ELSEVIER

Physica D 101 (1997) 1–16

PHYSICA D

An efficient QR based method for the computation of Lyapunov exponents

Hubertus F. von Bremen [a,1], Firdaus E. Udwadia [b,*], Wlodek Proskurowski [c]

[a] *Department of Mechanical Engineering, University of Southern California, Los Angeles, CA 90089-1453, USA*

[b] *Department of Mechanical Engineering, Civil Engineering and Decision Systems, University of Southern California, Los Angeles, CA 90089-1453, USA*

[c] *Department of Mathematics, University of Southern California, Los Angeles, CA 90089-1113, USA*

Received 7 June 1996; revised 13 September 1996; accepted 13 September 1996
Communicated by Y. Kuramoto

Abstract

An efficient and numerically stable method to determine all the Lyapunov characteristic exponents of a dynamical system is developed. Numerical experiments are presented highlighting some aspects of convergence, accuracy and efficiency in the computation of the Lyapunov characteristic exponents.

Keywords: Dynamical systems; Computation of Lyapunov characteristic exponents; Efficient and stable computational method; Error criteria

1. Introduction

The computation of Lyapunov characteristic exponents (LCEs) for nonlinear dynamical systems is often required to understand their dynamics. Benettin et al. [1] in their two-part paper place the computation of these exponents on a solid analytical basis; they were the first to propose a Gram–Schmidt orthogonalization type procedure to compute them [2].

Geist et al. [3] made a thorough comparison of several methods for computing LCEs and also presented the main ideas published in the previous decade [1,4–6]. For more recent additional references see [7,8]. Here we address the discrete methods only, as the continuous ones are computationally less efficient [3].

Discrete methods are based on the factorization of a matrix representation of the tangent map into a product of an orthogonal matrix Q and an upper triangular matrix R with positive diagonal elements. Such factorizations can be achieved using the Gram–Schmidt (GS) orthogonalization procedure (and its variants – modified GS (MGS),

* Corresponding author.

[1] Graduate student.

PII S0167-2789(96)00216-3

reorthogonalized GS (RGS)) or any of the so-called QR-factorization methods of which the variant (HQR) that uses the Householder transformation is more efficient than the one that uses Givens rotations.

The GS method is known to be numerically unstable since the Q matrix could seriously deviate from orthogonality due to the accumulation of roundoff errors. In spite of this there are no reports of poor accuracy in the LCEs while using GS, see [1,3]. Significantly improved numerical stability is achieved by using the MGS (see, for example [9]). It is therefore MGS that is advocated by several authors, see [8,10]. Moreover, the computational costs of MGS and GS are virtually the same; this makes MGS even more attractive.

Reorthogonalization (sometimes used even in conjuction with the MGS, for ill-conditioned problems) essentially doubles the cost of a factorization. In contrast with GS, the HQR method is known to be backward stable, see [11], since it uses unitary transformations that preserve the L_2-norm.

For a square $n \times n$ matrix the asymptotic (for large n) cost of MGS factorization is $2n^3$ flops (a flop is a floating point addition or multiplication [12]), while the HQR-factorization requires $\frac{4}{3}n^3$ flops to compute the upper triangular R, and an additional $2n^3$ flops if the orthogonal Q also is required.

In computing the LCEs one also needs to multiply each of the Q matrices (computed by the HQR-factorization) by the consecutive matrix representation of the tangent map. Therefore, it is the complete, and thus more costly variant of HQR that has been employed until now, see [3]. In this paper we show how to organize computation of the LCEs using HQR-factorization in such a manner that computational savings are obtained over the method based on MGS.

We report several numerical experiments in which we study convergence, accuracy and efficiency of our method.

2. Computation of LCEs using the QR-factorization

In smooth dynamical systems, we usually deal with maps of the form $x_t = T^t x$ (here x often belongs to a suitable compact connected manifold M, and the map is from a (finite) n-dimensional space to an n-dimensional space) with $T^t = T \circ T^{t-1}$. The parameter t denotes a nonnegative integer or a real number. The sequence of tangent maps dT_x^t is obtained through iterations by considering $dT_x^t = dT_{T^t x} \circ dT_x^{t-1}$. We shall consider the matrix representation of these operators in the standard bases (in fact, any orthonormal basis set would be sufficient). The LCEs can then be obtained from the QR-factorization of the product of the matrix representations of these tangent maps which are determined at the appropriate points x of M. For brevity we shall denote the matrix representation of the tangent map (evaluated at the appropriate point x of the manifold) corresponding to $t = i$ by J_i. To determine an approximation to all the LCEs, one then needs the QR-factorization of the matrix product $J_m J_{m-1} \cdots J_1$. This decomposition can be done sequentially as follows. Starting with $Q_0 = I$, we have

$$\begin{aligned} \mathrm{qr}[J_m J_{m-1} \cdots J_1] &= \mathrm{qr}[J_m J_{m-1} \cdots J_2 (J_1 Q_0)] = \mathrm{qr}[J_m J_{m-1} \cdots J_3 (J_2 Q_1)][R_1] \\ &= \mathrm{qr}[J_m J_{m-1} \cdots (J_3 Q_2)][R_2 R_1] = \cdots \\ &= \mathrm{qr}[J_m J_{m-1} \cdots (J_i Q_{i-1})][R_{i-1} R_{i-2} \cdots R_2 R_1] = \cdots \\ &= Q_m [R_m \cdots R_2 R_1] = Q_m R. \end{aligned} \tag{1}$$

Here we sequentially use the QR-factorization, and $\mathrm{qr}[\cdot]$ denotes the QR-factorization process. Starting with J_1, at each step i in the above sequence, we perform a premultiplication $B_i = J_i Q_{i-1}$ followed by a QR-factorization of $B_i = J_i Q_{i-1} = Q_i R_i, i = 1, 2, \cdots, m$. The matrix R is the product of the matrices $R_m \cdots R_2 R_1$ obtained in this sequential manner. Furthermore, each of the diagonal elements of R is simply the product of the corresponding diagonal elements of all the R_i's. Hence, approximations to the n LCEs are then obtained as: $\chi_k = (1/m) \sum_{i=1}^{m} \ln |R_i(k,k)|$, $k = 1, 2, \ldots, n$. The computation can be presented as follows.

Algorithm for computing all the LCEs of a dynamical system

Initialization
Initialize Q to be the $n \times n$ Identity Matrix
Initialize *LCEvector* to be a zero n-vector
for $i = 1$ to *m_iterations*
  $B = J_i Q$
  Compute the QR-factorization of B $(QR = B)$
  $LCEvector = LCEvector + \log(diag(|R|))$
end
$LCEvector = LCEvector/m\_iterations.$

The approximate values of the n LCEs after *m_iterations* are then given by the components of the n-vector denoted *LCEvector*. Here the successive maps J_i at each iteration are assumed to be known.

There are several ways of computing the QR-factorization (which is required at each step i) indicated in the loop above, the commonest ones being the GS, QR, MGS, and the HQR. The choice of method must be based upon factors such as accuracy of the procedure, storage requirements and simplicity of implementation. If the aim is ease of implementation, one could make use of computing environments which have built-in QR-factorizations. For example, MATLAB has a reliable and efficient QR-factorization using Householder reflectors.

