Mathematical and Computational Applications

MDPI

Article

Heavy Ball Restarted CMRH Methods for Linear Systems

Zhongming Teng [1] and Xuansheng Wang [2,*]

[1] College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou 350002, China; peter979@163.com

[2] School of Software Engineering, Shenzhen Institute of Information Technology, Shenzhen 518000, China

* Correspondence: 84780728@qq.com

Received: 1 February 2018; Accepted: 23 February 2018; Published: 25 February 2018

Abstract: The restarted CMRH method (changing minimal residual method based on the Hessenberg process) using fewer operations and storage is an alternative method to the restarted generalized minimal residual method (GMRES) method for linear systems. However, the traditional restarted CMRH method, which completely ignores the history information in the previous cycles, presents a slow speed of convergence. In this paper, we propose a heavy ball restarted CMRH method to remedy the slow convergence by bringing the previous approximation into the current search subspace. Numerical examples illustrate the effectiveness of the heavy ball restarted CMRH method.

Keywords: linear systems; Hessenberg; CMRH; GMRES; heavy ball methods

1. Introduction

In this paper, we are concerned with the CMRH method (changing minimal residual method based on the Hessenberg process) introduced in [1,2] for the solution of $n \times n$ linear system $Ax = b$. Given an initial approximation x_0, and letting the initial residual $r_0 = b - Ax_0$, the CMRH method is a Krylov subspace method based on the m-dimensional Krylov subspace

$$\mathcal{K}_m(A, r_0) = \text{span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\}. \quad (1)$$

It can also be considered as an alternative method of the generalized minimal residual method (GMRES) [3]. Nevertheless, they generate different basis vectors in different ways. The GMRES method uses the Arnoldi process to construct an orthonormal basis matrix of the m-dimensional Krylov subspace (1), while the CMRH method is based on the Hessenberg process [4], which requires half as much arithmetic work and less storage than the GMRES method. The corresponding convergence analysis of the CMRH method and its relation to the GMRES method can be found in [5,6]. There are a great deal of past and recent works and interests in developing the Hessenberg process and CMRH method for linear systems [7–13]. Specifically, in [7,8], Duminil presents an implementation for parallel architectures and an implementation of the left preconditioned CMRH method. A polynomial preconditioner and flexible right preconditioner for CMRH methods are considered in [9] and [10], respectively. In [12,13], the variants of the Hessenberg and CMRH methods are introduced for solving multi-shifted non-Hermitian linear systems.

Although the CMRH method is less expensive and needs less storage than the GMRES method per iteration, for large-scale linear systems, it is still very expensive for large m. In addition, the number of vectors requiring storage also increases as m increases. Hence, the method must be restarted. The restarted CMRH method denoted by CMRH(m) [1] is naturally developed to alleviate the possibly heavy memory burden and arithmetic operations cost. However, the price to pay for the restart is usually slower speed of the convergence. In the restarted GMRES methods, in order to

overcome the slow convergence, there are many other effectiveness restarting technologies [14–17] which are designed to improve the simplest version of the restarted GMRES methods. Nevertheless, the research on the restarted CMRH method is rather scarce. One of the reasons is because the basis vectors generated by the Hessenberg process are not orthonormal. This leads to the CMRH residual vector being not orthogonal to the subspace $\mathcal{K}_m(A, r_0)$ or $A \times \mathcal{K}_m(A, r_0)$. Thus, the whole augmented subspace containing the smaller Krylov subspace with Ritz vectors or harmonic Ritz vectors is not still a Krylov subspace. (See the subspace (2.4) of [17] for more details on this augmented subspace.) This means that the restarting strategy by including eigenvectors associated with the few smallest eigenvalues into the Krylov subspace [15–17] is not available in the restarted CMRH methods. In this paper, inspired by the locally optimal conjugate gradient (LOCG) methods for the eigenvalue problem [18–23] and the locally optimal and heavy ball GMRES methods for linear systems [14], we propose a heavy ball restarted CMRH method to keep the benefit and remedy the lost convergence speed of the traditional restarted CMRH method. For traditional restarted CMRH method (i.e., CMRH(m)), each CMRH cycle builds a Krylov subspace for computing the approximate solution of the cycle, which is used as the initial guess for the next cycle. As soon as the approximation is computed, the built Krylov subspace is thrown away. Nevertheless, in the heavy ball restarted CMRH method proposed in this paper, for salvaging the loss of the previous search space, we take the previous approximation into the current search to bring sufficient history information of the previous Krylov subspace.

