A collection of examples where Neville elimination outperforms Gaussian elimination

Pedro Alonso\textsuperscript{a}, Jorge Delgado\textsuperscript{b,*, c}, Rafael Gallego\textsuperscript{a}, Juan Manuel Peña\textsuperscript{c}

\textsuperscript{a}Departamento de Matemáticas, Universidad de Oviedo, 33203 Gijon, Spain
\textsuperscript{b}Departamento de Matemática Aplicada, Universidad de Zaragoza, 44003 Teruel, Spain
\textsuperscript{c}Departamento de Matemática Aplicada, Universidad de Zaragoza, 50009 Zaragoza, Spain

\textbf{ARTICLE INFO}

\textbf{Keywords:}
Iterative refinement
Neville elimination
Gaussian elimination

\textbf{ABSTRACT}

Neville elimination is an elimination procedure alternative to Gaussian elimination. It is very useful when dealing with totally positive matrices, for which nice stability results are known. Here we include examples, most of them test matrices used in MATLAB which are not totally positive matrices, where Neville elimination outperforms Gaussian elimination.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

Examples with an unusual behaviour have stimulated the theoretical research related to Gaussian elimination (GE) (see [12,30]). GE, with or without a pivoting strategy, is the most usual elimination procedure to transform a nonsingular matrix into an upper triangular matrix. But there are other alternative elimination procedures, such as Neville elimination (NE). NE has received a great attention in the last decades due to its nice properties when applied to a class of structured matrices called totally positive matrices and described below. This paper presents families of matrices whose structure is very far from that of totally positive matrices, but for which NE outperforms GE. Of course, Gaussian elimination with Partial Pivoting (GEPP) presents in general advantages over GE and NE without pivoting strategy. For this reason we do not compare our results with GEPP except for a particular family of matrices (denoted by gpp matrices). The pathological behaviour of GEPP with these matrices is well known (see [22]) and we also use in this case Neville elimination with Partial Pivoting (NEPP).

Roughly speaking, NE is an elimination procedure which produces zeros in each column by subtracting to each row an adequate multiple of the previous one, whereas GE produces zeros in each column by subtracting to each row an appropriated multiple of a fixed row, called pivot row. As recalled in Section 4, NE is very useful when dealing with Totally Positive (TP) matrices (matrices such that all their minors are nonnegative). These matrices are also called in the literature as totally nonnegative matrices. TP matrices, and their important subclass of Almost Strictly Totally Positive (ASTP) matrices, play an important role in many applications of Approximation Theory, Statistics or Computer Aided Geometric Design (CAGD): see [5], [13] or [24]. For these matrices and other related classes of matrices, NE has great advantages over GE in many important tasks, such as solving linear systems, calculating singular values or eigenvalues, or deriving tests to recognize total positivity (see [10,15]), a crucial task to recognize shape preserving representations in CAGD (see [24]). Besides, NE has nice stability properties when applied to ASTP matrices, similar to those known when using GE (see [3,6]). Convergence results for Iterative Refinement (IR) when using NE have been obtained in [2].

Most of the examples presented in this paper are provided by families of test matrices used in MATLAB for which one has no a priori reasons to expect with NE a better behaviour than with GE, in contrast to the mentioned case of ASTP matrices. In our examples the advantages of NE over GE have different nature: they can affect to the computational complexity, the
accuracy of the solution or to iterative refinement, depending on the family of examples. In the last two cases we often observe
that a better behaviour of NE is associated to a smaller growth factor than that of GE.

Section 2 presents basic notations on NE and IR. Section 3 includes the numerical examples with orthogonal matrices and
with matrices known as gpff (see [19,20]). Section 4 is devoted to TP matrices, and we show the low computational
complexity of applying NE to Lehmer matrices, which are not ASTP matrices but form a subclass of TP matrices. Finally, Section 5
discussed the theoretical explanations of the numerical experiments and collects the conclusions of the paper.

2. Neville elimination and iterative refinement

In this section we will describe basic notations related to NE and IR.