The Householder QR-factorization is known to be backward stable [11] (with regard to roundoff errors). Yet its direct application is (asymptotically, for large n) computationally more expensive than the GS (or the MGS) approach. However, efficiencies in the Householder-based QR-factorization in terms of both computation and storage can be achieved because:

(1) we need to compute and store only the diagonal entries of the matrix R, and
(2) the reflector (Householder) matrices which constitute Q can be sequentially assembled resulting in efficiencies in the computation of the action of the succeeding map J_{i+1} on Q.

In what follows we show that by modifications of the standard HQR-factorization we obtain a method that, while being more computationally efficient than the GS or the MGS approaches, seems also more stable with regard to roundoff errors.

3. Householder QR-based (HQRB) algorithm for the computation of LCEs

Consider the QR-factorization of the above-mentioned algorithm at the iteration index i. The HQR-factorization works along the following lines (for details see [13]). Given an $n \times n$ matrix B, one sequentially determines the matrices

$$B^{(s+1)} = H^{(s)} B^{(s)}, \quad s = 1, 2, \ldots, n-1; \qquad B^{(1)} \equiv B. \tag{2}$$

The Householder reflector matrices $H^{(s)}$ have the structure

$$H^{(s)} = I_n - w^{(s)}[w^{(s)}]^{\mathrm{T}}, \tag{3}$$

where the first $(s-1)$ elements of the n-vector $w^{(s)}$ are all zero. The matrix R of the QR-factorization of B is then obtained as

$$H^{(n-1)} \cdots H^{(2)} H^{(1)} B = R$$

and the matrix Q is given by

$$Q = H^{(1)}H^{(2)}\cdots H^{(n-1)}. \tag{4}$$

We note that this factorization is required to be done at each iteration i described in the aforementioned algorithm. At the next iteration (with index $i+1$) the matrix B is replaced by the matrix (we suppress the subscript in J_{i+1} and write J for simplicity)

$$JQ = \{JH^{(1)}H^{(2)}\cdots H^{(s)}\}H^{(s+1)}\cdots H^{(n-1)} = J^{(s)}H^{(s+1)}\cdots H^{(n-1)}. \tag{5}$$

We now make two important observations:

(1) The LCEs are obtained only through the determination of the diagonal elements of the R_i, $i = 1, 2, \ldots, m$ in Eq. (1) and therefore one needs to judiciously compute the elements in each of these upper triangular matrices; storage of only the diagonal elements is called for. We begin by noting that in the matrix $B^{(s)}$, the first $(s-1)$ columns have zeros below the main diagonal. Eq. (2) then becomes

$$B^{(s+1)} = \begin{bmatrix} I_{s-1} & 0 \\ 0 & \tilde{H}_{n-s+1} \end{bmatrix}\begin{bmatrix} b^{(s-1)} & c^{(n-s+1)} \\ 0 & d^{(n-s+1)} \end{bmatrix} = \begin{bmatrix} b^{(s)} & c^{(n-s)} \\ 0 & d^{(n-s)} \end{bmatrix}, \quad s = 2, \ldots, n-1, \tag{6}$$

where the matrix $b^{(s-1)}$ is an $(s-1)\times(s-1)$ upper triangular matrix, $c^{(n-s+1)}$ an $(s-1)\times(n-s+1)$, and $d^{(n-s+1)}$ is an $(n-s+1)\times(n-s+1)$ matrix. Thus $d^{(n-s)} = \tilde{H}_{n-s+1}d^{(n-s+1)}$, and it does not involve the matrix $c^{(n-s+1)}$. Furthermore, we are only interested in the submatrix $d^{(n-s)}$ when we continue on by premultiplication with the next reflector, $H^{(s+1)}$. Thus one does not need to compute nor store $c^{(n-s+1)}$, $s = 2, \ldots, n-1$.

(2) The action of the premultiplication of Q by J is given by Eq. (5) so that one does not need to compute Q explicitly. The matrix product $J^{(s)}H^{(s+1)}$ can be written as

$$J^{(s)}H^{(s+1)} = \begin{bmatrix} x^{(s)} & y^{(n-s)} \\ z^{(n-s)} & J^{(n-s)} \end{bmatrix}\begin{bmatrix} I_s & 0 \\ 0 & \tilde{H}_{n-s} \end{bmatrix},$$

where the matrix $x^{(s)}$ is $s\times s$, $y^{(n-s)}$ is $s\times(n-s)$, $z^{(n-s)}$ an $(n-s)\times s$ and $J^{(n-s)}$ is $(n-s)\times(n-s)$. This simplifies to

$$\begin{aligned} J^{(s)}H^{(s+1)} &= \begin{bmatrix} x^{(s)} & y^{(n-s)} \\ z^{(n-s)} & J^{(n-s)} \end{bmatrix}\begin{bmatrix} I_s & 0 \\ 0 & I_{n-s} - w^{(n-s)}(w^{(n-s)})^{\mathrm{T}} \end{bmatrix} \\ &= \begin{bmatrix} x^{(s)} & y^{(n-s)} - (y^{(n-s)}w^{(n-s)})(w^{(n-s)})^{\mathrm{T}} \\ z^{(n-s)} & J^{(n-s)} - (J^{(n-s)}w^{(n-s)})(w^{(n-s)})^{\mathrm{T}} \end{bmatrix}. \end{aligned} \tag{7}$$

Thus the first s columns of the matrix remain unchanged after the multiplication, and therefore do not need to be computed. The pseudo-code using the Householder-based method for computing the LECs having an $n\times n$ tangent map can be expressed as follows. [2]

Pseudo-code of the Householder QR based method (HQRB) for the computation of all LCEs

Initializations:
Initialize J to be the first tangent map.
Initialize $Jplus1$ to be the next tangent map.
Initialize *LCEvector* to be a zero vector

[2] A computer code implementing this pseudo-code can be obtained from the authors at fudwadia@alnitak.usc.edu, or at vonbrem@aludra.usc.edu.

for $i = 1$ to $m\_iterations$
QR-factorization part
for $k = 1$ to $n - 1$
Computation of the reflectors
The kth reflector is stored in the kth column of J. The diagonal elements of R are stored in the vector r.
The computations are as follows:

$$sigma = \sqrt{\sum_{s=k}^{n} J(s,k)^2},$$

$$gamma = sigma(sigma + |J(k,k)|)$$

$$r(k) = -sign(J(k,k))\ sigma$$

$$J(k,k) = J(k,k) - r(k)$$

Computation of reflectors with J as in (6):
for $j = k + 1$ to n

$$beta = \left(\sum_{s=k}^{n} J(s,k)J(s,j)\right)/gamma$$

for $s = k + 1$ to n

$$J(s,j) = J(s,j) - J(s,k)\ beta$$

end (s)
end (j)
Computation of action of $Jplus1$ *on* Q as in (7):
for $j = 1$ to n

$$beta = \left(\sum_{s=k}^{n} Jplus1(j,s)J(s,k)\right)/gamma$$

for $s = k$ to n

$$Jplus1(j,s) = Jplus1(j,s) - J(s,k)beta$$

end (s)
end (j)
end (k)

$$r(n) = J(n,n)$$

Set $J = Jplus1$
Set $Jplus1$ to be the next tangent map
For $q = 1$ to n

$$LCEvector(q) = LCEvector(q) + \log(|r(q)|)$$

end (q)
end (i)

$$LCEvector = LCEvector/m\_iterations$$

4. Numerical results

Computational cost of an iterative algorithm to find the LCEs is proportional to the number of iterations, which in turn often depends on the rate of convergence of the iterations. An additional issue is the choice of the stopping criterion based on the knowledge of convergence. Analyzing the convergence of the constant map B given by

