Numerical Solution of Large Lyapunov Equations

Youcef Saad

Abstract

In this paper we propose a few methods for solving large Lyapunov equations that arise in control problems. We consider the common case where the right hand side is a small rank matrix. For the single input case, i.e., when the equation considered is of the form $AX + XA^T + bb^T = 0$, where b is a column vector, we establish the existence of approximate solutions of the form $X = VGV^T$ where V is $N \times m$ and G is $m \times m$, with m small. The first class of methods proposed is based on the use of numerical quadrature formulas, such as Gauss-Laguerre formulas, applied to the controllability Grammian. The second is based on a projection process of Galerkin type. Numerical experiments are presented to test the effectiveness of these methods for large problems.

1 Introduction

Many of the matrix equations that arise in control problems can be successfully solved by well-known numerical techniques, when the matrices involved are small. In contrast there has been very little done to provide numerical methods for solving these same problems when the associated matrices are very large. Yet, there are now several applications in control which lead to matrix equations involving very large sparse matrices. These typically arise whenever the model involves a partial differential equation in several space dimensions such as when considering large space structures [3] or when a large network model, e.g. electrical network, [2] is involved.

Recently, there has been a few efforts directed towards large scale matrix problems in control. In [5] we proposed a method for partial pole placement, which consists of placing a few of the poles of the matrix, namely only those that are unstable. The methods proposed are based on projecting the problem onto a small invariant subspace of A associated with the unstable eigenvalues.

In this paper we focus on Lyapunov's equations. The numerical solution of these equations have gained considerable importance in the last few years due to the crucial role that they play in the so-called H_∞ analysis. The basic problem addressed by this theory is to find a low dimensional ODE model that approximates the original dynamical system governed by

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1}$$

Here, A is $N \times N$, B is $N \times p$ and C is $q \times N$. In the situation we consider N may be very large while p and q are very small. The closeness of this system from its

lower dimensional model is measured in terms of the singular values of the matrix XY where X and Y are solutions of the two Lyapunov equations,

$$AX + XA^T + BB^T = 0 \tag{2}$$
$$A^TY + YA + C^TC = 0 \tag{3}$$

These are termed Hankel singular values of (1). Typically, one would like to compute some of the largest Hankel singular values. In this paper we will provide the tools for computing approximations to X and Y. Moreover, these approximations are in such a form that they allow easy computation of the largest singular values of XY.

Throughout, we will assume that the eigenvalues of A have negative real parts. In this situation there is an explicit formula for the solution of (2),

$$X = \int_0^\infty e^{\tau A} BB^T e^{\tau A^T} d\tau \tag{4}$$

known as the controllability Grammian of (1).

Much of the theory in this paper will be established by using the above formula. The main observation which we will prove in the next section, is that there are accurate approximations to the system (2) *of small rank*. Extracting such approximations will be the subject of sections 3 and 4.

2 Low rank solutions to the Lyapunov equation

The purpose of this section is to identify the types of approximations that will be used in this paper. We will restrict ourselves to the case corresponding to the single input situation in (1), i.e., the equation (2) becomes

$$AX + XA^T + bb^T = 0 \tag{5}$$

where b is a single column vector, and its explicit solution is given by

$$X = \int_0^\infty e^{\tau A} bb^T e^{\tau A^T} d\tau. \tag{6}$$

One way to integrate formula (6), would be to start by replacing (6) by its approximation

$$X(s) = \int_0^s e^{\tau A} bb^T e^{\tau A^T} d\tau \tag{7}$$

where s is selected so that the error in the approximation is small enough. Note that we must integrate in the interval $[0, s]$ a function of the form $w(t)w(t)^T$ where $w(t)$ is the vector function $w(t) = e^{tA}b$. Assume that we approximate $w(t)$ in the interval $[0, s]$ as $w(t) \approx w_m(t) = q_m(At)b$, where q_m is a polynomial of degree $m-1$. The resulting approximation for $X(s)$ is

