Numer. Math. 54, 591–599 (1989)

Numerische Mathematik
© Springer-Verlag 1989

Rehabilitation of the Gauss-Jordan Algorithm

T.J. Dekker and W. Hoffmann
Department of Computer Systems, University of Amsterdam, Kruislaan 409, 1098 SJ Amsterdam, The Netherlands

Summary. In this paper a Gauss-Jordan algorithm with column interchanges is presented and analysed. We show that, in contrast with Gaussian elimination, the Gauss-Jordan algorithm has essentially differing properties when using column interchanges instead of row interchanges for improving the numerical stability. For solutions obtained by Gauss-Jordan with column interchanges, a more satisfactory bound for the residual norm can be given. The analysis gives theoretical evidence that the algorithm yields numerical solutions as good as those obtained by Gaussian elimination and that, in most practical situations, the residuals are equally small. This is confirmed by numerical experiments. Moreover, timing experiments on a Cyber 205 vector computer show that the algorithm presented has good vectorisation properties.

Subject Classifications: AMS(MOS): 65F05, 65G05, 15A06; CR: G1.3.

1. Introduction

With the advent of vector and parallel computers, the Gauss-Jordan algorithm has received renewed interest because of its supposedly good properties with respect to vectorization and parallelization.

The stability of the Gauss-Jordan algorithm with partial pivoting has been analysed by Peters and Wilkinson [6] who came to the following conclusion: *"in general the absolute error in the solution is strictly comparable with that corresponding to Gaussian elimination with partial pivoting plus back substitution; however, when the matrix is ill conditioned, the residual corresponding to the Gauss-Jordan solution will often be much greater than that corresponding to the Gaussian elimination solution."* These results hold true for the standard column pivoting strategy, where at each stage a pivot is selected in a certain column and correspondingly rows are interchanged to bring the pivot in diagonal position.

In this paper we show that Gauss-Jordan with row pivoting, and correspondingly interchanging of columns, is much more satisfactory. In most practical situations, the residual corresponding to the solution obtained by Gauss-Jordan with row pivoting is not larger than that corresponding to the Gaussian elimination solution.

The Gauss-Jordan algorithm with any pivoting strategy is equivalent to Gaussian elimination – with the same pivoting strategy – followed by a further reduction of the resulting upper triangular system to a diagonal system. With column pivoting this further reduction may yield arbitrarily large elements and, hence, a large residual, as is shown by Peters and Wilkinson. With row pivoting, however, the elements of the resulting upper triangular matrix are bounded by the diagonal elements in the corresponding rows.

For an error analysis it is convenient to consider Gauss-Jordan's algorithm with row scaling, i.e. at each stage the equation corresponding to the pivotal row is divided by the pivot. Then the resulting matrix U is unit upper triangular and its elements are bounded by 1. It follows that the growth of the elements in the further reduction to diagonal form is not much larger than the norm of the inverse of U. Consequently, the residual of the calculated solution is not much larger than that corresponding to the Gaussian elimination solution, except in those rare cases where U is ill conditioned.

In Sect. 2 we consider the Gauss-Jordan algorithm in more detail and present an error analysis. In Sect. 3 we give some numerical results, showing that the error and mostly also the residual are satisfactory, and some results of experiments on a Cyber 205 vector computer, showing that Gauss-Jordan is not slower than Gaussian elimination for systems of order up to 25, although it requires about 1.5 times more work.

