasing. The grey bulges are chased to the black bulges. The matrix under consideration is a 40×40 matrix. It is partitioned into 4 square blocks of size 20×20.

to chase the chain of bulges down some rows to the down right-hand corner of the contiguous diagonal block. We start from the lowest bulge of the block and chase one bulge at a time. The intra-block chasing can be performed locally on the process that owns this chain and simultaneously between different diagonal contiguous blocks that saves computation time. Figure 3 shows how the intra-block chasing are performed, the grey bulges are chased to the black bulges. After the bulge chasing within the diagonal block, the accumulated unitary matrices are sent to the corresponding processors in order to update the off-diagonal blocks. The off-diagonal blocks are then updated by matrix-matrix multiplications that use level 3 BLAS. The broadcasts are sent in parallel. In order to avoid conflicts in the intersecting parts, they are performed first in the row direction and then in the column direction (See Figure 3 for the intra-block chasing with $p_r = p_c = 2$ and $M_b = N_b = 20$).

For the inter-block chasing, for each diagonal blocks in which the bulge chains reside, we create copies of its neighbors and it becomes similar to the case of intra-block chasing. Figure 4 illustrates the procedure with $p_r = p_c = 2$ and $M_b = N_b = 20$, the grey bulges are chased to the black bulges. More precisely, the processor that stores the grey bulges creates a copy of the block on each side of the border. Then we can perform the chasing locally, just as in the intra-block chasing and broadcast the corresponding orthogonal factors to the blocks on both sides of the cross border. The updated neighboring blocks are sent to its owner. To update the corresponding off-diagonal blocks, we broadcast orthogonal matrix accumulated in the diagonal chasing stage to the corresponding rows/columns of processors that are involved in off-diagonal updating. Then each involved processor exchanges data blocks with its neighbor as illustrated in Figure 4 in the two large gray blocks. The off-diagonal blocks are then updated by multiplication with the accumulated orthogonal matrix. We perform inter-block chasing first for the odd-numbered blocks and then for the even-numbered blocks. This odd-even manner avoids conflicts between different tightly coupled chains [12].

For each diagonal block, the corresponding orthogonal transformations are accumulated into an orthogonal factor. Each orthogonal has the following form:

$$U = \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix} \tag{24}$$

U_{12} is a lower triangular matrix and U_{21} is a upper triangular matrix. As a result, matrix multiplications by U will be broken into two dense by dense matrix multiplications and two triangular by dense matrix multiplications. Computation time is saved because of the triangular structure and the required memory is the same (of the order of $0(N^2)$). The method presented in this section has already been published. However, the published papers are for real matrices [10, 12]. The implementation here concerns the complex matrices including the adaptation of the mixed-shift strategy. It is a novelty.
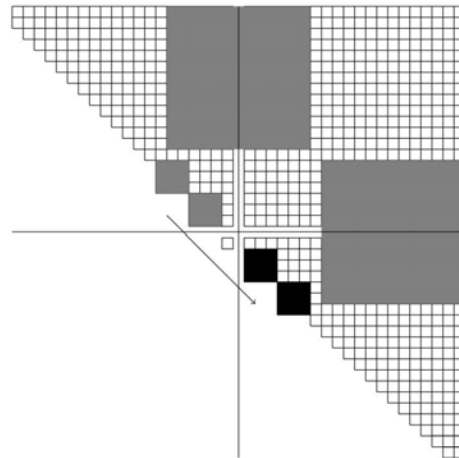
Figure 4. Inter-block parallel bulge chasing. The dark grey large parts exchange data blocks with neighbors when updating off-diagonal blocks.

3.4. Parallel AED

The AED algorithm was proposed in [13]. We divide the Hessenberg matrix H as follows:

$$H = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{pmatrix} \tag{25}$$

where H_{11} and H_{33} are $(N-k-1)$-and k-square matrices, respectively. We use the pipeline parallel QR algorithm to find the Schur decomposition of H_{33} with $H_{33} = VTV^H$ and perform the following similarity transformation:

$$\begin{pmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V^H \end{pmatrix} \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13}V \\ H_{21} & H_{22} & H_{23}V \\ 0 & s & T \end{pmatrix} \tag{26}$$

I is the $(N-k-1)$-square identity matrix and T an upper triangular k-square matrix. Now the matrix looks like as in Figure 5 where the spike s is denoted as the gray part.

