ISSN 1410 - 5888

SIGMA

JURNAL SAINS DAN TEKNOLOGI

Vol. 12, No. 2, Juli 2009

Jurnal sains dan teknologi **SIGMA** diterbitkan dua kali setahun, yaitu pada bulan Januari dan Juli, sebagai wahana komunikasi ilmiah di bidang sains dan teknologi serta lintas-ilmu yang terkait. Penyunting menerima karangan ilmiah dalam bidang-bidang tersebut berupa hasil penelitian, survai, atau telaah pustaka, yang belum pernah dipublikasikan dalam terbitan lain. Penyunting berhak untuk menyunting karangan yang akan dimuat tanpa mengubah isinya. Ketentuan tentang penulisan karangan dapat dibaca pada bagian dalam sampul belakang jurnal ini atau di http:/www.usd.ac.id/06/jurnal/sigma.htm.

DEWAN PENYUNTING

Alamat Penyunting:
Fakultas Sains dan Teknologi Universitas Sanata Dharma
Kampus III: Paingan, Maguwoharjo, Depok, Sleman.
Teromol Pos 29, Yogyakarta 55002
Telpon: (0274) 883968, 883037. Fax: (0274) 886529
E-mail: sigma@staff.usd.ac.id

Penerbit:
Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM)
Universitas Sanata Dharma
Kampus II: Jl. Affandi, Mrican
Teromol Pos 29, Yogyakarta 55002
Telpon: (0274) 513301, 515352. Fax: (0274) 562383
E-mail: lemlit@staff.usd.ac.id

SIGMA
JURNAL SAINS DAN TEKNOLOGI

Vol. 12, No. 2, Juli 2009 ISSN: 1410-5888

DAFTAR ISI

SIGMA, Vol. 12, No. 2, Juli 2009: 117-126
ISSN: 1410-5888

EXPERIMENTS ON GRAM-SCHMIDT PROCESS AND GRAM-SCHMIDT PROCESS WITH REORTHOGONALISATION

Sudi Mungkasi and Lusia Krismiyati Budiasih
Department of Mathematics, Science and Technology Faculty, Sanata Dharma University, Mrican, Tromol Pos 29, Yogyakarta 55002. E-mail address: sudi@staff.usd.ac.id

Abstract

This paper discusses the orthogonalisation process in Gram-Schmidt algorithms. Four variants of Gram-Schmidt process are presented, and the relation between matrix size and running time for computation is also discussed. Loss of orthogonality of the computed vectors in Gram-Schmidt process can be reduced to be close to the machine precision level by reorthogonalisation. Theoretically, the loss of orthogonality is bounded, and it is true that reorthogonalisation in Gram-Schmidt process works well when the computation is not overflow. However, when reorthogonalisation is applied, the backward error is becoming larger.

Keywords: *Gram-Schmidt orthogonalisation, QR factorisation, backward error*

1. Introduction

Let $\mathbf{A} = (\mathbf{a}_1, \mathrm{K}\ , \mathbf{a}_n)$ be a real $m \times n$ matrix ($m \geq n$) with full column rank. The Gram-Schmidt orthogonalisation process (Hogben, 2007: 5-8, 5-9) produces an orthogonal basis $\mathbf{Q} = (\mathbf{q}_1, \mathrm{K}\ , \mathbf{q}_n)$ of span($\mathbf{A}$) such that $\mathbf{A} = \mathbf{QR}$, where $\mathbf{R}$ is an $n \times n$ upper triangular matrix.

Throughout the paper, unless stated explicitly as in the attachment, $\| \mathbf{A} \|$ denotes the norm of matrix $\mathbf{A}$, $\kappa(\mathbf{A})$ refers the condition number of the matrix, and $\| \mathbf{x} \|$ denotes the Euclidean norm of vector $\mathbf{x}$. The unit roundoff is denoted by u. The terms $c_k(m,n)$ are low degree polynomials in the problem dimensions m and n, where k is nonnegative integer; they are independent of the condition number $\kappa(\mathbf{A})$ and the unit roundoff u, but they depend on details of the computer arithmetic.

There are several variants of the algorithms for Gram-Schmidt orthogonalisation. The first is Classical Gram-Schmidt (CGS) which is known to be unstable, and the second is Modified Gram-Schmidt (MGS) which is a stable algorithm (Trefethen & Bau, 1997: 48-61). Björck (1967) shows that although CGS and MGS are mathematically equivalent, due to round-off errors the set of vectors produced by either of these two methods can be sometimes far from orthogonal. In general, the loss of orthogonality of vectors computed by the CGS process is faster than the loss of that of vectors computed by the MGS (Giraud *et al.*, 2003).

