Computing Matrix-Valued Nevanlinna-Pick Interpolation

Guanrong Chen*
Department of Electrical Engineering
University of Houston
Houston, Texas 77204

and

Çetin Kaya Koç†
Department of Electrical and Computer Engineering
Oregon State University
Corvallis, Oregon 97331

Submitted by A. C. Antoulas

ABSTRACT

We describe a computational method, known as the Nevanlinna algorithm, for the matrix-valued Nevanlinna-Pick interpolation. The original interpolation problem formulated using the Carathéodory class of matrix-valued rational functions is first converted to an equivalent setting using the Schur class of rational functions. As a result, the necessary and sufficient Pick's condition for the interpolation becomes consistent with the scalar-valued formulation, so that some efficient techniques developed for the scalar-valued interpolation can be employed or modified for the matrix-valued case. We give a brief, yet sufficiently clear, derivation and a detailed arithmetic complexity analysis for the algorithm. We show that an n-point matrix-valued Nevanlinna-Pick interpolation using the new algorithm requires approximately $95nm^3$ complex arithmetic operations, where m is the matrix dimension.

1. INTRODUCTION

The classical Nevanlinna-Pick interpolation problem [22, 21] has recently been shown to be very useful in systems control engineering, particularly in

*Supported by the President's Research and Scholarship Fund, University of Houston.
†Supported in part by the U.S. Army Research Office grant DAAL03-91-G-0106.

655 Avenue of the Americas, New York, NY 10010 0024-3795/94/$7.00

the H^∞-control theory [4, 5], stable systems design [2], and system sensitivity minimization [26]. From a mathematical point of view, the Nevanlinna-Pick interpolation has been generalized to obtain very abstract results [20, 12, 23, 4, 11], which have also been directly and indirectly applied to systems control engineering and many other practical areas [10, 26, 4, 2, 3, 5]. On the other hand, numerical algorithms for such an interpolation have also been proposed [1].

We have investigated the computational complexity and parallelism of the Nevanlinna-Pick interpolation for the scalar-valued case in [18, 19]. Although the matrix-valued interpolation is more difficult, it is more important since it provides optimal solutions for many multiinput multioutput control systems, which are the most demanding settings in applications. In this paper, we describe a computational method, known as the Nevanlinna algorithm [9], for the calculation of the matrix-valued Nevanlinna-Pick interpolation, and give a detailed analysis of its computational complexity.

Let $M(z)$ be an $m \times m$ matrix-valued rational function in the complex variable z, and denote by

$$\mathscr{C} = \{M(z) : M(z) + M^*(z) \geqslant 0 \text{ in } |z| < 1\}$$

the so-called Carathéodory class of matrix-valued rational functions, where M^* is the complex conjugate transpose of M and $M + M^* \geqslant 0$ means that the matrix $M + M^*$ is nonnegative definite. Given n distinct points $z_1, z_2, \ldots, z_n$ in the unit disk $|z| < 1$ and n $m \times m$ complex matrices $W_1, W_2, \ldots, W_n$ in the Carathéodory class $\mathscr{C}$, the problem is to find an $m \times m$ rational matrix $\Phi(z)$ in $\mathscr{C}$ such that

$$\Phi(z_k) = W_k, \qquad k = 1, 2, \ldots, n.$$

It is well known [8, 9] that this problem is solvable if and only if the Pick matrix P_c defined by

$$P_c = \left[\frac{W_k + W_l^*}{1 - \bar{z}_k z_l}\right]_{(mn)\times(mn)}$$

is nonnegative definite. In general, this interpolation problem has nonunique solutions even in the scalar-valued setting, as shown in [19].

2. REFORMULATION OF THE PROBLEM

Since the Pick matrix P_c is different from that of the scalar-valued setting [5, 18, 19], we first reformulate the problem as follows: Introduce the so-called

Schur class $\mathscr{S}$ defined by

$$\mathscr{S} = \{M(z) : \|M\| \leqslant 1 \text{ in } |z| < 1\},$$

where $\|M\|$ is the spectral norm of the matrix $M(z)$ over the open unit disk $|z| < 1$; namely, the largest eigenvalue of the constant matrix $M_s := \sup_{|z|\to 1}[M(z)M^*(z)]^{1/2}$, in which $A^{1/2}$ indicates the Hermitian square root of the matrix A. Note that it is possible to use a different norm of the matrix (defined on the open unit disk), or even simply replace $\|M\| \leqslant 1$ by $M(z)M^*(z) \leqslant I$ in the above, with a routine modification in the following discussions.

