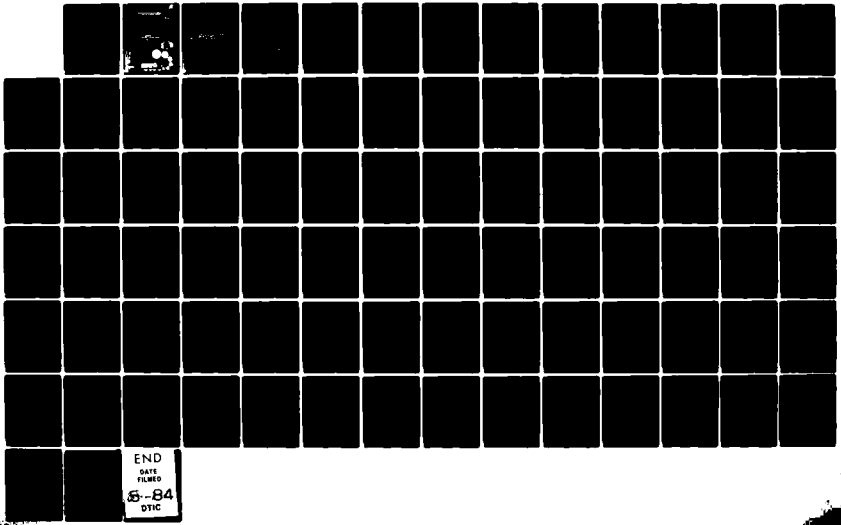
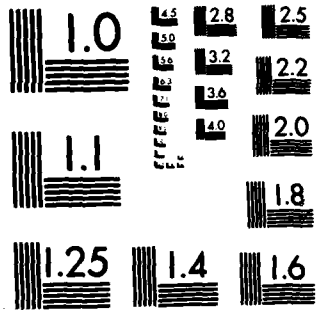
AD-A140 069 CONTRIBUTIONS TO THE COMPUTATION OF THE MATRIX 1/1
EXPONENTIAL REVISION(U) CALIFORNIA UNIV BERKELEY CENTER
FOR PURE AND APPLIED MATHEMATICS K C NG FEB 84 PAM-212
UNCLASSIFIED N00014-76-C-0013 F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A140069

CONTRIBUTIONS TO THE COMPUTATION OF THE MATRIX EXPONENTIAL

Kwok Choi Ng[†]

(revised)

UC Berkeley, February 1984

ABSTRACT

This thesis consists of two parts. First, a condition number for the exponential of a triangular matrix S is introduced. It measures how sensitive is e^S to relatively small perturbations in the elements of S. Second, a new technique *(matrix argument reduction)* for computing periodic matrix functions [illegible]scribed and discussed in detail. By applying this technique to the computation of e^S one can always reduce the problem to one in which the eigenvalues lie close to the real axis.

Accession For
NTIS GRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A-1

DTIC COPY INSPECTED 3

[†] The author gratefully acknowledges support by the office of Naval Research Contract N00014-76-C-0013.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Professor B.N. Parlett for giving his time, help and support during the writing of this thesis. Also, I would like to thank Professor W. Kahan for his many valuable comments. I am also grateful to Professor R. Fateman for reading this thesis.

TABLE OF CONTENTS

INTRODUCTION

The exponential matrix e^{tA} plays a role in several fields of mathematics and applied mathematics, particularly control theory. The possibility of computing e^{tA} was considered as soon as computers became available. Nevertheless the computation has proved harder than might be imagined for such a well behaved function as exp ! In 1976 Moler and Van Loan published a paper† with the title

"Nineteen dubious ways to compute the exponential of a matrix."

Among the approaches mentioned in the Moler-Van Loan article is the use of the Schur form. This seemed to promise a stable computation. Nevertheless there remained one weakness in this approach.

Let $S = \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix}$ be the Schur form. Clearly, if S_{11} and S_{22} have common eigenvalues than S should not be split up to facilitate the calculation. Less obvious is the fact that accuracy may also be lost when S_{11} and S_{22}, though far apart, have common eigenvalues in their exponentials. This has been the stumbling block, and the only stumbling block, to the success of Schur form techniques.

One of the contributions of this thesis is to remove this obstacle by a new technique that we call matrix argument reduction. The problem is thereby reduced to one in which the eigenvalues lie close the real axis. Consequently $\exp(S_{11}) \approx \exp(S_{22}) \iff S_{11} \approx S_{22}$. Now we have a stable procedure for computing exponentials of arbitrary square complex matrices. Just as the Singular Values Decomposition (SVD) is not always the method of choice for Least Square problems, our procedure may not always be

† Technical report 76-283. Dept. of Computer Science, Cornell University, Ithaca, New York. This appears in SIAM Review.

preferred for exponentiating matrices tA. Nevertheless, even if it is not the fastest technique, our approach furnishes a reliable procedure for tackling even the most difficult cases.

Part I.

A Measure of The Sensitivity of the Exponential of Triangular Matrices

1. Introduction

The sensitivity of e^{tA} to small changes in A is an important topic in the analysis of algorithms for solving linear systems of ODEs with constant coefficients. In finite precision arithmetic, one cannot expect to do better than to approximate the exponential of a slightly wrong matrix $A+\delta A$. Consequently, several researchers have obtained bounds on

$$\|e^{t(A+\delta A)}-e^{tA}\| \quad \text{or} \quad \frac{\|e^{t(A+\delta A)}-e^{tA}\|}{\|e^{tA}\|} \ , \tag{1.1}$$

where the matrix $A+\delta A$ is a small perturbation of A. See [3], [7] and [10].

We cannot improve on the bounds given in the above papers for a general matrix A. Our work stems from the realization that these days a preliminary reduction of A to Schur form S is a routine matter with moderate cost. Moreover, having reduced A to a triangular S, it is possible (see section 3.1) to compute e^S in floating point arithmetic to yield the exponential of some $S+\delta S$ where δS satisfies $|\delta S| \le Constant \cdot \varepsilon \cdot |S|$ (element-wise). Here ε is the arithmetic precision. Consequently, it suffices to examine the sensitivity of e^S with respect to small **relative** changes in S.

In practice, the computation of the Schur form of A will yield the *exact* Schur form of some $A+\delta A$ with a very small δA ($A+\delta A = USU^H$, where U is unitary). Our results give a realistic estimate for the difference $\|fl(e^S) - U^H e^{A+\delta A} U\|$. Of course, whether $e^{A+\delta A}$ is close to e^A depends on both A and δA. This question is not fully resolved. As a preliminary step, Appendix I.A offers an integral representation of $e^A - e^{A+\delta A}$; but it is too complicated to allow a realistic error bound in terms of the computed Schur factors.

The main result of the following sections is a bound (Theorem 3.4) on the spectral norm of the matrix $\mathrm{COND}(e^S; i,j)$, where the (p,q) element is defined by

$$[\mathrm{COND}(e^S; i,j)]_{p,q} = s_{p,q} \cdot \frac{\partial (e^S{}_{i,j})}{\partial s_{p,q}} .$$

This element is a measure of the change in the (i,j) element of e^S to a small *relative* change in $s_{p,q}$. To condense all this information to a single number, we propose the *condition number* of S *for exponentiation* to be

$$cond(S;\exp) = \frac{\| e^{(\mathrm{Re}(D)+|N|)} \|}{\| e^S \|},$$

where D is diagonal and N is strictly upper triangular and such that $S=D+N$. An alternative name is the exponential condition number of S.

2. A Representation of Functions of Triangular Matrices

2.1. Notations

Let Z denote the abscissae $Z=(\zeta_1,\zeta_2,\ldots,\zeta_n)$. We follow [5] and use $\Delta_i^k(Z)f$ for the k-th divided difference of f on $(\zeta_i,\zeta_{i+1},\ldots,\zeta_{i+k})$.

When Z has exactly $k+1$ elements, we suppress the subscript and use $\Delta^k(Z)f$ to denote the highest order (k-th) divided difference of f on Z.

For a matrix A, $|A|$ denotes the matrix all of whose elements are the absolute values of the elements of A, i.e., $|A|_{i,j}=|A_{i,j}|$. The notation $A\le B$ means that $A_{i,j}\le B_{i,j}$ for every i and j.

Finally, let $E_{i,j}^k$ be the set of multi-indices $\{\sigma=(\sigma_0,\sigma_1,\ldots,\sigma_k)$, where all σ's are integer and $i=\sigma_0<\sigma_1<\cdots<\sigma_k=j\}$. For examples,

$$E_{i,j}^1=\{(i,j)\},$$

$$E_{i,j}^2=\{(i,i+1,j),(i,i+2,j),\ldots,(i,j-1,j)\}, \text{ and so on.}$$

Notice that if $\sigma\in E_{i,j}^k$, then $i+l\le\sigma_l\le j-(k-l)$ for $l=1,2,\ldots,k-1$.

2.2. A Representation of f(S) in Terms of Divided Differences

Given f an analytic function and $S=(s_{i,j})$ an upper triangular matrix, every element of $f(S)$ can be written in terms of the exponential divided differences on eigenvalues of S. This representation (Theorem 1) can be found in Van Loan [9] (see also [6] and [8]). Here we give a different proof which is simpler than the one in [9].

For simplicity, let ζ denote an eigenvalue of S, i.e., $\zeta_i=s_{i,i}$, $i=1,2,\ldots,n$. According to [2], when f is a holomorphic function defined inside and on a

simple closed contour C in the complex plane (positively oriented), then the divided difference $\Delta_i^k(Z)f$ has the following representation:

$$\Delta_i^k(Z)f = \frac{1}{2\pi i}\int_C \frac{f(\omega)d\omega}{(\omega-\zeta_i)(\omega-\zeta_{i+1})\cdots(\omega-\zeta_{i+k})} \tag{2.2.1}$$

Another useful representation of the divided difference is Hermite-Genocchi formula (cf. [5])

$$\Delta_i^k(Z)f = \int_0^1\int_0^{\nu_1}\cdots\int_0^{\nu_{k-1}} f^{(k)}[\zeta_i+(\zeta_{i+1}-\zeta_i)\nu_1+\cdots+(\zeta_{i+k}-\zeta_{i+k-1})\nu_k]d\nu_k\cdots d\nu_1, \tag{2.2.2}$$

where $f^{(k)}$ denotes the k-th derivative of f. This formula will be used in later sections.

The key to the result is a formula for elements of the resolvent of S.

Lemma 2.1. Given z not equal to any of the eigenvalues of S, the resolvent of S, $X=(zI-S)^{-1}$, has the following representation:

$$x_{i,j} = \begin{cases} 0 & \text{if } j<i, \\ \dfrac{1}{(z-\zeta_i)} & \text{if } i=j, \\ \displaystyle\sum_{k=1}^{j-i}\sum_{\sigma\in E_{i,j}^k} \frac{s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-1},\sigma_k}}{(z-\zeta_{\sigma_0})(z-\zeta_{\sigma_1})\cdots(z-\zeta_{\sigma_k})} & \text{if } j>i\,. \end{cases} \tag{2.2.3}$$

Proof. Let $X=(x_{i,j})$ be defined according to (2.2.3) and let $C=(c_{i,j})=X\cdot(zI-S)$. We want to show that C is the identity matrix. Since S is upper triangular, it is obvious that $c_{i,j}=0$ for $j<i$ and $c_{i,i}=1$ for $i=1,\ldots,n$. To show that $c_{i,j}=0$ for $j>i$, note that for $j>i$,

$$\begin{aligned} c_{i,j} &= \sum_{m=i}^{j} x_{i,m}\cdot(zI-S)_{m,j} \\ &= x_{i,j}(z-\zeta_j) + \sum_{m=i}^{j-1} x_{i,m}\cdot(-s_{m,j})\,. \end{aligned}$$

It suffices to establish $x_{i,j}(z-\zeta_j)=\sum_{m=i}^{j-1} x_{i,m}\cdot s_{m,j}$. From section 2.1, if $\sigma\in E_{i,j}^k$, then $\sigma_k=j$ and $\sigma_0=i$. By the definition of $x_{i,j}$, we have

$$x_{i,j}(z-\zeta_j) = \sum_{k=1}^{j-i}\sum_{\sigma\in E_{i,j}^k} \frac{s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-1},\sigma_k}}{(z-\zeta_{\sigma_0})\cdots(z-\zeta_{\sigma_{k-1}})}$$

$$=\frac{1}{z-\zeta_i}\cdot s_{i,j} + \sum_{k=2}^{j-i}\sum_{\sigma\in E_{i,j}^k} \frac{s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-2},\sigma_{k-1}}}{(z-\zeta_{\sigma_0})\cdots(z-\zeta_{\sigma_{k-1}})}\cdot s_{\sigma_{k-1},\sigma_k}.$$

Set $m=\sigma_{k-1}$. Since $j-1\geq\sigma_{k-1}=m\geq i+k-1$ (see section 2.1), the right hand side becomes

$$\frac{1}{(z-\zeta_i)}\cdot s_{i,j} + \sum_{k=2}^{j-i}\sum_{m=i+k-1}^{j-1}\Big(\sum_{\sigma\in E_{i,m}^{k-1}} \frac{s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-2},m}}{(z-\zeta_{\sigma_0})\cdots(z-\zeta_{\sigma_{k-2}})(z-\zeta_m)}\Big)\cdot s_{m,j} \ ;$$

interchanging $\sum$'s, we get

$$x_{i,i}\cdot s_{i,j} + \sum_{m=i+1}^{j-1}\Big(\sum_{k=2}^{m-i+1}\sum_{\sigma\in E_{i,m}^{k-1}} \frac{s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-2},m}}{(z-\zeta_{\sigma_0})\cdots(z-\zeta_{\sigma_{k-2}})(z-\zeta_m)}\Big)\cdot s_{m,j} \ .$$

By the definition of $x_{i,m}$, the above expression is equal to

$$x_{i,j}\cdot s_{i,j} + \sum_{m=i+1}^{j-1} x_{i,m}\cdot s_{m,j}.$$

This proves the lemma. ▪

Let C be a simple closed (positively oriented) contour that encloses the eigenvalues of S. It can be shown (see [2]) that

$$f(S)=\frac{1}{2\pi i}\int_C f(z)(zI-S)^{-1}dz. \tag{2.2.4}$$

Theorem 2.2. Given S upper triangular with eigenvalues $\zeta_i=s_{i,i}$ and f analytic on some region containing $\zeta_1,\zeta_2,\ldots,\zeta_n$, we have

$$f(S)_{i,j} = \begin{cases} 0 & \text{if } j<i, \\ f(\zeta_i) & \text{if } i=j, \\ \sum_{k=1}^{j-i} \sum_{\sigma\in E_{i,j}^k} s_{\sigma_0,\sigma_1} \cdot s_{\sigma_1,\sigma_2} \cdots s_{\sigma_{k-1},\sigma_k} \cdot \Delta^k(Z_\sigma) f & \text{if } j>i \end{cases} \tag{2.2.5}$$

where $Z_\sigma=(\zeta_{\sigma_0},\zeta_{\sigma_1},\ldots,\zeta_{\sigma_k})$.

Proof. From Lemma 2.1 and (2.2.4), we have

$$f(S)_{i,j} = \frac{1}{2\pi i}\int_C f(z)\cdot x_{i,j}\,dz,$$

where $(x_{i,j})=X=(zI-S)^{-1}$. Consequently $f(S)_{i,j}=0$ when $j<i$ and $f(S)_{i,i}=f(\zeta_i)$. For $j>i$,

$$\begin{aligned} f(S)_{i,j} &= \frac{1}{2\pi i}\int_C \sum_{k=1}^{j-i} \sum_{\sigma\in E_{i,j}^k} \frac{s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-1},\sigma_k}}{(z-\zeta_{\sigma_0})\cdots(z-\zeta_{\sigma_k})}\cdot f(z)\,dz \\ &= \sum_{k=1}^{j-i} \sum_{\sigma\in E_{i,j}^k} s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-1},\sigma_k}\cdot\left(\frac{1}{2\pi i}\int_C \frac{f(z)\,dz}{(z-\zeta_{\sigma_0})(z-\zeta_{\sigma_1})\cdots(z-\zeta_{\sigma_k})}\right) \\ &= \sum_{k=1}^{j-i} \sum_{\sigma\in E_{i,j}^k} s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-1},\sigma_k}\cdot\Delta^k(Z_\sigma) f. \end{aligned}$$

This proves the theorem. ∎

The theorem has some interesting consequence.

Corollary 2.3. Let $S=D+N$ where D is diagonal and N is strictly upper triangular. For $j\geq i$ and $t\geq 0$,

$$|f(tS)_{i,j}| \leq (e^{t|N|})_{i,j}\cdot\max_{\xi\in\bar{\Omega}}|f^{(j-i)}(t\xi)|, \tag{2.2.6}$$

where $\bar{\Omega}$ is the convex hull of $\zeta_i,\ldots,\zeta_j$.

Proof. We first prove that (cf. [6]), for $k>0$,

$$f(S)_{i,j} = \begin{cases} 0 & \text{if } j<i, \\ f(\zeta_i) & \text{if } i=j, \\ \sum_{k=1}^{j-i} \sum_{\sigma\in E_{i,j}^k} s_{\sigma_0,\sigma_1}\cdot s_{\sigma_1,\sigma_2}\cdots s_{\sigma_{k-1},\sigma_k}\cdot \Delta^k(Z_\sigma)f & \text{if } j>i \end{cases} \tag{2.2.5}$$

where $Z_\sigma=(\zeta_{\sigma_0},\zeta_{\sigma_1},\ldots,\zeta_{\sigma_k})$.