To improve the orthogonality of the vectors computed by Gram-Schmidt process, reorthogonalisation can be applied. Reorthogonalisation here means the orthogonalisation step is iterated twice or several times. In some applications it may be important to generate a set of basis vectors which its orthogonality is on the level of the machine precision. Hoffmann (1989) conjectures that two steps of reorthogonalisation are enough for obtaining orthogonality which is close to the machine precision. Then, Giraud *et al.* (2005) represent the theoretical foundation for this observation. For convenience in this paper, Classical Gram-Schmidt with reorthogonalisation and Modified Gram-Schmidt with reorthogonalisation are denoted by CGS2 and MGS2 respectively.

The organisation of this paper is as follows. Section 2 gives a brief overview of how Gram-Schmidt process works and then discusses the loss of orthogonality of the vectors because of rounding error in the calculation. The discussion is divided into two parts, the first part represents the theoretical background in rounding error in the discussed algorithms, and the second part presents result for application of the algorithm for several types of matrices. Section 3 contains the relation between matrix size and running time for orthogonalisation. Subse-

quently, section 4 concludes the discussion. Several Scilab codes used in the work of this paper are attached.

2. Loss of Orthogonality in CGS

How CGS and its reorthogonalisation work are described in this section (reorthogonalisation in MGS can be done in an analogous way). This section also gives an overview of the theory related to Gram-Schmidt orthogonalisation, and presents the output of four variants in Gram-Schmidt orthogonalisation when they are being implemented in Scilab-4.1.1.

Giraud *et al.* (2005) represent details and thorough analysis of the CGS and CGS2 version. In both algorithm, CGS and CGS2, the matrix with orthonormal column $\mathbf{Q} = (\mathbf{q}_1, \mathrm{K}, \mathbf{q}_n)$ is assumed to be constructed column-by-column so that for each index $j = 1, \mathrm{K}, n$ span $(\mathbf{q}_1, \mathrm{K}, \mathbf{q}_j)$ = span $(\mathbf{a}_1, \mathrm{K}, \mathbf{a}_j)$. The CGS algorithm starts with $\mathbf{q}_1 = \mathbf{a}_1 / \| \mathbf{a}_1 \|$ and for $j = 2, \mathrm{K}, n$ it successively produces

$$\mathbf{v}_j = [\mathbf{I} - \mathbf{Q}_{j-1}\mathbf{Q}_{j-1}^T]\mathbf{a}_j \tag{1}$$

where $\mathbf{q}_j = \mathbf{a}_j / \| \mathbf{a}_j \|$. The corresponding column $\mathbf{r}_j$ in the upper triangular factor $\mathbf{R} = (\mathbf{r}_1, \mathrm{K}, \mathbf{r}_n)$ is given as $\mathbf{r}_j = (\mathbf{Q}_{j-1}^T \mathbf{a}_j, \| \mathbf{v}_j \|)^T$. In CGS2, the vectors are computed as follows. Starts from $\mathbf{q}_1 = \mathbf{a}_1 / \| \mathbf{a}_1 \|$, and for $j = 2, \mathrm{K}, n$ two vectors $\mathbf{v}_j$ and $\mathbf{w}_j$ are computed successively

$$\mathbf{v}_j = [\mathbf{I} - \mathbf{Q}_{j-1}\mathbf{Q}_{j-1}^T]\mathbf{a}_j \tag{2}$$

$$\mathbf{w}_j = [\mathbf{I} - \mathbf{Q}_{j-1}\mathbf{Q}_{j-1}^T]\mathbf{v}_j \tag{3}$$

Normalising $\mathbf{w}_j$ will get $\mathbf{q}_j$, so $\mathbf{q}_j = \mathbf{w}_j / \| \mathbf{w}_j \|$. The corresponding column $\mathbf{r}_j$ in the upper triangular factor is given as

$$\mathrm{r}_j = (\mathrm{r}_{j-1}, 0)^T + \mathrm{s}_j = (\mathrm{Q}_{j-1}^T \mathrm{a}_j, 0)^T + (\mathrm{Q}_{j-1}^T \mathrm{v}_j, \| \mathrm{w}_j \|)^T \tag{4}$$

where $\mathbf{r}_{j-1} = \mathbf{Q}_{j-1}^T \mathbf{a}_j$, and $\mathbf{s}_j = (\mathbf{Q}_{j-1}^T \mathbf{v}_j, \| \mathbf{w}_j \|)^T$.
Thorough analysis of the MGS version is not presented in this paper. However, the algorithm and some output of both MGS and MGS2 will be discussed.

2.1. Rounding Error in CGS

According to Giraud *et al.* (2005), the loss of orthogonality for CGS and CGS2 are bounded as stated in the two theorems below.

Theorem 1
Assuming $c_2(m,n)u\kappa^2(\mathbf{A}) < 1$, the loss of orthogonality of the vector $\overline{\mathbf{Q}}$ computed by the CGS algorithm is bounded by

$$\| \mathbf{I} - \overline{\mathbf{Q}}^{\mathbf{T}} \overline{\mathbf{Q}} \| \leq \frac{c_3(m,n)u\kappa^2(\mathbf{A})}{1 - c_2(m,n)u\kappa^2(\mathbf{A})} \tag{5}$$

where $c_3(m,n) = O(mn^2)$.