We first show that for a given $\Phi \in \mathscr{C}$, the matrix $I + \Phi(z)$ is nonsingular in $|z| < 1$. Indeed, if it is singular, then there is a constant vector $\mathbf{x} \neq 0$ such that $[I + \Phi]\mathbf{x} = 0$, which yields $\mathbf{x} = -\Phi\mathbf{x}$ and $\bar{\mathbf{x}}^{\mathsf{T}} = -\bar{\mathbf{x}}^{\mathsf{T}}\Phi^*$, so that

$$0 < 2\bar{\mathbf{x}}^{\mathsf{T}}\mathbf{x} = -\bar{\mathbf{x}}[\Phi + \Phi^*]\mathbf{x} \leqslant 0,$$

a contradiction. Let

$$\Psi(z) = [I - \Phi(z)][I + \Phi(z)]^{-1}, \qquad |z| < 1. \tag{1}$$

Then it follows that

$$I + \Psi = I + [I - \Phi][I + \Phi]^{-1} = 2[I + \Phi]^{-1},$$

which implies that the matrix $I + \Psi(z)$ is invertible in $|z| < 1$. Consequently, solving Equation (1) gives

$$\Phi(z) = [I - \Psi(z)][I + \Psi(z)]^{-1}, \qquad |z| < 1. \tag{2}$$

Observe that the two identities, (1) and (2), have the same pattern. More importantly, we notice that $\Psi \in \mathscr{S}$ if and only if $\Phi \in \mathscr{C}$, as pointed out in [8]. To verify this, we first note that

$$\begin{aligned} I - \Psi\Psi^* &= I - \left[2[I + \Phi]^{-1} - I\right]\left[2[I + \Phi^*]^{-1} - I\right] \\ &= 2[I + \Phi]^{-1}[\Phi + \Phi^*][I + \Phi^*]^{-1} \geqslant 0, \end{aligned}$$

which implies that $\Psi\Psi^* \leqslant I$ for $|z| < 1$, or $\|\Psi\| \leqslant 1$; namely, $\Psi \in \mathscr{S}$.

Conversely, given a $\Psi \in \mathscr{S}$, we have $\|\Psi\| \leqslant 1$, so that

$$\Phi + \Phi^* = [I - \Psi][I + \Psi]^{-1} + [I + \Psi^*]^{-1}[I - \Psi^*] \geqslant 0,$$

which means $\Phi \in \mathscr{C}$.

Now, we can reformulate the above problem as the following: Given n distinct points $z_1, z_2, \ldots, z_n$ in the unit circle $|z| < 1$ and n $m \times m$ complex matrices $W_1, W_2, \ldots, W_n$ in the Carathéodory class $\mathscr{C}$, find an $m \times m$ rational matrix Ψ in the Schur class $\mathscr{S}$ such that

$$\Psi(z_k) = [I - W_k][I + W_k]^{-1}, \qquad k = 1, 2, \ldots, n.$$

Here, we remark that since the matrix W_k is in the class $\mathscr{C}$, the matrix $I + W_k$ is invertible, $k = 1, 2, \ldots, n$, as argued above. Note also that this reformulated problem is solvable if and only if the new Pick matrix P_s defined by

$$P_s = \left[\frac{I - \Psi^*(z_l)\Psi(z_k)}{1 - \bar{z}_l z_k}\right]_{(mn)\times(mn)}$$

is nonnegative definite, which is the same as that in the scalar-valued setting [5, 18, 19]. To show this, it suffices to observe that

$$\begin{aligned} W_k + W_l^* &= [I - \Psi(z_k)][I + \Psi(z_k)]^{-1} \\ &\quad + \left[[I - \Psi(z_l)][I + \Psi(z_l)]^{-1}\right]^* \\ &= [I + \Psi(z_l)^*]^{-1}[[I + \Psi(z_l)^*][I - \Psi(z_k)] \\ &\qquad + [I - \Psi(z_l)^*][I + \Psi(z_k)]][I + \Psi(z_k)]^{-1} \\ &= 2[I + \Psi(z_l)^*]^{-1}[I - \Psi^*(z_l)\Psi(z_k)][I + \Psi(z_k)]^{-1}, \end{aligned}$$

which shows that P_c is nonnegative definite if and only if P_s is nonnegative definite.