Proof. From Lemma 2.1 and (2.2.4), we have

$$f(S)_{i,j} = \frac{1}{2\pi i}\int_C f(z)\cdot x_{i,j}\,dz,$$

where $(x_{i,j})=X=(zI-S)^{-1}$. Consequently $f(S)_{i,j}=0$ when $j<i$ and $f(S)_{i,i}=f(\zeta_i)$. For $j>i$,

$$\begin{aligned} f(S)_{i,j} &= \frac{1}{2\pi i}\int_C \sum_{k=1}^{j-i}\sum_{\sigma\in E_{i,j}^k} \frac{s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-1},\sigma_k}}{(z-\zeta_{\sigma_0})\cdots(z-\zeta_{\sigma_k})}\cdot f(z)\,dz \\ &= \sum_{k=1}^{j-i}\sum_{\sigma\in E_{i,j}^k} s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-1},\sigma_k}\cdot\left(\frac{1}{2\pi i}\int_C \frac{f(z)\,dz}{(z-\zeta_{\sigma_0})(z-\zeta_{\sigma_1})\cdots(z-\zeta_{\sigma_k})}\right) \\ &= \sum_{k=1}^{j-i}\sum_{\sigma\in E_{i,j}^k} s_{\sigma_0,\sigma_1}\cdots s_{\sigma_{k-1},\sigma_k}\cdot\Delta^k(Z_\sigma)f. \end{aligned}$$

This proves the theorem. ∎

The theorem has some interesting consequence.

Corollary 2.3. Let $S=D+N$ where D is diagonal and N is strictly upper triangular. For $j\geq i$ and $t\geq 0$,

$$|f(tS)_{i,j}| \leq (e^{t|N|})_{i,j}\cdot\max_{\xi\in\bar{\Omega}}|f^{(j-i)}(t\xi)|, \tag{2.2.6}$$

where $\bar{\Omega}$ is the convex hull of $\zeta_i,\ldots,\zeta_j$.

Proof. We first prove that (cf. [6]), for $k>0$,

$$(N^k)_{i,j} = \begin{cases} \sum_{\sigma \in E^k_{i,j}} s_{\sigma_0,\sigma_1} \cdot s_{\sigma_1,\sigma_2} \cdots s_{\sigma_{k-1},\sigma_k} & \text{if } j-i \geq k, \\ 0 & \text{otherwise}. \end{cases} \tag{2.2.7}$$

It is obvious that (2.2.7) is true when $k=1$ and that $(N^k)_{i,j}=0$ whenever $j<i+k$. Assume it is true for some $k \geq 1$, then for $j \geq i+k+1$

$$\begin{aligned} (N^{k+1})_{i,j} &= (N^k \cdot N)_{i,j} \\ &= \sum_{m=i+k}^{j-1} (N^k)_{i,m} \cdot N_{m,j} \\ &= \sum_{m=i+k}^{j-1} \Big(\sum_{\sigma \in E^k_{j,m}} s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},m} \Big) \cdot s_{m,j} \\ &= \sum_{\sigma \in E^{k+1}_{i,j}} s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},\sigma_k} \cdot s_{\sigma_k,j} \ . \end{aligned}$$

By the principle of induction, (2.2.7) is true for all k.

Now insert the inequality (see [5], section 1)

$$|\Delta^k(Z_\sigma)f| \leq \frac{1}{k!} \max_{\xi \in \Pi} |f^{(k)}(\xi)|, \text{ for all } \sigma \in E^k_{i,j},$$

into Theorem 2.2. For $j>i$,

$$|f(tS)_{i,j}| \leq p_{i,j}(t) \cdot \max_{\xi \in \Pi} |f^{(k)}(\xi)| \tag{2.2.8}$$

where

$$p_{i,j}(t) = \sum_{k=1}^{j-1} \Big(\frac{1}{k!} \cdot \sum_{\sigma \in E^k_{i,j}} |s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},\sigma_k}| \Big) \cdot t^k. \tag{2.2.9}$$

Because of (2.2.7), the polynomial in (2.2.9) can be written as

$$\begin{aligned} p_{i,j}(t) &= \sum_{k=1}^{j-1} \Big(\frac{1}{k!} |N|^k \Big)_{i,j} \cdot t^k \\ &= (e^{t|N|})_{i,j} \ . \end{aligned} \tag{2.2.10}$$

When $i=j$, both sides of (2 [illegible] equal; therefore, together with (2.2.8) and

(2.2.10), we have for $j \geq i$

$$|f(tS)_{i,j}| \leq (e^{t|N|})_{i,j} \cdot \max_{\xi \in [0]} |f^{(j-i)}(\xi)|. \quad \blacksquare$$

When f is specialized to exp we obtain the result we need in the sequel.

Corollary 2.4. Given $S=D+N$ as in Corollary 2.3, we have

$$|e^S| \leq e^{\mathrm{Re}(D)+|N|}. \tag{2.2.11}$$

Proof. Applying Theorem 2.2 to f =exp we get for $j>i$

$$|(e^S)_{i,j}| \leq \sum_{k=1}^{j-i} \sum_{\sigma \in E^k_{i,j}} |s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},\sigma_k}| \cdot |\Delta^k(Z_\sigma)\exp|.$$

Since

$$|\Delta^k(Z)\exp| \leq \Delta^k(\mathrm{Re}Z)\exp \tag{2.2.12}$$

(the reader can prove it directly from (2.2.2), or see [5]), we have

$$\begin{aligned} |e^S|_{i,j} &\leq \sum_{k=1}^{j-i} \sum_{\sigma \in E^k_{i,j}} |s_{\sigma_0,\sigma_1}| \cdots |s_{\sigma_k,\sigma_k}| \cdot \Delta^k(\mathrm{Re}Z_\sigma)\exp \\ &= (e^{(\mathrm{Re}(D)+|N|)})_{i,j}. \quad \blacksquare \end{aligned}$$

Remark. Corollary 2.3 is not new. It may be found in [9], though the proof is different. Corollary 2.4 is a sharpening of a result in [9], which asserts

$$|e^S| \leq e^{\alpha(S)} \cdot e^{|N|}$$

where $\alpha(S)=\max_i \mathrm{Re}\zeta_i$. Corollary 2.4 may be proved by using certain differential inequalities, see Appendix I.B.

3. A Condition Number for The Triangular Matrix Exponential

3.1. The Sensitivity of $(e^S)_{i,j}$

Let us define the matrix $\Gamma(S)=\mathrm{Re}(D)+|N|$ where D is diagonal and N is strictly upper triangular such that $S=D+N$. We shall examine the relation between the quantity $(e^{\Gamma(S)})_{i,j}$ and the change in $(e^S)_{i,j}$ under a small *relative* perturbation in S.

Given any function $g(x)$ and a small relative perturbation $(1+\delta)x$ of x, a good measure of the rate of change in $g(x)$ with respect to x relatively is

$$\lim_{\delta\to 0}\frac{g(x(1+\delta))-g(x)}{\delta} = x\cdot g'(x). \tag{3.1.2}$$

If we consider $(e^S)_{i,j}$ as a function of $s_{i,i}, s_{i,i+1}, \ldots, s_{j,j}$, then the corresponding measure of the rate of change in $(e^S)_{i,j}$ with respect to a particular element $s_{p,q}$ $(i\le p,q\le j)$ is

$$s_{p,q}\cdot\frac{\partial(e^S)_{i,j}}{\partial s_{p,q}}.$$

We may call the above measure the *sensitivity* of $(e^S)_{i,j}$ with respect to $s_{p,q}$. Associated with each $(e^S)_{i,j}$ is a matrix of sensitivities,

$$\mathrm{COND}((e^S)_{i,j}) = \begin{bmatrix} s_{1,1}\cdot\frac{\partial(e^S)_{i,j}}{\partial s_{1,1}} & s_{1,2}\cdot\frac{\partial(e^S)_{i,j}}{\partial s_{1,2}} & \cdots & s_{1,n}\cdot\frac{\partial(e^S)_{i,j}}{\partial s_{1,n}} \\ & s_{2,2}\cdot\frac{\partial(e^S)_{i,j}}{\partial s_{2,2}} & \cdots & \vdots \\ & & \ddots & \vdots \\ & & & s_{n,n}\cdot\frac{\partial(e^S)_{i,j}}{\partial s_{n,n}} \end{bmatrix}.$$

Our objective is to find a bound on $\|\mathrm{COND}((e^S)_{i,j})\|$ (see Theorem 3.4).

We shall use the following notations.

$$||S|| = (\text{max. eigenvalue of } S^H S)^{\frac{1}{2}} \quad \text{(spectral norm)},$$

$$||S||_1 = \max_i \sum_j |s_{i,j}| \quad \text{(1-norm)},$$

$$||S||_\infty = \max_j \sum_i |s_{i,j}| \quad (\infty\text{ - norm}),$$

$$G \times H = \{(g,h): g \in G, h \in H\} \ .$$

Recall $E^k_{i,j}$ is the set of multi-indices $\{\sigma: \sigma=(\sigma_0,\sigma_1,\ldots,\sigma_k),$ where $i=\sigma_0 \le \sigma_1 \le \cdots \le \sigma_k = j\}$. Let $E_{i,j} = \bigcup_{k=1}^{j-i} E^k_{i,j}$. Also recall $Z_\sigma = (\zeta_{\sigma_0},\ldots,\zeta_{\sigma_k})$ where $\zeta_k = s_{k,k}$, $k=1,\ldots,n$. By Theorem 2.2 $(e^S)_{i,j}$ has the following representation

$$(e^S)_{i,j} = \sum_{k=1}^{j-i} \sum_{\sigma \in E^k_{i,j}} s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},\sigma_k} \cdot \Delta^k(Z_\sigma)\exp. \tag{3.1.4}$$

Note that $(e^{\Gamma(S)})_{i,j}$ has the similar representation

$$(e^{\Gamma(S)})_{i,j} = \sum_{k=1}^{j-i} \sum_{\sigma \in E^k_{i,j}} |s_{\sigma_0,\sigma_1}| \cdots |s_{\sigma_{k-1},\sigma_k}| \cdot \Delta^k(\mathrm{Re} Z_\sigma)\exp.$$

We begin by proving the following two lemmas.

Lemma 3.1. For any integer m between i and $j-1$, let $R = [i,m] \times [m+1,j]$, i.e., $(p,q) \in R$ if and only if $i \le p \le m$, $m+1 \le q \le j$. Then

$$\sum_{(p,q)\in R} \left| s_{p,q} \cdot \frac{\partial (e^S)_{i,j}}{\partial s_{p,q}} \right| \le (e^{\Gamma(S)})_{i,j}. \tag{3.1.5}$$

Proof. Let $X_{p,q}$ denote all σ in $E_{i,j}$ such that for some l, $\sigma_l = p$ and $\sigma_{l+1} = q$. Since m lies between i and $j-1$, for any σ in $E_{i,j}$ there must be some σ_l and σ_{l+1} such that $\sigma_l \le m < \sigma_{l+1}$. Thus

$$E_{i,j} = \bigcup_{(p,q)\in R} X_{p,q}.$$

However, it is not difficult to see that for (p,q), $(r,s) \in R$,

$$X_{p,q} \cap X_{r,s} = \phi \qquad \text{if } (p,q) \neq (r,s).$$

So, $E_{i,j}$ is a disjoint union of $X_{p,q}$, $(p,q) \in R$. Since for $\sigma \in E_{i,j}$

$$s_{r,s} \frac{\partial}{\partial s_{r,s}} (s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},\sigma_k} \cdot \Delta^k(Z_\sigma)) = \begin{cases} s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},\sigma_k} \cdot \Delta^k(Z_\sigma) & \text{if } \sigma \in X_{r,s} \\ 0 & \text{otherwise,} \end{cases}$$

we have, using the representation of $(e^S)_{i,j}$,

$$\begin{aligned} \sum_{(p,q)\in R} \left| s_{p,q} \cdot \frac{\partial (e^S)_{i,j}}{\partial s_{p,q}} \right| &= \sum_{(p,q)\in R} \left| s_{p,q} \cdot \frac{\partial}{\partial s_{p,q}} \sum_{\sigma \in E_{i,j}} (s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},\sigma_k} \cdot \Delta^k(Z_\sigma)) \right| \\ &= \sum_{(p,q)\in R} \sum_{\sigma \in X_{p,q}} \left| (s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},\sigma_k} \cdot \Delta^k(Z_\sigma)) \right| \\ &= \sum_{\sigma \in E_{i,j}} \left| (s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},\sigma_k} \cdot \Delta^k(Z_\sigma)) \right|. \end{aligned}$$

Therefore, by (2.2.12)

$$\begin{aligned} \sum_{(p,q)\in R} \left| s_{p,q} \cdot \frac{\partial (e^S)_{i,j}}{\partial s_{p,q}} \right| &\leq \sum_{\sigma \in E_{i,j}} |s_{\sigma_0,\sigma_1}| \cdots |s_{\sigma_{k-1},\sigma_k}| \cdot \Delta^k(\mathrm{Re} Z_\sigma) \\ &= (e^{\Gamma(S)})_{i,j}. \quad \blacksquare \end{aligned}$$

Lemma 3.2. Recall $\zeta_m = s_{m,m}$ is the eigenvalue of S, $1 \leq m \leq n$. Then

$$\sum_{m=1}^{n} \left| \zeta_m \cdot \frac{\partial (e^S)_{i,j}}{\partial \zeta_m} \right| \leq \max_{i \leq p \leq j} |\zeta_p| \cdot (e^{\Gamma(S)})_{i,j}.$$

To prove this lemma, we need the following key result which brings in $\mathrm{Re}\zeta_i$ in place of ζ_i.

Lemma 3.3. For any $\sigma \in E^k_{i,j}$

$$\sum_{m=1}^{n} \left| \frac{\partial}{\partial \zeta_m} \Delta^k(Z_\sigma) \right| \leq \Delta^k(\mathrm{Re} Z_\sigma).$$

Proof of Lemma 3.3. For simplicity, we write Δ^k for $\Delta^k(Z_\sigma)$ whenever σ is fixed. From the Hermite-Genocchi formula (2.2.2), we have

$$\Delta^k = \int_0^1\int_0^{\nu_1}\cdots\int_0^{\nu_{k-1}} \exp[\zeta_{\sigma_0}+(\zeta_{\sigma_1}-\zeta_{\sigma_0})\nu_1+\cdots+(\zeta_{\sigma_k}-\zeta_{\sigma_{k-1}})\nu_k]d\nu_k\cdots d\nu_1.$$

Take the partial derivative of Δ^k with respect to ζ_{σ_p} and set $\nu_0=1$, $\nu_{k+1}=0$ to find that

$$\frac{\partial\Delta^k}{\partial\zeta_{\sigma_p}} = \int_0^1\cdots\int_0^{\nu_{k-1}} (\nu_p-\nu_{p+1})\cdot\exp[\zeta_{\sigma_0}+\sum_{j=1}^{k}(\zeta_{\sigma_j}-\zeta_{\sigma_{j-1}})\nu_j]d\nu_k\cdots d\nu_1.$$

Note that $1\geq\nu_1\geq\nu_2\geq\cdots\geq\nu_k\geq 0$; so $(\nu_p-\nu_{p+1})\geq 0$ and $\sum_{p=0}^{k}(\nu_p-\nu_{p+1})=1$. Hence

$$\begin{aligned}\sum_{p=0}^{k}\left|\frac{\partial\Delta^k}{\partial\zeta_{\sigma_p}}\right| &\leq \int_0^1\cdots\int_0^{\nu_{k-1}}\sum_p(\nu_p-\nu_{p+1})\cdot\left|\exp[\zeta_{\sigma_0}+\sum_{j=1}^{k}(\zeta_{\sigma_j}-\zeta_{\sigma_{j-1}})\nu_j]\right|d\nu_k\cdots d\nu_1\\ &= \int_0^1\cdots\int_0^{\nu_{k-1}}\exp[\mathrm{Re}\zeta_{\sigma_0}+\sum_{j=1}^{k}(\mathrm{Re}\zeta_{\sigma_j}-\mathrm{Re}\zeta_{\sigma_{j-1}})\nu_j]d\nu_k\cdots d\nu_1\\ &= \Delta^k(\mathrm{Re}Z_\sigma).\end{aligned}$$

To complete the proof, notice that $\frac{\partial\Delta^k}{\partial\zeta_m} = 0$ for $\sigma_m\neq\zeta_{\sigma_p}$, $0\leq p\leq k$, so

$$\sum_{m=1}^{n}\left|\frac{\partial\Delta^k}{\partial\zeta_m}\right| = \sum_{p=0}^{k}\left|\frac{\partial}{\partial\zeta_p}\Delta^k(Z_\sigma)\right| \leq \Delta^k(\mathrm{Re}Z_\sigma). \quad \blacksquare$$

Proof of Lemma 3.2. Using the representation of $(e^S)_{i,j}$ (3.1.4), we have

$$\begin{aligned}\sum_{m=1}^{n}\left|\zeta_m\cdot\frac{\partial(e^S)_{i,j}}{\partial\zeta_m}\right| &\leq \max_{i\leq p\leq j}|\zeta_p|\cdot\sum_{m=1}^{n}\left|\frac{\partial(e^S)_{i,j}}{\partial\zeta_m}\right|\\ &= \max_{i\leq p\leq j}|\zeta_p|\cdot\sum_{\sigma\in E_{i,j}}\left[|s_{\sigma_0\sigma_1}\cdots s_{\sigma_{k-1}\sigma_k}|\cdot\sum_{m=1}^{n}\left|\frac{\partial}{\partial\zeta_m}\Delta^k(Z_\sigma)\right|\right].\end{aligned}$$

By Lemma 3.3, it is less than

$$\max_{i\leq p\leq j}|\zeta_p|\cdot\sum_{\sigma\in E_{i,j}}|s_{\sigma_0\sigma_1}|\cdots|s_{\sigma_{k-1}\sigma_k}|\cdot\Delta^k(\mathrm{Re}Z_\sigma).$$

$$= \max_{i \le p \le j} |\zeta_p| \cdot (e^{\Gamma(S)})_{i,j}. \quad \blacksquare$$

Lemma 3.1 and Lemma 3.2 together yield the following theorem .