Table 1
Residual and error versus number of iterations for χ_1 using map B from Benettin et al. [1]

No. of iterations	10	100	1000	10 000	100 000
Residual	4.06×10^{-2}	6.64×10^{-4}	8.93×10^{-6}	1.12×10^{-7}	1.35×10^{-9}
Error	5.05×10^{-1}	7.6×10^{-2}	9.9×10^{-3}	1.2×10^{-3}	1.45×10^{-4}

Table 2
Operation count for the factorization

Method	Multiplications/divisions	Additions/subtractions	Sqrts	Total
GS	$n^3 + n^2$	$n^3 + n^2$	n	$2n^3 + 2n^2 + n$
MGS	$n^3 + n^2$	n^3	n	$2n^3 + n^2 + n$
HQR	$\frac{5}{3}n^3 + 3n^2 - \frac{5}{3}n - 3$	$\frac{5}{3}n^3 + \frac{3}{2}n^2 - \frac{1}{6}n - 3$	$n - 1$	$\frac{10}{3}n^3 + \frac{9}{2}n^2 - \frac{5}{6}n - 7$
HQRB	$\frac{2}{3}n^3 + \frac{1}{2}n^2 + \frac{11}{6}n - 3$	$\frac{2}{3}n^3 + \frac{7}{3}n - 3$	$n - 1$	$\frac{4}{3}n^3 + \frac{1}{2}n^2 + \frac{31}{6}n - 7$

Benettin et al. [1], as Table 1 indicates, the error e in χ_1 (the largest LCE) is inversely proportional to the number of iterations, m i.e., $e^{(m)} = k/m$. Thus the actual convergence rate is $e^{(m)}/e^{(m+1)} = (m-1)/m = 1 - 1/m$, i.e., one decimal gain in accuracy for each increase in the order of the number iterations. The same convergence rate was observed for the other LCEs. This shows that extremely slow convergence could occur, thus the need for efficient algorithms to minimize computing times.

4.1. Computational efficiency comparison

As a measure of efficiency of the algorithms under consideration we use the operation count of performing one QR-factorization of the tangent map (represented by an $n \times n$ matrix) followed by the action of multiplying the succeeding tangent map by the orthogonal matrix Q. We follow the GS and MGS algorithms given by Golub and Van Loan [12], and the HQR as given by Dahlquist and Björck [13]. The HQR presented computes R and Q explicitly, but in the computation of Q the explicit product of the reflector matrices (as in Eq. (4)) is avoided, see also [3]. The HQRB algorithm is the one described in this paper.

The main difference in efficiency of the various methods comes from the way the QR-factorization part of the method is performed. The ratios of the asymptotic constants (thus the dominant cost for the algorithm for sufficiently large values of n) in the operation count [3] in Table 2 are 3:5:2 for the GS (and MGS), HQR, and HQRB, respectively. The operation count for the action of multiplying the succeeding tangent map by the orthogonal matrix Q is roughly the same for all methods (see Table 3) and hence the cost associated with this step is nearly the same for all methods. The ratios for the total cost then become 6:8:5, see Table 4. Thus, for values of n larger than about 10 the savings by using HQRB versus MGS (the more efficient of the remaininig methods) in the number of operations are more than 10% (up to about 16% asymptotically). Since in the computation of LCEs smaller problems are also not uncommon, we include a plot of the total cost of the methods for the range below $n = 10$, see Fig. 1. Also here the HQRB method is advantageous, especially when compared with the HQR method. Since the actual computing time for different methods is environment and implementation dependent, we report here the operation count. The HQRB and HQR have the same accuracy. They differ only in the number of computations when it comes to computing the LCEs.

[3] $(n-1)$ absolute value and sign evaluations in the HQR and HQRB are not included in the tables.

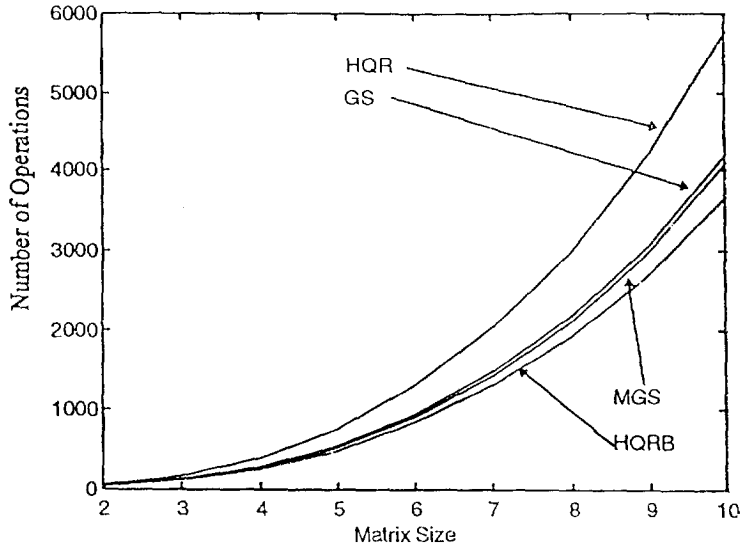

Fig. 1. Number of operations versus matrix size for GS, MGS, HQR and HQRB.

Table 3
Operation count for action of the multiplication by Q

Method	Multiplications/divisions	Additions/subtractions	Total
GS	n^3	n^3	$2n^3$
MGS	n^3	n^3	$2n^3$
HQR	n^3	n^3	$2n^3$
HQRB	$n^3 + 2n^2 - 3n$	$n^3 + n^2 - 2n$	$2n^3 + 3n^2 - 5n$

Table 4
Operation count for the factorization and action of multiplication by Q

Method	Multiplications/divisions	Additions/subtractions	Sqrts	Total
GS	$2n^3 + n^2$	$2n^3 + n^2$	n	$4n^3 + 2n^2 + n$
MGS	$2n^3 + n^2$	$2n^3$	n	$4n^3 + n^2 + n$
HQR	$\frac{8}{3}n^3 + 3n^2 - \frac{5}{3}n - 3$	$\frac{8}{3}n^3 + \frac{3}{2}n^2 - \frac{1}{6}n - 3$	$n - 1$	$\frac{16}{3}n^3 + \frac{9}{2}n^2 - \frac{5}{6}n - 7$
HQRB	$\frac{5}{3}n^3 + \frac{5}{2}n^2 - \frac{7}{6}n - 3$	$\frac{5}{3}n^3 + n^2 + \frac{1}{3}n - 3$	$n - 1$	$\frac{10}{3}n^3 + \frac{7}{2}n^2 + \frac{1}{6}n - 7$

4.2. Accuracy comparison

We use the following constant maps to assess the accuracy of the GS, MGS and HQRB methods.