The rest of this paper is organized as follows. In Section 2, we briefly review the Hessenberg process and the restarted CMRH method, then introduce the algorithmic framework and implementation details of the heavy ball restarted CMRH method. Numerical examples are given in Section 3 to show the convergence behavior of the improved method. Finally, we give our concluding remarks in Section 4.

Notation. Throughout this paper, $\mathbb{R}^{n\times m}$ is the set of all $n \times m$ real matrices, $\mathbb{R}^n = \mathbb{R}^{n\times 1}$ and $\mathbb{R} = \mathbb{R}^1$. I_n is the $n \times n$ identity matrix, and $e_j^{(n)}$ is its jth column. The superscript "$.^{\mathrm{T}}$" takes the transpose only of a matrix or vector. For a vector u and a matrix A, $u(j)$ is u's jth entry, $u(i:j)$ is the vector of components $u(i), \ldots, u(j)$, and $A(i,j)$ is A's (i,j)th entry. Notations $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ are the 1-norm, 2-norm, and ∞-norm of a vector or matrix, respectively.

2. The Heavy Ball Restarted CMRH Method

2.1. The Hessenberg Process with Pivoting

The CMRH method [1] for the linear systems $Ax = b$ is based on the Hessenberg process with pivoting as given in Algorithm 1 to construct a basis of the Krylov subspace $\mathcal{K}_m(A, r_0)$. Given an initial guess $x_0 \in \mathbb{R}^n$, the recursively computed basis matrix $L_m = [\ell_1, \ldots, \ell_m] \in \mathbb{R}^{n\times m}$ and the upper Hessenberg matrix $H_m \in \mathbb{R}^{m\times m}$ by Algorithm 1 satisfy

$$AL_m = L_{m+1}\bar{H}_m = L_m H_m + h_{m+1,m}\ell_{m+1}(e_m^{(m)})^{\mathrm{T}},$$

where $\bar{H}_m = \begin{bmatrix} H_m \\ h_{m+1,m}(e_m^{(m)})^{\mathrm{T}} \end{bmatrix}$ and PL_m is a unit lower trapezoidal matrix with $P^{\mathrm{T}} = [e_{p(1)}^{(n)}, \ldots, e_{p(n)}^{(n)}]$. In particular, the initial residual vector $r_0 = \beta_0 \times \ell_1 = \beta_0 \times L_{m+1}e_1^{(m+1)}$, where $\beta_0 = r_0(i_0)$ as defined in Line 2 of Algorithm 1.

Algorithm 1 Hessenberg process with pivoting

1: Choose an initial guess $x_0 \in \mathbb{R}^n$ and an integer $m \geq 1$.
2: Let $r_0 = b - Ax_0$. Set $p = [1, \ldots, n]$ and determine i_0 such that $|r_0(i_0)| = \|r_0\|_\infty$, $\beta_0 = r_0(i_0)$, $\ell_1 = r_0/\beta_0$, and $p(1) \leftrightarrow p(i_0)$ where "$\leftrightarrow$" means swap contents.
3: **for** $j = 1, 2, \ldots, m$ **do**
4: $u = A\ell_j$.
5: **for** $i = 1, 2, \ldots, j$ **do**
6: $h_{i,j} = u(p(i))$, $u = u - h_{i,j}\ell_i$.
7: **end for**
8: **if** ($j < n$ and $u \neq 0$) **then**
9: Determine $j_0 \in \{j+1, \ldots, n\}$ such that $|u(p(j_0))| = \|u\,(p(j+1:n))\,\|_\infty$.
10: $h_{j+1,j} = u(p(j_0))$, $\ell_{j+1} = u/h_{j+1,j}$ and $p(j+1) \leftrightarrow p(j_0)$.
11: **else**
12: $h_{j+1,j} = 0$. Stop.
13: **end if**
14: **end for**