$$X_m(s) = \int_0^s w_m(\tau) w_m(\tau)^T d\tau \tag{8}$$

This approximation can be made arbitrarily accurate by increasing the degree of q_m. Thus, by choosing a large enough s and and a large enough degree for q, we can make the approximation $X_m(s)$ to X as accurate as desired. Let $V_m = [b, Ab, \ldots, A^{m-1}b]$ and $q_m(t) = \alpha_0 + \alpha_1 t + ... + \alpha_{m-1} t^{m-1}$. Then $w_m(t)$ can be written as $w_m(t) = V_m z_m(t)$, $z_m(t) = [\alpha_0, \alpha_1 t, ..., \alpha_{m-1} t^{m-1}]^T$ and therefore

$$X_m(s) = V_m \left(\int_0^s z_m(\tau) z_m(\tau)^T d\tau \right) V_m^T \equiv V_m G_m V_m^T \tag{9}$$

where G_m is an $m \times m$ matrix whose entries can easily be shown to be equal to

$$g_{ij}^{(m)} = \frac{\alpha_{i-1}\alpha_{j-1}s^{i+j-1}}{i+j-1}. \tag{10}$$

Throughout the paper we will exploit approximations to the solution X which are of the form

$$X_m = V_m G_m V_m^T. \tag{11}$$

where $V_m = [v_1, v_2, ..., v_m]$ is a fixed set of vectors and G_m is an arbitrary $m \times m$ matrix. The set of all such matrices is clearly a subspace of the space of $N \times N$ matrices. The range of any matrix in this subspace is included in the subspace $K = span\{V_m\}$ while its kernel contains the orthogonal to K. The subspace of the matrices (11) is in fact uniquely defined by the range K of these matrices. Thus, we will denote by $\mathbf{Z}_m(K)$ the space of all matrices of the form (11), or $\mathbf{Z}_m$ if there is no ambiguity, where V_m is a basis of the subspace K. If we restrict the matrices to be symmetric, then $\mathbf{Z}_m$ is of dimension $(m(m+1))/2$. If one wants to deal with the more general situation where the right-hand-side is of a more general form bv^T rather than bb^T then G can no longer restricted to be symmetric and the dimension of $\mathbf{Z}_m$ is m^2.

What we have just shown above is that there are approximations from $\mathbf{Z}_m(K_m)$ where K_m is the Krylov subspace

$$K_m = span\{b, Ab, ..., A^{m-1}b\} \tag{12}$$

that will converge to X as m tends to infinity. Here we should mention that, rigorously speaking, this statement is trivial in the finite dimensional case since there is always an exact solution of the form (11) for $m \geq N$. However, 'convergence' is meant in the infinite dimensional context, in the same way that one speaks of the convergence of the conjugate gradient method which is known to be a finite process.

If we can find a good approximation to $w(t)$ in the interval $[0, s]$, by a low degree polynomial, we should be able to find a good approximation to X from $\mathbf{Z}_m$. However, the above formulas should not be used as a practical procedure since they are likely to be highly unstable. Section 4 presents a process that is mathematically (but not algorithmically) equivalent to the procedure outlined above with a specific choice for q_m. In the next section we exploit well-known numerical quadrature formulas to derive particular subspaces K and their corresponding approximations.

3 Use of numerical quadrature

The first procedure that we propose for approximating X is based upon evaluating the integral (6) by standard numerical quadrature formulas. We will describe two such procedures. First, assume that s has been chosen and that we can evaluate $w(t)$ at m equally spaced points

$$t_i = (i-1)\frac{s}{m-1}, i = 1, 2, ..., m \tag{13}$$

then an appropriate quadrature formula for evaluating (6) would yield,

$$X_m = \sum_{i=1}^{m} \delta_i w(t_i) w(t_i)^T \tag{14}$$

where the δ_i's are the quadrature coefficients. In other words,

$$X_m = W_m \Delta W_m^T \tag{15}$$

in which W_m is the $N \times m$ matrix

$$W_m = [w(t_1), w(t_2), ..., w(t_m)]$$

and

$$\Delta = Diag(\delta_1, ..., \delta_m).$$

Note that $w(t_1) = b$. In order to use the formula (14) we also need to compute and save $w(t_i), i = 2..., m$. One way in which this can be done is by solving the system of Ordinary Differential Equations

$$\dot{w} = Aw \tag{16}$$
$$w(0) = b \tag{17}$$

by any technique and save the vectors $w(t_i), i = 1, 2, .., m$ at the points t_i. Alternatively, one may compute $w(t_{i+1})$ from $w(t_i)$ by $w(t_{i+1}) = e^{A\Delta t_i} w(t_i)$, where $\Delta t_i = t_{i+1} - t_i$, in a number of efficient ways as has been recently suggested in [4]. The problem considered in [4] is to approximate the product of the exponential of a matrix times a vector. We will describe one such approach in some detail in Section 4.