Example 1.

$$A = \begin{bmatrix} (110 + 11\mu)/10 & 1 & 0 & 0 \\ -(100 + 121\mu)/10 & 0 & 1 & 0 \\ (110 + 11\mu)\mu/10 & 0 & 0 & 1 \\ -\mu^2 & 0 & 0 & 0 \end{bmatrix}, \quad 10^{-8} \le \mu \le 10^{-6},$$

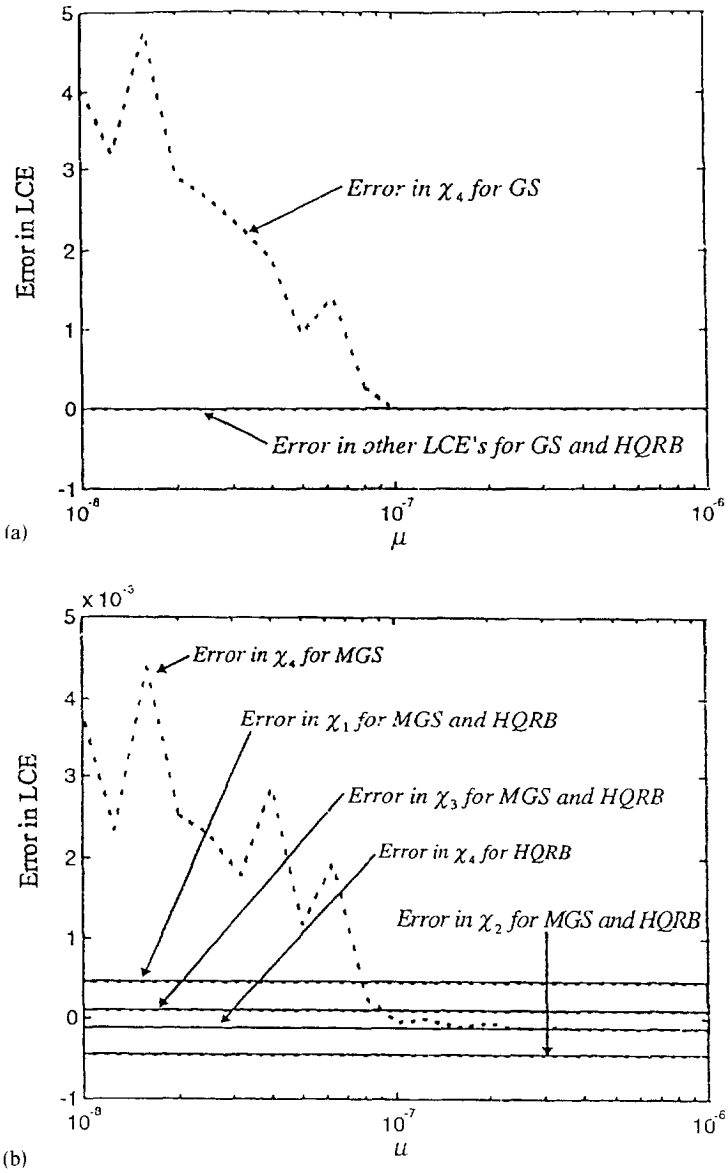

Fig. 2. Error $\chi_i^{\text{computed}} - \chi_i^{\text{exact}}$ versus μ using (a) HQRB and GS and (b) HQRB and MGS for Example 1 after 1000 iterations.

Example 2.

$$B = \begin{bmatrix} 3 & 5 & 6 & 9 \\ 3 & 5+\delta & 6 & 9 \\ 3 & 5 & 6+\delta & 9 \\ 2 & 4 & 5 & 2 \end{bmatrix}, \quad 10^{-8} \le \delta \le 10^{-6}.$$

All the computations were performed using MATLAB within the IEEE floating point standard, i.e., with a machine precision of about 2.22×10^{-16}.

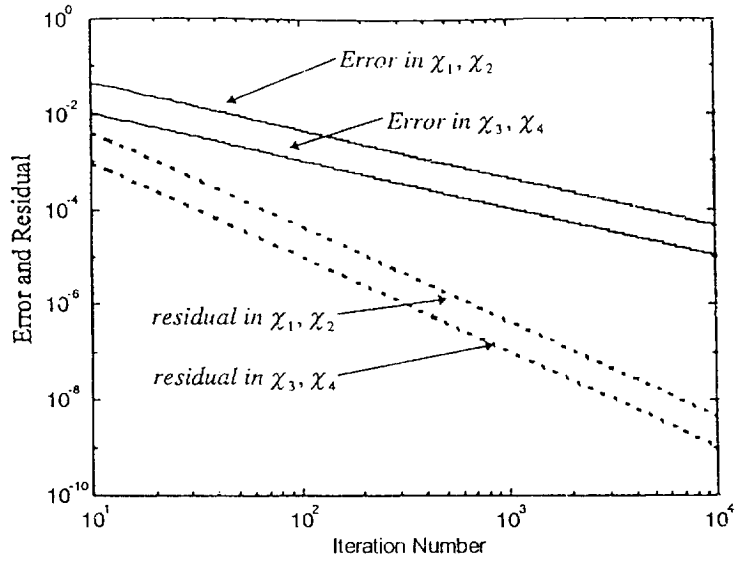

Fig. 3. Log of Error in LCE and Residual versus iteration number for Example 1 with $\mu = 10^{-8}$.

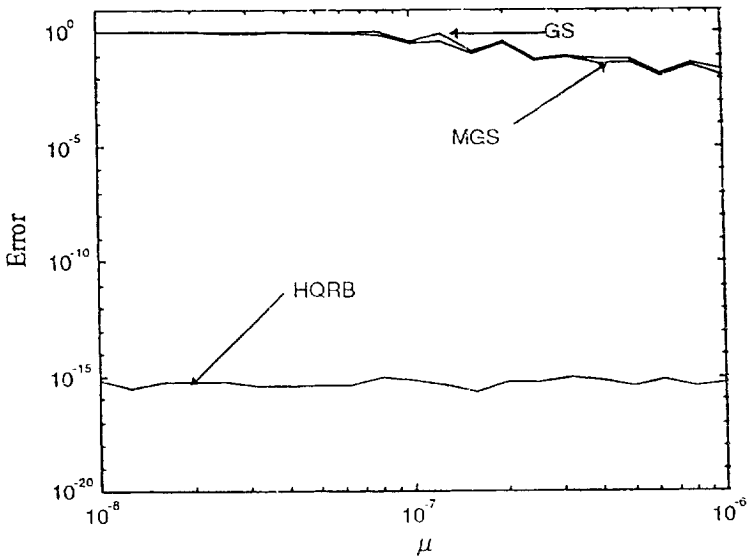

Fig. 4. Error $\tilde{a}_m$ versus μ for GS, MGS and HQRB, for Example 1 after 1000 iterations.