Theorem 2
Assuming $c_4(m,n)u\kappa(\mathbf{A}) < 1$, the loss of orthogonality of the vectors $\overline{\mathbf{Q}}$ computed by the CGS2 algorithm can bounded as

$$\| \mathbf{I} - \overline{\mathbf{Q}}^{\mathbf{T}} \overline{\mathbf{Q}} \| \leq c_5(m,n)u \tag{6}$$

where $c_4(m,n) = O(m^2 n^3)$ and $c_5(m,n) = O(mn^{3/2})$.

From the two theorems above, the CGS2 seems more accurate in terms of orthogonality than CGS. In their paper, they illustrate their theoretical results using a 200×100 matrices $\mathbf{A}_k$ generated by computing $\mathbf{A}_k = \mathbf{U}\mathbf{\Sigma}_k\mathbf{V}^{\mathbf{T}}$, where $\mathbf{U}$ and $\mathbf{V}$ are randomly chosen orthonormal matri-

ces and $\mathbf{\Sigma}_k$, diagonal matrix, contains the singular values of $\mathbf{A}_k$ uniformly distributed between 1 and 10^{-k} for $k = 1,\mathrm{K}\ ,8$.

The CGS and MGS algorithms can be written as follows.

CGS:

```
for j=1:n;
  V(:,j)=A(:,j);
     for i=1:j-1;
        R(i,j)=Q(:,i)'*A(:,j);
        V(:,j)=V(:,j)-R(i,j)*Q(:,i);
     end;
   R(j,j)=norm(V(:,j));
   Q(:,j)=V(:,j)/R(j,j);
end;
```

MGS:

```
for j=1:n;
  V(:,j)=A(:,j);
     for i=1:j-1;
        R(i,j)=Q(:,i)'*V(:,j);
        V(:,j)=V(:,j)-R(i,j)*Q(:,i);
     end;
  R(j,j)=norm(V(:,j))
  Q(:,j)=V(:,j)/R(j,j)
end;
```

In terms of the orthogonality of matrix Q, MGS has a better result than CGS; but in fact, in several cases the orthogonality is still not a satisfactory. In so called an orthogonal basis problem which requires the individual column vectors of **Q** matrix, preserving the orthogonality of the column vectors in **Q** matrix is becoming an important issue (Hoffmann, 1989). More details of CGS and MGS can be found in Trefethen & Bau (1997: 48-61).

Giraud *et al.* (2003) have proved that one reorthogonalisation step is enough for preserving the orthogonality of computed vectors close to machine precision level. Reorthogonalisation means each orthogonalisation step is performed exactly twice. Note that it is different from CGS which is performed twice. Therefore, CGS2 and MGS2 algorithms can be written as follows.

CGS2:

```
for j=1:n;
  V(:,j)=A(:,j);
     for i=1:j-1;
        R(i,j)=Q(:,i)'*A(:,j);
        V(:,j)=V(:,j)-R(i,j)*Q(:,i);
     end;

     W(:,j)=V(:,j);
     for i=1:j-1;
        R(i,j)=Q(:,i)'*V(:,j);
        W(:,j)=W(:,j)-R(i,j)*Q(:,i);
     end;
  R(j,j)=norm(W(:,j));
  Q(:,j)=W(:,j)/R(j,j);
end;
```

MGS2:

```
for j=1:n;
  V(:,j)=A(:,j);
     for i=1:j-1;
        R(i,j)=Q(:,i)'*V(:,j);
        V(:,j)=V(:,j)-R(i,j)*Q(:,i);
     end;

     W(:,j)=V(:,j);
     for i=1:j-1;
        R(i,j)=Q(:,i)'*W(:,j);
        W(:,j)=W(:,j)-R(i,j)*Q(:,i);
     end;
  R(j,j)=norm(W(:,j))
  Q(:,j)=W(:,j)/R(j,j)
end
```

2.2. Simulation Result

Implementing those four algorithms in Scilab-4.1.1 using $2\times 2,\mathrm{K}\ ,100\times 100$ random matrices where all the entries are uniformly distributed in the interval (0,1) as the test matrices generates output as shown in Figure 1 to Figure 4 as follows, where the digit of accuracy is calculated using the formula

$$^{10}\log \| \mathbf{I} - \mathbf{Q}^{\mathrm{T}}\mathbf{Q} \| \qquad (7)$$

and the norm is the infinity norm. Since the machine precision is about 10^{-16}, the ideal case is that the digit accuracy is -16, which means $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} \approx \mathbf{I}$.