Thus, the class of algorithms based on numerical quadrature would begin by choosing a quadrature rule and performing the following procedure.

Algorithm 1

(1) Start: Define $w(t_1) = b$;

(2) For $i = 2, ..., m$ do:

- Compute $w(t_i)$ from $w(t_{i-1})$;
- Save $w(t_i)$ in the i-th column of W_m.

It is clear that the actual matrix X_m should never be computed explicitly, since it is a dense $N \times N$ matrix in general. One only needs to store W_m and Δ_m. In fact, it is worth pointing out that one may not even need to store W_m if the original problem is to approximate Xv where v is some vector, rather than to compute an approximation to the matrix X itself. In this situation the approximation $x = X_m v$ can be accumulated as $x := x + (\delta_i w(t_i)^T v) w(t_i)$ every time that a new $w(t_i)$ is generated and one can discard the old $w(t_i)$'s.

The question that we would like to address next is which integration schemes to use and how to implement them. First, a restriction of the scheme is that the weights δ_i should be positive, since we know that the solution is semi-positive definite matrix. For example, this is not satisfied by all the Newton-Cotes formulas but it is true for the Gauss-Legendre formulas, see e.g., [6], pp. 145. We have implemented and tested two basic classes of quadrature formulas.

1. Closed Newton-Cotes formulas with 3 points (Simpson's rule), 5 points, and 7 points.

2. Gauss-Laguerre formulas with 9 points and 15 points.

Newton-Cotes formulas are fairly standard and the corresponding coefficients can be found in any elementary numerical analysis textbook. These use equally spaced points which may be an undesirable feature in our case because of the exponential behavior of the function to integrate. At the beginning of the interval $w(t)$ varies far more than for large t. Therefore it is natural to expand the intervals of integration as we proceed. One procedure we have experimented with is to double the size of

the intervals $[x_i, x_{i+k}]$ on which the Newton-Cotes formula is based at every time. For the 7-point Newton-Cotes formula we triple the size of the interval. This simple modification does provide better results than keeping a fixed interval.

The Gauss-Laguerre quadrature formula is a highly accurate scheme to evaluate integrals of functions of the form $e^{-t}g(t)$ on the infinite half line. The basic formula is,

$$\int_0^\infty e^{-\tau} g(\tau) d\tau \approx \sum_{i=1}^{m} \omega_i g(t_i) \tag{18}$$

where the t_i are the roots of the Laguerre polynomial of degree m and where the ω_i's are the quadrature weights. For general functions the formula is used as follows,

$$\int_0^\infty g(\tau) d\tau = \int_0^\infty e^{-\tau}(e^{\tau} g(\tau)) d\tau \approx \sum_{i=1}^{m} \omega_i e^{t_i} g(t_i) \equiv \sum_{i=1}^{m} \delta_i g(t_i). \tag{19}$$

The roots t_i and the coefficients δ_i for our numerical tests have been taken from [1].

An aspect which we found to be critical to the performance of the Gauss-Laguerre integration is a proper scaling of the variable τ. For some problems, such as the one in the numerical experiments section, the norm of A can be very large and as a result the vector $w(t_i)$ can be tiny for most of the t_i's, thus leading to a poor evaluation of the integral (6). For example assume that A is symmetric negative definite having $\lambda_1 = -10$ as its eigenvalue closest to zero, and consider Gauss Laguerre integration with 15 points. Then for the second root $t_2 = 1.21559..$ all the eigencomponents of $w(t_2)$ will be no larger than $e^{-12.1559..} \approx 10^{-06}$. For t_3 the components do not exceed $e^{-22.6694} \approx 1.38610^{-10}$, and beyond t_3 the $w(t_i)$'s become too small. A remedy is to use a change of variable $\xi = \alpha\tau$ to ensure that the value of $w(t)$ at the last root t_m is not too small or too large. For example, if an estimate of the rightmost eigenvalue λ_1 of A is available, we might choose α so that $|e^{\alpha t_m \lambda_1}| = tol$, where tol is some prescribed tolerance.