Example 1 is a perturbation of the transpose of the companion matrix corresponding to the characteristic polynomial with roots: 10, 1, μ, and $\mu/10$. The μ^2 term in the (2,1) entry of the matrix A was deleted, so that the matrix could be represented with no roundoff error. The exact LCEs were computed using 100-digit floating-point arithmetic utilizing MAPLE, and the results were then truncated to 16 digits. Fig. 2(a) shows a plot of the errors, $\chi_i^{\text{computed}} - \chi_i^{\text{exact}}$ as a function of the prameter μ, in the computation of the four LCEs at the end of 1000 iterations using HQRB and GS (the LCEs are ordered as $\chi_1 > \chi_2 > \chi_3 > \chi_4$). We observe that for sufficiently small values of the parameter μ, $\mu < 10^{-7}$, the performance of GS deteriorates significantly in computing the smallest LCE. This can be explained by the roundoff errors when μ^2 approaches the value of machine precision, i.e., when

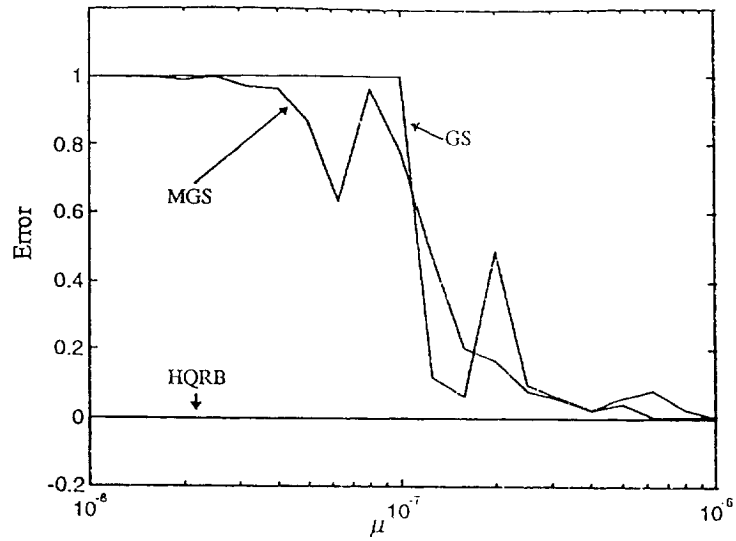

Fig. 5. Error a_m versus μ for GS, MGS and HQRB, for Example 1 after 1000 iterations.

$fl(1+\mu^2) = 1$. Also (not shown), there is no improvement in the smallest LCE by using GS when the number of iterations is increased. On the other hand, HQRB exhibits good stability, and remains stable with increased number of iterations.

Fig. 2(b) shows a similar comparison of the MGS and HQRB methods. Again HQRB appears to be the most stable, though the MGS gives errors in the smallest LCE, which may be considered to be negligibly small.

For most dynamical systems the errors $e_i^m = |\chi_i^m - \chi_i^{\text{exact}}|$ can seldom be measured. A computable quantity after m iterations is the residual $r_i^m = |\chi_i^m - \chi_i^{m-1}|$. However, the residual may not provide a good estimate of the error, see Table 1 and Fig. 3, and thus should not be used, in general, in establishing a proper stopping criterion for the iterations.

Fig. 3 confirms the earlier observation that to gain an additional digit in accuracy may require an increase in the order of iterations for a constant map. Dieci and Van Vleck [8] made an attempt in establishing a proper error criterion, at some cost.

A commonly used indicator of accuracy in the determination of LCEs [3] is the error in orthogonality $\tilde{a_m} = \log_{10} \| Q_m^{\mathrm{T}} Q_m - I \|_2$ of Q_m at the mth iteration. This measure reflects errors in the QR-factorization and one would expect that large deviations from orthogonality of the matrix Q_m would lead to large errors in the LCEs. Fig. 4 shows $\tilde{a}_m$ for $m = 1000$ as a function of μ for Example 1. We observe that the error in orthogonality for the HQRB method is much smaller than that for the GS. Yet, a comparison for these errors between the GS and MGS methods shows that they are close to each other over the entire range of μ, even though the computed LCEs using MGS are far superior to those obtained from GS for values of μ less than about 10^{-7} (see Figs. 2(a) and (b)). Most importantly, the GS and MGS errors increase with decreasing μ and give no indication of the abrupt loss of accuracy of GS around $\mu = 10^{-7}$. When plotted on regular scale the error $a_m = \| Q_m^{\mathrm{T}} Q_m - I \|_2$ shows the loss of orthogonality more clearly, see Fig. 5.

Another measure of the orhogonality of the matrix Q is the maximum value of the inner product between two of its columns. One could define the error in orthogonality as $b_m = \max_{\forall i,j; i \neq j} |q_i^{(m)^{\mathrm{T}}} q_j^{(m)}|$, where $q_i^{(m)}$ and $q_j^{(m)}$ are the ith and jth columns of the matrix Q_m after m iterations. Fig. 6 shows this error plotted for various values of μ for the GS, MGS and HQRB approaches. The HQRB method is more stable than GS and MGS over the range

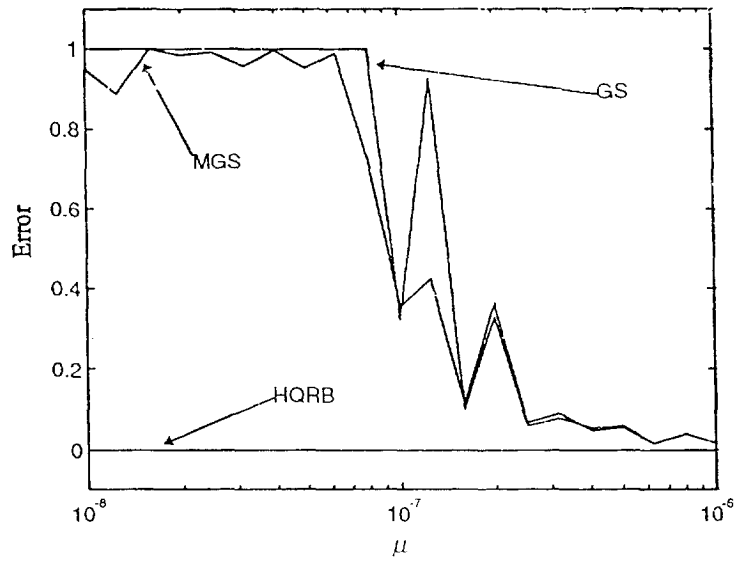

Fig. 6. Error b_m versus μ for GS, MGS and HQRB, for Example 1 after 1000 iterations.