The CMRH approximate solution x_m is obtained as $x_0 + z_m$, where z_m solves

$$\min_{z \in \mathcal{K}_m(A, r_0)} \|L_{m+1}^\dagger\,(b - A(x_0 + z))\,\|_2.$$

Here, $L_{m+1}^\dagger$ is the pseudo-inverse of L_{m+1}. In fact, any left inverse of L_{m+1} will work, but we use the pseudo-inverse here for simplicity. We can state equivalently that $x_m = x_0 + L_m y_m$, where $y_m = \arg\min_{y \in \mathbb{R}^m} \|\beta_0 e_1^{(m+1)} - \bar{H}_m y\|_2$. Like the quasi-minimal residual (QMR) method [24], the CMRH method is a quasi-residual minimization method.

2.2. The Heavy Ball Restarted CMRH (HBCMRH) Method

To alleviate a heavy demand on memory, the restarted CMRH method (or CMRH(m) for short) is implemented by fixing m and repeating the m-step CMRH method with the current initial guess $x_0^{(k)}$ being the previous approximation solution $x_m^{(k-1)}$, where the kth and $(k-1)$th CMRH cycles are indexed by the superscript "(k)" and "$(k-1)$", respectively. By limiting the number m in the Hessenberg process, although CMRH(m) successfully remedies possibly heavy memory burden and computational cost, at the same time it sometimes converges slowly. One of the reasons is that all the Krylov subspaces built in the previous cycles are completely ignored. Motivated by the locally optimal and heavy ball CMRES method [14] which is proposed by including the approximation before the last to bring in more information in the previous cycles, we naturally develop a heavy ball restarted CMRH method denoted by HBCMRH(m) to make up the loss of previous search spaces. For each cycle, HBCMRH(m) starts an approximation $x_0^{(k)}$ being the previous cycle approximation solution $x_m^{(k-1)}$ and seeks the next approximation solution $x_m^{(k)} = x_0^{(k)} + z_m^{(k)}$, where $z_m^{(k)} \in \mathcal{K}_m(A, r_0^{(k)}) + \text{span}\{x_0^{(k)} - x_0^{(k-1)}\}$. The actual implementation is as follows.

Let the vector $d = x_0^{(k)} - x_0^{(k-1)}$. Recall that we have L_m, $\bar{H}_m$, and p by the Hessenberg process satisfying $AL_m = L_{m+1}\bar{H}_m$, where PL_m is unit lower trapezoidal with $P^{\mathrm{T}} = [e_{p(1)}^{(n)}, \ldots, e_{p(n)}^{(n)}]$. Now we let d subtract multiples $\ell_1, \ldots, \ell_m$ to annihilate the m components $p(1), \ldots, p(m)$ of the vector d to obtain a new vector d_{new} as in Lines 8–10 in Algorithm 2. Suppose $d_{\text{new}} \neq 0$. We determine $j_0 \in \{m+1, \ldots, n\}$ such that $|d_{\text{new}}(p(j_0))| = \|d_{\text{new}}(p(m+1:n))\|_\infty$. Let

$$\hat{\ell}_{m+1} = d_{\text{new}}/d_{\text{new}}(p(j_0)), \quad \widehat{L}_{m+1} = [L_m, \hat{\ell}_{m+1}], \quad p(m+1) \leftrightarrow p(j_0).$$

It is clear that $\widehat{L}_{m+1}$ is a basis matrix of subspace $\mathcal{K}_m(A, r_0^{(k)}) + \text{span}\{d\}$.