Now, we return to the reformulation of the interpolation problem. Once a solution $\Psi(z)$ for the reformulated problem is found, a solution for the original problem is obtained via

$$\Phi(z) = [I - \Psi(z)][I + \Psi(z)]^{-1}.$$

Indeed, we have

$$\begin{aligned}
\Phi(z_k) &= [I - \Psi(z_k)][I + \Psi(z_k)]^{-1} \\
&= \left[I - [I - W_k][I + W_k]^{-1}\right]\left[I + [I - W_k][I + W_k]^{-1}\right]^{-1} \\
&= [[I + W_k] - [I - W_k]][I + W_k]^{-1} \\
&\quad \times \left[[[I + W_k] + [I - W_k]][I + W_k]^{-1}\right]^{-1} \\
&= 2W_k[I + W_k]^{-1}\left[2[I + W_k]^{-1}\right]^{-1} \\
&= W_k \qquad \text{for} \quad k = 1, 2, \ldots, n.
\end{aligned}$$

3. DERIVATION OF THE ALGORITHM

We now derive a simple, yet efficient, algorithm for the aforementioned matrix-valued Nevanlinna-Pick interpolation problem. The derivation that we present here preserves the basic idea of [9]. First, we point out that under the nonnegative definiteness condition of the Pick matrix P_c, we must have $\|W_k W_k^*\| < 1$; otherwise the interpolation problem is not solvable. A verification of this has been given in [19] and hence is omitted here. For each $k = 1, 2, \ldots, n$, let $\Psi_k(z)$ be an $m \times m$ matrix-valued rational function such that $\|\Psi_k\| \leqslant 1$, which will be determined later, and define

$$\begin{aligned}
M_k := \frac{z_k(1 - \bar{z}_k z)}{|z_k|(z_k - z)}&[I - W_k W_k^*]^{-1/2}[\Psi_k - W_k] \\
&\times [I - W_k^* \Psi_k]^{-1}[I - W_k^* W_k]^{1/2}, \qquad (3)
\end{aligned}$$

where, as before, $A^{1/2}$ denotes the Hermitian square root of A, and $A^{-1/2}$ is the inverse of $A^{1/2}$. Since both its scalar product (the coefficient) and its matrix product are Blaschke products, M_k is analytic in $|z| < 1$ with $\|M_k\| \leqslant 1$ for $|z| < 1$. It then follows form the Schwarz lemma that M_k belongs to $\mathscr{S}$. For n arbitrarily chosen matrices $K_1, K_2, \ldots, K_n$ in the Schur class $\mathscr{S}$, define

$$\Psi_{k+1} = [I - K_k K_k^*]^{-1/2}[M_k - K_k][I - K_k^* M_k]^{-1}[I - K_k^* K_k]^{1/2}. \quad (4)$$

From (4), we have

$$[I - K_k K_k^*]^{1/2}\Psi_{k+1}[I - K_k^* K_k]^{-1/2} = [M_k - K_k][I - K_k^* M_k]^{-1}. \quad (5)$$

Using the notation

$$U_k := [I - K_k K_k^*]^{1/2}\Psi_{k+1}[I - K_k^* K_k]^{-1/2}$$

and solving (5) for M_k, we obtain

$$M_k = [U_k + K_k][I + U_k K_k^*]^{-1}. \quad (6)$$

On the other hand, it follows from (3) that

$$\frac{|z_k|(z_k - z)}{z_k(1 - \bar{z}_k z)}[I - W_k W_k^*]^{1/2} M_k [I - W_k^* W_k]^{-1/2}$$

$$= [\Psi_k - W_k][I - W_k^* \Psi_k]^{-1}. \quad (7)$$

Then, using the notation

$$V_k := \frac{|z_k|(z_k - z)}{z_k(1 - \bar{z}_k z)}[I - W_k W_k^*]^{1/2} M_k [I - W_k^* W_k]^{-1/2} \quad (8)$$

and solving (7) for Ψ_k, we obtain

$$\Psi_k = [V_k + W_k][I + V_k W_k^*]^{-1}, \quad (9)$$

which is a Blaschke product and so satisfies $\|\Psi_k\| \leqslant 1$, as required.