Table 5
Computed LCEs and errors after 1000 and 10 000 iterations for Example 1 with $\mu = 10^{-6.8}$

LCE	GS	MGS	HQRB	Exact	E-GS	E-MGS	E-HQRB
χ_1	2.30303702	2.30303702	2.30303702		4.52×10^{-4}	4.52×10^{-4}	4.52×10^{-4}
	2.30263028	2.30263028	2.30263028	2.30258509	4.52×10^{-5}	4.52×10^{-5}	4.52×10^{-5}
χ_2	−0.00045193	−0.00045193	−0.00045193		-4.52×10^{-4}	-4.52×10^{-4}	-4.52×10^{-4}
	−0.00004519	−0.00004519	−0.00004519	3.87418×10^{-8}	-4.52×10^{-5}	-4.52×10^{-5}	-4.52×10^{-5}
χ_3	−15.6574732	−15.6574732	−15.6574732		1.05×10^{-4}	1.05×10^{-4}	1.05×10^{-4}
	−15.6575680	−15.6575680	−15.6575680	−15.6575786	1.05×10^{-5}	1.05×10^{-5}	1.05×10^{-5}
χ_4	−17.9602681	−17.9602690	−17.9602690		-1.04×10^{-4}	-1.05×10^{-4}	-1.05×10^{-4}
	−17.9601741	−17.9601742	−17.9601742	−17.9601637	-1.04×10^{-5}	-1.05×10^{-5}	-1.05×10^{-5}

Table 6
Computed LCEs and errors after 1000 and 10 000 iterations for Example 1 with $\mu = 10^{-8}$

LCE	GS	MGS	HQRB	Exact	E-GS	E-MGS	E-HQRB
χ_1	2.30303702	2.30303702	2.30303702		4.52×10^{-4}	4.52×10^{-4}	4.52×10^{-4}
	2.30263028	2.30263028	2.30263028	2.30258509	4.52×10^{-5}	4.52×10^{-5}	4.52×10^{-5}
χ_2	−0.00045193	−0.00045193	−0.00045193		-4.52×10^{-4}	-4.52×10^{-4}	-4.52×10^{-4}
	−0.00004519	−0.00004519	−0.00004519	2.44444×10^{-9}	-4.52×10^{-5}	-4.52×10^{-5}	-4.52×10^{-5}
χ_3	−18.4205753	−18.4205753	−18.4205753		1.05×10^{-4}	1.05×10^{-4}	1.05×10^{-4}
	−18.4206702	−18.4206702	−18.4206702	−18.4206807	1.05×10^{-5}	1.05×10^{-5}	1.05×10^{-5}
χ_4	−16.6941783	−20.7193529	−20.7233711		4.0291	3.91×10^{-3}	-1.05×10^{-4}
	−16.6943953	−20.7228745	−20.7232763	−20.7232658	4.0270	3.91×10^{-4}	-1.05×10^{-5}

Table 7
Computed LCEs and errors after 1000 and 10 000 iterations for Example 2 with $\delta = 10^{-6.8}$

LCE	GS	MGS	HQRB	Exact	E-GS	E-MGS	E-HQRB
χ_1	2.975244707	2.975244707	2.975244707		-1.25×10^{-3}	-1.25×10^{-3}	-1.25×10^{-3}
	2.976370819	2.976370819	2.976370819	2.976495942	-1.25×10^{-4}	-1.25×10^{-4}	-1.25×10^{-4}
χ_2	1.284414938	1.284414938	1.284414938		-1.77×10^{-3}	-1.77×10^{-3}	-1.77×10^{-3}
	1.286007039	1.286007039	1.286007039	1.286183929	-1.77×10^{-4}	-1.77×10^{-4}	-1.77×10^{-4}
χ_3	−15.65700657	−15.65700657	−15.65700657		5.72×10^{-4}	5.72×10^{-4}	5.72×10^{-4}
	−15.65752142	−15.65752142	−15.65752142	−15.65757863	5.72×10^{-5}	5.72×10^{-5}	5.72×10^{-5}
χ_4	−17.43289971	−17.43290368	−17.43290369		2.45×10^{-3}	2.45×10^{-3}	2.45×10^{-3}
	−17.43510663	−17.43510704	−17.43510705	−17.43535185	2.45×10^{-4}	2.45×10^{-4}	2.45×10^{-4}

Table 8
Computed LCEs and errors after 1000 and 10 000 iterations for Example 2 with $\delta = 10^{-8}$

LCE	GS	MGS	HQRB	Exact	E-GS	E-MGS	E-HQRB
χ_1	2.975244702	2.975244702	2.975244702		-1.25×10^{-3}	-1.25×10^{-3}	-1.25×10^{-3}
	2.976370814	2.976370814	2.976370814	2.976495938	-1.25×10^{-4}	-1.25×10^{-4}	-1.25×10^{-4}
χ_2	1.284414947	1.284414947	1.284414947		-1.78×10^{-3}	-1.77×10^{-3}	-1.77×10^{-3}
	1.286007039	1.286007039	1.286007039	1.286183938	-1.77×10^{-4}	-1.77×10^{-4}	-1.77×10^{-4}
χ_3	−18.42010868	−18.42010869	−18.42010869		5.72×10^{-4}	5.72×10^{-4}	5.72×10^{-4}
	−18.42062353	−18.42062354	−18.42062354	−18.42068074	5.72×10^{-5}	5.72×10^{-5}	5.72×10^{-5}
χ_4	−16.88335438	−20.19600564	−20.19600575		3.3151	2.45×10^{-3}	2.45×10^{-3}
	−16.87315174	−20.19820899	−20.19820910	−20.19845397	3.3253	2.45×10^{-4}	2.45×10^{-4}

$10^{-8} \leq \mu \leq 10^{-6}$. At the same time, for the GS and MGS methods one can see changes in orthogonality in the transition region where we could expect accumulation of the roundoff errors as μ becomes such that $fl(1+\mu^2) = 1$.

The absolute value of the determinant of Q_m could also be used as a measure of the error in orthogonality in the form $c_m = |1 - |\mathrm{Det}(Q_m)||$. Departure of c_m from zero would then be an indication of lack of orthogonality. This error criterion (see Fig. 7) behaves similar to b_m (Fig. 6) in showing the zone in μ where inaccurate LCEs may be expected. Each of these error criteria which deal with orthogonality seem to indicate that HQRB is more stable than GS and MGS over the range of μ considered.

In order to verify how well the error criteria a_m, b_m and c_m capture the inaccuracy in determining the LCEs, we computed the LCEs with all three methods and compared them with the exact values. Tables 5 and 6 show the values of the LCEs using the different methods after 1000 (top value) and 10 000 (botton value) iterations. The errors $\chi_i^{\text{computed}} - \chi_i^{\text{exact}}$ in LCEs for the three methods are denoted as E-GS, E-MGS, E-HQRB.

As expected (see Figs. 5–7), the LCEs show similar small errors for all three methods when $\mu = 10^{-6.8}$ (Table 5). For $\mu = 10^{-8}$, large errors in the smallest LCE for the GS are observed compared to the errors in MGS and HQRB. These errors do not reduce when the number of iterations increases (Table 6). For the smallest LCE the MGS shows a slightly larger error than the HQRB, although it does decrease with an increase in the number of iterations. These result for Example 1 indicate that the measure of error in orthogonality b_m and c_m capture the inaccuracy in determining the LCEs about as well as the usually used a_m criterion.

Flop counts for the computation of a_m, b_m and c_m are shown in Fig. 8. As seen from the figure, the flop count for c_m is the lowest. As each of the error criteria are similar in their behavior as far as indicating errors in LCEs are concerned, the criterion c_m may be preferable from the viewpoint of computational efficiency.