Next, we run the similar procedure to annihilate the $m+1$ components $p(1), \ldots, p(m+1)$ of the vector $A\hat{\ell}_{m+1}$, and then obtain $\tilde{\ell}_{m+2}$ and $\bar{H}_{m+1} \in \mathbb{R}^{(m+2)\times(m+1)}$ as Lines 14–20 in Algorithm 2. Let $\widetilde{L}_{m+2} = [L_{m+1}, \tilde{\ell}_{m+2}]$. We have the relationship

$$A\widehat{L}_{m+1} = \widetilde{L}_{m+2}\bar{H}_{m+1}. \tag{2}$$

At the same time, $P\widetilde{L}_{m+2}$ keeps the structure of unit lower trapezoid.

The new approximation solution of HBCMRH(m) is $x_m^{(k)} = x_0^{(k)} + z_m^{(k)}$, where $z_m^{(k)}$ is computed by solving

$$\min_{z \in \mathcal{K}_m(A, r_0^{(k)}) + \mathrm{span}\{d\}} \left\| \widetilde{L}_{m+2}^{\dagger} \left(b - A(x_0^{(k)} + z) \right) \right\|_2.$$

Here, $\widetilde{L}_{m+2}^{\dagger}$ is the pseudo-inverse of $\widetilde{L}_{m+2}$. For any $z \in \mathcal{K}_m(A, r_0^{(k)}) + \mathrm{span}\{d\}$ can be expressed by $z = \widehat{L}_{m+1}y$ for some $y \in \mathbb{R}^{m+1}$. It is followed by (2) that

$$\begin{aligned} b - Ax_m^{(k)} &= b - Ax_0^{(k)} - A\widehat{L}_{m+1}y = r_0^{(k)} - \widetilde{L}_{m+2}\bar{H}_{m+1}y \\ &= \beta_0 \widetilde{L}_{m+2} e_1^{(m+2)} - \widetilde{L}_{m+2}\bar{H}_{m+1}y \\ &= \widetilde{L}_{m+2}\left(\beta_0 e_1^{(m+2)} - \bar{H}_{m+1}y \right). \end{aligned}$$

Thus, the new HBCMRH(m) approximation solution $x_m^{(k)}$ is obtained as $x_0^{(k)} + \widehat{L}_{m+1}y_m^{(k)}$, where $y_m^{(k)} = \arg\min_{y \in \mathbb{R}^{m+1}} \|\beta_0 e_1^{(m+2)} - \bar{H}_{m+1}y\|_2$. We summarize the HBCMRH(m) method mentioned in this subsection in Algorithm 2. A few remarks regarding Algorithm 2 are in order:

1. In Algorithm 2, we only simply consider the case $d \neq 0$ and $u \neq 0$ in Line 11 and 18, respectively. In fact, in the case $d \neq 0$ and $u = 0$, by a simple modification of the above process, the new approximation $x_m^{(k)} = x_0^{(k)} + \widehat{L}_{m+1}y_m^{(k)}$ with $y_m^{(k)} = \arg\min_{y \in \mathbb{R}^{m+1}} \|\beta_0 e_1^{(m+1)} - H_{m+1}y\|_2$ where $H_{m+1} \in \mathbb{R}^{(m+1)\times(m+1)}$ is obtained by deleting the last row of $\bar{H}_{m+1}$. Similarly, if $d = 0$, then $y_m^{(k)}$ is the optimal argument of $\min_{y \in \mathbb{R}^m} \|\beta_0 e_1^{(m+1)} - \bar{H}_m y\|_2$.
2. In comparison with the CMRH(m) method, the HBCMRH(m) method only requires one extra matrix-vector multiplication with A, but it represents a significant improvement in the speed of convergence, as shown in our numerical examples.