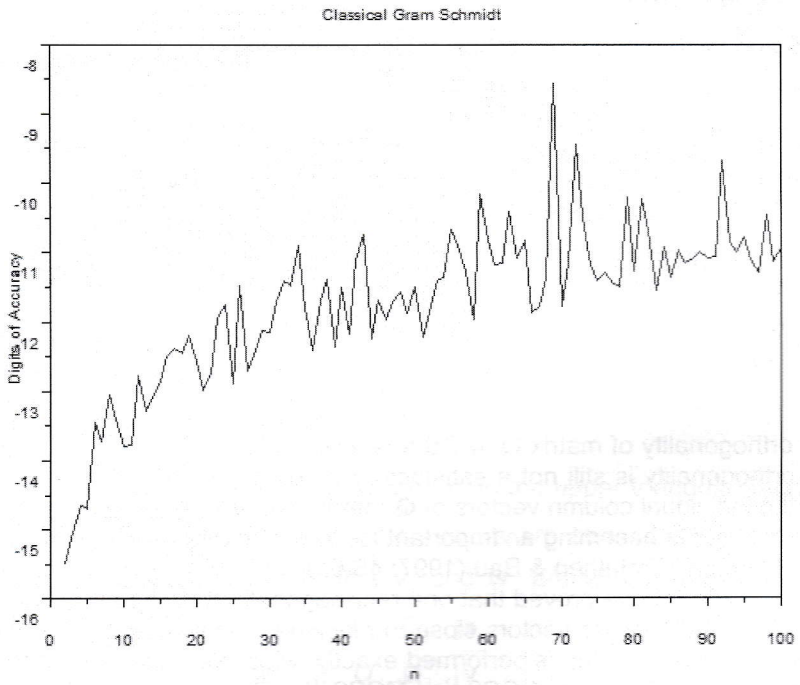

Figure 1. Digit of accuracy for CGS

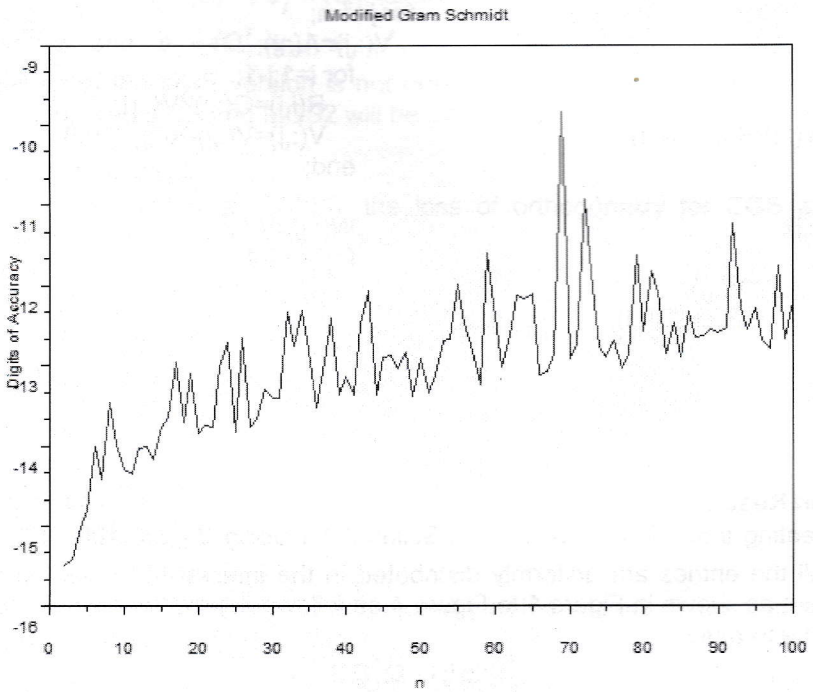

Figure 2. Digit of accuracy for MGS

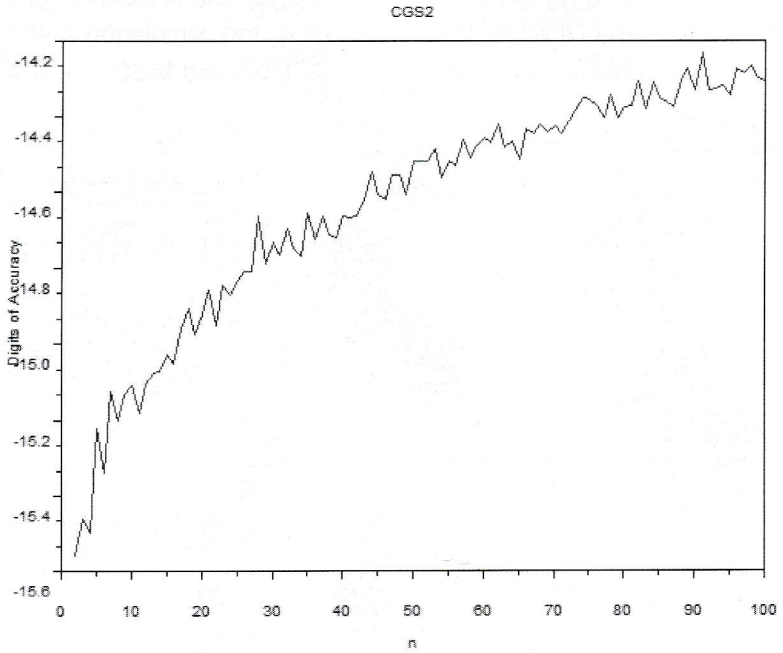

Figure 3. Digit of accuracy for CGS2

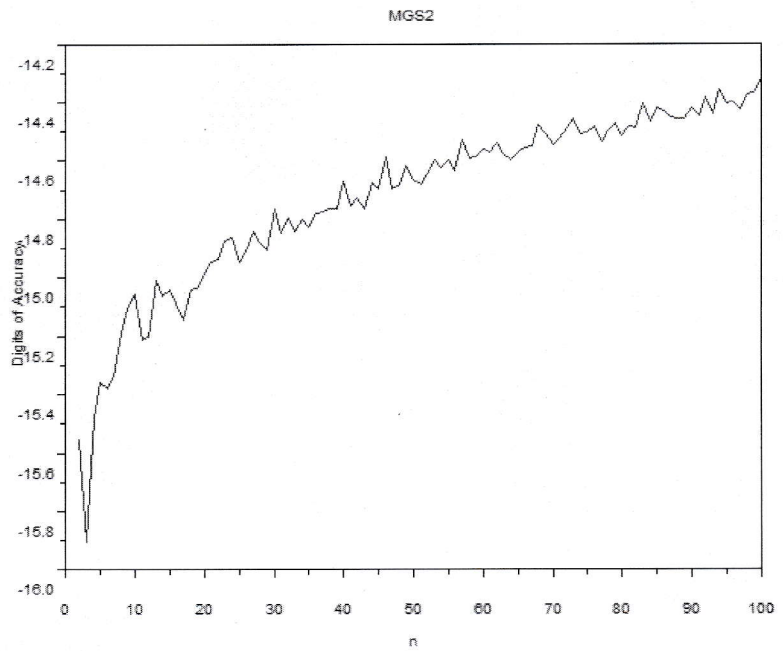

Figure 4. Digit of accuracy for MGS2

Digit of accuracy describes the level of orthogonality. Figure 1 to Figure 4 show that the level of orthogonality for CGS remains in the range between machine precision and 10^{-8}, and that for MGS remains between machine precision and 10^{-9}; while the level of orthogonality for CGS2 and MGS2 remain in the same range, that is between machine precision and $10^{-14.2}$. Therefore, reorthogonalisation in CGS and MGS is worthwhile to keep the orthogonality of the computed vectors in matrix **Q** to be close to the machine precision. In other words, the theoretical proof of this, which is presented in Giraud *et al.* (2005), is matching with the obtained result.

After finding the expected result in terms of orthogonality, the discussion will be continued to investigate the backward error of the related factorisation. The backward error here is computed using formula $|| \mathbf{A} - \mathbf{QR} || / || \mathbf{A} ||$. The norm used in this simulation is the infinity norm. Here the logarithmic scale is not used because for $n = 2$, CGS and MGS give $|| \mathbf{A} - \mathbf{QR} ||= 0$.