2. Error Analysis of Gauss-Jordan with Row Pivoting

Let A be a given matrix of order n and b a given right-hand side vector. The application of the Gauss-Jordan elimination on the given system is equivalent with performing n successive transformations, starting from the original matrix $A^{(1)}=A$ and right-hand side $b^{(1)}=b$. The total effect is the transformation of $A^{(1)}$ with permuted columns into the identity matrix. This is described in the following algorithm.

```
For $k=1$ (1) $n$ do
    Determine $p$ such that $k \leq p \leq n$ and $|A_{kp}^{(k)}| = \max_{k \leq j \leq n} |A_{kj}^{(k)}|$

    $P_k$ $:= I-(e_k-e_p)(e_k-e_p)^T$
        {permutation matrix for interchanging colums $p$ and $k$}
    $\delta_k$ $:= A_{kp}^{(k)}$
    $D_k$ $:= I+(\delta_k-1)\, e_k e_k^T$    $\{=\mathrm{diag}(1, \ldots, 1, \delta_k, 1, \ldots, 1)\}$
    $g_k$ $:= \delta_k e_k - A^{(k)} e_p$
    $G_k$ $:= g_k e_k^T$
    $A^{(k+1)} := (I+G_k)\, D_k^{-1} A^{(k)} P_k$
    $b^{(k+1)} := (I+G_k)\, D_k^{-1} b^{(k)}$

enddo
```

Fig. 1. Gauss-Jordan algorithm with column interchanges

The application of this algorithm results in $A^{(n+1)}=I$ and $b^{(n+1)}=P_n^{-1}\dots P_1^{-1}x$.

Summarizing, with the use of $P=P_1\dots P_n$, the effect of all elimination-steps is given by:

$$(I+G_n)D_n^{-1}\dots(I+G_1)D_1^{-1}(AP|b)=(I|P^{-1}x).$$

For our theoretical analysis we introduce the following notation.

Define m_k to be equal to the lower part of g_k (below the k-th element) and v_k equal to the upper part of g_k such that $[m_1,m_2,\dots,m_n]$ is strictly lower triangular and $[v_1,v_2,\dots,v_n]$ strictly upper triangular.

Furthermore

$$M_k:=m_k e_k^T \quad\text{and}\quad V_k:=v_k e_k^T.$$

Then we have

$$g_k=m_k+v_k,\qquad G_k=M_k+V_k,$$

and

$$(I+G_k)=(I+V_k)(I+M_k).$$

We observe that

$$(I+M_i)D_i^{-1}(I+V_j)=(I+V_j)(I+M_i)D_i^{-1},\qquad \text{for } j<i,$$

so that

$$(I+G_n)D_n^{-1}\dots(I+G_1)D_1^{-1}=(I+V_n)\dots(I+V_1)(I+M_n)D_n^{-1}\dots(I+M_1)D_1^{-1}.$$

If L and V are defined by

$$\begin{aligned}L&:=[(I+M_n)D_n^{-1}\dots(I+M_1)D_1^{-1}]^{-1}=D_1(I-M_1)\dots D_n(I-M_n)\\&=(D_1\dots D_n-M_1-\dots-M_n),\\V&:=(I+V_n)\dots(I+V_1)=I+V_1+\dots+V_n,\end{aligned}$$

then the Gauss-Jordan elimination is symbolically given by:

$$VL^{-1}(AP|b)=(I|P^{-1}x).$$

Let the upper triangular matrix U and vector y, which are intermediate results during the calculation, be defined by

$$U:=L^{-1}AP;\qquad y:=L^{-1}b,$$

then the error analysis of (standard) Gaussian elimination shows that these calculated L, U and y satisfy:

$$LU=AP+E_1,\qquad \text{with } \|E_1\|\leqq\phi_1(n)\,g\,\|A\|\,\mu, \tag{2.1}$$

and

$$(L+E_2)\,y=b,\qquad \text{with } \|E_2\|\leqq\phi_2(n)\,\|L\|\,\mu, \tag{2.2}$$

where $\phi_1(n)$ and $\phi_2(n)$ are low-degree polynomials in n, g is the growth factor and μ is a small, arithmetic-dependent, constant times the machineprecision (see e.g. [3, 7]).

For the rest of our rounding error analysis we have to examine the remaining part of the algorithm. This is the part where V is calculated such that

$$V(U\,|\,y)=(I\,|\,P^{-1}x).$$

The pivoting strategy and row-scaling in the first part of the algorithm have ensured that $|U_{ij}|\leqq 1$ for $j>i$ and $U_{ii}=1$.