3.5. Hardware and Software Platforms

For the sequential case, there are two stages: the Hessenberg matrix implementation and the triangular decomposition of the Hessenberg matrix. Each of these stages requires one matrix and a few buffers with "negligible" sizes for intermediate calculations. For the parallel QR case, the matrix is distributed over the computation nodes. It is necessary to store it somewhere and to update it regularly. We must therefore store at most two matrices and we need a few buffers for the intermediate calculations. The computer memory required is proportional to N^2.

The experiments are performed on the machine Poincare hosted by IDRIS national computing center in France (http://www.idris.fr/). This machine is IBM computer, composed mainly iDataPlex dx360 M4 servers: 92 nodes equipped with 2 Sandy Bridge E5-2670 processors (2.60 GHz, 8 cores every processor, 16 cores every node) and 32 Gb memory every node.

The following tools are installed: mkl 11.0, intelmpi 4.0.3, lapacke 3.5_gnu47. The program is written in Fortran and compiled with Intel compiler ifort. Our implementation is based on a simple imitation of the ScaLAPACK routine PDHSEQR and we use this to a complex implementation with our own shift strategy. We adopt the recommended values such as the size of the deflation window and the tuning parameter NIBBLE that determines when to skip a QR sweep and to perform AED in [12]. For all the following experiments, $M_b = N_b = 50$.
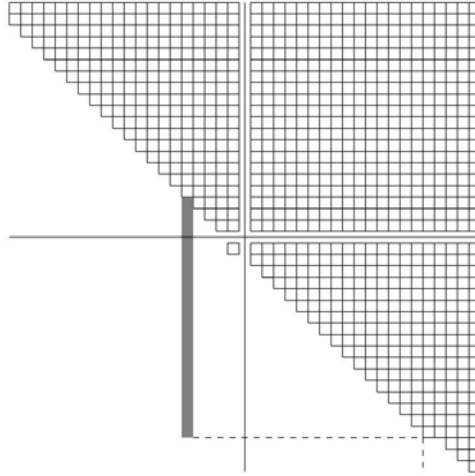
Figure 5. Aggressive early deflation. The grey spike contains the vector s.

4. NUMERICAL RESULTS

4.1. Surfaces under Consideration

The parallel QR algorithm presented in Section 3 can be used for analyzing crossed gratings or two-dimensional random gratings. As an illustration, we consider in this section periodic random surfaces having a Gaussian height distribution. The correlation function is Gaussian and isotropic. We note σ_s the root mean square height of the surface and l_s, the correlation length, respectively. For an incident wave in (a) polarization and a scattered wave in (b) polarization, the average scattered intensity in the upper medium is derived from the scattering amplitudes $A_n^{(ba)}$ associated with angles (θ_n, φ_n) [18]. The average scattered intensity is estimated by averaging the scattering amplitudes over N_r realizations. The interface realization is obtained by filtering of uncorrelated white noise [18, 19].

4.2. Reduction on the Computational Time

To compare the computation times of sequential algorithm and parallel algorithm, we present Figure 6 that is based on numerical result of one realization. The perfectly conducting surface we consider is of 64 square wavelengths and $\sigma_s = \lambda^{(1)}$ and $l_s = \sqrt{2}\lambda^{(1)}$. The incident angle is chosen as $\theta_0 = 30°$ and $\varphi_0 = 0°$. The truncation order M is equal to 28 and the matrix size is $N = 6498$.

Figure 6 gives the results of the pairs $(p_r; p_c) = (1; 1)$ $(2; 2)$ $(4; 4)$ $(6; 6)$ and $(8; 8)$. Considering one core, the proposed algorithm with respect to the sequential one leads to a substantial reduction in the computational time (factor higher than 5). With the new version of parallel strategy by using 16 cores for one simulation, the solutions for 100 eigenvalues problems associated with 100 surface realizations require approximately 12 minutes if we use 1600 cores. The naive parallel strategy based on using the sequential algorithm and on solving of individual realizations on individual processors requires approximately 330 minutes. The proposed parallel strategy can be significantly more efficient than the naive parallel strategy.