Theorem 3.4. The sensitivity matrix $\mathrm{COND}((e^S)_{i,j})$ of each element $(e^S)_{i,j}$ of e^S satisfies

$$||\mathrm{COND}((e^S)_{i,j})|| \le (1 + \max_{i \le l \le j} |\zeta_l|) \cdot (e^{\Gamma(S)})_{i,j}. \tag{3.1.6}$$

Proof. Since $\dfrac{\partial (e^S)_{i,j}}{\partial s_{p,q}} = 0$ for $p<i$ or $q>j$,

$$\begin{aligned} ||\mathrm{COND}((e^S)_{i,j})||_1 &= \max_{1 \le m \le n} \sum_{l=1}^{m} |s_{l,m} \cdot \frac{\partial (e^S)_{i,j}}{\partial s_{l,m}}| \\ &\le \sum_{l=i}^{j} |\zeta_l \cdot \frac{\partial (e^S)_{i,j}}{\partial \zeta_l}| + \sum_{(p,q) \in R} |s_{p,q} \cdot \frac{\partial (e^S)_{i,j}}{\partial s_{p,q}}| \end{aligned}$$

where $R=[i,m]\times[m+1,j]$. Hence, Lemma 3.1 and Lemma 3.2 imply

$$||\mathrm{COND}((e^S)_{i,j})||_1 \le (1 + \max_{i \le l \le j} |\zeta_l|) \cdot (e^{\Gamma(S)})_{i,j}.$$

Similarly

$$||\mathrm{COND}((e^S)_{i,j})||_\infty \le (1 + \max_{i \le l \le j} |\zeta_l|) \cdot (e^{\Gamma(S)})_{i,j}.$$

Since

$$\begin{aligned} ||B||^2 &= \text{max. eigenvalue of } (B^H B) \\ &\le ||B^H||_1 \cdot ||B||_1 \\ &= ||B||_\infty \cdot ||B||_1, \end{aligned}$$

the theorem follows. $\blacksquare$

The theorem and lemmas in this section suggest that $(e^{\Gamma(S)})_{i,j}$ is the essential quantity that measures the norm of the matrix $\mathrm{COND}((e^S)_{i,j})$. In fact, if we allow no perturbation in the diagonals ζ_l's (i.e., replacing the diagonal of $\mathrm{COND}((e^S)_{i,j})$ by zeros), then $||\mathrm{COND}((e^S)_{i,j})|| \le (e^{\Gamma(S)})_{i,j}$. The factor

$\max_i |\zeta_i|$ in (3.1.6) comes only from the effect of small relative changes on the diagonal. This factor can be reduced considerably. For example, consider the one dimensional case: compute e^{-1000}. The logarithmic derivative of e^{-1000} is

$$\frac{-1000}{e^{-1000}} \cdot e^{-1000} = -1000,$$

which indicates correctly that one ulp's (unit in the last place) change in the argument -1000 will result in a thousand ulp's change in e^{-1000}. However, modern implementations of exp have taken the trouble to do *better* than getting $e^{x(1+\varepsilon)}$.

As a matter of fact, in [5] we exhibit an algorithm for computing the exponential divided differences with guaranteed accuracy

$$|fl(\Delta_i^k(Z)\exp) - \Delta_i^k(Z)\exp| \le \varepsilon \cdot K_k \cdot \Delta_i^k(\mathrm{Re}Z)\exp,$$

where ε is the precision of the arithmetic and K_k depends on k only. Thus, a typical error analysis will show that if one computes e^S by formula (3.1.4), then for some K'_{j-i} depends only on the difference $j-i$,

$$|fl((e^S)_{i,j}) - (e^S)_{i,j}| \le \varepsilon \cdot K'_{j-i} \cdot (e^{\Gamma(S)})_{i,j} \tag{3.1.7}$$

Conclusion. In view of the above discussion, we define the absolute *sensitivity* of $(e^S)_{i,j}$ to be $(e^{\Gamma(S)})_{i,j}$ (ignore the factor $(1+\max_i|\zeta_i|)$); and the *relative* sensitivity (relative to $(e^S)_{i,j}$) to be $\rho_{i,j} = \frac{(e^{\Gamma(S)})_{i,j}}{|(e^S)_{i,j}|}$, provided $(e^S)_{i,j} \neq 0$.

3.2. Perturbation Bounds

Since the matrix exponential problem is closely related to the solution of linear systems of O.D.E.'s one might suspect that similar results concerning the sensitivity of e^{tA} would have been in the O.D.E. literature . For example, a simple corollary of the following integral representation (cf. Bellman [1]) yields perturbation bounds similar to our results in Theorem 3.4.

Theorem 3.5. Let A and B be square matrices.

$$e^{t(A+B)} - e^{tA} = \int_0^t e^{(t-\tau)A}\cdot(B)\cdot e^{\tau(A+B)}d\tau \tag{3.2.1}$$

Proof. Let $X(t)$ be the left hand side of (3.2.1). It is straightforward to see that $X(t)$ satisfies the O.D.E.

$$\frac{d}{dt}[e^{-tA}\cdot X(t)] = e^{-tA}\cdot B\cdot e^{tA}.$$

The theorem follows by integrating the above equation from 0 to t. •

Let δS be a triangular perturbation of an upper triangular matrix S.

Corollary 3.6. If $|\delta S| \le \varepsilon |S|$†, then, as $\varepsilon \to 0$,

$$|e^{S+\delta S} - e^S| \le \varepsilon\cdot(\max_i |s_{i,i}|\cdot I + \frac{1}{2}\Gamma(S))\cdot e^{\Gamma(S)} + O(\varepsilon^2) \tag{3.2.2}$$

Proof. (3.2.1) implies, as $\varepsilon \to 0$,

$$e^{t(S+\delta S)} - e^{tS} = \int_0^t e^{(t-\tau)S}\cdot(\delta S)\cdot e^{\tau S}d\tau + O(\varepsilon^2).$$

Note that $\Gamma(S)$ dominates $|S|$ except on the diagonal. In fact

$$|\delta S| \le \varepsilon\cdot|S| \le \varepsilon(\Gamma(S) + 2\max_i |s_{i,i}|\cdot I).$$

† Recall $|A|$ denotes the matrix whose elements are the absolute values of the elements of A.

Since $|e^S| \le e^{\Gamma(S)}$ (Corollary 2.4) and $\Gamma(S)e^{\Gamma(S)} = e^{\Gamma(S)}\Gamma(S)$, we get

$$
\begin{aligned}
|e^{S+\delta S} - e^S| &\le \int_0^1 e^{(1-\tau)\Gamma(S)} \cdot |\delta S| \cdot e^{\tau\Gamma(S)} d\tau + O(\varepsilon^2) \\
&\le \int_0^1 e^{(1-\tau)\Gamma(S)} \cdot e^{\tau\Gamma(S)} (\Gamma(S) + 2\max_i |s_{i,i}| \cdot I) d\tau + O(\varepsilon^2) \\
&\le \varepsilon \cdot (\max_i |s_{i,i}| \cdot I + \frac{1}{2}\Gamma(S)) \cdot e^{\Gamma(S)} + O(\varepsilon^2). \quad \blacksquare
\end{aligned}
$$

We should mention that Corollary 2.4, namely, $|e^S| \le e^{\Gamma(S)}$, can be derived without using the explicit representation of e^S given in Theorem 2.2. In fact, it is just a corollary of the following typical differential inequality in O.D.E. .

Theorem 3.7. If $\psi(0) \ge |\varphi(0)|$ and $\psi'(t) \ge |\varphi'(t)|$ $(t \ge 0)$, then

$$\psi(t) \ge \varphi(t) \qquad (t \ge 0). \tag{3.2.3}$$

Proof. It follows from integrating ψ' and φ' from 0 to t. ▪

Corollary 3.8.

$$|e^S| \le e^{\Gamma(S)}.$$

The proof is given in Appendix I.B.

It is possible that someone has already obtained results similar to (3.2.2) in works related to O.D.E. ; but our search of the literature has not revealed it. To compare the result of our approach (using the explicit representation of $(e^S)_{i,j}$) with Corollary 3.6, we give a perturbation bound derived from Theorem 2.2. Recall $(e^S)_{i,j}$ has the following representation (for simplicity, we drop the reference of exp in $\Delta^k(Z_\sigma)$exp)

$$(e^S)_{i,j} = \sum_{k=1}^{i-j} \sum_{\sigma \in E^k_{i,j}} s_{\sigma_0,\sigma_1} \cdots s_{\sigma_{k-1},\sigma_k} \cdot \Delta^k(Z_\sigma). \tag{3.2.4}$$

Theorem 3.9. If $|\delta S| \le \varepsilon |S|$, then, as $\varepsilon \to 0$,

$$|(e^{S+\delta S} - e^{S})_{i,j}| \le \varepsilon(j-i+\max_i |s_{i,i}|)(e^{\Gamma(S)})_{i,j} + O(\varepsilon^2) \tag{3.2.5}$$

Proof. A typical component in the right hand side of (3.2.4) is of the form

$$a_1 \cdot a_2 \cdots a_k \cdot \Delta^k(X) \tag{3.2.6}$$

where $X=\{\xi_0, \xi_1, \ldots, \xi_k\}$ is a subset of the eigenvalues of S. The corresponding component in the perturbed element $(e^{S+\delta S})_{i,j}$ is

$$(a_1+\delta a_1)\cdot(a_2+\delta a_2)\cdots(a_k+\delta a_k)\cdot\Delta^k(X+\delta X) \tag{3.2.7}$$

where $|\delta a_i| \le \varepsilon |a_i|$ and $|\delta \xi_i| \le \varepsilon |\xi_i|$. Since Δ^k is a smooth function of each of its argument, we can expand $\Delta^k(X+\delta X)$ at X to get

$$\Delta^k(X+\delta X) = \Delta^k(X) + \delta\Delta^k(X) + O(\varepsilon^2) \tag{3.2.8}$$

where

$$\delta\Delta^k(X) = \sum_{i=0}^{k} \xi_i \cdot \frac{\partial}{\partial \xi_i} \Delta^k(X).$$

It follows from Lemma 3.3 that

$$|\delta\Delta^k(X)| \le \max_i |\xi_i| \Delta^k(\mathrm{Re}X).$$

Now, subtract (3.2.6) from (3.2.7) to get

$$\left| \prod_{i=1}^{k}(a_i+\delta a_i)\Delta^k(X+\delta X) - \prod_{i=1}^{k} a_i \cdot \Delta^k(X) \right| =$$

$$= \left| \prod_{i=1}^{k} a_i \cdot \left[\prod_{i=1}^{k}(1+\frac{\delta a_i}{a_i})\cdot(\Delta^k(X)+\delta\Delta^k(X)) - \Delta^k(X) \right] + O(\varepsilon^2) \right.$$

$$\le \prod_{i=1}^{k} |a_i| \cdot \Delta^k(\mathrm{Re}X)\cdot[(1+\varepsilon)^k - 1 + \varepsilon \cdot \max_i |\xi_i|] + O(\varepsilon^2)$$

$$\le \prod_{i=1}^{k} |a_i| \cdot \Delta^k(\mathrm{Re}X)\cdot(\varepsilon k + \varepsilon \cdot \max_i |\xi_i|) + O(\varepsilon^2).$$

As a consequence, we have, as $\varepsilon \to 0$,

$$
\begin{aligned}
|(e^{S+\delta S} - e^{S})_{i,j}| &\le \\
&\le \varepsilon(j-i+\max_i |s_{i,i}|)\cdot \sum_{k=1}^{j-i} \sum_{\sigma\in E^k_{i,j}} |s_{\sigma_0,\sigma_1}|\cdots|s_{\sigma_{k-1},\sigma_k}|\cdot \Delta^k(\mathrm{Re}Z_\sigma) + O(\varepsilon^2) \\
&\le \varepsilon(j-i+\max_i |s_{i,i}|)\cdot (e^{\Gamma(S)})_{i,j} + O(\varepsilon^2). \quad \bullet
\end{aligned}
$$

The reader should notice that the perturbation bounds (3.2.1) and (3.2.5) are similar but different. It is not clear which one is sharper in general.

3.3. An Exponential Condition Number

Recall that $\Gamma(S) \equiv \mathrm{Re}(D)+|N|$ where $S=D+N$, D is diagonal and N is strictly upper triangular. In section 3.1 we defined

$$\rho_{i,j}(S) = \frac{(e^{\Gamma(S)})_{i,j}}{|(e^{S})_{i,j}|} \tag{3.3.1}$$

to be the relative sensitivity of $(e^S)_{i,j}$. In case of $(e^S)_{i,j}=0$ and $(e^{\Gamma(S)})_{i,j}\neq 0$, we set $\rho_{i,j}=\infty$. The reason is simple : when $(e^S)_{i,j}=0$, a tiny perturbation in S will turn $(e^S)_{i,j}$ into a non-zero element and hence the relative change of $(e^S)_{i,j}$ is ∞. When $(e^{\Gamma(S)})_{i,j}=0$, we define $\rho_{i,j}=1$. For, it can be shown that from the representation of $(e^{\Gamma(S)})_{i,j}$, $(e^{\Gamma(S)})_{i,j}=0$ implies S is reducible : some integer m exists such that $s_{p,q}$ is equal to zero for $i\le p\le m$ and $m+1\le q<j$; i.e., the sub-matrix

$$
\begin{bmatrix}
s_{i,i} & s_{i,i+1} & \cdot & s_{i,j} \\
 & s_{i+1,i+1} & \cdot & \cdot \\
 & & \cdot & \cdot \\
 & & & s_{j,j}
\end{bmatrix}
\to
\begin{bmatrix}
S_1 & 0 \\
 & S_2
\end{bmatrix}.
$$

In this situation, only perturbations in S_1 and S_2 are considered and $(e^S)_{i,j}$ remains zero.

The measure $\rho_{i,j}$ is quite realistic, in the sense that the relative error in the computed $(e^S)_{i,j}$ (by some some of our best numerical method) agrees with $\rho_{i,j}$ in magnitude. Let us consider the following example.

$$S = \begin{bmatrix} 1 & 3 & 2.7727 \\ & 0 & -2 \\ & & -1 \end{bmatrix}.$$

The exponential of S and $\Gamma(S)$ are (correct to 5 significant decimal digits)

$$e^S = \begin{bmatrix} 2.7183 & 5.1548 & -3.4592_{10}-6 \\ & 1 & -1.2642 \\ & & 0.36788 \end{bmatrix} \quad e^{\Gamma(S)} = \begin{bmatrix} 2.7183 & 5.1548 & 6.5190 \\ & 1 & 1.2642 \\ & & 0.36788 \end{bmatrix}$$

The sensitivity of the (1,3) element is

$$\rho_{1,3} = \frac{6.5190}{|-3.4592_{10}-6|} \approx 1.88_{10}6,$$

which indicates that $(e^S)_{1,3}$ is sensitive to rounding error. For instance, if one uses Parlett's recurrence ([5]) using 5 significant decimal digits arithmetic, one obtains

$$fl(e^S) = \begin{bmatrix} 2.7183 & 5.1549 & 4.7670_{10}-6 \\ & 1 & -1.2642 \\ & & 0.36788 \end{bmatrix}.$$

Notice that not even one digit is correct in the (1,3) element. This is expected since a perturbation of 2.7727 to 2.772707 will yield an exponential equal to the above $fl(e^S)$ up to five significant decimal digits.

In the above example, the (1,3) element is so small that it is negligible compared to the other elements. Therefore, the inaccuracy of that element will not have much effect on the whole matrix. A reasonable single measure

for the matrix as a whole is the following :

Definition. The *condition number* for the exponential of a triangular matrix S with respect to a given norm is defined to be

$$cond(S)=cond(S,\exp)\equiv\frac{||e^{\Gamma(S)}||}{||e^{S}||}\ . \tag{3.3.2}$$

In the above example, $cond(S) \approx 1.8$.

Remark 1. In practice we are only interested in *cond's* order of magnitude. Numerical examples show that the number of decimal digits lost in $fl(e^S)$ is usually smaller than $\log_{10}cond(S)$. Since there is no need to compute $cond(S)$ accurately, a fast method can be employed to compute $e^{\Gamma(S)}$ for $cond(S)$ (notice that $\Gamma(S)$ is real). The cost of computing $e^{\Gamma(S)}$ adds about 20% to the cost of the Schur form and the cost of e^S.

Remark 2. We have performed numerous experiments on various matrices based on the numerical algorithm (for the matrix exponential) suggested in [5]. We compared the relative error in $fl(e^S)$, namely

$$\frac{||fl(e^S)-e^S||}{||e^S||}\ ,$$

with $\varepsilon\cdot cond(S)$ (where ε is the machine precision) and found that $cond(S)$ is a good predictor of the error in magnitude, provided that the matrix S is first reduced to an equivalent matrix that has eigenvalues bounded in their imaginary parts by a small number like π (cf. Part II Matrix Argument Reduction).

In general, if $\Gamma(S) \approx S$ ($cond(S) \approx 1$), then we expect e^S can be computed accurately.

4. Numerical Results

Two sets of examples are considered in this section: the first one contains matrices that approximate $\ln(J_\zeta)$ where J_ζ is the Jordan block with ζ on the diagonal. The second set contains matrices of dimension 6 with various imaginary parts on the diagonal elements.