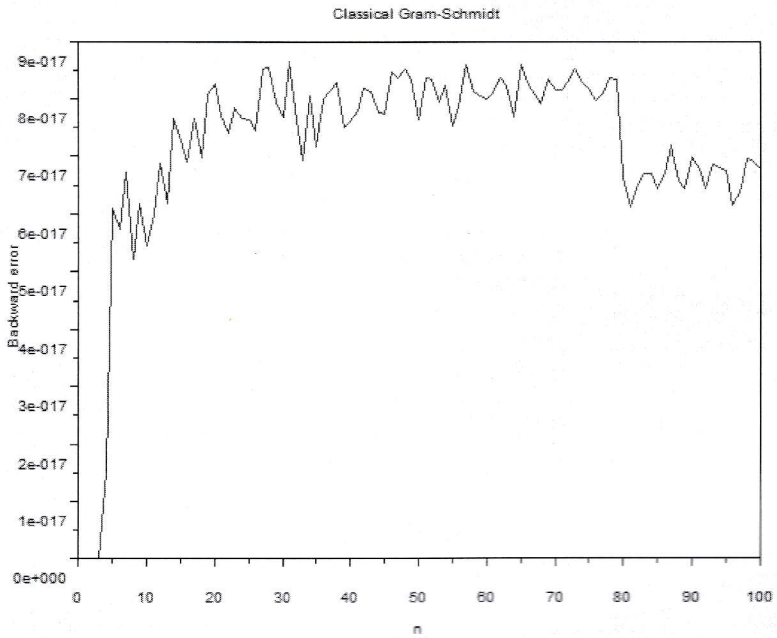

Figure 5. Backward error for CGS

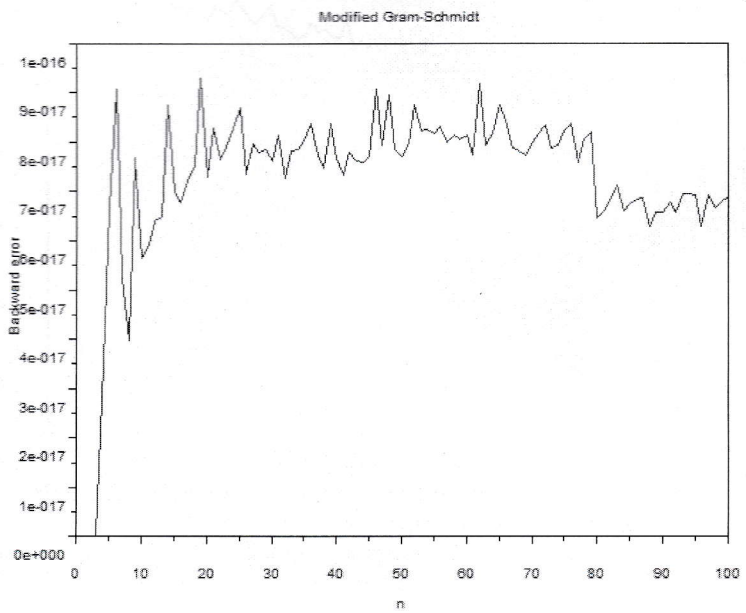

Figure 6. Backward error for MGS

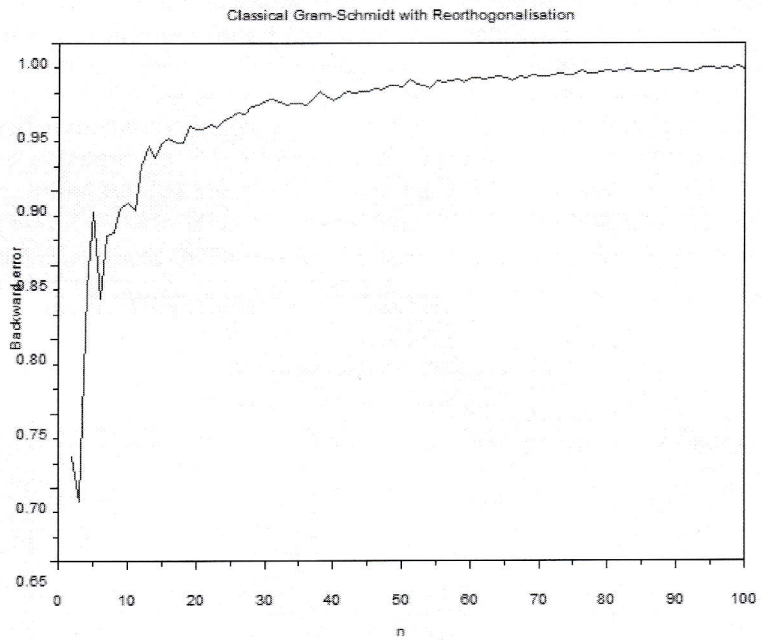

Figure 7. Backward error for CGS2

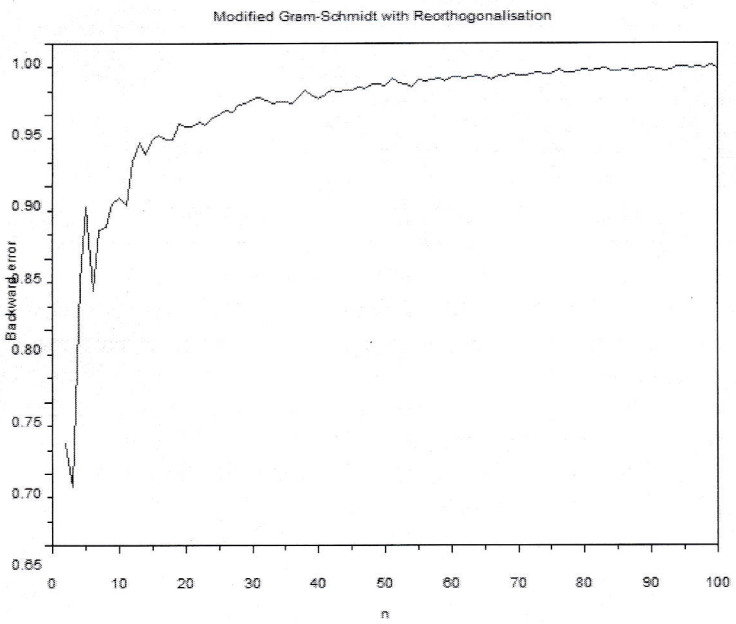

Figure 8. Backward error for MGS2

By implementing the same test matrices as those are used before, the backward error are shown in Figure 5 to Figure 8. The backward errors in CGS and MGS are very small, and they are even smaller than the machine precision. These errors lie between 0 and 10^{-16}. In contrast, the backward errors produced by CGS2 and MGS2 range between 0.65 and 1. These errors are large enough, and it means that the resulted factorisation using reorthogonalisation algorithms, $\mathbf{A} = \mathbf{QR}$, is not so accurate.