We also compare the computation cost of sequential algorithm and parallel algorithm when the truncation order M is varying. Figure 7 shows this comparison based on a single surface realization. For $M = 12$, the matrix size is $N = 1250$, for $M = 36$, $N = 10658$, respectively. The surface parameters are the same as above. For the parallel QR algorithm, the number of cores is fixed with $p_r = p_c = 4$. This curves shows that the reduction in the computational time is important. For instance, the ratio is close to 25 when $M = 28$.
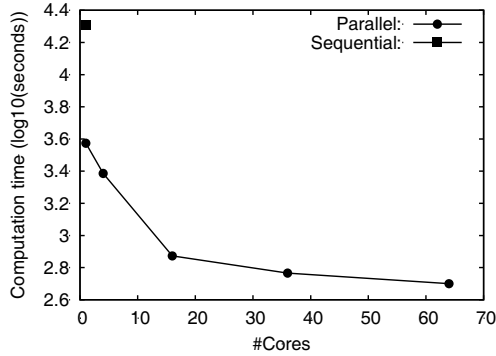
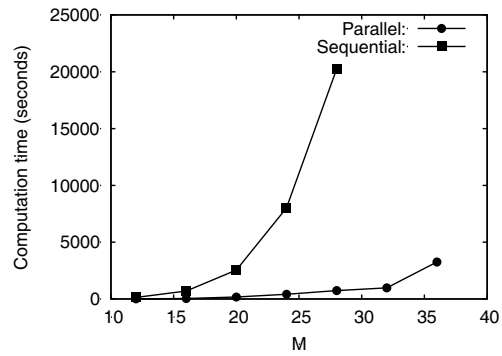**Figure 6.** Computation times with respect to number of cores.

Figure 7. Computation times with respect to truncation order.

4.3. Comparison with Experimental Data

We use the parallel QR algorithm for analyzing random grating and we compare numerical results with published experimental data considering random rough surfaces. We analyze moderately rough and very rough surfaces. For all simulations, $D = 8\lambda$, $M = 28$ and $N_r = 200$. First, we consider a surface with a moderate roughness, $\sigma_s = 0.352\lambda^{(1)}$ and $l_s = 2.21\lambda^{(1)}$, and illuminated under $\theta_0 = 55°$ and $\varphi_0 = 0°$. The reflective index of the lower medium is $\nu^{(2)} = 1.62 - 0.001j$. Figure 8 shows the (vv)-polarized differential reflection coefficient (DRC) as a function of the observation angle in the incidence plane [20]. It is noteworthy that the DRC curve presents a minimum similar to the Brewster angle for a planar surface (By analogy with reflection from a smooth surface, a lossless dielectric with a refractive index equal to 1.62 provides a Brewster angle close to 58°). The comparison with experimental data that come from [20] is excellent. The comparison is also conclusive for (hh)-results.
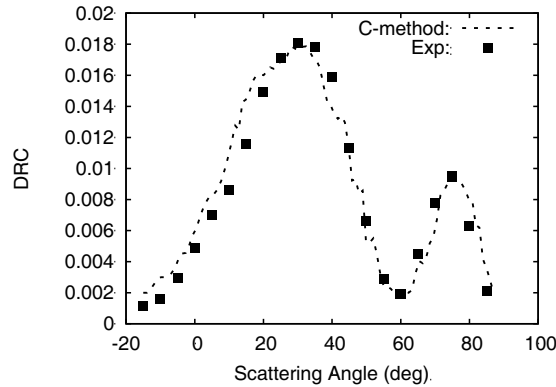

Figure 8. Differential reflection coefficient versus observation angle in the incidence plane. Moderately rough surface. Polarization (vv).