Now, the computational algorithm works as follows: To start with, pick an arbitrary matrix Ψ_{n+1} such that $\|\Psi_{n+1}\| \leqslant 1$ and n arbitrarily chosen matrices $K_1, K_2, \ldots, K_n$ from the Schur class $\mathscr{S}$. For example, we may choose

$$\Psi_{n+1} = I_{m\times m} \quad \text{and} \quad K_1 = K_2 = \cdots = K_n = 0_{m\times m}$$

for simplicity of calculation. Then, we recursively compute $\Psi_i(z)$ for $i = n, n - 1, \ldots, 1$. The final result is $\Psi(z) = \Psi_1(z)$.

Given the complex numbers z_i and the $m \times m$ complex matrices W_i for $1 \leqslant i \leqslant n$, and a particular value of z, the matrix-valued Nevanlinna-Pick interpolation algorithm shown in Table 1 computes the complex matrix $\Psi(z)$. Since the intermediate matrices V_n are functions of the complex value z, the

TABLE 1
THE MATRIX-VALUED NEVANLINNA-PICK INTERPOLATION ALGORITHM

Input:	z and z_i, W_i for $1 \leqslant i \leqslant n$.
Output:	$\Psi(z)$.
Step 1.	Compute the $m \times m$ diagonal matrix V_n and the $m \times m$ matrix Ψ_n by $V_n = \dfrac{\lvert z_n \rvert (z_n - z)}{z_n(1 - \bar{z}_n z)} I_{m\times m}$, $\Psi_n = [V_n + W_n][I + V_n W_n^*]^{-1}$.
Step 2.	For $i = n-1, n-2, \ldots, 1$ compute the $m \times m$ matrices V_i and Ψ_i by $V_i = \dfrac{\lvert z_i \rvert (z_i - z)}{z_i(1 - \bar{z}_i z)} [I - W_i W_i^*]^{1/2} \Psi_{i+1} [I - W_i^* W_i]^{-1/2}$, $\Psi_i = [V_i + W_i][I + V_i W_i^*]^{-1}$.
Step 3.	The final matrix $\Psi_1(z)$ is the desired solution $\Psi(z)$.

algorithm needs to be repeated whenever $\Psi(z)$ is required to be evaluated at another value of z.

4. COMPLEXITY OF THE ALGORITHM

Generally speaking, when a new computational method is introduced, there are several fundamental issues which are worthy of being investigated:

(1) *Numerical analysis:* the numerical stability, accuracy, and the convergence rate. In this category, one can include issues related to approximation order and best (or exact) coefficients of bounds involved in the algorithm.

(2) *Parallelization:* parallel complexity of the algorithm on a generic model of parallel computer; issues related to implementation on specific parallel computers such as shared and distributed memory multiprocessors. Systolic array implementations are also in this category.

(3) *Computational complexity:* a detailed count of the number of arithmetic operations.

In this paper, we are mainly concerned with the computational complexity problem of the proposed algorithm, to make our earlier investigation [18, 19] complete. We will address numerical-analysis issues in the future. We are interested in a detailed arithmetic operation count, since this directly relates to the running time of the algorithm in terms of the input size, i.e., the order of the interpolation. In many applications such as robust control of multiinput multioutput biomedical and engineering systems (neural networks, computer networks, power networks, industrial automatic assembly-lines etc.), the interpolation can be of very high order.

The matrix-valued Nevanlinna-Pick interpolation algorithm that we proposed uses the following matrix operations: addition, subtraction, multiplication, inversion, square root, and inverse square root. Following the conventional method of analysis, we first count the total of arithmetic operations required by the algorithm. For iterative algorithms, the arithmetic operations per iteration step are counted. The number of iteration steps, on the other hand, is a function of the convergence properties of the algorithm as well as the input values and the size of the input matrix.