2.1. Neville elimination

For a nonsingular matrix A of order n the NE procedure consists of n − 1 successive steps, resulting in a sequence of matrices
as follows:

\[ A = A^{(1)} \rightarrow \tilde{A}^{(1)} \rightarrow A^{(2)} \rightarrow \tilde{A}^{(2)} \rightarrow \ldots \rightarrow A^{(n)} = \tilde{A}^{(n)} = U, \]
where U is an upper triangular matrix. For each \( t \), \( 1 < t \leq n \), the matrices \( A^{(t)} = (a_{ij}^{(t)})_{1 \leq i,j \leq n} \) and \( \tilde{A}^{(t)} = (\tilde{a}_{ij}^{(t)})_{1 \leq i,j \leq n} \) have zeros
below their main diagonal in the first \( t − 1 \) columns and also one has

\[ \tilde{a}_{tt}^{(t)} = 0, \quad i > t \Rightarrow \tilde{a}_{it}^{(t)} = 0 \quad \forall h \geq i. \]  

(1)

\( \tilde{A}^{(t)} \) is obtained from the matrix \( A^{(t)} \) by reordering the rows \( t \), \ldots , \( n \) according to a row pivoting strategy satisfying (1). To get
\( A^{(t+1)} \) from \( \tilde{A}^{(t)} \) we produce zeros in the column \( t \) below the main diagonal by subtracting a multiple of the \( t \)th row to the
\( (i+1) \)th for \( i = n−1, n−2, \ldots , t \), according to the following formula: for any column \( j \)

\[ a_{ij}^{(t+1)} = \begin{cases} \tilde{a}_{ij}^{(t)}, & \text{if } 1 \leq i \leq t, \\ \tilde{a}_{ij}^{(t)} - \frac{a_{ij}^{(t)}}{a_{ii}^{(t)}} \tilde{a}_{i+1,t}^{(t)}, & \text{if } t + 1 \leq i \leq n \text{ and } \tilde{a}_{i+1,t}^{(t)} \neq 0, \\ \tilde{a}_{ij}^{(t)}, & \text{if } t + 1 \leq i \leq n \text{ and } \tilde{a}_{i+1,t}^{(t)} = 0. \end{cases} \]  

(2)

Observe that in the third case \( \tilde{a}_{i+1,t}^{(t)} = 0 \) implies \( \tilde{a}_{i+1}^{(0)} = 0 \) by (1). In this process one has \( A^{(n)} = \tilde{A}^{(n)} = U \), and when no row exchanges are needed, then \( A^{(t)} = \tilde{A}^{(t)} \) for all \( t \).

Analogously to the case of GE, in order to keep the absolute values of the multipliers less than or equal to 1, we perform
NEPP when we reorder the rows of the matrix \( A^{(t)} \) to get a matrix \( A^{(t)} \) with

\[ |\tilde{a}_{tt}^{(t)}| \geq |\tilde{a}_{t+1,t}^{(t)}| \geq \ldots \geq |\tilde{a}_{nn}^{(t)}|, \]

(3)

(see [17] for more details).

2.2. Iterative refinement

In this subsection we recall the IR technique and apply it to NE. Let us assume that the system \( Ax = b \) has been solved
using NE with floating point arithmetic, obtaining an approximate solution \( \tilde{x} = x^{(1)} \). If the matrix \( A \) is not well conditioned,
the computed solution \( x^{(1)} \) might not be accurate enough. When \( x^{(1)} \) has been computed, we will obtain the corresponding residual, which is defined as

\[ r^{(1)} = A(\tilde{x} - x^{(1)}) = Ax - Ax^{(1)} = b - Ax^{(1)}. \]

It is convenient to compute this vector in extra precision in order to improve the accuracy of \( x^{(1)} \). Observe that Skeel proves in
[25,26] that in the Gaussian case it is not always necessary to compute the residual in extra precision to obtain a stable
procedure, because stability might be achieved in working precision.

Next we will solve the linear system \( A \tilde{e} = r^{(1)} \) and for this purpose we will use the matrices \( \tilde{L} \) and \( \tilde{U} \) obtained in the NE
process. Then the system to be solved will be \( \tilde{L} \tilde{U} \tilde{e} = r^{(1)} \) that is equivalent to \( (A + E) \tilde{e} = r^{(1)} \) (see [3]). The solution obtained
will not be exactly \( e \), but an approximation, which will be denoted by \( e^{(1)} \), and then \( x^{(2)} = x^{(1)} + e^{(1)} \) will be a new approximation
to the exact solution.

This procedure is commonly known as IR and, as we have pointed out in the introduction, it was first presented by Wilkinson in
[27,28]. In other works, as for example, [11,21,23,26], several features of IR have been studied.