The computation of e^S consists of two steps. First, S is reduced to a triangular matrix C whose eigenvalues are near the real axis. Then, using the Newton polynomial of $\exp(C)$,

$$\exp(S) = \exp(C) = \Delta_1^0(Z)\exp\cdot I + \sum_{k=1}^{n-1}\Delta_1^k(Z)\exp\cdot\prod_{j=1}^{k}(C-\zeta_j I).$$

Here $Z=(\zeta_1, \ldots, \zeta_n)$ is the diagonal (eigenvalues) of C. The coefficients $\Delta_1^k(Z)\exp$ are computed by a hybrid algorithm (SH) suggested in [5]. The details of the computation will be given in another paper. Because the exponential divided differences can be computed accurately, Newton's method is quite reliable (although it is slower than Parlett's recurrence [6]).

In what follows, we use single precision arithmetic† to compute $fl(e^S)$ and $cond(S)$. The relative error in $fl(e^S)$ and the number $cond(S)\cdot\varepsilon$ ($\varepsilon=2^{-24}$) will be compared .

† All computations in this section are done on a Vax 11/780. The single precision arithmetic is ~7 decimal significant digits and the double is ~16 . Vax is a trademark of the Digital Equipment Corp.

Example (I).

The general form of the matrix considered in this example is the matrix

$$S_\zeta = fl(\ln(J_\zeta)) = fl\left(\begin{bmatrix} \ln\zeta & \frac{1}{\zeta} & -\frac{1}{2\zeta^2} & \frac{1}{3\zeta^3} & \cdots & \frac{(-1)^n}{(n-1)\zeta^{n-1}} \\ & \ln\zeta & \frac{1}{\zeta} & -\frac{1}{2\zeta^2} & \cdots & \cdot \\ & & \ln\zeta & \frac{1}{\zeta} & \cdots & \cdot \\ & & & \ln\zeta & \cdots & \cdot \\ & & & & \cdots & \cdot \\ & & & & & \ln\zeta \end{bmatrix}\right),$$

where

$$J_\zeta = \begin{bmatrix} \zeta & 1 & & \\ & \zeta & 1 & \\ & & \cdots & 1 \\ & & & \zeta \end{bmatrix}.$$

Because of the roundoff, $\exp(S_\zeta)$ is close but not equal to J_ζ. Our results are summarized in Table (4.1) and (4.2). In Table (4.1), the first column is the value of ζ; the second column is the dimension of S_ζ; the third column is

$$cond = cond(S_\zeta) = \frac{\|fl(\exp(\Gamma(S_\zeta)))\|_1}{\|fl(\exp(S_\zeta))\|_1};$$

and the last column is err/ε ($\varepsilon=2^{-24}$), where err is the relative error in $fl(\exp(S_\zeta))$ in norm

$$err = err(fl(\exp(S_\zeta))) = \frac{\|fl(\exp(S_\zeta))-\exp(S_\zeta)\|_1}{\|\exp(S_\zeta)\|_1}.$$

Table (4.2) contains the details of the elements of S_ζ and $\exp(S_\zeta)$ when $n=5$ and $\zeta=0.25$. The reader should notice that the condition number of each element is a good prediction of the relative error in that element

ζ	n	$cond(S_\zeta)$	err/ε
1	5	2.5	0.3
	10	5.0	1.0
	15	7.5	1.7
0.5	5	10.4	4.0
	10	341	170
	15	10921	1615
0.25	5	68	32
	10	7×10^4	4×10^4
	15	6×10^7	3×10^6

Table 4.1. Condition numbers and the relative error in $\exp(S_\zeta)$.

i,j	$S_{i,j}$	$fl(e^S)_{i,j}$	correct value	$\rho_{i,j}$	rel err / ε
(1,1)	-1.386294e+00	2.500000e-01	2.500000e-01	1.0e+00	6.4e-02
(1,2)	4.000000e+00	1.000000e+00	1.000000e+00	1.0e+00	6.4e-02
(1,3)	-8.000000e+00	0.000000e+00	0.000000e+00	4.0e+00	1.0e+00
(1,4)	2.133333e+01	0.000000e+00	1.589457e-07	1.0e+08	1.7e+07
(1,5)	-6.400000e+01	2.861023e-06	6.357830e-07	1.0e+08	5.9e+07
(2,2)	-1.386294e+00	2.500000e-01	2.500000e-01	1.0e+00	6.4e-02
(2,3)	4.000000e+00	1.000000e+00	1.000000e+00	1.0e+00	6.4e-02
(2,4)	-8.000000e+00	0.000000e+00	0.000000e+00	4.0e+00	1.0e+00
(2,5)	2.133333e+01	0.000000e+00	1.589457e-07	1.0e+08	1.7e+07
(3,3)	-1.386294e+00	2.500000e-01	2.500000e-01	1.0e+00	6.4e-02
(3,4)	4.000000e+00	1.000000e+00	1.000000e+00	1.0e+00	6.4e-02
(3,5)	-8.000000e+00	0.000000e+00	0.000000e+00	4.0e+00	1.0e+00
(4,4)	-1.386294e+00	2.500000e-01	2.500000e-01	1.0e+00	6.4e-02
(4,5)	4.000000e+00	1.000000e+00	1.000000e+00	1.0e+00	6.4e-02
(5,5)	-1.386294e+00	2.500000e-01	2.500000e-01	1.0e+00	6.4e-02

Table 4.2 The elements of $\exp(S_\zeta)$ and their sensitivities (with n =5 and ζ=0.25).

Example (II).

$$S_k = \begin{bmatrix} -20ki & 0.25 & -12.5 & 75 & -99 & 183.5 \\ & -10ki & 0.25 & -12.5 & 75 & -99 \\ & & -5ki & 0.25 & -12.5 & 75 \\ & & & 0 & 0.25 & -12.5 \\ & & & & 5ki & 0.25 \\ & & & & & 10ki \end{bmatrix}.$$

In this example we apply the argument reduction (cf. part II) on S before the matrix exponential is called. The reduction matrix C is triangular and its eigenvalues are near the real axis. It may worth to mention that when our matrix exponential is applied to S_k directly, the relative error in $fl(\exp(S_k))$ is much larger than those in $fl(\exp(C_k))$. We summarize our results in Table (4.3) and (4.4). In Table (4.3), we list the relative error in $fl(\exp(C_k))$ in the third column

$$rel\ err = \frac{\|fl(\exp(C_k)) - \exp(S_k)\|_1}{\|\exp(S_k)\|_1}$$

In the second column, we give the exponential condition number *cond* of S_k. The first column is the value of k. In Table (4.4), we list the details (the elements of the matrix exponential and the condition number of each element) of $\exp(S_5)$. Again the $\rho_{i,j}$ gives a fairly good indication on the size of the relative error in the i,j-th element (except the diagonals and the superdiagonals, which are computed by special formula, cf [5] for the special formula of the first divided difference).

k	*cond*	*rel err* : ε
0	1.7	0.3
1	50.	8.0
2	96.	8.1
3	138	34.
4	192	44.
5	492	96.

Table 4.3. *cond* vs. *err* / ε on S_k's exponential.

i,j	$fl(\exp(S_5))$		*correct value*		$\rho_{i,j}$	*rel err* / ε
(1,1)	8.623189e-01	5.063657e-01	8.623189e-01	5.063656e-01	1.0e+00	2.6e-01
(1,2)	-1.219954e-03	-5.132358e-04	-1.219954e-03	-5.132358e-04	1.9e+02	7.7e-01
(1,3)	6.233682e-02	2.148104e-02	6.233648e-02	2.148092e-02	1.9e+02	9.0e+01
(1,4)	-3.798594e-01	-1.032820e-01	-3.798592e-01	-1.032819e-01	2.0e+02	9.9e+00
(1,5)	5.085295e-01	1.025526e-01	5.085271e-01	1.025522e-01	3.8e+02	8.0e+01
(1,6)	-9.670361e-01	-1.285248e-01	-9.670222e-01	-1.285228e-01	1.2e+03	2.4e+02
(2,2)	9.649660e-01	2.623748e-01	9.649660e-01	2.623749e-01	1.0e+00	5.0e-01
(2,3)	-1.300231e-03	-2.623678e-04	-1.300231e-03	-2.623678e-04	1.9e+02	9.1e-01
(2,4)	6.559493e-02	8.758656e-03	6.559459e-02	8.758609e-03	1.9e+02	8.9e+01
(2,5)	-3.948144e-01	-2.624261e-02	-3.948142e-01	-2.624261e-02	2.0e+02	9.0e+00
(2,6)	5.222191e-01	-1.132139e-08	5.222166e-01	-4.065758e-19	3.8e+02	7.8e+01
(3,3)	9.912028e-01	1.323518e-01	9.912028e-01	1.323518e-01	1.0e+00	3.4e-01
(3,4)	-1.323517e-03	-8.797189e-05	-1.323518e-03	-8.797188e-05	1.9e+02	1.1e+00
(3,5)	6.617710e-02	-9.313226e-10	6.617675e-02	-3.252607e-19	1.9e+02	8.8e+01
(3,6)	-3.948144e-01	2.624262e-02	-3.948142e-01	2.624261e-02	2.0e+02	7.8e+00
(4,4)	1.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	1.0e+00	9.3e-10
(4,5)	-1.323517e-03	8.797187e-05	-1.323518e-03	8.797188e-05	1.9e+02	1.1e+00
(4,6)	6.559493e-02	-8.758656e-03	6.559459e-02	-8.758609e-03	1.9e+02	8.7e+01
(5,5)	9.912028e-01	-1.323518e-01	9.912028e-01	-1.323518e-01	1.0e+00	3.4e-01
(5,6)	-1.300231e-03	2.623678e-04	-1.300231e-03	2.623678e-04	1.9e+02	6.6e-01
(6,6)	9.649660e-01	-2.623748e-01	9.649660e-01	-2.623749e-01	1.0e+00	5.0e-01

Table 4.4 The elements of $\exp(S_5)$ and their sensitivities .
(Complex number is represented as a pair of integers.)

Appendix I.A. Stability Analysis of The Schur Decomposition

Given any square matrix A, there exist (by Schur's lemma) an unitary matrix U and a triangular matrix S (with the eigenvalues arranged in any desired order along the diagonal) such that $=U^H AU$. S is called the Schur form of A (associated with U), and sometime we call (S, U) a Schur decomposition of A. Since $f(A)=Uf(S)U^H$ for any analytic function f, e^A may be computed by $Ue^S U^H$; accordingly, it suffices to work on S. The worry in using Schur decomposition to compute e^A is whether the whole process is stable. In order words, if the computed S and U satisfy $A \approx USU^H$ and $UU^H \approx I$ (whether or not S is close to the exact Schur form), does $e^A \approx Ue^S U^H$? We will answer this question in this section.

Notations

We fix some notations. Let $K(X)$ denotes the condition number of X (for matrix inversion), i.e., $K(X)=||X||\cdot||X^{-1}||$. Here $||\cdot||$ will always denote the 2-norm, $||A||=\max_{||x||=1} ||Ax||_2$. We also use the spectral abscissa (or the index)

$$\alpha(A)=\max\{\,\mathrm{Re}(\zeta):\ \zeta\in spectrum(A)\,\} \tag{a1}$$

and the log norm

$$\mu(A)=\max\{\mu\,|\,\mu\in spectrum((A^H+A)/2)\}. \tag{a2}$$

The log norm has the following properties (see [3]):

$$\begin{aligned} &|\mu(A)|\le||A||,\\ &\mu(A+B)\le\mu(A)+\mu(B), \text{ and}\\ &\alpha(A)\le\mu(A). \end{aligned} \tag{a3}$$

Bounds on exp(A)

We summarize some results on bounding e^{tA}, which will be used later. All the following bounds can be found in [3], [9] and [10].

(a) Lower bound.

$$||e^{tA}|| \geq e^{t\alpha(A)} \qquad (t \geq 0) \tag{a4}$$

(b) Norms and log norms.

$$||e^{tA}|| \leq e^{t\mu(A)} \leq e^{t||A||} \qquad (t \geq 0). \tag{a5}$$

(c) Jordan canonical form. If $A = XJX^{-1}$ and m is the maximum size of the Jordan blocks in J, then

$$||e^{tA}|| \leq mK(X) \cdot \max_{0 \leq r \leq m-1} \frac{t^r}{r!} e^{t\alpha(A)}. \tag{a6}$$

(d) Schur decomposition bound. If $U^H AU = D + N$ where U is unitary, D is diagonal and N is strictly upper triangular, then

$$||e^{tA}|| \leq e^{t\alpha(A)} \sum_{k=0}^{n-1} \frac{||tN||^k}{k!}. \tag{a7}$$

Remark. These bounds have been discussed in detail in [3]. One can always find examples that favor one bound over the others. For later use the following slightly general form

$$||e^{tA}|| \leq p_A(t) \cdot e^{tg(A)} \tag{a8}$$

will be used to represent the bounds (a5-7). Thus $g(A)$ can be the index or the log norm of A, and $p_A(t)$ can be a constant or a polynomial in t.

Theorem A.1.

If $A-USV=E$ and $I-VU=F$, then

$$e^{tA}-Ue^{tS}V=\int_0^t e^{(t-\tau)A}(EU-USF)e^{\tau S}\cdot V\,d\tau. \tag{a9}$$

Proof. Let $X(t)=e^{tA}-Ue^{tS}V$, we have

$$\begin{aligned}X'(t)=\frac{d}{dt}X(t)&=Ae^{tA}-USe^{tS}V\\&=Ae^{tA}-US\cdot(F+VU)\cdot e^{tS}V\\&=Ae^{tA}-USFe^{tS}V-USV\cdot(Ue^{tS}V)\\&=Ae^{tA}-USFe^{tS}V-(A-E)\cdot(Ue^{tS}V)\\&=AX(t)+(EU-USF)\cdot e^{tS}\cdot V\end{aligned}$$

Thus

$$e^{-tA}\cdot(X'(t)-AX(t))=e^{-At}(EU-USF)\cdot e^{tS}\cdot V$$

or

$$(e^{-tA}X(t))'=e^{-At}(EU-USF)\cdot e^{tS}\cdot V.$$

Taking the integral of both sides from 0 to t, we have

$$e^{-tA}X(t)=\int_0^t e^{-\tau A}(EU-USF)e^{\tau S}\,d\tau.$$

The theorem follows by multiplying e^{tA} from the left to both sides of the above equation. ∎

Corollary A.2.

$$||e^{tA}-Ue^{tS}V||\le||V||\,||U||(||E||+||SF||)t\cdot e^{t\cdot\max(\mu(A),\mu(S))}; \tag{a10}$$

$$||e^{tA}-Ue^{tS}V||\le||V||\,||U||(||E||+||SF||)t\cdot p(t)\cdot e^{t\cdot\max(\alpha(A),\alpha(S))}, \tag{a11}$$

where $p(t)$ is either

(a). a polynomial of degree less than m_1+m_2-1, where m_1 and m_2 are the maximum sizes of Jordan block of A and S respectively,

(b). a polynomial of degree less than $2n-1$ with the constant term equal to 1.

Proof. Assuming $||e^{tA}||\le p_A(t)e^{tg(A)}$, Theorem A.1 implies

$$||e^{tA}-Ue^{tS}V||\le\int_0^t p_A(t-\tau)e^{(t-\tau)g(A)}||EU-USF||p_S(\tau)\cdot e^{\tau g(S)}||V||d\tau$$
$$\le||EU-USF||||V||\int_0^t e^{(t-\tau)g(A)+\tau g(S)}\cdot p_A(t-\tau)\cdot p_S(\tau)d\tau.$$

Since

$$(t-\tau)g(A)+\tau g(S)\le((t-\tau)+\tau)\max\{g(A),g(S)\}$$

and

$$||EU-USF||\le||V||||U||(||E+||SF||),$$

we have

$$||e^{tA}-Ue^{tS}V||\le||V||||U||(||E||+||SF||)\int_0^t p_A(t-\tau)p_S(\tau)d\tau\cdot e^{\max\{g(A),g(S)\}}.$$

The corollaries follow by choosing suitable $p_A(t)$ and $g(A)$ according to (a5-7). ∎

Stability of Schur Decomposition.

When the Schur decomposition USU^H is used, the computed U satisfies $||U||\cdot||U^H||\approx1$. Corollary A.2 implies that if $||A-U^HSU||\le\varepsilon||A||$ and $||I-UU^H||\le\varepsilon$, then, for some polynomial $p(t)$,

$$\frac{||e^{tA}-Ue^{tS}U^H||}{\max\{||e^{tA}||,||e^{tS}||\}}\le 2\varepsilon t\cdot p(t)\cdot\max\{||A||,||S||\}\ .$$

This shows that whether S accurately approximated the Schur form of A does not matter; as long as USU^H is close to A, $Ue^S U^H$ is close to e^A.

The corollary also shows the possible advantage of using Schur decomposition to Jordan decomposition (in computing e^A) when A has an ill-conditioned eigensystem. If one uses Jordan decomposition, then $||U||||V||=||R||||R^{-1}||\approx K(R)$ where R is the matrix that transforms A to Jordan normal form : $R^{-1}AR=J$. Here $K(R)$ could be enormous.

Appendix B. A Different Proof of Corollary 2.4.

It is well known (see, e.g. Bellman [1]) that a fundamental matrix Φ for a system of O.D.E.s with constant coefficients $x'=Sx$ is given by

$$\Phi(t) = e^{tS} \qquad (|t| < \infty).$$

The solution φ (vector function) with given initial vector $\varphi(0)$ looks as follows

$$\varphi(t) = e^{tS}\cdot\varphi(0) \qquad (|t| < \infty).$$

Similarly, $\psi(t)=e^{t\Gamma(S)}\cdot\psi(0)$ is the solution of $x'=\Gamma(S)x$. It is not difficult to see then that $|e^S|\le e^{\Gamma(S)}$ is equivalent to $\psi(t)\ge|\varphi(t)|$ for any $\psi(0)\ge|\varphi(0)|$. Thus Corollary 2.4 follows from the theorem below.