It can be easily verified that the matrix addition (or subtraction) algorithm requires a total of m^2 arithmetic operations for two $m \times m$ matrices. The operation count for the matrix multiplication algorithm is not that simple. The standard matrix multiplication algorithm requires $m^2(2m - 1) = O(m^3)$ arithmetic operations. However, there exist faster algorithms. For example, the Strassen algorithm [25] uses only $O(m^{\log_2 7}) = O(m^{2.807})$ arithmetic operations by reducing an $m \times m$ matrix product problem to seven $(m/2) \times (m/2)$ matrix product problems plus several matrix additions and subtractions. More advanced algorithms use generalizations of Strassen's technique as well as entirely new techniques to further reduce the total number of arithmetic operations [17, 7]. The best algorithm to date, proposed by Coppersmith and Winograd, requires $O(m^{2.376})$ arithmetic operations [6]. It is now customary to say that the multiplication of two $m \times m$ matrices requires $O(m^\alpha)$ arithmetic operations, where $2 < \alpha \leqslant 3$, leaving the optimal value of α to future development. However, we must remark that there are instability problems associated with fast matrix multiplication algorithms, and that these algorithms perform faster only if m is large, usually when $m > 250$.

The computation of $\Psi_i = (V_i + W_i)(I + V_i W_i^*)^{-1}$ requires the use of matrix addition, multiplication, and inversion subprograms. This computation is to be performed for all i from n down to 1. In order to invert an $m \times m$ matrix, we can use the Gaussian elimination algorithm, which requires $\frac{8}{3}m^3 + O(m^2)$ arithmetic operations.

The computation of V_n is simple and easily accomplished by using only scalar operations. However, after the first step, we need to compute V_i by computing the square root and the inverse square root of the matrices $1 + W_i W_i^*$ and $1 + W_i^* W_i$, respectively. The square root of a complex matrix can be computed by using Newton's method or any of its variants [15, 16]. In order to compute the inverse square root, we have essentially two approaches:

1. First, compute the square root of the matrix using Newton's method, then invert the resulting matrix.
2. Combine the square-root and inversion steps to obtain an iterative algorithm for the inverse square root. One approach was described in [24].

TABLE 2
ARITHMETIC OPERATIONS REQUIRED BY THE MATRIX SUBPROGRAMS

Subprograms	Arithmetic operations
Addition (subtraction)	m^2
Multiplication	$2m^3 - m^2$
Inversion	$\frac{8}{3}m^3 + O(m^2)$
Square root	$\frac{14}{3}m^3 + O(m^2)$ per step

The square-root and inverse square-root algorithms are iterative techniques. In the case of Newton's method for computing the square root, the convergence rate is quadratic and the algorithm uses one matrix inversion and one matrix multiplication per step. The total number of arithmetic operations per step is thus equal to $\frac{14}{3}m^3 + O(m^2)$. The number of iteration steps for Newton's algorithm is a function of the size and the spectral distribution of the matrix, and is usually in the range $6 \leqslant k \leqslant 10$ for matrices of reasonable size and stiffness. The inverse square-root algorithm (scheme II) described in [24] requires three matrix multiplications and a single matrix inversion, thus a total of $\frac{26}{3}m^3 + O(m^2)$ arithmetic operations per step. However, we will assume that the inverse square root is computed via the square root and that Newton's method requires k iteration steps to compute the square root of a given $m \times m$ matrix.

Using the operation count summarized in Table 2, we calculate the higher-order terms (m^3 terms) in the number of arithmetic operations required by the matrix-valued Nevanlinna-Pick interpolation as follows:

Step 1 requires 2 multiplications and 1 inversion: $\frac{20}{3}m^3$ arithmetic operations.
Step 2 requires 6 multiplications, 2 inversions, and 2 square-root operations: $12m^3 + \frac{16}{3}m^3 + \frac{28}{3}km^3$ arithmetic operations for each $i = n - 1, n - 2, \ldots, 1$.

Thus, we obtain a total of $\frac{1}{3}[20 + (28k + 52)(n - 1)]m^3$ arithmetic operations. Taking an approximate value for $k = 8$, we find that the matrix-valued Nevanlinna-Pick interpolation algorithm requires approximately $95nm^3$ arithmetic operations.