Algorithm 2 Heavy ball restarted CMRH method (HBCMRH(m))

Choose an initial guess $x_0^{(1)} \in \mathbb{R}^n$ and an integer $m \geq 1$.
for $k = 1, 2, \ldots$ until convergence **do**
  call Algorithm 1 to get p, L_{m+1}, and $\bar{H}_m$.
  if $k = 1$ **then**
    Compute $x_m^{(k)} = x_0^{(k)} + L_m y_m^{(k)}$, where $y_m^{(k)} = \arg\min_{y \in \mathbb{R}^m} \|\beta_0 e_1^{(m+1)} - \bar{H}_m y\|_2$, and $x_0^{(k+1)} = x_m^{(k)}$.
  else
    $d = x_0^{(k)} - x_0^{(k-1)}$.
    for $i = 1, 2, \ldots, m$ **do**
      $d = d - d(p(i))\ell_i$.
    end for
    if $d \neq 0$ **then**
      Determine $j_0 \in \{m+1, \ldots, n\}$ such that $|d(p(j_0))| = \|d\,(p(m+1:n))\,\|_\infty$, $\hat{\ell}_{m+1} = d/d(p(j_0))$ and $p(m+1) \leftrightarrow p(j_0)$.
    end if
    Compute $u = A\hat{\ell}_{m+1}$.
    for $i = 1, 2, \ldots, (m+1)$ **do**
      $h_{i,m+1} = u(p(i))$, $u = u - h_{i,m+1}\ell_i$.
    end for
    if $u \neq 0$ **then**
      Determine $j_0 \in \{m+2, \ldots, n\}$ such that $|u(p(j_0))| = \|u\,(p(m+2:n))\,\|_\infty$, $h_{m+2,m+1} = u(p(j_0))$, $\tilde{\ell}_{m+2} = u/h_{m+2,m+1}$ and $p(m+2) \leftrightarrow p(j_0)$.
    end if
    Let $\widehat{L}_{m+1} = [L_m, \hat{\ell}_{m+1}]$. Compute $x_m^{(k)} = x_0^{(k)} + \widehat{L}_{m+1} y_m^{(k)}$ where $y_m^{(k)} = \arg\min_{y \in \mathbb{R}^{m+1}} \|\beta_0 e_1^{(m+2)} - \bar{H}_{m+1} y\|_2$, and $x_0^{(k+1)} = x_m^{(k)}$.
  end if
end for

3. Numerical Examples

In this section, we present some numerical examples to illustrate the convergence behavior of the HBCMRH(m) method (i.e., Algorithm 2) with the initial vector $x_0 = [0, \ldots, 0]^{\mathrm{T}}$ and $m = 30$. In demonstrating the quality of computed approximations, we monitor the normalized residual norms

$$\frac{\|b - Ax\|_2}{\|A\|_1 \|x\|_2 + \|b\|_2}$$

against the number of cycles. All our experiments were performed on a Windows 10 (64 bit) PC-Intel(R) Core(TM) i7-6700 CPU 3.40 GHz, 16 GB of RAM using MATLAB version 8.5 (R2015a) with machine epsilon 10^{-16} in double precision floating point arithmetic.

Example 1. We first consider a $n \times n$ dense matrix

$$A = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 & 1 \\ \alpha_1 & 1 & 1 & \ldots & 1 & 1 \\ \alpha_1 & \alpha_2 & 1 & \ldots & 1 & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \alpha_1 & \alpha_2 & \alpha_3 & \ldots & \alpha_{n-1} & 1 \end{bmatrix}$$

appearing in [1] and [25] with $n = 100$, $a_i = 1 + i \times \varepsilon$, and $\varepsilon = 10^{-2}$. The right hand side $b = \mathsf{rand}(n, 1)$ where rand is a MATLAB built-in function. In order to be fair in comparing algorithms, we tested the HBCMRH(m) method and the CMRH($m+1$) method. Recalling the remark of Algorithm 2, we know the CMRH($m+1$) method requires the same matrix-vector multiplications as HBCMRH(m), and it presents a better convergence behavior than CMRH(m). The normalized residual norms against