Theorem B.1. If $\varphi(t)$ and $\psi(t)$ are the corresponding solution of the differential systems $x' = Sx$ and $x' = \Gamma(S)x$ (S is a n by n triangular matrix), and if $\psi(0) \ge |\varphi(0)|$ (element-wise), then

$$\psi(t) \ge |\varphi(t)| \qquad (t\ge 0). \tag{b1}$$

Proof. Since

$$\frac{d\varphi_i(t)}{dt} = a_{i,i}\cdot\varphi_i(t) + \sum_{i<j} a_{i,j}\cdot\varphi_j(t) \qquad (i=1,2,\ldots,n), \tag{b2}$$

we have, multipling both sides of (b2) by $e^{-a_{i,i}t}$ and rearranging the terms,

$$\frac{d}{dt}[e^{-a_{i,i}t}\cdot\varphi_i(t)] = \sum_{i<j} a_{i,j}\cdot[e^{-a_{i,i}t}\cdot\varphi_j(t)]. \tag{b3}$$

Similarly,

$$\frac{d}{dt}[e^{-\mathrm{Re}a_{i,i}t}\cdot\psi_i(t)] = \sum_{i<j} |a_{i,j}|\cdot[e^{-\mathrm{Re}a_{i,i}t}\cdot\psi_j(t)]. \tag{b4}$$

When $i=n$,

$$\psi_n(t)=e^{-\mathrm{Re}a_{n,n}\cdot t}\cdot\psi_n(0)\geq|e^{-a_{n,n}t}\cdot\varphi_n(0)|=|\varphi_n(t)|.$$

So for $k=n$ $\psi_k(t)\geq|\varphi_k(t)|$ is true. Assume that the inequality $\psi_k(t)\geq|\varphi_k(t)|$ is true for $i<k\leq n$. Then

$$\begin{aligned}\frac{d}{dt}[e^{-\mathrm{Re}a_{i,i}t}\psi_i(t)] &= \sum_{i<j}|a_{i,j}|\cdot e^{-\mathrm{Re}a_{i,i}t}\cdot\psi_j(t)\\ &\geq |\sum_{i<j}a_{i,j}\cdot e^{-a_{i,i}t}\cdot\varphi_j(t)|\\ &= |\frac{d}{dt}[e^{-a_{i,i}t}\varphi_j(t)]|.\end{aligned}$$

Since $\psi_i(0)\geq|\varphi_i(0)|$, by Theorem 3.7,

$$e^{-\mathrm{Re}a_{i,i}t}\cdot\psi_i(t)\geq|e^{-a_{i,i}t}\varphi_i(t)|\ .$$

Which implies

$$\psi_i(t)\geq|\varphi_i(t)|.$$

By the principle of induction, $\psi_i(t)\geq|\varphi_i(t)|$ for every i ; hence proving the theorem. ▪

REFERENCES

[1] R. Bellman, *Introduction to Matrix Analysis*, McGraw-Hill, New York, 1969.

[2] A.O. Gel'fand, *Calculus of Finite Differences*, Hindustan, India (1971).

[3] Bo Kågström, *Bounds and perturbation bounds for the matrix exponential*, Bit 17 , 1977, 39-57.

[4] C.C. MacDuffee, "The Theory of Matrices", Chelsea, New York, 1956.

[5] A. McCurdy, K.C. Ng and B.N. Parlett, *Accurate Computation of Divided Differences of the Exponential Function*. 1983. (to appear)

[6] B.N. Parlett, *Computation of functions of triangular matrices*, ERL-M481, UC Berkeley, 1974.

[7] J.R. Roche, *On the sensitivity of the matrix exponential problem*, R.A.I.R.O. Analyse numérique/Numercal Analysis, Vol. 15, n°3, 1981, p.249-255 .

[8] J.D. Stafney, *Functions of a Matrix and their norms*, Linear Algebra and its Application 20, 87-94 (1978).

[9] C. VanLoan, *A study of the matrix exponential*, University of Manchester numerical analysis report 10, 1975, Manchester, England.

[10] C. VanLoan, *The sensitivity of the matrix exponential*, SIAM J. NUMER. ANAL., Vol 14, No.6 December 1977, p971-981.

[11] J.H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice Hall (1963).

Part II.

Matrix Argument Reduction and its Application to Computing Exp(S)

5. Introduction

One common and important technique in computing periodic functions (like exp, sin and cos) is argument reduction, which reduces a given value x to an equivalent value in a standard range. E.g.

$$\sin(100)=\sin(100-32\pi)=\sin(-0.530984\cdots).$$

This technique can be extended to matrices. For periodic functions like exp and sin, one can reduce a matrix B to some C whose eigenvalues lie in a standard range. This technique is useful in the matrix exponential problem, for exp is periodic on the imaginary axis and by working with C one needs only consider matrices whose eigenvalues are near the real axis.

In this part, we restrict our attention to triangular matrices[†]. We begin with a formal definition of matrix argument reduction. Then, in section 7, a numerical method for the reduction on triangular matrices is described. Finally we discuss the application of argument reduction to the computation of the matrix exponential in section 8.

[†] Any matrix B is unitarily similar to its Schur form S, which is triangular.

6. Argument Reduction for Matrix Functions

The objective of this section is to define an equivalence relation "congruence modulo ω" on square matrices. It characterizes matrices that have the same image under certain periodic matrix functions. More precisely, if C is *congruent* to B modulo ω, then $f(C)=f(B)$ for function f of period ω. As a consequence, a given argument B may be replaced (for the purpose of computing $f(B)$) by some other matrix with more desirable properties.

6.1. Integral Representation of Congruence

Given complex numbers η, ζ and $\omega\neq 0$, we say η is *congruent* to ζ *modulo* ω if $(\eta-\zeta)/\omega$ is an integer; and using the notation of Gauss we write

$$\eta\equiv\zeta \pmod{\omega}\,. \tag{6.1.1}$$

(If ζ is not congruent to η, we shall write $\zeta\not\equiv\eta \pmod{\omega}$.) When there is no doubt concerning the modulus, the mod ω of the formula may be omitted . Notice that if $\eta\equiv\zeta \pmod{\omega}$ then some integer k must exist such that $\eta=\zeta-k\omega$, and conversely. Hence, for any periodic function f with period ω, we have

$$f(\eta)=f(\zeta-k\omega)=f(\zeta). \tag{6.1.2}$$

By the Cauchy integral formula, η does have the following rather complicated representation

$$\eta=\zeta-k\omega=\frac{1}{2\pi i}\int_{\Gamma}\frac{(z-k\omega)}{z-\zeta}dz\,, \tag{6.1.3}$$

where Γ is a simple closed contour (positively oriented) that contains ζ in its interior. For matrices, an analogue of the Cauchy integral formula is the

Dunford-Taylor integral. For any polynomial $p(t)$ it can be proved that

$$p(B) = \frac{1}{2\pi i}\int_{\Gamma} p(z)(zI-B)^{-1}dz$$

where Γ is a closed contour that enclosed all eigenvalues of A (see Kato [0], p.44). By analogy, this integral is used to define $p(B)$, even if $p(t)$ is not a polynomial but an analytic function with no singularity inside nor on Γ, as follows. Let $\lambda_1, \ldots, \lambda_m$ be the eigenvalues, not necessarily all distinct, of a given $n \times n$ matrix B. Let $\Gamma_1, \Gamma_2, \ldots, \Gamma_m$ be a collection of positively oriented simple closed contours. We do not assume these curves are disjoint; they may cross. We do assume that each eigenvalue λ_i lies on no curve Γ_j, and lies inside just one of those contours.

Let $\varphi(t)$ be a function analytic inside and on each curve Γ_i; if φ is multivalued then choose a branch of φ for each Γ_i, not necessarily the same branch for every Γ_i. Then

$$\varphi(B)=\frac{1}{2\pi i}\int_{\cup\Gamma_i} \varphi(\zeta)\cdot(\zeta I-B)^{-1}d\zeta.$$

Of course, different choices of branches for φ on Γ_i may result in different value of $\varphi(B)$. In particular, we can choose φ to be a "ω-shift" function $\varphi(z)=z-k_j\omega$ inside and on Γ_j, for any integers k_j, to obtain the following definition :

Definition 6.1. $C \equiv B \pmod{\omega}$ just when C is a ω-shift function of B :

$$C=\varphi(B)=\frac{1}{2\pi i}\int_{\cup\Gamma_i} \varphi(\zeta)\cdot(\zeta I-B)^{-1}d\zeta. \tag{6.1.4}$$

Note that the eigenvalues of C are $\varphi(\lambda_j)=\lambda_j-k_j\omega$. Therefore, each member of the class of matrices congruent to B is determined by the set of integers $\{k_1, \ldots, k_m\}$.

EXAMPLE. For any matrix B and any integer k,

$$B-k\omega I\equiv B \pmod{\omega}.$$

This follows from choosing $\varphi(z)=z-k\omega$ inside and on every Γ_i.

In order to show that " $\equiv \pmod{\omega}$ " is an equivalence relation on matrices, we need the following lemma.

Lemma 6.1. For any square matrix T,

$$g(\zeta)=g_1(g_2(\zeta)) \quad \text{implies} \quad g(T)=g_1(g_2(T)) \tag{6.1.5}$$

where g_2 is holomorphic in a domain that contains the eigenvalues of T, and g_1 is holomorphic in a domain that contains all the eigenvalues $g_2(\lambda_k)$ of $g_2(T)$, so $g(T)$ may be defined by the Dunford integral.

Proof. See Kato [0], p.45 . ▪

Theorem 6.2. The congruence relation on matrices is an equivalence relation; i.e.,

$$B\equiv B \pmod{\omega}, \tag{6.1.6}$$

$$C\equiv B \pmod{\omega} \quad \text{implies} \quad B\equiv C \pmod{\omega}, \tag{6.1.7}$$

$$A\equiv B \pmod{\omega} \text{ and } B\equiv C \pmod{\omega} \quad \text{implies} \quad A\equiv C \pmod{\omega}. \tag{6.1.8}$$

Proof. (6.1.6) is obvious ($k=0$). For (6.1.7), let $g_2=\varphi$ and define $g_1=\varphi^{-1}$: $\varphi^{-1}(z)=z-(-k_j)\omega$ inside and on Γ_j. Then $g_1(g_2(z))=\varphi^{-1}\varphi(z)=z$ for z in Γ_i, $i=1,\ldots,m$. Thus $C\equiv B \pmod{\omega}$ => $C=\varphi(B)=g_2(B)$. So

$$B=g_1(g_2(B))=g_1(C)=\varphi^{-1}(C).$$

Since φ^{-1} is a ω-shift function, by definition $B\equiv C \pmod{\omega}$. The proof of (6.1.8) is similar : if $A=\varphi_1(B)$ and $B=\varphi_2(C)$, then $\varphi(z)=\varphi_1(\varphi_2(z))$ is also a ω-shift function. Hence $A=\varphi(C)$ implies $A\equiv C \pmod{\omega}$. ▪

The usefulness of the congruence relation on matrices lies in the following theorem.

Theorem 6.3. For any f holomorphic in the complex plane with period ω, if $C \equiv B \pmod{\omega}$ then

$$f(C)=f(B). \tag{6.1.9}$$

Proof. The theorem follows from choosing $g_1=f$ and $g_2=\varphi$ in Lemma 6.1. Since the period of f is ω, $f(z)=f(\varphi(z))$ for any ω-shift function φ. Thus, $f(C)=f(\varphi(B))=f(B)$. ∎

6.2. Argument Reduction for periodic Function

Let MOD(B, ω) denote the class of matrices that are congruent to B modulo ω. For any holomorphic function f with period ω, the computation of $f(B)$ might be difficult when B is large in certain norm; especially when series or polynomials must be used. Theorem 6.3 enables us to pick any C from MOD(B,ω) and compute $f(C)$ instead. Thus, one can reduce the given argument B by choosing from MOD(B,ω) a matrix smaller in norm. We call such a process *argument reduction*. For example, one may choose C to have eigenvalues ζ_i that are smaller in magnitude, yet congruent to the corresponding λ_i; for instance, take $\zeta_i=\lambda_i-k_i\omega$ where k_i is chosen to minimize $|\lambda_i-k\omega|$ over all integers k. More generally, we may ask the following question :

Given a matrix norm, can we characterize the matrix C of minimum norm in MOD(B,ω) ?

It turns out that such a C depends on B and the given norm in a complicated way and may not be unique.

EXAMPLE. Let

$$\omega=1, \quad B=\begin{bmatrix} 0.8 & t \\ 0 & 0.4 \end{bmatrix}.$$

Since B is triangular, any $C\in\mathrm{MOD}(B,\omega)$ is of the following form :

$$C=\begin{bmatrix} 0.8-k_1 & t\times\dfrac{(0.8-k_1)-(0.4-k_2)}{0.8-0.4} \\ 0 & 0.4-k_2 \end{bmatrix}.$$

This form is dictated by commutativity $CB=BC$, see equation (6.2.1). Now, for different values of t and norm, the minimizing C is as follows

Case (a). $t=0$, any of the norms $\|\cdot\|_j$, $j=1,2,\infty$;

$$C=\mathrm{diag}(-0.2,0.4) \quad (k_1=1 \text{ and } k_2=0).$$

Case (b). $t=0.4$, 1-norm ($\|C\|_1=\max\limits_{1\le i\le n}\sum\limits_{j=1}^{n}|a_{i,j}|$);

$$C=B=\begin{bmatrix} 0.8 & 0.4 \\ 0 & 0.4 \end{bmatrix} \quad (k_1=k_2=0).$$

Case (c). $t=1$, ∞-norm ($\|C\|_\infty=\max\limits_{1\le j\le n}\sum\limits_{i=1}^{n}|a_{i,j}|$);

$$C=\begin{bmatrix} -0.2 & 1 \\ 0 & -0.6 \end{bmatrix} \quad (k_1=k_2=1).$$

In general, if $B=\begin{pmatrix} \lambda_1 & t \\ 0 & \lambda_2 \end{pmatrix}$ and $\omega=1$, then the (1,2) element of C is equal to

$$t\times\frac{(\lambda_1-k_1)-(\lambda_2-k_2)}{(\lambda_1-\lambda_2)}=t\times(1-\frac{k_1-k_2}{\lambda_1-\lambda_2}).$$

Thus when $t\neq0$ and λ_1 and λ_2 are sufficiently close, the (1,2) element must be large unless $k_1=k_2$. This suggests that it may be a good idea to constrain

$k_1=k_2$ if λ_1 and λ_2 are close together.

In computing $f(B)$, if the Schur form S of B is used, then argument reduction of a triangular matrix is required. For a triangular S any C in MOD(S,ω) can be computed by a variation of Parlett's recurrence [5], provided the eigenvalues (diagonal) of C are given. We shall be content with a C that minimizes the sizes of the eigenvalues under the constrain $k_i=k_j$ if λ_i is close to λ_j. The numerical method in the next section is based on the following consequences of Definition 6.1 :

If $C\equiv B \pmod{\omega}$, then

$$(a).\quad BC=CB; \tag{6.2.1}$$

$$(b).\quad (Q^{-1}CQ)\equiv Q^{-1}BQ \pmod{\omega}. \tag{6.2.2}$$

Both follow from the fact that $C=\varphi(B)$: $B\varphi(B)=\varphi(B)B$ implies (a) and $Q^{-1}\varphi(B)Q=\varphi(Q^{-1}BQ)$ implies (b).

6.3. Remark

It is possible to define congruence via the Jordan form. Let J_λ denote a Jordan block with eigenvalue λ. Given any B, let

$$Q^{-1}BQ = J = \mathrm{diag}(J_{\lambda_1}, J_{\lambda_2}, \ldots, J_{\lambda_t})$$

be its Jordan normal form (in this case, some of the λ_i may be equal), then by Lemma 6.4 below the only matrices that are congruent to J are $J'=\mathrm{diag}(J_{\lambda_1-k_1\omega}, \ldots, J_{\lambda_t-k_t\omega})$, where k_i are some integers (the only requirement on these integers is $k_i=k_j$ if $\lambda_i=\lambda_j$). Consequently, $C\equiv B \pmod{\omega}$ if and only if C is of the form $QJ'Q^{-1}$.

Lemma 6.4. The only matrices that are congruent to J_λ are $J_{\lambda-k\omega}$, $k=0,\pm1,+12,\cdots$.

Proof. From (6.1.4), if C is congruent to J_λ, then for any simple closed contour Γ that encloses λ, we have, for some integer k,

$$C=\frac{1}{2\pi i}\int_\Gamma (z-k\omega)\cdot(zI-J_\lambda)^{-1}\,dz = \frac{1}{2\pi i}\left[\int_\Gamma z(zI-J_\lambda)^{-1}dz - \int_\Gamma k\omega(zI-J_\lambda)^{-1}dz\right]$$

$$= J_\lambda - k\omega I.$$

Therefore, the only matrices that are congruent to J_λ are $J_\lambda - k\omega I = J_{\lambda-k\omega}$ $(k=0,\pm1,\pm2,\cdots)$. ∎

7. A Numerical Method for Matrix Argument Reduction on Triangular Matrices

In this section a basic method for computing $C=\varphi(S)\in\mathrm{MOD}(S,\omega)$ for a given triangular matrix S and a given φ (see (6.1.4)) is described. This method is unstable if some eigenvalues of S are almost equal but not contiguous on the diagonal. To avoid this situation the diagonal elements must be moved into a suitable order (by unitary similarity transformations).