4 A Krylov subspace technique

In this section we present a method that is based on a global approximation to $w(t)$ in $[0, +\infty)$. Explicit integration of this approximation will then yields an approximation to X. This, as will be seen, is equivalent to a projection process of Galerkin-type.

The first question that we address is how to approximate $e^A b$ for a given vector b. This was considered in [4] where polynomial approximation to the exponential was used. The approximation to $e^A b$ is taken of the form

$$e^A b \approx p_{m-1}(A) b \tag{20}$$

where p_{m-1} is a polynomial of degree $m-1$.

Clearly, one can use other types of approximations, e.g., rational, which are usually more accurate. The attraction of polynomials is the fact that they do not require solving linear systems. One way in which a good polynomial can be found is by attempting to minimize some norm of the error $e^z - p_{m-1}(z)$ on a continuum in the complex plane that encloses the spectrum of A. For example, Chebyshev approximation can be used. The disadvantage of this approach is that it requires some approximation to the spectrum of A. In [4] we considered a technique based on Arnoldi's method which does not require any eigenvalue information. This technique is now summarized.

The approximation (20) to $e^A b$ is an element of the Krylov subspace (12). In this approach we need to generate an orthonormal basis $V_m = [v_1, v_2, v_3, \ldots, v_m]$ of K_m via the well-known Arnoldi algorithm starting with $v_1 = b/\|b\|_2$.

Algorithm 2: Arnoldi

1. *Initialize:* Compute $v_1 := b/\|b\|_2$.
2. *Iterate:* Do $j = 1, 2, ..., m$
    1. Compute $w := Av_j$
    2. Compute a set of j coefficients h_{ij} so that $w := w - \sum_{i=1}^{j} h_{ij} v_i$ is orthogonal to all previous v_i's.
    3. Compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$.

By construction the above algorithm produces an orthonormal basis

$$V_m = [v_1, \ldots, v_m],$$

of the Krylov subspace K_m. If we denote the $m \times m$ upper Hessenberg matrix consisting of the coefficients h_{ij} computed by the algorithm by H_m we have the relation

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T \tag{21}$$

from which we get $H_m = V_m^T A V_m$. Therefore H_m represents the projection of the linear transformation A onto the subspace K_m, with respect to the basis V_m. As is well-known in the particular case where A is symmetric, Arnoldi's algorithm simplifies to the Lanczos process in which case H_m becomes tridiagonal symmetric.

We can write the desired approximation to $x = e^A b$ as $x_m = p_m(A)v$ or equivalently $x_m = V_m y$ where y is an m-vector. In [4], the choice $y = \beta e^{H_m} e_1$ with $\beta = \|b\|_2$ was suggested, leading to the following formula for arbitrary t,

$$e^{tA} b \approx \beta V_m e^{tH_m} e_1 \tag{22}$$

The quality of this approximation was also analyzed in [4] and the following result was shown.

Theorem 4.1 *Let A be any square matrix and let $\rho = \|A\|_2$. Then the error of the approximation (22) is such that*

$$\|e^{tA} b - \beta V_m e^{tH_m} e_1\|_2 \leq 2\beta \frac{(t\rho)^m e^{t\rho}}{m!}. \tag{23}$$

Experiments reported in [4], reveal that this approximation can be very accurate even for moderate values of the degree m. The theorem shows convergence of the approximation (22) for fixed t, as m increases to infinity. However, note that the above approximation is exact when $m = N$, see [4].

Let us now substitute the expression (22) in (6). We obtain the approximation

$$X_m = V_m \left(\int_0^\infty e^{\tau H_m} (\beta e_1)(\beta e_1)^T e^{\tau H_m^T} d\tau \right) V_m^T \equiv V_m G_m V_m^T. \tag{24}$$

Assuming that H_m has eigenvalues with negative real parts, we note that G_m is the solution of the $m \times m$ Lyapunov equation

$$H_m G_m + G_m H_m^T + \beta^2 e_1 e_1^T = 0 \tag{25}$$

In other words, modulo the polynomial approximation made on the exponential of A, we have reduced the original problem into one of dimension m. This raises the question as to whether or not the process just described is mathematically equivalent to a projection-type method such as a Galerkin process.