In a general way, we will define the vectors \( r^{(k)} \) and \( x^{(k+1)} \) through the following equations

\[ r^{(k)} = b - Ax^{(k)}, \quad x^{(k+1)} = x^{(k)} + e^{(k)}, \]

(4)

where \( e^{(k)} \) is the approximate solution obtained when solving the system \( (\tilde{L} \tilde{U}) \tilde{e} = r^{(k)} \), namely \( e^{(k)} = (\tilde{L} \tilde{U})^{-1} r^{(k)} \).
Bringing the expression of \( a^{(k)} \) to (4) we have that
\[
{x}^{(k+1)} = {x}^{(k)} + (\hat{L}\hat{U})^{-1}{r}^{(k)}.
\]
If the vector sequence \( \{x^{(k)}\} \) can be formed without additional roundoff errors, we deduce that
\[
x^{(k+1)} = x^{(k)} + (\hat{L}\hat{U})^{-1}(b - Ax^{(k)}) = x^{(k)} + (\hat{L}\hat{U})^{-1}A(x - x^{(k)}),
\]
and so
\[
x^{(k+1)} - x = (I - (\hat{L}\hat{U})^{-1}A)(x^{(k)} - x),
\]
and repeating the process
\[
x^{(k+1)} - x = (I - (\hat{L}\hat{U})^{-1}A)^{k}(x^{(1)} - x).
\]
Considering (5) we can state that
\[
\|x^{(k+1)} - x\|_{\infty} \leq \|I - (\hat{L}\hat{U})^{-1}A\|_{\infty}^{k}\|x^{(1)} - x\|_{\infty},
\]
so for \( k + 1 \) to converge towards \( x \), or in other words, for
\[
\lim_{k \to \infty} \|x^{(k+1)} - x\|_{\infty} = 0,
\]
it is enough that the following inequality holds:
\[
\|I - (\hat{L}\hat{U})^{-1}A\|_{\infty} < 1.
\]

3. Numerical experiments

In this section we carry out numerical tests for solving two kinds of linear systems of equations through the IR technique by NE and GE.

On one hand, the first type of systems has as coefficient matrix three types of orthogonal matrices. These matrices were used by Higham in [19,20], which describe the version 3.0 of the Test Matrix Toolbox for MATLAB 4.2. The toolbox contains, among other things, a collection of test matrices including such orthogonal matrices. The current versions of MATLAB already have built-in support for these orthogonal matrices without the necessity of installing the toolbox. On the other hand, the second kind of systems has as coefficient matrices those introduced in Example 45, p. 212 of [29], and they are called gfpp matrices. The gfpp matrices have the maximum growth factor \( 2^{n^{-1}} \) possible for GEPP (see [22]). In addition, the gfpp matrices belong to a more general class of matrices introduced in [22]. Nevertheless, these matrices are not included in the basic installation of the current versions of MATLAB, but they are available through the Test Matrix Toolbox for MATLAB (see Section 2 of [20] for a list of web addresses where the toolbox is available).

We will show that for the mentioned orthogonal matrices NE outperforms GE getting a more accurate solution and presenting usually a smaller growth factor. For gfpp matrices NEPP outperforms GEPP from three points of view: getting a more accurate solution, presenting a lower growth factor and having a lower computational cost.

In our experiments, we consider linear systems of equations of the form \( Ax = b \) for the two different kinds of matrices mentioned above. The numerical tests are performed on a PC Core 2 Duo E6750 with the Kernel Linux 2.6.22-14-generic in C programming language with the compiler gcc in double precision. The Intel Core 2 Duo processor we use implements IEEE 754 standard for binary floating point arithmetic. Hence the difference between 1.0 and the smaller number greater than 1.0 representable by the computer is \( 1.19209290 \times 10^{-7} \) in simple precision and \( 1.110223046251568 \times 10^{-16} \) in double precision. In addition, gcc uses as a rounding strategy choosing the closest representable number. For more details on the compiler, see http://gcc.gnu.org.

We compute the Wilkinson growth factor
\[
\rho_n^{W}(A) := \max_{i,j} \frac{|a^{(k)}_{ij}|}{\max_{ij} |a_{ij}|},
\]
the relative errors and the componentwise backward error of the solutions obtained. In the tables containing these data for the different linear systems of equations considered we present the number of iterations of IR necessary to observe the general behaviour. Let us recall that the componentwise backward error, \( \omega_{\|A\|,\|b\|}(y) \), of the solutions \( y \) obtained, is defined by
\[
\omega_{\|A\|,\|b\|}(y) := \max_{i} \frac{|r_i|}{\|A\|\|y\| + |b|},
\]
where \( Ax = b \) is the system and \( r = b - Ay \) is the residual (see [21, p. 122] for more details). On the other hand, in the figures with the numerical results we only show the relative errors. Finally, we want to remark that all the results presented both in figures and tables have been performed in double precision, including the computation of the residual. Nevertheless, we have checked that the results do not change significantly by computing the residual with extra precision.