7.1 The Recurrence Method

Let S be given. For convenience, let $\lambda_i=s_{i,i}$ $(i=1,\ldots,n)$ be the eigenvalues of S. Let the ω-shift function φ be defined by $\varphi(\lambda_i)=\lambda_i-k_i\omega$ $(i=1,\ldots,n)$, with the constrain $k_i=k_j$ if $\lambda_i=\lambda_j$. Since S is triangular, $C=\varphi(S)$ is also triangular and the eigenvalues (diagonal elements) of C are $c_{i,i}=\varphi(\lambda_i)$ $(i=1,\ldots,n)$. Given the set of integers $\{k_1,\ldots,k_n\}$, a simple way to compute C is to use a recurrence based on the commutativity of C and S (cf. [5]). There are three cases :

case 1. *Distinct eigenvalues.* When all λ_i are distinct, C is completely determined by its diagonal elements : comparing the (i,j) element (with $j>i$) of both sides of $0=SC-CS$,

$$0=\sum_{k=0}^{j-i}(c_{i,i+k}\cdot s_{i+k,j}-s_{i,j-k}\cdot c_{j-k,j}).$$

Hence

$$c_{i,j}=\sum_{k=0}^{j-i-1}(c_{i,i+k}\cdot s_{i+k,j}-s_{i,j-k}\cdot c_{j-k,j})/(\lambda_i-\lambda_j). \tag{7.1.1}$$

Thus C can be generated from the diagonal to the upper right hand corner.

case 2. *Contiguous coincident eigenvalues.* When $\lambda_r=\lambda_{r+1}=\lambda_s$, the element $c_{i,j}$ is just equal to $s_{i,j}$ (for $r\le i<j\le s$). To see that, it suffices to consider the case when $r=1$ and $s=n$ (i.e., all the eigenvalues of S are equal). There is only one simple contour Γ that encloses all of the λ_i. By definition,

$$\begin{aligned} C &= \frac{1}{2\pi i}\int_\Gamma (z-k2\pi i)\cdot(zI-B)^{-1}dz \\ &= S-\omega kI. \end{aligned} \tag{7.1.2}$$

Hence,

$$c_{i,j}=s_{i,j} \quad \text{for } i<j. \tag{7.1.3}$$

case 3. *Scattered coincident eigenvalues.* When equal eigenvalues are not contiguous, the recurrence breaks down. For example, formula (7.1.1) yields

$$c_{1,3} = \frac{0}{0}$$

when

$$S = \begin{bmatrix} 2 & 1 & 1 \\ & 0 & 1 \\ & & 2 \end{bmatrix}, \qquad \omega=1 \text{ and } c_{i,i}=0 \text{ for } i=1,2,3.$$

But the (1,3) element of C is well-defined ! To remedy that, we need to move the diagonals of S so that the coincident eigenvalues are contiguous (and hence formula (7.1.3) can be employed). There exist (complex) plane rotations which exchange *adjacent* diagonal elements and preserve triangularity. So it is possible to find a sequence of rotations in plane $(i,i+1)$ that bring any coincident eigenvalues together (see section 7.2). Thus the computation reduces to case 1 and 2.

Remark 1. In general (7.1.3) is true as long as $k_i=k$ for any $r\le i\le s$. That

means that if $k_i, k_{i+1}, \ldots, k_j$ are all equal, the (i,j) element of C is given by (7.1.3), at no expense.

EXAMPLE.

To see how (7.1.1) and (7.1.3) work, let us compute C where $\omega=1$ and

$$S=\begin{bmatrix} 2.101 & 1 & 0 & 0 \\ & 0 & 1 & 0 \\ & & 2.1 & 1 \\ & & & 2.1 \end{bmatrix}. \tag{7.1.4}$$

We pick the diagonal of C to be (0.101, 0.0, 0.1, 0.1) (corresponding to $k_1=2$, $k_2=0$, $k_3=2$ and $k_4=2$, which are chosen to minimize the absolute values of $s_{i,i}-k_i\cdot\omega$). Using (7.1.3) for the (2,3) element and (7.1.1) for the rest, we have (answers are correct to 5 significant decimal digits)

$$C=\begin{bmatrix} 0.101 & 0.048072 & 0.45330 & -0.21586 \\ & 0 & 0.047619 & 0.45351 \\ & & 0.1 & 1 \\ & & & 0.1 \end{bmatrix}. \tag{7.1.5}$$

Remark 2. When 5 significant decimal digit floating point arithmetic is used in the above example, cancellation occurs and instead of (7.1.5) we get

$$fl(C)=\begin{bmatrix} 0.101 & 0.048072 & 0.45300 & 0.51000 \\ & 0 & 0.047619 & 0.45351 \\ & & 0.1 & 1 \\ & & & 0.1 \end{bmatrix}. \tag{7.1.6}$$

Notice that the (1,4) element lost all its digits! This shows that the recurrence is unstable if close eigenvalues are not adjacent to each other. To avoid this problem eigenvalues that are close together should be treated as if they were equal and therefore must be contiguous and have the same k so

that formula (7.1.3) can be used (cf Remark 1).

7.2. Complex Plane Rotation

Reordering the diagonals of S can be done by successive complex rotations (or reflections) in planes $(i,i+1)$. Let

$$\begin{bmatrix} a & c \\ 0 & b \end{bmatrix} = \begin{bmatrix} s_{i,i} & s_{i,i+1} \\ 0 & s_{i+1,i+1} \end{bmatrix}.$$

Our problem is to find an unitary matrix U to swap a and b:

$$U^H \cdot \begin{bmatrix} a & c \\ 0 & b \end{bmatrix} \cdot U = \begin{bmatrix} b & c \\ 0 & a \end{bmatrix}. \tag{7.2.1}$$

Theorem 7.1. Suppose $a \neq b$ (otherwise $U=I$), let $r = \frac{c}{b-a}$. The following U satisfies (7.2.1)

$$U = \frac{1}{\sqrt{1+|r|^2}} \begin{bmatrix} r & -r/\bar{r} \\ 1 & r \end{bmatrix}. \tag{7.2.2}$$

Proof. First of all U is unitary, for

$$\frac{1}{\sqrt{1+|r|^2}} \begin{bmatrix} r & -r/\bar{r} \\ 1 & r \end{bmatrix} \cdot \frac{1}{\sqrt{1+|r|^2}} \begin{bmatrix} \bar{r} & 1 \\ -\bar{r}/r & \bar{r} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

To show

$$\begin{bmatrix} a & c \\ 0 & b \end{bmatrix} \cdot U = U \cdot \begin{bmatrix} b & c \\ 0 & a \end{bmatrix},$$

we multiply out both sides,

$$\begin{bmatrix} ar+c & cr-ar/\bar{r} \\ b & br \end{bmatrix} = \begin{bmatrix} br & rc-ar/\bar{r} \\ b & c+ar \end{bmatrix}.$$

Since $r=\frac{c}{b-a}$, we have $br=ar+c$ and hence establishes the above equality . ▪

Remark. There are many U to swap a and b. For examples,

$$\text{if} \quad U=\frac{1}{\sqrt{1+|r|^2}}\begin{bmatrix} r & r/\bar{r} \\ 1 & -r \end{bmatrix}, \text{ then } \quad U^H\cdot\begin{bmatrix} a & c \\ 0 & b \end{bmatrix}\cdot U=\begin{bmatrix} b & -c \\ 0 & a \end{bmatrix}; \tag{7.2.3}$$

$$\text{if} \quad U=\frac{1}{\sqrt{1+|r|^2}}\begin{bmatrix} r & \frac{r}{|r|} \\ \frac{r}{|r|} & -r \end{bmatrix}, \text{ then } \quad U^H\cdot\begin{bmatrix} a & c \\ 0 & b \end{bmatrix}\cdot U=\begin{bmatrix} b & c\frac{|r|}{r} \\ 0 & a \end{bmatrix}. \tag{7.2.4}$$

The proof of (7.2.3) and (7.2.4) is almost the same as in Theorem 7.1. The U in (7.2.4) has an extra property that it is symmetric.

From now on, we call the similar transformation that exchange adjacent diagonal elements in S a *swap* :

$$\begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & U^H & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}\cdot S\cdot\begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & U & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} = \textit{new } S\ .$$

It is elementary to see that each swap takes $4(n-2)$ ops; here an arithmetic operation is a complex multiplication or division.

7.3. A Variation of the Recurrence Method

Recall that k_i is defined by

$$c_{ii}=\varphi(\lambda_i)=\lambda_i-k_i\omega.$$

When equal k_i's are contiguous, a better way (than (7.1.3) and (7.1.1)) to compute C is by

$$C=S-\omega X \tag{7.3.1}$$

where $X=(x_{i,j})$ is a triangular matrix defined as follow:

$$x_{i,j} = \begin{cases} k_i & \text{if } i=j, \\ 0 & \text{if } k_i=k_{i+1}=\cdots=k_j, \\ \sum_{k=0}^{j-i-1}(x_{i,i+k}s_{i+k,j} - s_{i,j-k}x_{j-k,j})/(\lambda_i-\lambda_j) & \text{if } \lambda_i\neq\lambda_j. \end{cases} \tag{7.3.2}$$

The formulae in (7.3.2) come from forcing $SX=XS$. To verify $C=S-\omega X$, notice that by definition

$$\mathrm{diag}(C)=\mathrm{diag}(S-\omega X).$$

That is, $c_{i,j}=(S-\omega X)_{i,j}$ when $j-i=0$. Assume $c_{i,j}=(S-\omega X)_{i,j}$ for $j-i<k$, $1\leq k$. Consider $(S-\omega X)_{i,j}$ for $i+j=k$. There are two cases :

(1) if $k_i=k_{i+1}=\cdots=k_j$, then (7.3.2) and remark 1 in §7.1 imply

$$(S-\omega X)_{i,j} = s_{i,j} - 0 = c_{i,j};$$

(2) when $k_i\neq k_j$, the relation $XS=SX$ implies $S(S-\omega X)=(S-\omega X)S$ and hence the element $(S-\omega X)_{i,j}$ is determined uniquely by the recurrence formula (7.1.3). Therefore by induction assumption $c_{i,j}=(S-\omega X)_{i,j}$.

Both cases yield $c_{i,j}=(S-\omega X)_{i,j}$ for $j-i=k$; hence, by induction, the equality holds for all $j-i\geq 0$.

Formula (7.3.1) is somewhat preferable to (7.1.1) and (7.1.3). Often ω is not exactly representable in a computer, and (7.3.1) puts the arithmetic that involves ω into the last step, thus preventing the propagation of roundoff error due to the inexact representation of ω.

When the equal k_i's are not contiguous, we can find an unitary transformation U (e.g. a product of complex plane rotations) such that $S'=U^HSU$ has confluent diagonals. Let X' be computed according to (7.3.2) with S replace by S'. We have

$$C=\varphi(US'U^H)=U\varphi(S')U^H$$
$$=U(S'-\omega X')U^H=S-\omega UX'U^H.$$

Thus C retains the form $C=S-\omega X$ where $X=UX'U^H$.

As an example, we compute the example in §7.1 again (using 5 significant digit floating point arithmetic only) by swapping the first and the second diagonal elements before the reduction. Recall

$$S=\begin{bmatrix} 2.101 & 1 & 0 & 0 \\ & 0 & 1 & 0 \\ & & 2.1 & 1 \\ & & & 2.1 \end{bmatrix}.$$

We have

$$U^HSU = S' = \begin{bmatrix} 0 & 1 & 0.90293 & 0 \\ & 2.101 & -0.42976 & 0 \\ & & 2.1 & 1 \\ & & & 2.1 \end{bmatrix}$$

where (using (7.2.1) with $\tau=-0.47596$)

$$U = \begin{bmatrix} -0.42976 & -0.90293 & 0 & 0 \\ 0.90293 & -0.42976 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Thus the corresponding X' of S' is equal to (by (7.3.2))

$$X' = \begin{bmatrix} 0 & 0.95193 & 1.0548 & -0.50229 \\ & 2 & 0 & 0 \\ & & 2 & 0 \\ & & & 2 \end{bmatrix}.$$

Transform it back and subtract it from S to find

$$fl(C)=S-UX'U^H=\begin{bmatrix} 0.101 & 0.04807 & .45331 & -.21586 \\ & 0 & 0.047589 & .45353 \\ & & 0.1 & 1 \\ & & & 0.1 \end{bmatrix}.$$

Recall the correctly rounded C (to five decimal significant digits) is

$$C=\begin{bmatrix} 0.101 & 0.048072 & 0.45330 & -0.21586 \\ & 0 & 0.047619 & 0.45351 \\ & & 0.1 & 1 \\ & & & 0.1 \end{bmatrix}.$$

Here that the (1,4)-th element is correct up to the last digit (compare the completely wrong answer in (7.1.6)). Each element in $fl(C)$ is in absolute error by at most 3×10^{-5}, which is approximately the rounding error of the largest element in C.

It seems that the recurrence formula (7.1.1) always yields accurate results as long as λ_i and λ_j are not close to each other. It worths to re-apply (7.1.1) once more for computing those $x_{i,j}$ such that $k_i \neq k_j$ *after* the whole X has been computed.

7.4. An Algorithm for Computing C

Section 7.3 suggests the following algorithm to compute C:

Algorithm MAR (Matrix Argument Reduction).

[1] Determine k_i $(i=1,\ldots,n)$ so that $\lambda_i - k_i\omega$ is small in magnitude.

[2] Order the (k_i) so that equal k's are contiguous;

[3] Find U so that $U^H SU=S'$ has its diagonals in the new ordering;

[4] compute X' according to (7.3.2) for S';

[5] transform back $X:=UX'U^H$;

[6] recompute X by (7.3.2) on applicable elements;

[7] compute $C := S - \omega X$.

Step [*1*]. As we have mentioned in section 7.1, each k_i should be chosen to minimize $|\lambda_i - k_i\omega|$ under the constrain $k_i = k_j$ whenever λ_i is close to λ_j. In general, we say λ_i and λ_j are close together if $\lambda_i - \lambda_j$ is less than 0.1ω.

Step [*2*]. There are many ways to permute $\{k_1, k_2, \ldots, k_n\}$ so that equal k_i's are contiguous (let us call them *confluent permutations* of $\{k_1, k_2, \ldots, k_n\}$). Since every swap takes $4(n-2)$ (complex) floating point operations (cf. §7.2), one would like to find a confluent permutation that requires fewest swaps to bring the k_i to the new permutation. It turns out that this task is equivalent to the acyclic subgraph problem (cf.[2]), which is known to be NP complete (i.e., exponential time is probably needed to find the minimal solution). In Appendix II.A, we show that one can easily find a confluent permutation which requires fewer than $n(n-1)/4$ swaps. For our purpose this is quite satisfactory.

Step [*3*]. Apply the complex rotations (in §7.2) to transform S to $S' = U^H S U$ according to the new permutation of $\{k_1, k_2, \ldots, k_n\}$. The information of the transformation matrix U should be stored for later use (step [5]).

Step [*4*], [*5*]. These steps are easy to implement : X' can be computed according to (7.3.2) ; then perform the inverse transformation on X' to get $X = UX'U^H$.

Step [*6*]. This step can be best described by the following algorithm :

for $i=1$ to n , $j=i$ to n do

if $(k_i \neq k_j)$ then $x_{i,j} = \sum_{k=0}^{j-i-1} (x_{i,i+k} \cdot s_{i+k,j} - s_{i,j-k} \cdot x_{j-k,j}) / (\lambda_i - \lambda_j)$.

Step [7]. Finally compute $C=S-\omega X$.

Operation count and storage. The major work is in steps [3], [4] and [5]. Since the upper bound on the number of swaps is $n(n-1)/4$ (cf. Appendix II.A) and each swap takes $4(n-2)$ ops, there are at most $n(n-1)(n-2)$ ops for each of [3] and [5]. Step [4] and [6] together take only $\frac{2n^3}{3}$ ops . Hence, the total number of ops needed is at most $2\frac{2}{3}n^3$.(This is the worst case analysis. In general the number of swaps needed is very few, possibly at most $O(n^2)$ in average†. For example, if all k_i are distinct, then no swap is needed !)

For storage, it is possible to implement MAR without using any matrix storage besides S provided one uses (7.1.1) and (7.1.3) instead of (7.3.1-2) in step [4]. However, formulae (7.3.1-2) are recommanded and it would be convenient to have an extra working matrix. This storage requirement is not unreasonable since the objective of applying MAR is the computation of $f(S)$, so there should be at least an available array waiting for the function.

A fortran program of this algorithm is given in Appendix II.B.

7.5. A Simplified Formula for $x_{1,3}$ when $k_1=k_3$.

When $k_1=k_3$ but $k_1\neq k_2$ (if $k_1=k_2=k_3$ then $x_{13}=0$), it is possible to compute x_{13} by a single formula which suffers no cancellation.

† If one uses QR algorithm to get the Schur form, then the equal eigenvalues tend to cluster together on the diagonal. The worst situation simply won't happen.