To see that this is the case we need to define the subspace of approximants and the inner product. The subspace of approximants will be the subspace $\mathbf{Z}_m(K_m)$ as defined in Section 2. This is the same as the subspace of all matrices of size N of the form $V_m G V_m^T$ where V_m is the orthogonal basis of K_m as constructed from the Arnoldi process. Recall that from our definition V_m *is fixed* and therefore the actual variable is the $m \times m$ matrix G. For the inner product we take the usual inner product in R^{N^2}, i.e., the inner product of two matrices taken as vectors of N^2 elements. It is a simple exercise to show that this can also be defined by

$$< X, Y >= tr(XY^T) \tag{26}$$

The Galerkin condition defines an approximation $\tilde{X}$ by stipulating that $\tilde{X}$ belong to $\mathbf{Z}_m$ and that the residual $R(\tilde{X}) = A\tilde{X} + \tilde{X}A^T + bb^T$ be orthogonal to all of the subspace of approximants. This second condition gives,

$$tr[Z R^T(\tilde{X})] = 0, \quad \forall Z \ \in \mathbf{Z_m} \tag{27}$$

and therefore,

$$0 = tr[V_m G V_m^T R(\tilde{X})^T] = tr[G V_m^T R(\tilde{X})^T V_m] \ \ \forall\ G \tag{28}$$

Taking matrices G of the form $G = e_i e_j^T, i, j = 1, ..., m$ leads immediately to

$$V_m^T R(\tilde{X})^T V_m = 0. \tag{29}$$

Let us now substitute $\tilde{X} = V_m G_m V_m^T$ in (29). Remembering that V_m is orthogonal and that we have $b = \beta v_1$, where v_1 is the first column of V_m, we get

$$\begin{aligned} 0 &= V_m^T R(\tilde{X})^T V_m = V_m^T (A V_m G_m V_m^T + V_m G_m V_m^T A^T + bb^T) V_m \\ &= H_m G_m + G_m H_m^T + \beta^2 e_1 e_1^T \end{aligned} \tag{30}$$

which is exactly (25). We have therefore just proved the following result.

Theorem 4.2 *The Galerkin method applied to the Lyapunov equation (5) over the subspace* $\mathbf{Z}_m$ *is mathematically equivalent to approximating the solution* X *by evaluating the integral (6) in which* $w(\tau) = e^{\tau A} b$ *is replaced by its approximation (22).*

This theorem can allow one to establish error bounds for the approximation provided by a Galerkin-type process onto the subspace $\mathbf{Z}_m$.

5 Numerical Experiments

In this section we describe a few numerical experiments in order to test and compare the methods described in Sections 3 and 4. All the tests have been performed in double precision arithmetic on an Ardent Titan super workstation. Our example is derived from the discretization of a partial differential equation of the form:

$$\frac{\partial u}{\partial t} = \Delta u + F(x, y) g(t) \tag{31}$$

in a rectangular domain, with Dirichlet boundary conditions. Here $\Delta = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ is the Laplacean operator. If we discretize the rectangle using $n_x + 2$ points in the x direction and $n_y + 2$ points in the y direction, the above equations lead to a matrix problem of the form:

$$\dot{u} = Au + bg \tag{32}$$

m	$\|Res\|_F$	Time (sec)
5	1.10 E-04	0.18
10	5.40 E-06	0.23
15	7.92 E-07	0.35
20	1.92 E-07	0.45

Table 1: Performance of the Krylov subspace method.

where A is square of dimension $N = n_x n_y$. In this experiment we took $n_x = 20$ and $n_y = 40$ leading to a matrix of size 800. The corresponding matrix has a 1-norm of $3,528.0$. We have taken b to be simply e_1 the first column of the identity. Tests with other choices for b showed similar results. First we would like to show the behavior of the residual achieved by the Krylov subspace method described in Section 4, as the degree m varies. Table 1 shows the scaled Frobenius norm of the residual, i.e., the quantity $\|AX_m + X_m A^T + bb^T\|_F$, where $\|Z\|_F = (tr[Z^T Z]/N)^{1/2}$. This is done for $m = 5, 10, 15, 20$.