the number of cycles of these two methods are collected in Figure 1, which clearly shows that the HBCMRH(m) method converges much faster than CMRH($m+1$). In fact, as shown in Figure 1, to reach about 10^{-8} in normalized residual norms on this example, the HBCMRH(m) method takes 34 cycles, while the CMRH($m+1$) method is seen to need much more cycles to get there.

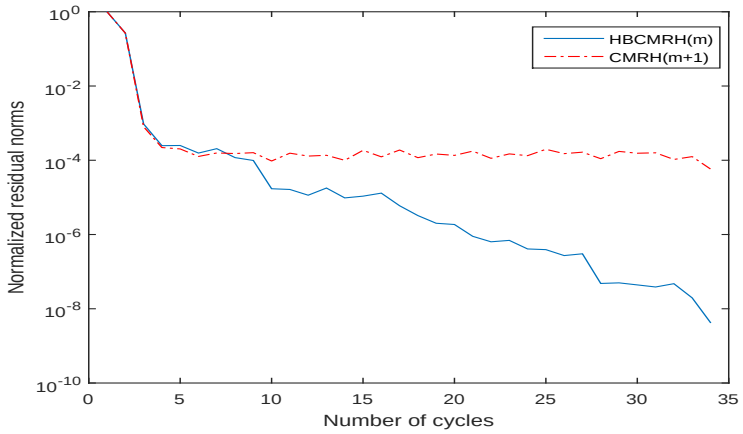

Figure 1. Convergence behavior of the heavy ball restarted changing minimal residual method (HBCMRH)(m) and CMRH($m+1$) method.

Example 2. In this example, we consider a sparse matrix A and right hand side b which are extracted from `raefsky1` taken from the University of Florida sparse matrix collection [26]. In such a problem, A is not symmetric with order $n = 3242$ and contains 293,409 nonzero entries. Similarly, we compare the normalized residual norms of the HBCMRH(m) method with the CMRH($m+1$) method in Figure 2. Similar comments to the ones we made at the end of Example 1 are valid here as well.

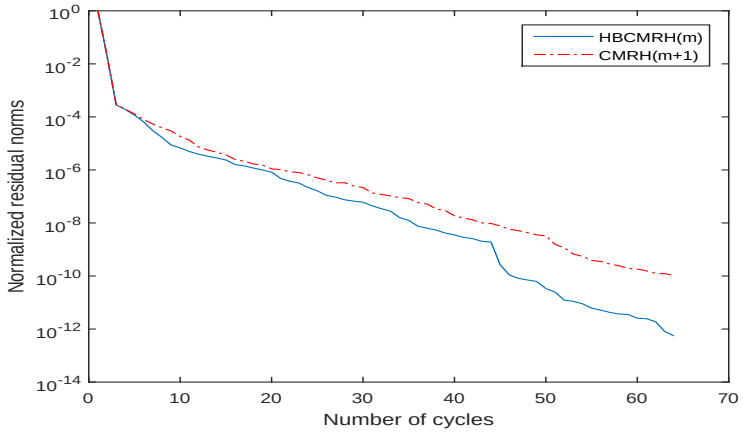

Figure 2. Convergence behavior of the HBCMRH(m) and CMRH($m+1$) method.

Example 3. In this example, we use the symmetric Hankle matrix appearing in [8] with elements

$$A(i,j) = \frac{0.5}{n-i-j+1.5}$$

and $n = 1000$. Let the right rand side $b = A \times \text{ones}(n,1)$, where ones is a MATLAB built-in function. We compare the HBCMRH(m) method with the heavy ball restarted GMRES method denoted by HBGMRES(m) and compute the associated normalized residual norms in Figure 3. As shown in

Figure 3, in such a case, the HBCMRH(m) and HBGMRES(m) method are competitive in the number of cycles, but less computation cost and storage requirement are needed in the HBCMRH(m) method.