3.1. Orthogonal matrices

Here we have worked with systems \( Ax = b \) where \( A = (a_{ij})_{1 \leq i,j \leq n} \) are the orthogonal matrices given by

\[
a_{ij} = \sqrt{\frac{2}{n+1}} \sin \left( \frac{ij\pi}{n+1} \right), \quad 1 \leq i, j \leq n.
\]

In the current versions of MATLAB "gallery (‘orthog’, n, 1)" returns the orthogonal matrix \( A \) of order \( n \). Then we have taken the solutions \( x \) as random vectors from the uniform distribution on the interval \([-100, 100]\) and we have computed in a straightforward way the independent vectors \( b = Ax \). Finally we have solved the corresponding systems with IR by both NE and GE.

In Fig. 1 we can see the general behaviour of IR by GE and NE with these systems. In this figure we represent the logarithm of the relative errors of the solutions versus the dimension of the coefficient matrix for six consecutive iterations. With iteration 0 \((i = 0)\) we mean that the solution is obtained by GE and NE without applying IR, whereas strictly positive \( i \)'s show the number of iterations of IR applied. In Fig. 1 \( i = 0 \) we see that both NE and GE provide a very similar accurate solution. But in the case of NE the solution gets more accurate as the number of iterations of IR increases in contrast to GE. For example, with five iterations of IR, NE provides an accurate enough solution for all the systems, while for GE this only can be assured for the systems with dimension at most 37 or 38.

As representation of the behaviour above commented we have chosen the orthogonal matrices \( A \) of order 25, 35 and 50, presenting the numerical results in Tables 1–3, respectively.

In Table 1 iteration 0 (that is, without IR) we can see that, although the growth factor of GE is greater than that of NE, both GE and NE provide a solution with very similar accuracy. However, with one iteration of IR by NE we obtain a solution with a
previous examples. These orthogonal matrices NE outperforms IR by GE with respect to the goal of obtaining an accurate solution. We have also performed other numerical tests with the previous orthogonal matrices. In one case we have obtained an approximation to the solution vector as integer random vectors from the uniform distribution on the intervals \([0,1]\) and \([0,100]\) and then we have computed in a straightforward way more than with NE. Observe in Table 1 that in the resolution of the system of order 25 through IR by GE we have obtained almost the same results for the componentwise backward error of the solutions than the ones obtained by Higham for the same kind of systems in Table 11.1, p. 239, of [21], although Higham chooses \(b\) as a random vector from the uniform distribution on the interval \([0,1]\).

Table 2
Orthogonal matrix of order 35.

| i | Alg. | \(\rho^n(A)\) | \(|x^{(i)} - x^{(i)}|/|x^{(i)}|\) | \(\omega_{A|x|^2}(x^{(i)})\) |
|---|------|----------------|---------------------------------|----------------------------|
| 0 | GE   | 8.243766e+14   | 1.041200e-01                    | 3.947879e-02               |
|   | NE   | 8.485724e+08   | 5.831958e-02                    | 2.121932e-02               |
| 1 | GE   | 1.635846e-03   | 7.873518e-04                    |                            |
|   | NE   | 2.868878e-12   | 7.841756e-13                    |                            |
| 2 | GE   | 3.564976e-06   | 2.021920e-06                    |                            |
|   | NE   | 2.842171e-16   | 1.130131e-16                    |                            |
| 3 | GE   | 8.292431e-08   | 3.179888e-08                    |                            |
|   | NE   | 3.552714e-16   | 1.050056e-16                    |                            |

Table 3
Orthogonal matrix of order 50.