Defining $U^{(1)}=U$ and $y^{(1)}=y$, the calculation is carried out according to the following rules. Note that these rules are part of the algorithm described in Fig. 1.

```
For $k=1\ (1)\ n$ do

    $v_k\quad := e_k - U^{(k)} e_k$
    $V_k\quad := v_k e_k^T$
    $U^{(k+1)} := (I+V_k)\, U^{(k)}$
    $y^{(k+1)} := (I+V_k)\, y^{(k)}$

enddo
```

Fig. 2. Inversion and solution of triangular system

The result of this calculation is $U^{(n+1)}=I$ and $y^{(n+1)}=P^{-1}x$.

For the calculated quantities we observe that for each k an error matrix $F^{(k)}$ exists such that

$$U^{(k+1)}=U^{(k)}+V_k U^{(k)}+F^{(k)}, \qquad \text{with } F_{ij}^{(k)}=0 \text{ for } j\leqq k \text{ and } i\geqq k.$$

A simple rounding error analysis yields:

$$\max|F_{ij}^{(k)}|\leqq 3 \max_{i<k} |U_{ij}^{(k)}|\,\mu.$$

Since $(I+V_i)\,F^{(k)}=F^{(k)}$ for $i\geqq k$, we obtain

$$I=U^{(n+1)}=(I+V_n)\ldots(I+V_1)\,U+F^{(1)}+F^{(2)}+\ldots+F^{(n)}.$$

For V this implies

$$VU+E_3=I, \qquad \text{with } E_3=F^{(1)}+F^{(2)}+\ldots+F^{(n)}. \tag{2.3a}$$

For an estimate of E_3 we need a bound for $\max\limits_{i<k}|U_{ij}^{(k)}|$.

From

$$U^{(k)}=(I+V_1+\ldots+V_{k-1})\,U$$

we find

$$\max_{i<k} |U_{ij}^{(k)}| \leqq \max_{i<k} |U_{ij}| \left\{ 1 + \sum_{h=1}^{k-1} \max_{i,j} |(V_h)_{ij}| \right\}$$
$$\leqq \{1 + (k-1) \cdot \max |V_{ij}|\} \leqq \{1 + (k-1) \, \|V\|\}.$$

Using $\|F^{(k)}\| \leqq n \max_{i,j} (|F_{ij}^{(k)}|)$ we find for $\|E_3\|$

$$\|E_3\| \leqq \sum_{k=1}^{n} \|F^{(k)}\| \leqq 3n \sum_{k=1}^{n} \{1 + (k-1) \, \|V\|\} \, \mu,$$

which gives

$$\|E_3\| \leqq \phi_3(n) \, \|V\| \, \mu \tag{2.3b}$$

for a low-degree polynomial ϕ_3 in n.

With respect to the error in the calculated solution $z = P^{-1} x$, we notice that this calculation is numerically equivalent with multiplying y from the left by V, hence

$$(V + E_4) \, y = z, \quad \text{with } \|E_4\| \leqq \phi_4(n) \, \|V\| \, \mu, \tag{2.4}$$

for a low-degree polynomial ϕ_4 in n.

The combination of formulae (2.3) and (2.4) yields:

$$y = U(I - E_3 + E_4 U)^{-1} z,$$

which in combination with (2.1) and (2.2) gives

$$b = (A + E_1 + E_2 U)(I - E_3 + E_4 U)^{-1} z. \tag{2.5}$$

If we put

$$w = (I - E_3 + E_4 U)^{-1} z, \tag{2.6a}$$

then this results in

$$b = (A + E_1 + E_2 U) \, w. \tag{2.6b}$$

If we furthermore use E_5 for $(E_3 - E_4 U)$ then the distance between z and w satisfies

$$\|z - w\| / \|z\| \leqq \|E_5\| / (1 - \|E_5\|) \text{ provided that } \|E_5\| < 1. \tag{2.7}$$

For $\|E_5\|$ we find the following bound

$$\|E_5\| \leqq \|E_3\| + \|E_4\| \, \|U\| \leqq (\phi_3(n) + \phi_4(n) \, \|U\|) \, \|V\| \, \mu$$
$$\leqq \phi_5(n) \, \|V\| \, \mu \tag{2.8}$$

for a low-degree polynomial ϕ_5.