Figure 9 shows the (vv)-polarized normalized bistatic cross section as a function of the observation angle in the incidence plane for a perfectly conducting surface [21]. Figure 10 gives results for the (hv)-polarized bistatic cross section. We consider a very rough surface for which the predictions of the standard analytic methods are inaccurate [7]. Roughness parameters are: $\sigma_s = \lambda^{(1)}$ and $l_s = \sqrt{2}\lambda^{(1)}$ and the surface is illuminated under $\theta_0 = 20°$ and $\varphi_0 = 0°$. The C-method used with $M = 28$ leads to an error on the power balance smaller than 5% for all the realizations. The mean error estimated over the 200 realizations is smaller than 2×10^{-3}.

The early shift is used. The approximations of real eigenvalues by (23) are good. As an example,

Figure 9. Normalized bistatic scattering cross section versus observation angle in the incidence plane. Very rough surface. Polarization (*vv*).
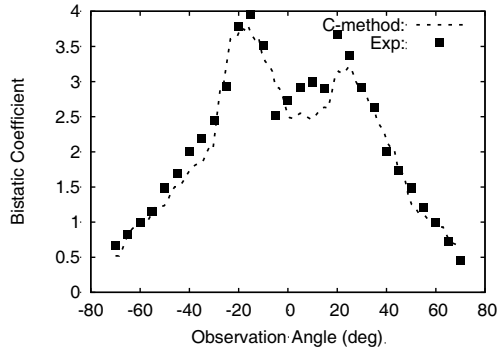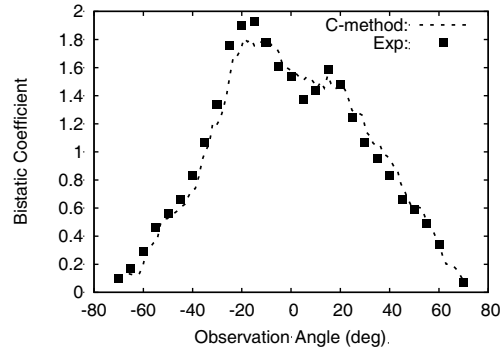

Figure 10. Normalized bistatic scattering cross section versus observation angle in the incidence plane. Very rough surface. Polarization (*hv*).

for the first surface realization, the relative errors are small and vary within the range $[10^{-8}, 10^{-3}]$. With early shift, the computational time of parallel multi-shift and AED decreases by approximately 16%.

It is noteworthy that the studied surface exhibits backscattering enhancement in both co-polarized and cross-polarized returns. We recall that the truncation order is a numerical parameter of the C-method. The Mth-order truncation removes the evanescent waves in the field components as shown in Equations (15) and (16). The evanescent waves contribute to the near field and take part in the couplings between propagating waves. For the roughness parameters under consideration, when we use the C-method with $M = 28$, the proportion of the evanescent wave functions is sufficient to describe the electromagnetic couplings and to analyze the backscattering enhancement. The comparison with experimental data coming from [21] is very good and the backscattering peaks coincide well. The comparison is also conclusive for (*hh*)- and (*vh*)-results.

5. CONCLUSION

In this paper, we have proposed a parallel QR algorithm that is specifically designed for the C-method applied to the scattering of electromagnetic plane wave by a rough surface. From numerical experiments, we have observed that some eigenvalues can be approximated efficiently by the propagation coefficients of elementary plane waves with respect to the Oz axis. We design the "early shift" algorithm to take advantage of this property. We plug this "early shift" method, together with Wilkinson shift and exceptional shift, into the new parallel QR algorithm. This algorithm uses multiple chains of tightly coupled bulges chasing technique to parallelize the conventional bulge chasing and the aggressive early deflation technique to detect deflation quickly. We apply this specifically designed parallel QR algorithm to the matrix characterizing the scattering problem. We also compare the computation time with that of the sequential code. The proposed algorithm leads to a substantial reduction in the computational time.

This parallel QR algorithm can be used for analyzing crossed gratings or two-dimensional random gratings. In this paper, we have analyzed two-dimensional random rough surfaces by a grating approach. Comparisons with experimental data are conclusive in both co- and cross-polarized components and validate the grating approach for moderately rough and very rough surfaces under consideration.