Therefore, the application of reorthogonalisation has advantages and disadvantages. On one hand, the process without reorthogonalisation will be faster and the backward error is very small, but the orthogonality may be absent. On the other hand, the process with reorthogonalisation will be slower and the backward error is large enough, but the orthogonality of the computed vectors can be preserved close to the machine precision level.

An interesting result can be found by applying the same algorithms to either Hilbert matrix or inverse of Hilbert matrix. In this paper, only the inverse of Hilbert matrices are used as the second set of test matrices (the result for Hilbert matrices is almost the same as the result for their inverses). Orthogonalisation and reorthogonalisation in Gram-Schmidt process can not be applied for big-size $\mathbf{H}_{n\times n}^{-1}$ inverse Hilbert matrix. When $n\times n$ inverse Hilbert matrix, for $n = 2,\mathrm{K}\ ,100$ is taken as the test matrix, the result is presented in Figure 9.

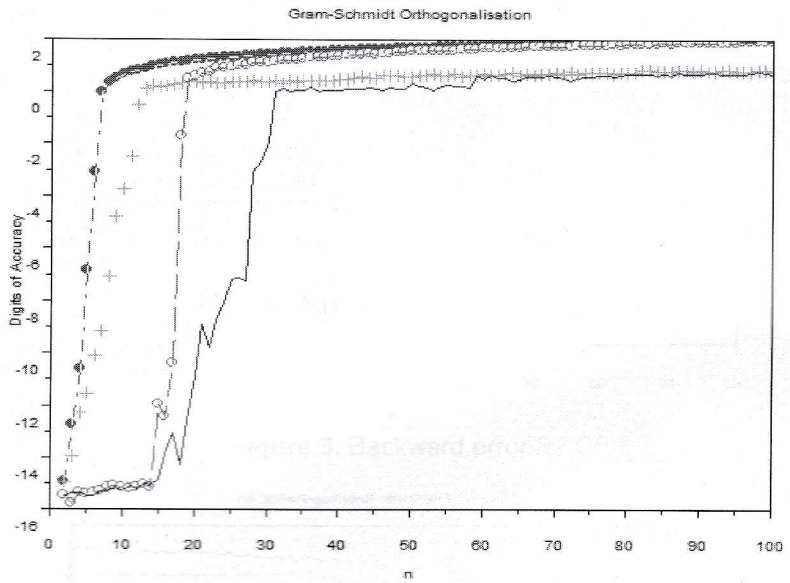

Figure 9. Digit of accuracy using inverse Hilbert matrix
Note •-.-.•-.-.• : CGS ++++++ : MGS •--•--•--• : CGS2 _____ : MGS2

This result can be considered as one limitation of Gram-Schmidt orthogonalisation and its reorthogonalisation. This failure of orthogonalisation and its reorthogonalisation might be because the computation in the orthogonalisation of the vectors is overflow, or exceeding the range of number that scilab can take in to account. This prediction can be double checked by multiplying $\mathbf{H}_{n\times n}^{-1}$ inverse Hilbert matrix and $\mathbf{H}_{n\times n}$ Hilbert matrix for some large n, and in fact, the result is not an identity matrix.

3. Running Time for Calculating the Vectors

This section is also divided into two parts. The first part gives an overview of the theoretical result of the operation count for Gram-Schmidt process, and the second part presents the relation between matrix size and running computation for CGS2.

3.1. Theoretical Background

Given $\mathbf{A}_{m\times n}$ matrix, the operation count for the CGS or MGS algorithm is $2mn^2$ flops (Trefethen & Bau, 1997: 59), while that for CGS2 or MGS2 is $4mn^2$, twice as much as CGS or MGS because of the reorthogonalisation operation. Therefore, if the matrix size is $n\times n$, the running time for those processes, t, will agree with the relation $t = cn^3$, where c is some constant.