| i | Alg. | \(\rho^n(A)\) | \(|x^{(i)} - x^{(i)}|/|x^{(i)}|\) | \(\omega_{A|x|^2}(x^{(i)})\) |
|---|------|----------------|---------------------------------|----------------------------|
| 0 | GE   | 2.175242e+18   | 3.820225e+02                    | 9.171884e-01               |
|   | NE   | 6.973971e+11   | 4.592481e+01                    | 9.603081e-01               |
| 1 | GE   | 2.196718e+02   | 8.297208e-01                    |                            |
|   | NE   | 5.063479e-04   | 1.447356e-04                    |                            |
| 2 | GE   | 8.782735e+02   | 8.694895e-01                    |                            |
|   | NE   | 5.870045e-13   | 2.410904e-13                    |                            |
| 3 | GE   | 2.626648e+03   | 9.716075e-01                    |                            |
|   | NE   | 5.869083e-14   | 1.978457e-14                    |                            |

relative error of the order of the unit roundoff, in contrast to the relative error of order \(10^{-13}\) of GE. The table, in addition, shows that for obtaining a solution with a precision of the order of the unit roundoff with IR by GE it is necessary one iteration more than with NE. Observe in Table 1 that in the resolution of the system of order 25 through IR by GE we have obtained almost the same results for the componentwise backward error of the solutions than the ones obtained by Higham for the same kind of systems in Table 11.1, p. 239, of [21], although Higham chooses \(b\) as a random vector from the uniform distribution on the interval \([0,1]\).

Table 2 shows that, although GE and NE without IR behave in a similar way, in the case of IR by NE two iterations are sufficient for obtaining a solution with an accuracy of the order of the unit roundoff, whereas with two iterations of IR by GE we get a solution with a relative error of order \(10^{-6}\). Therefore we can conclude that IR by NE converges faster to the solution than IR by GE.

In Table 3 iteration 0 we can see that both GE and NE provide a solution with a very similar and poor accuracy. However, we can see that with IR by GE the relative error of the solution even increases with the number of iterations. Therefore, for this system IR by GE diverges. On the other hand, the same table shows that IR by NE leads, with only three iterations, to a solution with a relative error of order \(10^{-14}\), which is good enough.

Remark 1. We have also performed other numerical tests with the previous orthogonal matrices. In one case we have chosen \(b\) as integer random vectors from the uniform distribution on the interval \([-100,100]\) and with Mathematica we have obtained an approximation to the solution vector \(x\). In the remaining cases we have chosen the solution \(x\) as real random vectors from the uniform distribution on the intervals \([0,1]\) and \([0,100]\) and then we have computed in a straightforward way \(b\) as \(Ax\). In the three cases we got results very similar to those presented in Fig. 1 and in the previous tables, that is, IR by NE outperforms IR by GE with respect to the goal of obtaining an accurate solution.

We have found other two kind of orthogonal matrices where IR by NE also outperforms IR by GE, analogously to the previous examples. These orthogonal matrices \(B = (b_{ij})_{1 \leq i, j \leq n}\) and \(C = (c_{ij})_{1 \leq i, j \leq n}\) are defined by

\[
b_{ij} = \frac{2}{\sqrt{2n+1}} \sin \left( \frac{2ij \pi}{2n+1} \right), \text{ for all } i,j \in \{1, \ldots, n\},
\]

and

\[
c_{ij} = \frac{\sqrt{2}}{n} \cos \left( \frac{(i-1/2)(j-1/2)\pi}{n} \right), \text{ for all } i,j \in \{1, \ldots, n\}.
\]

In the current versions of MATLAB "gallery (‘orthog’,n,a)" return the orthogonal matrices \(B (a = 2)\) and \(C (a = 6)\) of order \(n\), respectively. The results we have obtained with the matrices \(B\) and \(C\) are very similar to those presented in Fig. 1 and Tables 1–3.
3.2. Gfpp matrices

Now let us consider gfpp matrices \( A = (a_{ij})_{1 \leq i,j \leq n} \) defined by

\[
a_{ij} = \begin{cases} 
  -1, & \text{if } j < i, \\
  1, & \text{if } i = j \text{ or } j = n, \\
  0, & \text{otherwise.}
\end{cases}
\]

First we shall analyze the number of elementary operations performed when applying NE and GE to linear systems of equations where the coefficients matrices are gfpp matrices, comparing the computational cost of both elimination procedures. We will restrict ourselves to compute the number of elementary operations for transforming the gfpp matrix into the corresponding upper triangular matrix and the number of elementary operations performed on the right hand side vector in order to obtain the upper triangular matrix of NE the entries of \( A \) in positions \( (n,1), (n-1,1), \ldots, (2,1) \) are made zeros according to the procedure introduced in Section 2.1 obtaining the following matrix

\[
A^{(2)} = \begin{pmatrix}
1 & 0 & \cdots & \cdots & \cdots & 0 & 1 \\
0 & 1 & 0 & \cdots & \cdots & 0 & 2 \\
0 & -2 & 1 & \ddots & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & 0 & -2 & 1 & 0 \\
0 & \cdots & \cdots & 0 & \cdots & \cdots & 0 & -2 & 0
\end{pmatrix}
\]