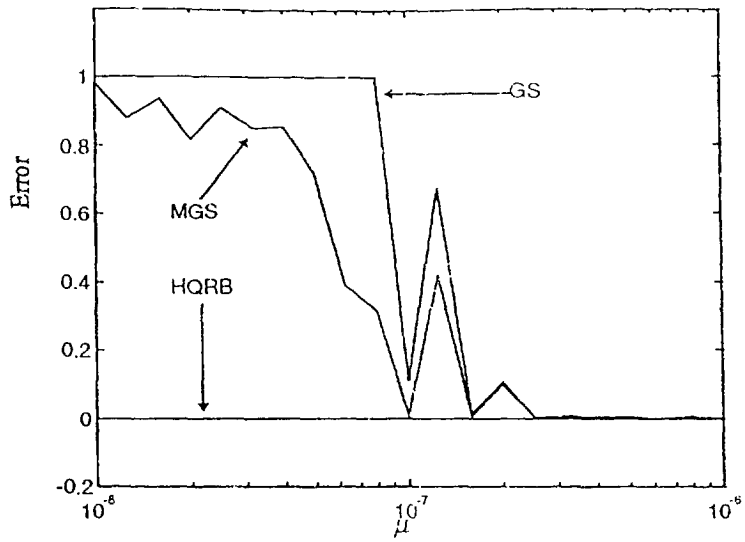

Fig. 7. Error c_m versus μ for GS, MGS and HQRB, for Example 1 after 1000 iterations.

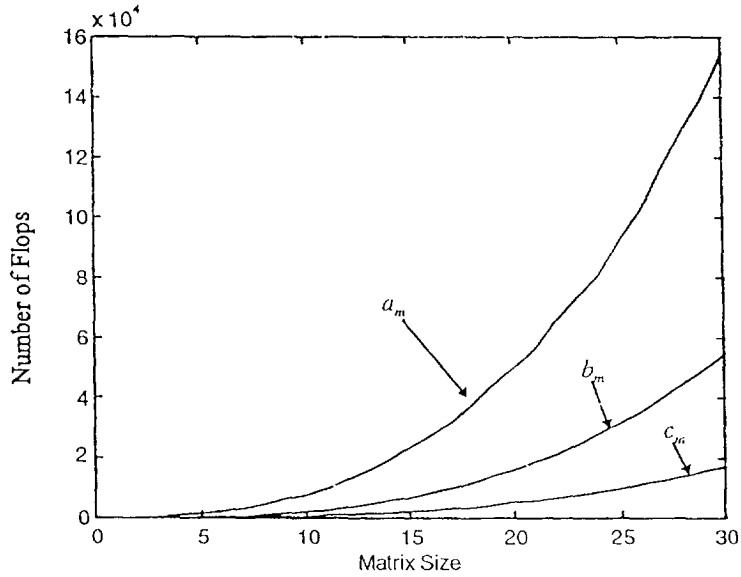

Fig. 8. Flop count of a_m, b_m and c_m, versus matrix size.

In Example 2 the exact LCEs were computed as in Example 1 using MAPLE. The significant difference in comparison with Example 1 is that here the performances of HQRB and MGS are almost identical, see Figs. 9(a) and (b), while as before GS produces larger errors for the range $\delta < 10^{-7.5}$.

Tables 7 and 8 show the LCEs and their errors for the three methods after 1000 and 10 000 iterations for $\delta = 10^{-6.8}$ and $\delta = 10^{-8}$, respectively. As expected, no significant difference in the errors in LCEs is observed among the three methods for $\delta = 10^{-6.8}$ (Table 7). When $\delta = 10^{-8}$, only the error for χ_4 using GS is significantly greater than the errors obtained from MGS and HQRB. This behavior is well predicted by the error criteria c_m, see Fig. 10(a). The superior stability of the HQRB method is depicted by $\tilde{a}_m$, see Fig. 10(b).

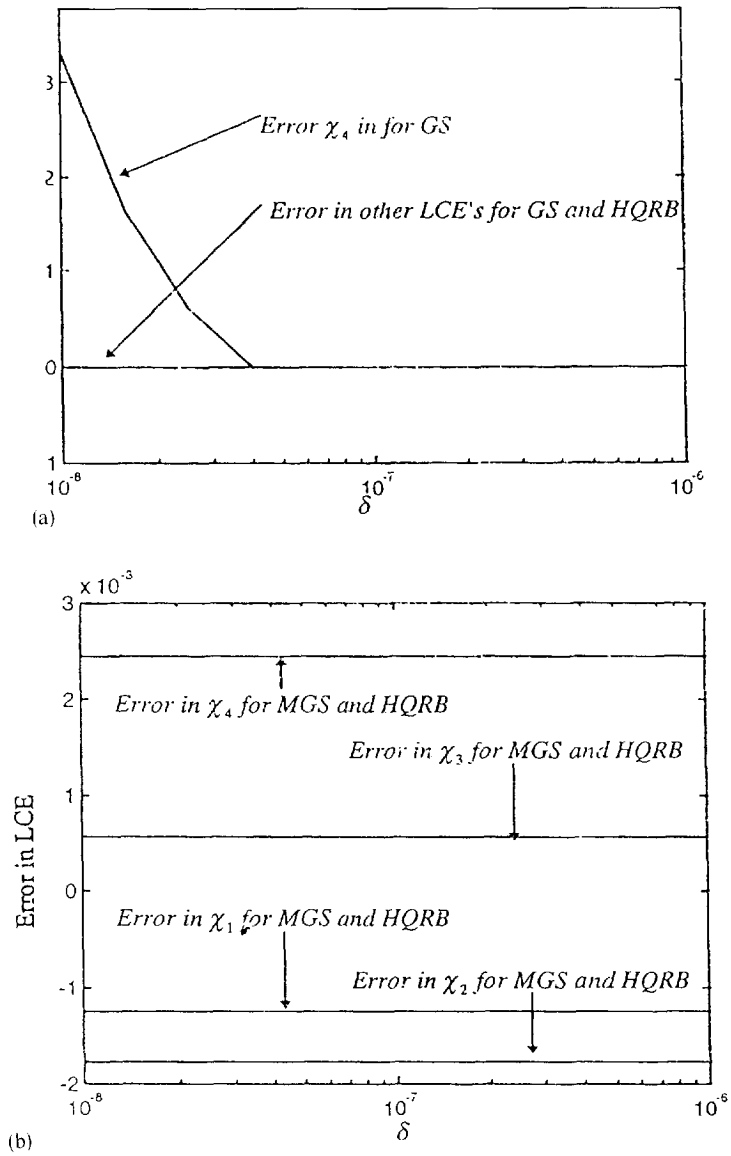

Fig. 9. Error $\chi_i^{\text{computed}} - \chi_i^{\text{exact}}$ versus δ using (a) HQRB and GS and (b) HQRB and MGS for Example 2 after 1000 iterations.