Using (2.3a) for a bound on $\|V\|$ this can be written as

$$\|E_5\| \leqq \phi_5(n) \, \|U^{-1}\| \, \mu / \{1 - \phi_3(n) \, \|U^{-1}\| \, \mu\}, \tag{2.9}$$

provided that the denominator is positive.

Summarizing, the calculated solution $z = P^{-1} x$ is close to a vector w, which is the exact solution of a nearby problem as specified in formulae (2.6a) and (2.6b).

For the residual $r := b - A z$ we have according to these formulae

$$r = (A + E_1 + E_2 U) w - A(I - E_5) w,$$

which can be bounded by

$$\|r\| \leqq (\|E_1\| + \|E_2 U\| + \|A\| \, \|E_5\|) \, \|z\| / (1 - \|E_5\|). \tag{2.10}$$

In this bound the contribution $\|A\| \, \|E_5\| / (1 - \|E_5\|)$ creates the essential difference with the formula for the residual bound for Gaussian elimination.

As long as $\|E_5\| \ll 1$, this term has order of magnitude $\|A\| \, \|U^{-1}\| \, \mu$. As a consequence of our pivoting strategy, U will mostly be well-conditioned, even in cases where A itself is ill-conditioned, so that the contribution of this term is harmless. However, a well known example of an ill-conditioned unit triangular matrix is given in the next section in experiments series d.

3. Numerical Experiments

Experiments on accuracy and timing were carried out on the Cyber 205 computer (one vector pipe) of the Academic computer centre SARA in Amsterdam; the arithmetic precision of this machine is about 10^{-14}.

For a large number of linear systems we compared the solution obtained via Gauss-Jordan with row pivoting with the solution from Gaussian elimination. These experiments are described hereafter and listed in Table 1.

For timing results we compared the CP time for our Gauss-Jordan algorithm with the CP time for LINPACK-routines SGESL and SGEFA [2] and with the CP time for the NUMVEC implementation of $LD^{-1}U$ factorization with

Table 1. Overview of experiments on accuracy

Series	Order	# Syst.	A	x	b	Cond. nr	# correct digits (rel. to x)			
							solution		residual	
							Gauss	G.-J.	Gauss	G.-J.
a1	25	200	$U\Sigma V^T$	random	Ax	10^{15}	<2	same	>12	same
a2	50	1800	$U\Sigma V^T$	random	Ax	10^6–10^{15}	<8	same	>12	same
a3	50	200	$U\Sigma V^T$	$A^{-1}b$	random	10^{10}	3–7	same	>12	same
a4	50	200	$U\Sigma V^T$	v_{50}	$\sigma_{50} u_{50}$	10^{10}	3–7	same	>12	same
b1	25	200	upper	random	Ax	$>10^7$	<2	same	>12	same
b2	25	200	upper	$A^{-1}b$	random	$>10^7$	<2	same	>12	same
c1	50	200	W	random	Ax	1700	0–4	0–1	0–4	0–1
c2	30	200	W	random	Ax	<1700	5–10	5–7	5–9	5–6
d1	50	200	Δ	random	Ax	$>10^{14}$	0–4	0–1	>12	0–1
d2	30	200	Δ	random	Ax	$\cong 10^{10}$	5–10	5–7	>12	6–7

row pivoting (which is equivalent with Gaussian elimination) followed by forward and backward substitution [4, 5]. An overview of these results is given in Table 2.

Our implementation of the Gauss-Jordan algorithm is a slight modification of the algorithm described above. In each step the factor D_k^{-1} is omitted so that the resulting matrix is given by $A^{(n+1)}=D=\operatorname{diag}(\delta_1, \ldots, \delta_n)$. In this form the algorithm is more efficient on the Cyber 205 vector computer, because no extra updating of the pivotal row in each step is required. The error analysis remains essentially the same.