Hence, obtaining \( A^{(2)} \) from \( A^{(1)} \) requires \( n - 1 \) quotients (for the computation of the corresponding \( n - 1 \) multipliers), and \( (n - 1)(n - 1) \) sums/subtractions and \( (n - 1)(n - 1) \) products (for modifying the rows \( n, n - 1, \ldots, 2 \)). Finally, at this step \( n - 1 \) sums/subtractions and \( n - 1 \) products are performed on the vector \( b \).

By the structure of the matrix \( A^{(2)} \) we deduce that at each of the remaining steps \( k = 2, \ldots, n - 1 \) it is only necessary to make zero the element at the position \( (k + 1, k) \). Therefore, we only have to compute a multiplier, and to modify the elements of the corresponding row and the corresponding element of the right hand side vector in order to obtain the upper triangular matrix

\[
U = A^{(n)} = \begin{pmatrix}
1 & 0 & \cdots & \cdots & \cdots & 0 & 1 \\
0 & 1 & 0 & \cdots & \cdots & 0 & 2^1 \\
0 & \cdots & \cdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & 0 & 1 & 0 & 2^{n-3} \\
0 & \cdots & \cdots & 0 & 1 & 0 & 2^{n-2} \\
0 & \cdots & \cdots & 0 & \cdots & \cdots & 0 & 2^{n-1}
\end{pmatrix}
\]

(8)

On the other hand, if we apply GE to gfpp matrices, then we cannot save any operation.

In Table 4 we can see a comparison of the number of elementary operations performed by GE and NE for these systems. We have seen that in this case NE has a computational cost of \( O(n^2) \) operations in contrast to the computational cost of \( O(n^3) \) operations of GE.

**Remark 2.** Let us observe that in the case of NE we can even save more elementary operations taking into account that, at steps \( k = 2, \ldots, n - 1 \), with a careful programming it is only necessary to update the elements \( (k + 1, k) \) and \( (k + 1, n) \) in the matrix \( A^{(k)} \). Let us also observe that by the structure of gfpp matrices NE and NEPP procedures are equivalent in

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Number of operations when applying NE and GE on gfpp systems.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NE</strong></td>
<td><strong>GE</strong></td>
</tr>
<tr>
<td>Quotients</td>
<td>( 2n - 3 )</td>
</tr>
<tr>
<td>Sums/Subtractions</td>
<td>( 3n^2 - 2n - 2 )</td>
</tr>
<tr>
<td>Products</td>
<td>( 3n^2 - 2n - 2 )</td>
</tr>
<tr>
<td>Total</td>
<td>( 3n^2 - n - 5 )</td>
</tr>
</tbody>
</table>
computational cost. Nevertheless, although GE coincides with GEPP for this kind of matrices, in the case of NEPP, at steps $k = 2, \ldots, n - 1$, the exchange of the rows $k$ and $k + 1$ is necessary and hence the matrix $U = A^{(n)}$ obtained by NEPP is not the matrix given by (8). In addition, we can check that the entries of the matrices $A^{(2)}, \ldots, A^{(n)}$ of NEPP with the highest absolute value are 2’s and $-2$’s. Therefore, the growth factor of NEPP for gfpp matrices is 2 for any order $n$ in contrast to the growth factor $2^{n-1}$ of GEPP.

In addition, in this case we have tested IR by NEPP and GEPP with systems of the form $Ax = b$, where the coefficients matrix $A$ is a gfpp matrix, $b$ is chosen as an integer random vector from the uniform distribution on the interval $[-100, 100]$ and the vector solution $x$ is computed exactly with Mathematica by the nature of the gfpp matrices. In our numerical tests we have seen that for the systems of order at most 50 the solutions obtained by both elimination procedures are good enough without applying IR. For the systems of order between 50 and 100 NEPP leads to a good solution without IR whereas GEPP without IR does not provide an accurate solution and it needs one iteration of IR in order to get a good enough solution. For the systems with order between 100 and 170 with NEPP without IR we obtain an accurate solution in contrast to the inaccurate solution provided by GEPP. In addition, in this case the IR does not help to GEPP to improve the solution. These tendencies can be checked in Fig. 2.