Theorem 7.2. For $k_1=k_3$ and $k_1\neq k_2$,

$$x_{13} = \frac{s_{12}s_{23}(k_2-k_1)}{(s_{22}-s_{11})(s_{22}-s_{33})} . \tag{7.5.1}$$

Proof. First, we swap s_{33} and s_{22} in S. Take

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & u_{11} & u_{12} \\ 0 & u_{21} & u_{22} \end{bmatrix}$$

where

$$\begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} = \frac{1}{\sqrt{1+|\tau|^2}} \begin{bmatrix} \tau & -\tau/\bar{\tau} \\ 1 & \tau \end{bmatrix} , \qquad \tau = \frac{s_{23}}{s_{33}-s_{22}} .$$

We have

$$S' = P^H S P = \begin{bmatrix} s_{11} & * & * \\ & s_{33} & s_{23} \\ & & s_{22} \end{bmatrix} .$$

According to (7.3.2) (with $k_1=k_3$ in mind),

$$S' = \begin{bmatrix} s_{11} & * & * \\ & s_{33} & s_{23} \\ & & s_{22} \end{bmatrix} \rightarrow X' = \begin{bmatrix} k_1 & 0 & x'_{13} \\ & k_3 & x'_{23} \\ & & k_2 \end{bmatrix}$$

for some x'_{13} and x'_{23}. Transforming it back we have

$$PX'P^H = X = \begin{bmatrix} k_1 & x'_{13}\bar{u}_{12} & x'_{13}\bar{u}_{22} \\ & k_2 & * \\ & & k_1 \end{bmatrix} = \begin{bmatrix} k_1 & x_{12} & x_{13} \\ & k_2 & x_{23} \\ & & k_1 \end{bmatrix} .$$

Since $x_{12} = s_{12}\dfrac{k_2-k_1}{s_{22}-s_{11}} = x'_{13}\bar{u}_{12}$ and

$$\frac{\bar{u}_{22}}{\bar{u}_{12}} = \frac{\bar{\tau}}{\left(-\frac{\bar{\tau}}{\tau}\right)} = -\tau = \frac{-s_{23}}{s_{33}-s_{22}} ,$$

we have

$$
\begin{aligned}
x'_{13}\bar{u}_{22} &= x'_{13}\bar{u}_{12}\cdot\left(\frac{\bar{u}_{22}}{\bar{u}_{12}}\right) \\
&= s_{12}\frac{(k_2-k_1)}{s_{22}-s_{11}}\cdot\frac{\bar{u}_{22}}{\bar{u}_{12}} \\
&= \frac{s_{12}s_{23}(k_2-k_1)}{(s_{22}-s_{11})(s_{22}-s_{33})}\ . \ \blacksquare
\end{aligned}
$$

8. Application to Matrix Exponentials

8.1. The Matrix Exponential

It is well known that

$$e^{z+2\pi i} = e^{z} ;$$

therefore, with $\omega=2\pi i$ Theorem 2.3 asserts

$$e^{S} = e^{C} \qquad \text{where } C=\varphi(S) \in \mathrm{MOD}(S,\omega).$$

So one can apply the algorithm in the previous section and replace S by C for computing the matrix exponential (in this case, $C=\varphi(S)$ is chosen to have eigenvalues close to the real axis). There are two advantages:

(a) The norm of C is usually smaller than the norm of S; therefore, C is less *sensitive than S to the roundoff error in the calculation of matrix* exponentials; especially when the eigenvalues of S have large variation of imaginary parts;

(b) the imaginary parts of the eigenvalues of C are bounded in magnitude.

Property (b) is important in the accurate computation of the exponential divided differences (cf. [4]). Because of the bounded imaginary parts, the coefficients $\Delta_1^k(Z)\exp$ of the Newton polynomial

$$e^{C} = \Delta_1^0(Z)\exp\cdot I + \sum_{k=1}^{n-1}\Delta_1^k(Z)\exp\cdot\prod_{j=1}^{k}(C-\zeta_j I) \tag{8.1.1}$$

can be computed accurately; consequently (8.1.1) becomes one of the most accurate methods for computing the matrix exponential.

As an illustration, we compute† the exponential of the following S using

† The calculation is done on a Vax 11/780 using single precision arithmetic (the machine precision is $\varepsilon=2^{-24}$, which is approximately 7 decimal significant digits).

i,j	ReS	ImS	Re C	Im C
1,1	0.0000000e+00	1.0000000e+02	0.0000000e+00	-5.3096491e-01
1,2	5.0000000e+00	0.0000000e+00	-2.6548386e-02	0.0000000e+00
1,3	-1.2500000e+01	0.0000000e+00	-2.8269482e-01	1.7453294e-02
1,4	4.1687500e+01	0.0000000e+00	5.2827454e-01	0.0000000e+00
1,5	-1.5625000e+02	0.0000000e+00	8.2632446e-01	-1.0206868e-01
1,6	6.2500000e+02	0.0000000e+00	6.2440491e+02	2.8962694e+01
1,7	-2.6042500e+03	0.0000000e+00	1.2380859e+01	3.1803755e+01
2,2	0.0000000e+00	5.0000000e+01	0.0000000e+00	-2.6548243e-01
2,3	5.0000000e+00	0.0000000e+00	2.8761101e-01	0.0000000e+00
2,4	-1.2500000e+01	0.0000000e+00	-2.8269482e-01	-1.7453294e-02
2,5	4.1687500e+01	0.0000000e+00	-2.2161484e-01	6.4788714e-02
2,6	-1.5625000e+02	0.0000000e+00	8.2351685e-01	1.7664604e-01
2,7	6.2500000e+02	0.0000000e+00	-3.3146973e+00	-3.1911498e-01
3,3	0.0000000e+00	1.0000000e+01	0.0000000e+00	-2.5663707e+00
3,4	5.0000000e+00	0.0000000e+00	-2.6548386e-02	0.0000000e+00
3,5	-1.2500000e+01	0.0000000e+00	3.5196972e-01	-9.5199691e-03
3,6	4.1687500e+01	0.0000000e+00	9.4289780e-01	2.8333304e-02
3,7	-1.5625000e+02	0.0000000e+00	-1.2384033e+00	3.5637300e-04
4,4	0.0000000e+00	-4.0000000e+01	0.0000000e+00	-2.3008881e+00
4,5	5.0000000e+00	0.0000000e+00	-2.3598766e-01	0.0000000e+00
4,6	-1.2500000e+01	0.0000000e+00	-1.5802860e-01	-7.4799755e-03
4,7	4.1687500e+01	0.0000000e+00	2.1535873e-01	6.2332950e-03
5,5	0.0000000e+00	-1.0000000e+02	0.0000000e+00	5.3096491e-01
5,6	5.0000000e+00	0.0000000e+00	-2.6548386e-02	0.0000000e+00
5,7	-1.2500000e+01	0.0000000e+00	6.6370964e-02	0.0000000e+00
6,6	0.0000000e+00	1.0000000e+02	0.0000000e+00	-5.3096491e-01
6,7	5.0000000e+00	0.0000000e+00	-2.6548386e-02	0.0000000e+00
7,7	0.0000000e+00	2.0000000e+02	0.0000000e+00	-1.0619298e+00

Table 8.1.2. Matrix S and the computed C .

$$cond(S) \approx 3.1\,, \quad cond(C) \approx 1$$

Newton's interpolating polynomial method (the coefficients are computed by algorithms in [4]) on $fl(C)$. Every pair of numbers in the following tables represent the real and imaginary part of the corresponding element. As a summary, we have†

$$\frac{\| e^S - fl(e^C)\|_1}{\| e^S\|_1} \approx 5.7\varepsilon\,, \quad \varepsilon = 2^{-24} \tag{8.1.3}$$

† If we apply our exponential directly to S (using algorithm SH(II) in [4] for the divided differences), then $\frac{\| e^S - fl(e^S)\|_1}{\| e^S\|_1} \approx 1310.0\varepsilon$; almost 200 times larger than the error in (8.1.3).

For reference, we give the exponential condition number of S and C (see part I) :

$$cond(S) \approx 31.0 \quad , \quad cond(C) \approx 1$$

i,j	correctly rounded e^S		computed e^C	
(1,1)	8.623189e-01	-5.063657e-01	8.623189e-01	-5.063656e-01
(1,2)	-2.439908e-02	1.026472e-02	-2.439908e-02	1.026472e-02
(1,3)	7.868409e-03	2.396153e-01	7.868374e-03	2.396156e-01
(1,4)	7.220490e-02	-4.624961e-01	7.220464e-02	-4.624946e-01
(1,5)	7.280552e-01	-2.936441e-02	7.280577e-01	-2.936442e-02
(1,6)	5.530223e+02	-2.910502e+02	5.530224e+02	-2.910502e+02
(1,7)	2.462661e+01	1.804374e+01	2.462648e+01	1.804387e+01
(2,2)	9.649660e-01	-2.623748e-01	9.649660e-01	-2.623749e-01
(2,3)	3.520579e-02	-2.255047e-01	3.520578e-02	-2.255047e-01
(2,4)	-8.053124e-02	2.258143e-01	-8.053132e-02	2.258146e-01
(2,5)	-1.700779e-01	-1.040713e-02	-1.700736e-01	-1.040661e-02
(2,6)	8.959118e-01	-2.774917e-01	8.958906e-01	-2.774836e-01
(2,7)	-2.805674e+00	1.929768e+00	-2.805659e+00	1.929756e+00
(3,3)	-8.390715e-01	-5.440211e-01	-8.390715e-01	-5.440211e-01
(3,4)	2.010921e-02	1.721335e-02	2.010920e-02	1.721335e-02
(3,5)	1.143518e-01	-1.989942e-01	1.143519e-01	-1.989943e-01
(3,6)	3.831667e-02	-7.865904e-01	3.831670e-02	-7.865940e-01
(3,7)	2.745636e-01	1.096910e+00	2.745654e-01	1.096918e+00
(4,4)	-6.669381e-01	-7.451131e-01	-6.669381e-01	-7.451132e-01
(4,5)	-1.042899e-01	1.274381e-01	-1.042899e-01	1.274381e-01
(4,6)	-2.586806e-02	1.337202e-01	-2.586810e-02	1.337204e-01
(4,7)	-1.971859e-02	-1.977346e-01	-1.971872e-02	-1.977361e-01
(5,5)	8.623189e-01	5.063657e-01	8.623189e-01	5.063656e-01
(5,6)	-2.531828e-02	0.000000e+00	-2.531828e-02	-7.368618e-19
(5,7)	5.779887e-02	-1.574674e-02	5.779857e-02	-1.574666e-02
(6,6)	8.623189e-01	-5.063657e-01	8.623189e-01	-5.063656e-01
(6,7)	-1.834658e-02	1.875656e-02	-1.834658e-02	1.875656e-02
(7,7)	4.871877e-01	-8.732973e-01	4.871877e-01	-8.732973e-01

Table 8.1.4. The computed e^C and the correctly rounded e^S.

$$\| e^S - fl(e^C) \| \approx 5.7 \times 2^{-24} \approx 10^{-6}.$$

8.2. Stability analysis of Step [4]

When X' is computed according to $X'S'=S'X'$, it may happen that the computed X' is nowhere close to the exact answer even though $X'S'\approx S'X'$. In this section, we show that the commutativity play an important role and is the essential condition determining whether e^S is close to e^C. Thus it is not essential that X' be accurate, only that it must nearly commute with S'. Recall C is computed by $S-\omega X$ ($\omega=2\pi i$) and $X=UX'U^H$. Let $E=X'S'-S'X'$, we have $E=U^HXSU-U^HSXU=U^H(XS-SX)U$; hence $\|E\|=\|XS-SX\|$, since U is unitary.

Theorem 8.1. With C,X, and E as given above

$$\|e^S-e^C\|\le\frac{1}{2}\|E\|\,e^{2\pi\|X\|+\|C\|}. \tag{8.2.1}$$

We first prove the following two lemmas.

Lemma 8.2. For square conformable matrices A and B

$$e^{t(A+B)}-e^{tA}\cdot e^{tB}=\int_0^t e^{(t-\tau)(A+B)}(Be^{\tau A}-e^{\tau A}B)e^{\tau B}\,d\tau$$

Proof of Lemma 8.2. Let $Y(t)=e^{t(A+B)}-e^{tA}e^{tB}$. The derivative of $Y(t)$ is

$$\begin{aligned}\frac{d}{dt}Y(t)&=(A+B)e^{t(A+B)}-Ae^{tA}e^{tB}-e^{tA}Be^{tB}\\&=(A+B)Y(t)+(Be^{tA}-e^{tA}B)e^{\tau B}.\end{aligned}$$

Therefore

$$\frac{d}{dt}(e^{-t(A+B)}Y(t))=e^{-t(A+B)}(Be^{tA}-e^{tA}B)e^{tB}.$$

Since $Y(0)=0$,

$$Y(t)=\int_0^t e^{(t-\tau)(A+B)}(Be^{\tau A}-e^{\tau A}B)e^{\tau B}\,d\tau. \quad \blacksquare$$

Lemma 8.3.

$$Be^{tA}-e^{tA}B=\int_0^t e^{(t-\tau)A}(BA-AB)e^{\tau A}d\tau.$$

Proof of Lemma 8.3. Let $Z(t)=Be^{tA}-e^{tA}B$. We have

$$\begin{aligned}\frac{d}{dt}Z(t)&=BAe^{tA}-Ae^{tA}B\\&=BAe^{tA}-ABe^{tA}+ABe^{tA}-Ae^{tA}B\\&=AZ(t)+(BA-AB)e^{tA}.\end{aligned}$$

Therefore,

$$e^{-tA}\left(\frac{d}{dt}Z(t)-AZ(t)\right)=e^{-tA}(BA-AB)e^{tA}$$

$$\frac{d}{dt}(e^{-tA}Z(t))=e^{-tA}(BA-AB)e^{tA}.$$

Since $Z(0)=0$

$$e^{-tA}Z(t)=\int_0^t e^{-\tau A}(BA-AB)e^{\tau A}d\tau$$

and finally

$$Z(t)=\int_0^t e^{(t-\tau)A}(BA-AB)e^{\tau A}d\tau. \bullet$$

Proof of Theorem 8.1. After the transformation in step [3], the diagonal of X' is of form

$$\mathrm{diag}(X')=(a,a,\ldots,a,b,b,\ldots,c,c,\ldots,c),$$

where $a,b,\ldots,c$ denote distinct integers. According to (7.3.2), X' has the following form

$$X' = \begin{bmatrix} a & 0 & 0 & * & * & * & * \\ & \ddots & 0 & * & * & * & * \\ & & a & * & * & * & * \\ & & & \ddots & * & * & * \\ & & & & c & 0 & 0 \\ & & & & & \ddots & 0 \\ & & & & & & c \end{bmatrix}$$

We claim that

$$\exp(2\pi i X') = I.$$

It suffices to consider the case that diag(X') has only two distinct integers,

$$X' = \begin{bmatrix} a & 0 & 0 & 0 & * & * & * \\ & \ddots & 0 & 0 & * & * & * \\ & & a & 0 & * & * & * \\ & & & a & * & * & * \\ & & & & b & 0 & 0 \\ & & & & & \ddots & 0 \\ & & & & & & b \end{bmatrix} = \begin{bmatrix} aI_1 & (*) \\ & bI_2 \end{bmatrix}.$$

It is obvious that $F=\exp(2\pi i X')$ is of form

$$\exp(2\pi i X') = F = \begin{bmatrix} I_1 & F_{12} \\ & I_2 \end{bmatrix}.$$

Since $FX'=X'F$, by comparing the (1,2)-th block of FX'

$$(*)I_1+F_{12}\cdot bI_2 = aI_1F_{12}+(*)I_2.$$

Since $a \neq b$, $F_{12}=0$.

From step [5] of MAR, $X=UX'U^H$; hence

$$e^{2\pi i X}=U\cdot I\cdot U^H=I.$$

Therefore, if we write

$$e^S-e^C=e^{C+2\pi i X}-e^{2\pi i X}\cdot e^C,$$

then Lemma 8.2 implies (with $t=1$)

$$e^S - e^C = \int_0^1 e^{(1-\tau)S} \cdot (Ce^{\tau 2\pi i X} - e^{\tau 2\pi i X} C) \cdot e^{\tau C} \, d\tau. \tag{8.2.2}$$

Using $\| e^S \| \le e^{\| S \|}$, Lemma 8.3 implies (for $\tau \ge 0$)

$$\begin{aligned} &\| Ce^{\tau 2\pi i X} - e^{\tau 2\pi i X} C \| \\ &\quad \le \| C(2\pi i \tau X) - (2\pi i \tau X) C \| \cdot \int_0^\tau e^{(\tau-\xi)\| 2\pi i X \|} \cdot e^{\xi \| 2\pi i X \|} \, d\xi \\ &\quad \le 2\pi \| (C + 2\pi i X)(X) - (X)(C + 2\pi i X) \| \cdot \tau \cdot e^{2\pi\tau \| X \|} \\ &\quad \le 2\pi \| SX - XS \| \, \tau \cdot e^{\tau \| X \|}. \end{aligned}$$

Since $\| E \| = \| SX - XS \|$ and $\| S \| \le 2\pi \| X \| + \| C \|$, (8.2.2) implies

$$\| e^S - e^C \| \le \| E \| \, e^{(2\pi \| X \| + \| C \|)} \cdot \int_0^1 \tau \, d\tau;$$

and the bound (8.2.1) follows. ▪

Remark. Using different bounds on $\| e^S \|$ yields different results. For example, if one uses $\| e^S \| \le e^{\mu(S)}$, where $\mu(S)$ is the log norm of S ($\mu(S) := \max\{\mathrm{Re}(\lambda_i) : \lambda_i \in \lambda(S)\}$), then the same reasoning as above yields

$$\| e^S - e^C \| \le \frac{\| E \|}{2} e^{2\pi\mu(X) + \mu(C)}.$$

See part I Appendix I.A for other useful bounds on $\| e^S \|$.

Appendix II.A. (A Bound on The Number of Swaps)

Our intention here is to examine how many adjacent swaps are needed to reorder n integers $\mathbf{k}=(k_1,k_2,\ldots,k_n)$ so that coincident k_i's go together. We call such an ordering *confluent* (or a *confluent permutation* of $\{k_1,\ldots,k_n\}$). To clarify our meaning of swapping, we will use *swap* to denote the exchange of *adjacent* elements, and *exchange* to denote the exchange of any two elements.

There are many ways to reorder $\mathbf{k}$. For example, if $\mathbf{k}=(2,3,2,5,3,2)$, then there are six confluent permutations of them:

$$(2,2,2,5,3,3),\ (5,3,3,2,2,2),\ \cdots,\ (3,3,5,2,2,2).$$

Our problem here is to determine a confluent permutation $\mathbf{k}'$ for $\mathbf{k}$ such that the transformation from $\mathbf{k}$ to $\mathbf{k}'$ requires a nearly minimal number of swaps. To formulate the problem clearly, we follow [1] in using the notion of *multiset*. A multiset is like a set (where a set is a collection of distinct elements) except that it can have repetitions of identical elements. For example,

$$M=\{a,a,a,b,b,c,d,d,d,d\},$$

which contain 3 a's, 2 b's, 1 c, and 4 d's. We may also indicate the multiplicities of elements in another way, namely

$$M=\{3\cdot a,2\cdot b,c,4\cdot d\}.$$

A permutation of M is an arrangement of its elements , e.g. ,

$$c\ a\ b\ d\ d\ a\ b\ d\ a\ d\ .$$

From another point of view we could call this a string of letters, containing 3 a's, 2b's, 1c, and 4 d's. For convenience, we use per(M) to denote the set of all permutations of M where $\mathbf{x}=(x_1,\ldots,x_n)$ is a typical element.