5. CONCLUSION

In this paper, we have described the Nevanlinna algorithm for the matrix-valued Nevanlinna-Pick interpolation by reformulating the original problem to an equivalent problem which is consistent with the scalar-valued

interpolation. We have given a brief description of the derivation and a detailed arithmetic complexity analysis of the algorithm. We will investigate other important issues of the matrix-valued Nevanlinna-Pick interpolation, such as its numerical stability and parallel computation, elsewhere in the future.

REFERENCES

1 A. C. Allison and N. J. Young, Numerical algorithms for the Nevanlinna-Pick problem, *Numer. Math.* 42(2):125–145 (1983).
2 A. C. Antoulas and B. D. O. Anderson, On the problem of stable rational interpolation, *Linear Algebra Appl.* 122/124:301–329 (1989).
3 A. C. Antoulas, J. A. Ball, J. Kang, and J. C. Willems, On the solution of the minimal rational interpolation problem, *Linear Algebra Appl.* 137/138:511–573 (1990).
4 J. A. Ball, Nevanlinna-Pick interpolation: Generalizations and applications, in *Survey of Some Recent Results in Operator Theory I* (J. B. Conway and B. B. Morrel, Eds.), Pitman, Boston, 1988, pp. 51–94.
5 C. K. Chui and G. Chen, *Signal Processing and Systems Theory: Selected Topics*, Springer-Verlag, New York, 1992.
6 D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progressions, *J. Symbolic Comput.* 9(3):23–52 (1990).
7 T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, Mass., 1990.
8 Ph. Delsarte, Y. Genin, and Y. Kamp, Schur parametrization of positive definite block-Toeplitz systems, *SIAM J. Appl. Math.* 36(1):34–46 (1979).
9 Ph. Delsarte, Y. Genin, and Y. Kamp, The Nevanlinna-Pick problem for matrix-valued functions, *SIAM J. Appl. Math.* 36(1):47–61 (1979).
10 Ph. Delsarte, Y. Genin, and Y. Kamp, On the role of the Nevanlinna-Pick problem in circuit and system theory, *Circuit Theory Appl.* 9:177–187 (1981).
11 H. Dym, *J Contractive Matrix Functions, Reproducing Kernel Hilbert Spaces and Interpolations*, Vol. 71, Amer. Math. Soc., 1989.
12 I. P. Fedčina, A criterion for the solvability of the Nevanlinna-Pick tangent problem (in Russian), *Mat. Issled.* 7:223–227 (1972).
13 F. Fenyves, *Beitrag zur Realisierung von Zweipolen mit Vorgegebener Charakteristik*, Atheneum, Budapest, 1938.
14 J. B. Garnett, *Bounded Analytic Functions*, Academic, 1981.
15 G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins U.P., Baltimore, 1983.
16 N. J. Higham, Newton's method for the matrix square root, *Math. Comp.* 46(174):537–549 (1986).
17 D. E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 2nd ed., Addison-Wesley, Reading, Mass., 1981.
18 Ç. K. Koç and G. Chen, Parallel algorithms for Nevanlinna-Pick interpolation: The scalar case, *Internat. J. Comput. Math.* 40(1 + 2):99–115 (1991).

19 Ç. K. Koç and G. Chen, A fast algorithm for scalar Nevanlinna-Pick interpolation, *Numer. Math.* 64(1):115–126 (1993).
20 B. Sz.-Nagy and A. Korányi, Relations d'un problème de Nevanlinna et Pick la théorie des opérateurs de l'espace Hilbertien, *Acta Math. Acad. Sci. Hungar.* 7:295–302 (1956).
21 R. Nevanlinna, Über beschränkte analytische Funkionen, *Ann. Acad. Sci. Fenn.* A32:1–75 (1929).
22 G. Pick, Über die beschränkungen analytischer Funktionen welche durch vorgegebene Funktionswerte bewirkt werden, *Math. Ann.*, 77:7–23 (1916).
23 M. Rosenblum and J. Rovnyak, *Hardy Classes and Operator Theory*, Oxford U.P., 1985.
24 N. Sherif, On the computation of a matrix inverse square root, *Computing* 46(4):295–305 (1991).
25 V. Strassen, Gaussian elimination is not optimal, *Numer. Math.* 13(4):354–356 (1969).
26 M. Vidyasagar, *Control System Synthesis: A Factorization Approach*, MIT Press, 1985.

Received 24 November 1992; final manuscript accepted 11 September 1993