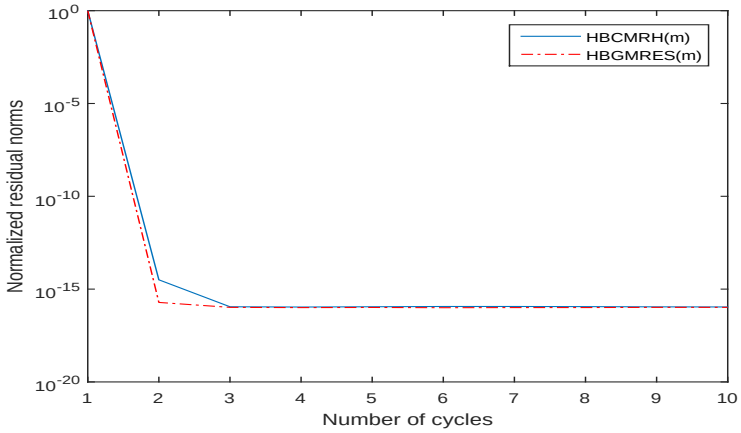

Figure 3. Convergence behavior of the HBCMRH(m) and heavy ball restarted generalized minimal residual (HBGMRES)(m) method.

4. Conclusions

In this paper, we proposed a heavy ball restarted CMRH method (HBCMRH(m) for short; i.e., Algorithm 2) for a linear system $Ax = b$. Compared to the traditional restarted CMRH method (i.e., CMRH(m)), $x_0^{(k-1)}$ is built in the search space of the HBCMRH(m) method to compute the next approximation $x_m^{(k)}$ for alleviating the loss of previous Krylov subspace. Therefore, one more matrix-vector multiplication of A is required in the HBCMRH(m) method. However, as shown in our numerical examples, HBCMRH(m) presents a better convergence behavior than CMRH($m+1$).

We have focused on the heavy ball restarted CMRH method. In fact, we can easily give the locally optimal restarted CMRH method as the locally optimal restarted GMRES method. We omit the details. In addition, while the HBCMRH(m) method has been developed for real linear systems, the algorithm can be rewritten to work for complex linear systems. This is done by simply replacing all $\mathbb{R}$ by $\mathbb{C}$ and each matrix/vector transpose by complex conjugate transpose.

Acknowledgments: The authors are grateful to the anonymous referees for their careful reading, useful comments, and suggestions for improving the presentation of this paper. The work of the first author is supported in part by National Natural Science Foundation of China NSFC-11601081 and the research fund for distinguished young scholars of Fujian Agriculture and Forestry University No. xjq201727. The work of the second author is supported in part by National Natural Science Foundation of China Grant NSFC-11601347, and the Shenzhen Infrastructure Project No. JCYJ20170306095959113.

Author Contributions: Both authors contributed equally to this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sadok, H. CMRH: A new method for solving nonsymmetric linear systems based on the Hessenberg reduction algorithm. *Numer. Algorithms* **1999**, *20*, 303–321.
2. Heyouni, M.; Sadok, H. A new implementation of the CMRH method for solving dense linear systems. *J. Comput. Appl. Math.* **2008**, *213*, 387–399.
3. Saad, Y.; Schultz, M.H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.* **1986**, *7*, 856–869.
4. Heyouni, M.; Sadok, H. On a variable smoothing procedure for Krylov subspace methods. *Linear Algebra Appl.* **1998**, *268*, 131–149.
5. Sadok, H.; Szyld, D.B. A new look at CMRH and its relation to GMRES. *BIT Numer. Math.* **2012**, *52*, 485–501.