In order to illustrate the best behaviour of NEPP versus GEPP above commented, we show in Tables 5 and 6 the results obtained for the particular systems of order 110 and 170, respectively.

**Remark 3.** We have also performed other numerical tests with the gfpp matrices. In these cases we have chosen the solutions $x$ as real random vectors from the uniform distribution on the intervals $[0, 1], [0, 100]$ and $[-100, 100]$ and then we have computed in a straightforward way $b$ as $Ax$. In the three cases we have obtained results very similar to those showed in Fig. 2 and Tables 5 and 6.

![Fig. 2. Solving Ax = b with IR by GEPP and NEPP.](image)

<table>
<thead>
<tr>
<th>i</th>
<th>Alg.</th>
<th>$\rho_A(W)$</th>
<th>$|x^{(0)} - x|<em>\infty / |x|</em>\infty$</th>
<th>$\omega_{A,b}(x^{(0)})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GEPP</td>
<td>6.490371e+32</td>
<td>9.916748e−01</td>
<td>1.145957e−01</td>
</tr>
<tr>
<td></td>
<td>NEPP</td>
<td>2.000000e+00</td>
<td>9.928460e−17</td>
<td>1.164824e−16</td>
</tr>
<tr>
<td>1</td>
<td>GEPP</td>
<td>6.602273e−01</td>
<td>4.753710e−02</td>
<td>5.824121e−17</td>
</tr>
<tr>
<td></td>
<td>NEPP</td>
<td>4.964230e−16</td>
<td>6.178011e−17</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>GEPP</td>
<td>3.702862e−01</td>
<td>1.660087e−02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NEPP</td>
<td>6.949922e−16</td>
<td>6.178011e−17</td>
<td></td>
</tr>
</tbody>
</table>
4. Totally positive matrices

It has been shown that NE process is very useful with totally positive matrices, sign-regular matrices and other related types of matrices (see [13,24]). A real matrix is called Totally Positive (TP) if all its minors are nonnegative. TP matrices arise in a natural way in many areas of Mathematics, Statistics, Economics, etc. In particular, it is very interesting their application to Approximation Theory and Computer Aided Geometric Design (CAGD). For example, coefficient matrices of interpolation or least square problems with a lot of representations in CAGD (the Bernstein basis, the B-spline basis, etc.) are TP (see [6,24]). Some recent applications of such kind of matrices to CAGD can be found in [13,24,9]. For applications of TP matrices to other fields, see [5,13]. In [10,18,15,16] it has been proved that NE is a very useful alternative to GE when working with TP matrices. In addition, studies which prove the high performance computing of NE have been performed (see [1]).

If a matrix $A$ is nonsingular and totally positive, then $A = LU$ with nonnegative triangular factors $L$ and $U$. Hence, one can deduce that the growth factor of GE for these matrices is optimal (see [22, pp. 164–165]). For the case of NE we also have an optimal growth factor taking into account Remark 3.5 of [7] and the argument of Theorem 6.1 of [8].

A nonsingular TP matrix $A$ is said to be Almost Strictly Totally Positive (ASTP) if it satisfies the following condition: each minor $\det(C)$ of $A$ is positive if and only if all the diagonal elements of $C$ are positive. These matrices form an important class of TP matrices (see [13,14]) which include that of strictly totally positive matrices and many collocation matrices used in CAGD, such as B-spline collocation matrices (see [24]). For nonsingular ASTP matrices we have (see Proposition 3.5 of [13]) that for sufficiently high finite precision arithmetic the NE of $A$ can be carried out without row exchanges and the computed matrices $A^{(t)}$ $(t = 1, 2, \ldots, n)$ are nonnegative. Besides, it was proved for ASTP matrices in [3] that NE (without any pivoting strategy) presents very small backward errors, analogously to GE without pivoting (see [6]). Moreover, let us remind that, in the case of applying NE to these matrices, no row exchanges is the best pivoting strategy (see [17]). Finally, convergence results for IR when using NE have been obtained in [2]. All these results can be applied to ASTP matrices. However, the family of examples included in this section does not belong to ASTP matrices, and we shall focus on computational complexity.