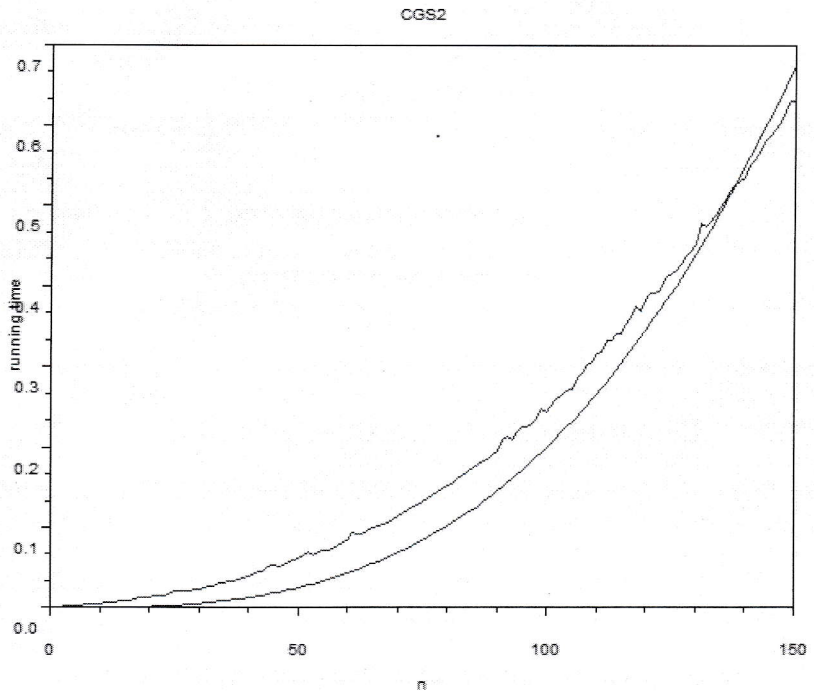

Figure 10. Relation between matrix size and running time

3.2. Simulation Result

In this part, a set of more number of random matrices is used. The set of test matrices which is used is $2\times 2,\mathrm{K},150\times 150$ random matrices, where all the entries are uniformly distributed in the interval (0,1). Here, recorded running time for calculating vectors is presented in Figure 10, and only the result for CGS2 is presented because the result for MGS2 is not much different, and so are the CGS and MGS. The unsmooth solid line is the relation between matrix size and the running computation in Scilab. The smooth solid line is $y = 2\times 10^{-7}x^3$. From this result, the relation between matrix size and the running time is approximately approaching the relation $t = cn^3$, where c is some constant, though $c = 2\times 10^{-7}$ is not the best choice in this case.

4. Conclusion

Reorthogonalisation in Gram-Schmidt process is worthwhile to preserve the orthogonality of the computed vectors close to the machine precision level. Reorthogonalisation does not work, if the computation in the machine is overflow. However, applying the reorthogonalisation leads to a higher backward error. To choose which algorithm is appropriate to be applied in a particular problem, considering the real problem is needed. That is if the problem is focusing more on the orthogonality, Gram-Schmidt algorithms with reorthogonalisation will be more appropriate.

Acknowledgment

The authors would like to thank Dr. Linda Stals (Australian National University) and Iwan Binanto, S.Si., M.Cs. (Sanata Dharma University) for the fruitful discussions.

Bibliography

Björck, Å. 1967. "Solving Linear Least Squares Problems by Gram-Schmidt Orthogonalisation." *BIT,* **7**: 1-21.

Björck, Å. 1992. "Loss and Recapture of Orthogonality in the Modified Gram-Schmidt Algorithm." *SIAM J. Matrix Anal. Appl.,* **13** (1): 176–190.

Ciarlet, P. G. 1989. *Introduction to Numerical Linear Algebra and Optimisation*. Cambridge: Cambridge University Press.

Cullen, C. G. 1994. *An Introduction to Numerical Linear Algebra*. Boston: PWS Publishing Company.

Giraud, L. and Langou, J. 2002. "When Modified Gram-Schmidt Generates a Well-Conditioned Set of Vectors." *IMA Journal of Numerical Analysis,* **22** (4): 521–528.

Giraud, L. *et al.* 2003. "On the Loss of Orthogonality in the Gram-Schmidt Orthogonalization Process." *CERFACS Technical Report No. TR/PA/03/25.*

Giraud, L. *et al.* 2005. "Rounding Error Analysis of the Classical Gram-Schmidt Orthogonalization Process." *Numer. Math.*, **101**: 87–100.

Heath, M. T. 2002. *Scientific Computing: An Introductory Survey*. New York McGraw-Hill.

Hogben, L. 2007. *Handbook of Linear Algebra*. Boca Raton: Chapman & Hall/CRC.

Kelley, C. T. 1998. *Iterative Methods for Linear and Nonlinear Equations*. Philadelphia: SIAM.

Noble, B., Daniel, J.W. 1977. *Applied Linear Algebra*. Englewood Cliffs: Prentice-Hall.

Rice, J. R. 1966. "Experiments on Gram-Schmidt Orthogonalization." *Math. Comp.*, **20**: 325-328.

Trefethen, L. N. and Bau, D. III. 1997. *Numerical Linear Algebra*. Philadelphia: SIAM.

SUDI MUNGKASI

Graduated from Gadjah Mada University S1 Mathematics study program in 2004 and from the Australian National University with Master of Mathematical Sciences (M.Math.Sc.) degree in 2008. Joined the Mathematics Department of Sanata Dharma University as a faculty member since 2005.

LUSIA KRISMIYATI BUDIASIH

Graduated from Satya Wacana Christian University S1 Mathematics study program in 1999 and from Bandung Institute of Technology with Magister Sains (M.Si) degree in 2005. Joined the Mathematics Department of Sanata Dharma University as a faculty member since 2000.