6. Tebbens, J.D.; Meurant, G. On the convergence of Q-OR and Q-MR Krylov methods for solving nonsymmetric linear systems. *Bit Numer. Math.* **2016**, *56*, 77–97.
7. Duminil, S. A parallel implementation of the CMRH method for dense linear systems. *Numer. Algorithms* **2013**, *63*, 127–142.
8. Duminil, S.; Heyouni, M.; Marion, P.; Sadok, H. Algorithms for the CMRH method for dense linear systems. *Numer. Algorithms* **2016**, *71*, 383–394.
9. Lai, J.; Lu, L.; Xu, S. A polynomial preconditioner for the CMRH algorithm. *Math. Probl. Eng.* **2011**, *2011*, doi:10.1155/2011/545470.
10. Zhang, K.; Gu, C. A flexible CMRH algorithm for nonsymmetric linear systems. *J. Appl. Math. Comput.* **2014**, *45*, 43–61.
11. Alia, A.; Sadok, H.; Souli, M. CMRH method as iterative solver for boundary element acoustic systems. *Eng. Anal. Bound. Elem.* **2012**, *36*, 346–350.
12. Gu, X.M.; Huang, T.Z.; Carpentieri, B.; ImakuraWen, A.; Zhang, K.; Du, L. Variants of the CMRH method for solving multi-shifted non-Hermitian linear systems. Available online: https://arxiv.org/abs/1611.00288 (accessed on 24 February 2018).
13. Gu, X.M.; Huang, T.Z.; Yin, G.; Carpentieri, B.; Wen, C.; Du, L. Restarted Hessenberg method for solving shifted nonsymmetric linear systems. *J. Comput. Appl. Math.* **2018**, *331*, 166–177.
14. Imakura, A.; Li, R.C.; Zhang, S.L. Locally optimal and heavy ball GMRES methods. *Jap. J. Ind. Appl. Math.* **2016**, *33*, 471–499.
15. Morgan, R.B. Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations. *SIAM J. Matrix Anal. Appl.* **2000**, *21*, 1112–1135.
16. Morgan, R.B. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.* **1995**, *16*, 1154–1171.
17. Morgan, R.B. GMRES with deflated restarting. *SIAM J. Matrix Anal. Appl.* **2002**, *24*, 20–37.
18. Bai, Z.; Li, R.C. Minimization Principle for Linear Response Eigenvalue Problem, I: Theory. *SIAM J. Matrix Anal. Appl.* **2012**, *33*, 1075–1100.
19. Bai, Z.; Li, R.C. Minimization Principle for Linear Response Eigenvalue Problem, II: Computation. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 392–416.
20. Bai, Z.; Li, R.C. Minimization principles and computation for the generalized linear response eigenvalue problem. *BIT Numer. Math.* **2014**, *54*, 31–54.
21. Bai, Z.; Li, R.C.; Lin, W.W. Linear response eigenvalue problem solved by extended locally optimal preconditioned conjugate gradient methods. *Sci. China Math.* **2016**, *59*, 1–18.
22. Knyazev, A.V. Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method. *SIAM J. Sci. Comput.* **2001**, *23*, 517–541.
23. Knyazev, A.V.; Argentati, M.E.; Lashuk, I.; Ovtchinnikov, E.E. Block locally optimal preconditioned eigenvalue xolvers (blopex) in hypre and petsc. *SIAM J. Sci. Comput.* **2007**, *29*, 2224–2239.
24. Freund, R.W.; Nachtigal, N.M. QMR: A quasi-minimal residual method for non-Hermitian linear systems. *Numer. Math.* **1991**, *60*, 315–339.
25. Gregory, R.T.; Karney, D.L. *A Collection of Matrices for Testing Computational Algorithms*; Wiley: New York, NY, USA, 1969.
26. Davis, T.; Hu, Y. The University of Florida Sparse Matrix Collection. *ACM Trans. Math. Softw.* **2011**, *38*, doi:10.1145/2049662.2049663.