We consider the multiset of integers. Let $M=\{k_1,k_2,\ldots,k_n\}$ where k_i's are integers; then M is a multiset. If $a_1,\ldots,a_l$ denote the distinct integers in M, then M can be written as

$$M=\{n_1\cdot a_1,\ n_2\cdot a_2,\ \ldots,\ n_l\cdot a_l\},$$

where n_i is the multiplicity of a_i. A *confluent permutation* of M is a permutation of M, $(a_p)\in\mathrm{per}(M)$, such that

$$\mathbf{a}_p=(n_{p(1)}\cdot a_{p(1)},\ n_{p(2)}\cdot a_{p(2)},\ \ldots,\ n_{p(l)}\cdot a_{p(l)}),$$

where $p:p(i),\ i=1,2,\ldots,l$ is a permutation of $1,2,\ldots,l$, i.e., $p\in\mathrm{per}\{1,2,\ldots,l\}$. Since we are interested in the number of swaps needed to transform one permutation to another, we define $\tau(\mathbf{x},\mathbf{y})$ to be the minimal number of swaps needed to change $\mathbf{x}$ to $\mathbf{y}$.

For example, if $\mathbf{x}=(2,3,2,5,3,2)$ and $\mathbf{y}=(2,2,2,5,3,3)$, then $\tau(\mathbf{x},\mathbf{y})=5$. Each new line is obtained from the previous one by a swap

$$\begin{aligned}\mathbf{x} &= (2,\ 3,\ 2,\ 5,\ 3,\ 2)\\ &\to (2,\ 2,\ 3,\ 5,\ 3,\ 2)\\ &\to (2,\ 2,\ 5,\ 3,\ 3,\ 2)\\ &\to (2,\ 2,\ 5,\ 3,\ 2,\ 3)\\ &\to (2,\ 2,\ 5,\ 2,\ 3,\ 3)\\ &\to (2,\ 2,\ 2,\ 5,\ 3,\ 3) = \mathbf{y}.\end{aligned}$$

With the above notation, we restate our problem as follow

Problem. Given any permutation $\mathbf{k}=(k_1,\ldots,k_n)$ of M, find $p\in\mathrm{per}\{1,2,\ldots,l\}$ (l is the number of distinct integers in M) so that the $\tau(\mathbf{k},\mathbf{a}_p)$ is minimal over $\mathrm{per}\{1,2,\ldots,l\}$.

This is a NP-complete problem, that is, it probably requires exponential time (in l) to find the minimal solution (for one thing, there are $l!$ choice of

p). In fact, this problem is related to the *acyclic subgraph problem*, (see [2]), which is well known to be NP-complete. We'll establish the connection between our problem and the acyclic subgraph problem later. First, we show that we can always find a p such that $r(\mathbf{k},\mathbf{a}_p)$ is less than $\frac{n^2}{4}(1-\frac{1}{l}) \le n(n-1)/4$. Let $-p$ denotes the reverse permutation of p: if $p=p(1),p(2),\ldots,p(l)$, then $-p=p(l),p(l-1),\ldots,p(1)$ (i.e., $(-p)(i)=p(l-i)$). Also we use I to denote the identity permutation of per$\{1,2,\ldots,l\}$. Note that $-p$ is not the inverse of p.

Theorem A.1. For any given p,

$$\min\{r(\mathbf{k},\mathbf{a}_p),\ r(\mathbf{k},\mathbf{a}_{-p})\} \le \frac{n^2}{4}(1-\frac{1}{l}).$$

Proof. It suffices to consider $p=I$, for one can always replace a_i by $a_{p(i)}$. Let $\eta_{i,j}$ denote the number of a_j in front of a_i. By that we mean for each a_i, we count how many a_j's are on the left of this a_i (in $\mathbf{k}$), then take the sum over all a_i and call it the *number* of a_j *before* a_i. For example, if

$$\mathbf{k}=(7,8,7,5,8,7) \quad \text{where } a_1=7,\ a_2=8,\ a_3=5,$$

then $\eta_{2,3}=1$, $\eta_{1,3}=1$ and $\eta_{1,2}=3$. ($\eta_{i,j}$ can be considered as the number of *inversions* between a_i and a_j. This is a generalization of the inversions defined in [1] p.11: given (i,j) in order, and a permutation $\mathbf{k}=(k_1,\ldots,k_n)$, we call a pair (k_{m1},k_{m2}) an inversion of the permutation with respect to (a_i,a_j) if $k_{m1}=a_j$ is before (to the left of) $k_{m2}=a_i$. $\eta_{i,j}$ is just the total number of inversions with respect to (a_i,a_j).)

Lemma A.2.

$$r(\mathbf{k},\mathbf{a}_I)) = \sum_{i<j}\eta_{i,j}. \tag{a1}$$

Proof of Lemma A.2. By the definition of $\eta_{i,j}$, there are $\eta_{i,j}$ a_j before a_i, therefore at least $\eta_{i,j}$ swaps are needed between a_i and a_j in order to bring all a_i before a_j. It is then obvious that the number of swaps needed to bring all a_1 to the left of all $a_2, a_3, \ldots, a_l$ is at least $\sum_{i=2}^{n}\eta_{1,i}$. However, since there are only $\sum_{i=2}^{n}\eta_{1,i}$ non-a_1 before a_1, we can move all a_1 to the far left in exactly $\sum_{i=2}^{n}\eta_{1,j}$ swaps. The same reasoning shows that there are $\sum_{i=3}^{n}\eta_{2,i}$ swaps needed to bring all a_2 to the left of all $a_3, a_4, \ldots, a_l$, and so on. The lemma is proved. ∎

Notice that Lemma A.2 also implies that $\sum_{i>j}\eta_{i,j}$ is equal to $r(\mathbf{k}, \mathbf{a}_{-I})$.

Lemma A.3. $\sum_{i\neq j}\eta_{i,j} \leq \frac{n^2}{2}(1-\frac{1}{l})$

Proof of Lemma A.3. Recall n_i is the multiplicity of a_i. We first show that

$$\eta_{i,j}+\eta_{j,i}=n_i n_j \tag{a2}$$

for $i\neq j$. There are always $n_j \cdot a_j$ on both sides of each of a_i. Therefore, the number of swappings to bring each of a_i to the left of all a_j and to the right of all a_j is n_j. Since there are n_i of a_i, the total swapping is $n_i n_j$. Because of (a2), we have

$$\begin{aligned}
\sum_{i\neq j}\eta_{i,j} &= \sum_{i>j}\eta_{i,j} + \sum_{i>j}\eta_{j,i} \\
&= \sum_{i>j}(\eta_{i,j} + \eta_{j,i}) = \sum_{i>j} n_i n_j \\
&= \sum_{1\leq i,j\leq n}\frac{1}{2}n_i n_j - \frac{1}{2}\sum_{i=1}^{l} n_i^2 \\
&= \frac{1}{2}(n_1+\cdots+n_l)^2 - \frac{1}{2}\sum_{i=1}^{l} n_i^2
\end{aligned}$$

$$= \frac{1}{2}n^2 - \frac{1}{2}\sum_{i=1}^{l} n_i^2$$
$$\leq \frac{n^2}{2}(1-\frac{1}{l}).$$

The last inequality follows from Cauchy-Schwartz inequality

$$(n_1+\cdots+n_l)^2 \leq l\cdot(n_1^2+\cdots n_l^2). \ \blacksquare$$

Lemma A.3 implies that either $\sum_{j>i}\eta_{i,j}$ or $\sum_{i>j}\eta_{i,j}$ is $\leq \frac{n^2}{4}(1-\frac{1}{l})$; therefore, by Lemma A.2, either $r(\mathbf{k},\mathbf{a}_l)$ or $r(\mathbf{k},\mathbf{a}_{-l})$ is less than $\frac{n^2}{4}(1-\frac{1}{l})$. This completes the proof of the theorem. ▪

Remark 1. Theorem A.1 implies $\frac{n^2}{4}(1-\frac{1}{l})$ is an upper bound on $\min_p r(\mathbf{k},\mathbf{a}_p)$. This bound is attainable, e.g., $(k_1,\ldots,k_n)=(1,2,\ldots,\frac{n}{2},\frac{n}{2},\ldots,1)$ when n is even. One can easily see that this example takes at least $n(n-2)/4$ swaps to bring all pair numbers together.

Remark 2. From the proof of the theorem our problem is equivalent to

$$\min_p r(\mathbf{k},\mathbf{a}_p), \tag{a3}$$

or,

$$\min_p \sum_{p(i)<p(j)} \eta_{p(i),p(j)}. \tag{a4}$$

If we change min to max, then (a4) is exactly *the quadratic assignment problem* ([2]), a NP-complete problem. Various methods (see [2]) have been suggested to find the maximal (or minimal) solution. However, in our Application, there is little incentive for a minimal solution. Since either a_l or a_{-l}

takes less than $n(n-1)/4$ swaps, it is quite sufficient to choose the 'smaller' one (the example in Remark 1 shows that one cannot in general expect a further reduction on the number of swaps).

In Appendix II.B, a Fortran program for matrix argument reduction is given in which the values of a_i's are set up according to the first appearance of distinct integer in $\mathbf{k}$ from left to right, and then a comparison on $\tau(\mathbf{k},\mathbf{a}_I)$ and $\tau(\mathbf{k},\mathbf{a}_{-I})$ determines which permutation will be used.

Appendix II.B: Program Listing and Usage

The subroutine for matrix argument reduction is "modt", which will return mod(T) in the upper part of T (the lower part will store the original T). The user must provide subroutine "mod1", the one dimensional argument reduction function. The details of the numerical method can be found in section 7.

Subroutine modt(m,n,t,f,z,w,tau).

Given a triangular matrix T, this subroutine computes mod(tau*T) according to the algorithm MAR in section 7. The resulted matrix will be stored in the strict upper part of T with the diagonal in $z(1),\ldots,z(n)$. The original matrix will be saved in the lower T. The user must provide the subroutine mod1. There is a subprogram "swap".

m	the global dimension of matrix t,f
n	the dimension of t,f
t	input complex matrix T
f	working complex matrix
z	output complex vector, store the diagonal of mod(tau*T)
p,q	integer working array.
tau	input complex parameter.

```
      subroutine modt(m,n,t,f,z,p,q,tau)
      complex t(m,n),f(m,n),z(n),tau
      integer p(n),q(n)
c This subroutine computes the argument of tau*t. Result is stored in strict
c upper t with z the diagonal. The original triangular will be stored in
c the lower t with diagonal. An user provided subroutine "mod1" is required.
      complex x,inf,w
      inf=(10e37,10e37)
c ####
      w= user provided (the period)
c ####
      do 10 i=1,n
      p(i)=0
      z(i)=tau*t(i,i)
      call mod1(z(i),x,q(i))
10    continue
      do 20 i=1,n
      do 20 j=1,n
      x=z(i)-z(j)
      if(abs(real(x))+abs(imag(x)).gt.0.1e0) goto 20
      if(q(i).gt.q(j)) q(i)=q(j)
      if(q(i).lt.q(j)) q(j)=q(i)
```

```
20      continue
c ...find out the ordering
        p(1)=1
        l=1
        in=0
        do 40 j=2,n
        do 30 i=1,j-1
        if(p(j).eq.0) then
                if(q(i).eq.q(j)) p(j)=p(i)
                if(q(i).ne.q(j)) in=in+1
        else
                if(p(j).gt.p(i)) in=in+1
                if(p(j).lt.p(i)) in=in-1
        endif
        f(i,j)=tau*t(i,j)
        t(j,i)=inf
30      continue
        if(p(j).ne.0) goto 40
        l=l+1
        p(j)=l
40      continue
c ...swapping forward
        itest=0
        isq=sign(1,in)
        do 50 i=2,n
        do 50 j=n,i,-1
        if(isq*(p(j)-p(j-1)).ge.0) goto 50
        x=f(j-1,j)/(z(j)-z(j-1))
        t(j,i-1)=conjg(x)
        call swap(m,n,f,z,q,j,x)
        itest=1
        l    =p(j)
        p(j)  =p(j-1)
        p(j-1)=l
50      continue
c ...reduction
        do 70 j=1,n-1
        do 70 i=1,n-j
        if(p(i).eq.p(i+j)) then
                f(j+i,i)=f(i,j+i)
                f(i,i+j)=0
        else
                x=f(i,i+j)*cmplx(q(i+j)-q(i))
                do 60 k=1,j-1
60              x=x+f(i+k,i)*f(i+k,i+j)-f(i,i+j-k)*f(i+j,i+j-k)
                f(i+j,i)=f(i,i+j)
                f(i,i+j)=x/(z(i+j)-z(i))
        endif
70      continue
c ...back swapping
        do 80 j=n-1,1,-1
        do 80 i=j+1,n
        x=t(i,j)
```

```
      if(x.eq.inf) goto 80
      call swap(m,n,f,z,q,-i,x)
80    continue
c ...final reduction
      if(itest.eq.0) goto 110
      do 100 i=n-1,1,-1
      itest=0
      do 100 j=i+1,n
      if(q(i).eq.q(j)) then
            if(itest.eq.0) f(i,j)=0
            if(itest.ne.0.and.j-i.eq.2) f(i,j)=t(i,i+1)*t(i+1,i+2)*
     *      cmplx(q(i+1)-q(i))/(t(i+1,i+1)-t(i,i))/(t(i+1,i+1)-t(j,j))
      else
            itest=1
            x=t(i,j)*cmplx(q(j)-q(i))
            do 90 k=1,j-i-1
90          x=x+t(i,i+k)*f(i+k,j)-f(i,j-k)*t(j-k,j)
            f(i,j)=x*tau/(z(j)-z(i))
      endif
100   continue
110   do 120 i=1,n
      call mod1(z(i),x,k)
      z(i)=x+cmplx(q(i)-k)*w
      do 120 j=i+1,n
      t(j,i)=t(i,j)
120   t(i,j)=tau*t(j,i)-f(i,j)*w

      return
      end



      subroutine swap(m,n,f,z,q,k,x)
      complex f(m,n),z(n),u11,u12,u21,x
c This subroutine swap the |k|-th and |k|-1-th diagonal of input matrix f.
c If input k is positive, than we perform U^H FU, else
c the inverse transformation UFU^H, where U is defined as
c in section 7.2.
      integer q(n)
      u21=cmplx(1e0/sqrt(1e0+real(conjg(x)*x)))
      u11=x*u21
      u12=u21
      if(x.eq.(0e0,0e0)) goto 5
      if(k.gt.0) then
        u12=-u11/conjg(x)
      else
        k=-k
        u21=-u11/conjg(x)
      endif
5     continue
      do 10 i=1,k-2
      x      =f(i,k-1)*u11+f(i,k)*u21
      f(i,k)  =f(i,k-1)*u12+f(i,k)*u11
```

```
10      f(i,k-1)=x
        u11=conjg(u11)
        u21=conjg(u21)
        u12=conjg(u12)
        do 20 j=k+1,n
        x      =f(k-1,j)*u11+f(k,j)*u21
        f(k,j)  =f(k-1,j)*u12+f(k,j)*u11
20      f(k-1,j)=x
        i    =q(k)
        q(k)  =q(k-1)
        q(k-1)=i
        x    =z(k)
        z(k)  =z(k-1)
        z(k-1)=x
        return
        end


        subroutine mod1 (z,x,k)
        complex x,z
        integer k
        ###
        w= user given
        ###

        .....User provided.....
        Given argument x, return the reduced argument z so that
        |z|=|x-kw| is minimized over all integers k.

        return
        end
```

REFERENCES

[0] T. Kato, *Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1966.

[1] D.E. Knuth, *The Art of Computer Programming Vol. 3*, Addison Wesley, 1973.

[2] H.W. Lenstra, Jr., *The acyclic subgraph problem*, Afdeling Mathematicshe Besliskunde, BW 26/73, July.

[3] F.R. Gantmacher *Theory of Matrices Vol I*, Chelsea, New York, 1959.

[4] A. McCurdy, K.C. Ng and B.N. Parlett, *Accurate Computation of Divided Differences of the Exponential Function*. 1983. (to appear)

[5] B.N. Parlett, *Computation of Functions of Triangular Matrices*, Memorandum No ERL-M481, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, 1974.

[6] B.N. Parlett, *A Program to Swap the Diagonal Block*, Memorandum No ERL-M77/66, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, 1977.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A140069	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CONTRIBUTIONS TO THE COMPUTATION OF THE MATRIX EXPONENTIAL		5. TYPE OF REPORT & PERIOD COVERED Unclassified
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Kwok Choi Ng†		8. CONTRACT OR GRANT NUMBER(s) N00014-76-0013
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California Berkeley, CA 94720		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE February 1984
		13. NUMBER OF PAGES 72
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis consists of two parts. First, a condition number for the exponential of a triangular matrix S is introduced. It measures how sensitive is e^S to relatively small perturbations in the elements of S. Second, a new technique (*matrix argument reduction*) for computing periodic matrix functions is described and discussed in detail. By applying this technique to the computation of e^S one can always reduce the problem to one in which the eigenvalues lie close to the real axis.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This report was done with support from the Center for Pure and Applied Mathematics. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of the Center for Pure and Applied Mathematics or the Department of Mathematics.

LMED

-8