Let us consider Lehmer matrices $A_{ij} = (a_{ij})_{1 \leq i,j \leq n}$, which are defined by

$$a_{ij} = \begin{cases} i/j, & \text{if } j \geq i, \\ j/i, & \text{if } j < i. \end{cases}$$

These matrices have been considered, for example, by Higham in [19,20]: the version 3.0 of the Test Matrix Toolbox for MATLAB 4.2 also includes Lehmer matrices. In the current versions of MATLAB “gallery (‘lehmer’,n)” returns the Lehmer matrix of order $n$.

The Lehmer matrices form a particular case of Example (g) in p. 214 of [5]. With the notation of that paper, it is sufficient to take the sequences $(b_i)_{i=1}^n$ and $(c_i)_{i=1}^n$ defined by $b_i = i$ and $c_i = 1/i$ for all $i \in \{1, \ldots, n\}$ since $(b_i/c_i)_{i=1}^n$ is an increasing sequence, and the matrix $A = (a_{ij})_{1 \leq i,j \leq n}$ given by $a_{ij} = b_{\max(i,j)}c_{\max(i,j)}$ is the Lehmer matrix of order $n$. Therefore Lehmer matrices are TP, in addition to nonsingular matrices. However, observe that Lehmer matrices are not ASTP. For instance, the minor involving the first and second columns and the last but one rows is null in spite that all its diagonal entries are nonzero.

For this type of matrices we have performed several tests with $n$ ranging from 10 to 120. In these tests we have observed that the results provided by the four procedures (GE, GEPP, NE and NEPP) have a very similar behaviour with respect to the error, both with and without IR.

Finally, let us compute the number of operations when applying NE and GE on such systems and let us see that NE has a lower computational cost than GE. If we apply GE without partial pivoting to such a system the matrix $A^{(2)}$ turns out to be already an upper triangular matrix by the particular structure of the Lehmer matrices, that is, $A^{(2)} = A^{(3)} = \ldots = A^{(n)} = U$. Therefore, the application of NE to these systems consists of $n - 1$ quotients, $n(n - 1)$ sums/subtractions and $n(n - 1)$ products corresponding to the first step of the elimination procedure. On the other hand, if we apply GE without partial pivoting we cannot avoid any operation. Hence, NE for these particular systems has a computational cost of $c(n^2)$ operations in contrast to the computational cost of $c(n^3)$ operations corresponding to GE.
5. Discussion and conclusions

Gaussian elimination is the most usual procedure to transform a nonsingular matrix into an upper triangular matrix. The relationship between large growth factors and loss of accuracy when using Gaussian elimination is well known (see [21,28]). Partial pivoting strategy is used when applying Gaussian elimination in order to control the magnitude of the growth factor: with this pivoting strategy the growth factor is bounded above by $2^{n-1}$. Moreover, the relative and the componentwise backward error of the computed solution are considered as a posteriori indicators of the performance of the method.

In [30] Wright showed with examples arising in real applications how Gaussian elimination with partial pivoting can fail when solving some linear systems. Gaussian elimination with partial pivoting provides very low growth factors in general, but for some particular matrices it can become great. For instance, in [22] the so-called gfpp matrices were presented, for which the maximum growth factor is attained with Gaussian elimination with partial pivoting.

Backward stability of Gaussian elimination with partial pivoting depends on the growth factor $\rho_{\text{NE}}^W$ (see Section 9.3 of [21]). A backward and forward error analysis of Neville elimination are performed in [3,4] respectively. The growth factor is again involved in these error analysis.

In this paper we have shown examples where Neville elimination outperforms Gaussian elimination. This has to do with both the accuracy of the solution of the iterative refinement and the computational cost of the method. Hence, one of the reasons of the better behaviour of Neville elimination in contrast to Gaussian elimination with the orthogonal and gfpp matrices is the lower size of the growth factors associated with Neville elimination. In Tables 1–3 and 5,6 it is seen how the relative errors corresponding to the Neville elimination are much smaller than those of Gaussian elimination. The same phenomenon holds if we consider the backward componentwise error.

Finally, we have also checked that Neville elimination has a computational cost lower than Gaussian elimination for gfpp and Lehmer matrices. A lower computational cost is always desirable from an algorithmic perspective and it might explain in some particular cases the lower backward error.

Acknowledgements

This work has been partially supported by the Spanish Research Grant MTM2009-07315, by Gobierno de Aragón and under MEC and FEDER Grant TIN2007–61273.

References