For testing the methods, other examples were also considered. Among them we explored the constant map [14]:

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \varepsilon & 0 & 0 & 0 \\ 0 & \varepsilon & 0 & 0 \\ 0 & 0 & \varepsilon & 0 \end{bmatrix}.$$

This example showed fluctuations in a_m, b_m and c_m for GS as the iteration number was varied for a large range of ε (say for $10^{-8} \leq \varepsilon \leq 10^{-6}$). Such fluctuations in the errors in orthogonality should warn us to use the indicators

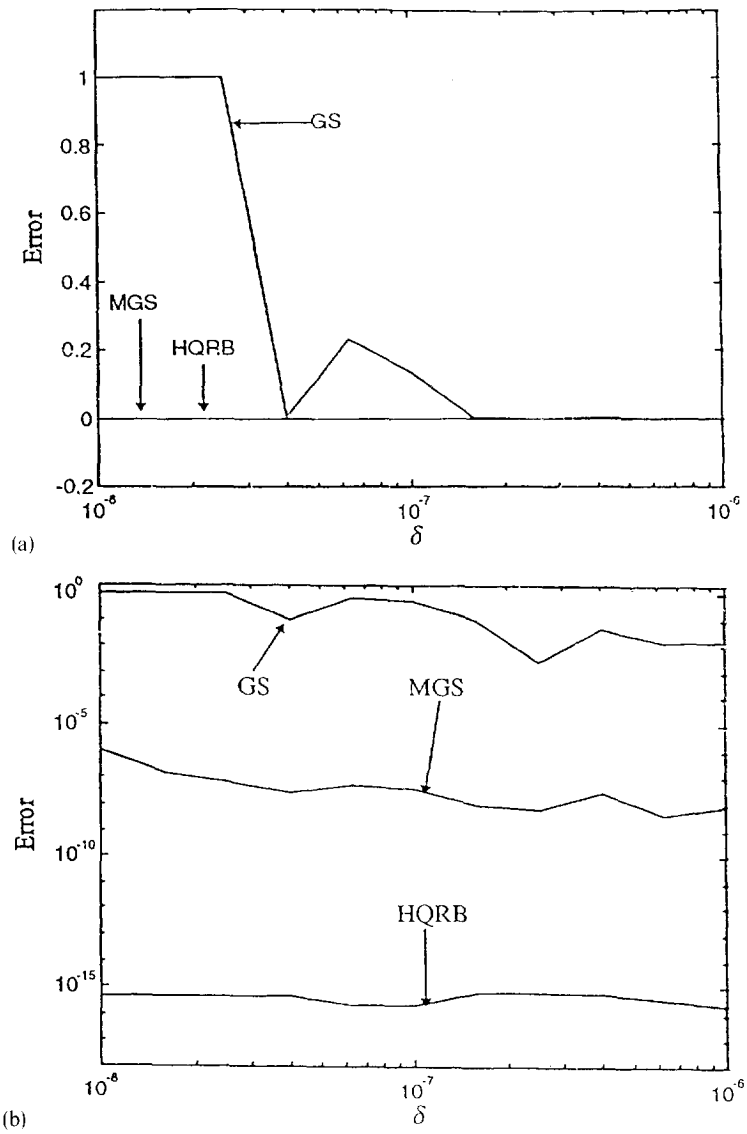

Fig. 10. (a) Error c_m versus δ for GS, MGS and HQRB, for Example 2 after 1000 iterations. (b) Error in orthogonality $\tilde{a}_m$ versus δ for Example 2 after 1000 iterations.

a_m, b_m and c_m with caution. For, one could sample the errors in orthogonality at places where the error is "small", and come to wrong conclusions about the validity of the computed LCEs.

5. Conclusions

In this paper we have provided a computationally efficient and robust algorithm for computing the LCEs of a dynamical system. We base our approach on the recognition that:

(1) approximations to the LCEs can be obtained in a direct manner from the diagonal elements of the R matrix in the QR-factorization of the tangent map, and

(2) such a factorization can be efficiently done through a modification in the use of the HQR algorithm.

The approach proposed here for determining LCEs is shown to be computationally superior to the GS, MGS, and far superior to standard HQR methods. In addition, the numerical experiments reported here show that the algorithm is more stable with respect to roundoff errors than both the GS and MGS algorithms.

Numerical experiments suggest that it may be misleading to use the residual error in the LCEs as a measure of the accuracy of their computed values. Three measures of the error in orthogonality of Q_m at the mth iteration are considered: $a_m = \| Q_m^{\mathrm{T}} Q_m - I \|_2$; $b_m = \max_{i,j;i\neq j} |q_i^{(m)^{\mathrm{T}}} q_j^{(m)}|$ where $q_i^{(m)}$ and $q_j^{(m)}$ are the ith and jth columns of the matrix Q_m; and $c_m = |1 - |\mathrm{Det}(Q_m)||$. It appears that a_m, b_m and c_m are comparable in their ability to detect the lack of orthogonality; since c_m is computationally by far the cheapest to determine, it might be more useful when computation time is an important factor.

References

[1] G. Benettin, L. Galgani, A. Giorgilli and J.M. Strelcyn, Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. Parts I and II, Meccanica 15 (1980) 9–30.
[2] G. Benettin, L. Galgani, A. Giorgilli and J.M. Strelcyn, Tous les nombres characteristiques de Lyapunov sont effectivement calculables, Comptes Rendues Acad. Sci. Paris 286A (1978) 431–433.
[3] K. Geist, U. Parlitz and W. Lauterborn, Comparison of different methods for computing Lyapunov exponents, Progr. Theoret. Phys. 83 (5) (1990) 875–893.
[4] I. Shimada and T. Nagashima, A numerical approach to ergodic problem of dissipative dynamical systems, Progr. Theoret. Phys. 61 (1979) 1605–1616.
[5] A. Wolf, J. Swift, H. Swinney and J. Vastano, Determining Lyapunov exponents from a time series, Physica D 16 (1985) 285–317.
[6] J.P. Eckmann and D. Ruelle, Ergodic theory of chaos and strange attractors, Rev. Modern Phys. 57 (1985) 617–656.
[7] G. Barna and A.I. Tsuda, New method for computing Lyapunov exponents, Phys. Lett. A 175 (6) (1993) 421–427.
[8] L. Dieci and E. Van Vleck, Computation of a few Lyapunov exponents for continuous and discrete dynamical systems, Appl. Numer. Math. 17 (3) (1995) 275–291.
[9] Å. Björck, Numerics of Gram–Schmidt orthogonalization, Linear Algebra Appl. 197 (1994) 297–316.
[10] T. Parker and L. Chua, Practical Numerical Algorithms for Chaotic Systems (Springer, Berlin, 1989).
[11] J.H. Wilkinson, The Algebraic Eigenvalue Problem (Oxford University Press, Oxford, 1965).
[12] G. Golub and C. Van Loan, Matrix Computations, 2nd Ed. (Johns Hopkins University Press, Baltimore, MD, 1993).
[13] G. Dahlquist and Å. Björck, Numerical methods, 2nd Ed. (1997), to be published.
[14] P. Läuchli, Jordan-Elimination und Ausgleichung nach kleinsten Quadraten, Numer. Math. 3 (1961) 226–240.