We have used the Arnoldi process instead of the Lanczos algorithm on purpose, despite the fact that the matrix is symmetric. This is in order to give an idea of the increase in time in the more general nonsymmetric case. The table indicates that the accuracy of the Krylov subspace approximation to the Lyapunov equation is good for very small m and then improves slowly. The times reported in this table and in the next one are in seconds on the Titan and have been obtained using the -O3 compiling option.

Next we would like to test the methods based on numerical integration described in Section 3. Just as in the previous test we show the residual norm and the time to compute the approximate solution for various choices of accuracy and step size. In all of the methods we used the same change of variable for scaling purposes as described at the end of Section 3: we scaled the variable t by $2.5/\|A\|_1$. To approximate $e^{A\Delta t_i} w_i$ we used the formula (20) of degree 5 for the Newton Cotes formulas and 10 for the Gauss-Laguerre formulas. In general the Newton-Cotes formulas do not perform as well as those based on Gauss-Laguerre. The disadvantage of Gauss-Laguerre formulas is that if their accuracy is not sufficient one must restart all over again since the roots of the Laguerre polynomials of different degrees are different. With the Newton Cotes formulas it suffices to refine the previous intervals and the previous work can be saved and used.

Method	m	$\|Res\|_F$	Time
Laguerre(9)	9	4.21 E-06	0.57
Laguerre(15)	15	7.08 E-08	0.92
NC-3 $\Delta t = 0.3$	6	3.47 E-04	0.23
NC-3 $\Delta t = 0.1$	11	1.17 E-04	0.30
NC-3 $\Delta t = 0.05$	16	8.53 E-05	0.44
NC-5 $\Delta t = 0.40$	9	1.59 E-04	0.26
NC-5 $\Delta t = 0.20$	12	5.97 E-05	0.35
NC-5 $\Delta t = 0.20$	16	1.45 E-05	0.41
NC-7 $\Delta t = 0.5$	7	3.76 E-04	0.23
NC-7 $\Delta t = 0.25$	13	4.96 E-05	0.36
NC-7 $\Delta t = 0.20$	18	9.60 E-06	0.46

Table 2: Behavior of the integration techniques.

6 Conclusion

We have proposed and tested two types of techniques for approximating solutions to large Lyapunov equations that arise in control problems. The attraction of the Krylov subspace method is its simplicity and its overall effectiveness. The Gauss-Laguerre formula may be particularly useful when only the product of X by a vector is desired and one cannot afford to store the vectors $w(t_i)$. The use of Newton-Cotes formulas is not recommended without incorporating a proper change of variables to improve effectiveness of the integration scheme. Although we have relied on the integral formulation of the solution which requires that the eigenvalues of A all have negative real parts, the Galerkin approach may be applied to more general situations. The theory for this is not established however. Another case which we have not addressed is the general situation of equations of the form $AX + XA^T + B = 0$, where B is may be sparse but not necessarily of small rank. A completely different approach may be required in this situation, one that might exploit any possible sparsity in the solution.

Acknowledgements.

Work herein supported was by Cooperative Agreement NCC 2-387 between the National Aeronautics and Space Administration (NASA) and the Universities Space Research Association (USRA).

The author is indebted to the referee for pointing out an important result concerning Gaussian quadrature.

References

[1] M. Abaramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* National Bureau of Standards, Washington, DC, 1964.

[2] P. Kokotovic B. H. Krogh. Feedback control of overloaded networks. *IEEE Trans. Aut. Cont.*, AC-29:704–711, 1984.

[3] M. J. Balas. Trends in large space structure control theory: fondest dreams, wildest hopes. *IEEE Trans. Aut. Contr.*, AC-2:522–535, 1982.

[4] E. Gallopoulos and Y. Saad. On the parallel solution of parabolic equations. In R. De Groot, editor, *Proceedings of the International Conference on Supercomputing 1989, Heraklion, Crete, June 5-9, 1989.* ACM press, 1989.

[5] Y. Saad. Projection and deflation methods for partial pole assignment in linear state feedback. *IEEE Trans. Aut. Cont.*, 33:290–297, 1988.

[6] J. Stoer and R. Bulirsch. *Introduction to numerical analysis.* Springer Verlag, New York, Heidelberg, 1980.