REFERENCES

1. Chandezon, J., D. Maystre, and G. Raoult, "A new theoretical method for diffraction gratings and its numerical application," *J. Optics (Paris)*, Vol. 11, 235–241, 1980.
2. Li, L. and J. Chandezon, "Improvement of the coordinate transformation method for surface-relief gratings with sharp edges," *J. Opt. Soc. Am. A*, 2247–2255, 1996.

3. Granet, G., "Analysis of diffraction by surface-relief crossed gratings with use of the Chandezon method: Application to multilayer crossed gratings," *J. Opt. Soc. Am. A*, Vol. 15, 1121–1131, 1998.

4. Aït Braham, K., R. Dusséaux, and G. Granet, "Scattering of electromagnetic waves from two-dimensional perfectly conducting random rough surfaces — Study with the curvilinear coordinate method," *Waves Random Complex Media*, Vol. 18, 255–274, 2008.

5. Dusséaux, R., K. Aït Braham, and G. Granet, "Implementation and validation of the curvilinear coordinate method for the scattering of electromagnetic waves from two-dimensional dielectric random rough surfaces," *Waves Random Complex Media*, Vol. 18, 551–570, 2008.

6. Dusséaux, R., E. Vannier, O. Taconet, and G. Granet, "Study of backscatter signature for seedbed surface evolution under rainfall — Influence of radar precision," *Progress In Electromagnetics Research*, Vol. 125, 415–437, 2012.

7. Elfouhaily, T. M. and C. A. Guérin, "A critical survey of approximate scattering wave theories from random rough surfaces," *Waves in Random and Complex Media*, Vol. 14, R1–10, 2004.

8. Bai, Z., J. W. Demmel, J. J. Dongarra, A. Ruhe, and H. van Der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems. Software, Environments, and Tools*, SIAM, 2000.

9. Golub, G. H. and F. Uhlig, "The QR algorithm: 50 years later its genesis by John Francis and Vera Kublanovskayaand subsequent developments," *IMA J. Numer. Anal.*, Vol. 29, 467–485, 2009.

10. Golub, G. H. and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.

11. Braman, K., R. Byers, and R. Mathias, "The multi-shift QR algorithm. Part I: Maintaining well-focused shifts and level 3 performance," *SIAM J. Matrix Anal. Appl.*, Vol. 23, 929–947, 2002.

12. Granat, R., B. Kagström, and D. Kressner, "A novel parallel QR algorithm for hybrid distributed memory HPC systems," *SIAM J. Sci. Comput.*, Vol. 32, 2345–2378, 2010.

13. Braman, K., R. Byers, and R. Mathias, "The multi-shift QR algorithm. Part II: Aggressive early deflation," *SIAM J. Matrix Anal. Appl.*, Vol. 23, 948–972, 2002.

14. "MPI — Messaging passing interface," See http://www.mcs.anl.gov/research/projects/mpi/.

15. "OpenMP — Open multi-processing," See http://openmp.org/wp/.

16. "BLAS — Basic linear algebra subprograms," See http://www.netlib.org/blas/.

17. "ScaLAPACK — Scalable linear algebra package," See http://www.netlib.org/scalapack/.

18. Kong, J. A., K. H. Ding, and C. O. Ao, *Scattering of Electromagnetic Waves — Numerical Simulations*, Wiley-Interscience, New York, 2001.

19. Afifi, S. and R. Dusséaux, "Scattering by anisotropic rough layered 2D interfaces," *IEEE Trans. Antennas Propag.*, Vol. 60, 5315–5328, 2012.

20. Berginc, G., "Small-slope approximation method: A further study of vector wave scattering from two-dimensional surfaces and comparison with experimental data," *Progress In Electromagnetics Research*, Vol. 37, 251–287, 2002.

21. Johnson, J. T., L. Tsang, R. T. Shin, K. Pak, C. H. Chan, A. Ishimaru, and Y. Kuga, "Backscattering enhancement of electromagnetic waves from two-dimensional perfectly conducting random rough surfaces: A comparison of Monte-Carlo simulations with experimental data," *IEEE Trans. Antennas Propag.*, Vol. 44, 748–756, 1996.