Experiments on Accuracy and Residuals

a) In test series a1–a4 we use linear systems with prescribed condition. The matrices are constructed from a given diagonal matrix (the singular values chosen) which is pre- and post-multiplied by random orthogonal matrices. These left and right orthogonal factors are the product of $\sqrt{n}$ random Householder reflections. The singular values are chosen in various ways; the largest always $+1$, the smallest 10^{-6} or smaller and the remaining ones either distributed equally, or clustered on one end of the spectrum, or on the other end.

In series a1 we use very ill-conditioned matrices of order 25; the right-hand side vector b is constructed by taking the product of the coëfficiënt-matrix A and a random vector x.

In series a2 we use matrices of order 50 of the same type and with the same type of right-hand side vector.

In series a3 we use a different type of right-hand side vector. Firstly the linear system is solved with a random right-hand side vector. With the solution x_0 of this system, the vector $b_0=Ax_0$ is calculated. This vector b_0 serves as right-hand side vector in the test system. For ill-conditioned matrices the right-hand side vector constructed in this way is in general "rich" in the least singular vector of the matrix, so that the solution is very sensitive for perturbations.

In series a4 the left singular vector corresponding to the least singular value is taken as right-hand side vector.

All these series yield solutions with accuracy as expected in view of the condition number of the matrix, and small residuals both for Gaussian elimination and Gauss-Jordan factorization with no significant difference.

b) In series b1–b2 we use upper triangular matrices. The diagonal elements have the value $+1$ except A_{33} and A_{44} which have the value 10^{-7}. The elements in the strictly upper triangular part have random values between -1 and $+1$. This type of matrices is used by Peters and Wilkinson [6] to show that Gauss-Jordan with column pivoting can produce larger residual vectors than Gaussian elimination. The choices for the right-hand sides in b1 and b2 are made in the same way as in series a1 and a3 respectively. The results of Gauss-Jordan factorization with row pivoting and Gaussian elimination are fully comparable and as accurate as can be expected in view of the condition of the matrices.

We also tested Gauss-Jordan with column pivoting on these matrices. The accuracy of the solution is comparable with the accuracy in the other solutions, but the residual is much larger (of the same size as the error in the solution), which confirms the analysis in [6].

c) In series c1–c2 we use a matrix W for which maximal growth in its elements is obtained during Gaussian elimination with partial pivoting. For our situation where row pivoting is performed, this matrix is given by $W_{ij} = -1$ for $j > i$; $W_{jj} = W_{nj} = 1$ for all j and $W_{ij} = 0$ elsewhere.

$$W = \begin{pmatrix} 1 & -1 & \dots & & -1 \\ 0 & 1 & & & -1 \\ \vdots & & & & \vdots \\ 0 & & 0 & 1 & -1 \\ 1 & \dots & & 1 & 1 \end{pmatrix}.$$

For $n = 50$, as used in series c1, the conditionnumber of W roughly equals 1700 and the element growth is 2^{49}. As right-hand side vector we have chosen the product Wx for a random vector x.

In series c2 we use the same experiment but now for the order $n = 30$.

The results of Gaussian elimination are for some cases slightly better than with the Gauss-Jordan factorization.

Note that for this matrix, a result obtained with column pivoting is correct to almost full working accuracy; this is true for both Gaussian elimination and Gauss-Jordan factorization. An implementation of a variant of Gaussian elimination where this dangerous element-growth is detected and can be cured is given by Hoffmann and Lioen [5]. The technique used is presented in a paper by Businger [1] and can also be applied to the Gauss-Jordan algorithm.

d) In series d1–d2 we use a unit upper triangular matrix Δ having all elements in the strictly upper triangular part equal to -1.

$$\Delta = \begin{pmatrix} 1 & -1 & \dots & & -1 \\ 0 & 1 & & & \\ \vdots & & & & \vdots \\ & & & 1 & -1 \\ 0 & \dots & & 0 & 1 \end{pmatrix}.$$

The least singular value of Δ is less than $2^{-n}\sqrt{3}$. As right-hand side vector, we take matrix times random vector. For this type of matrices, which have an increasing bad condition for growing values of the order n, the Gauss-Jordan factorization for large values of n produces a solution with a much larger residual vector than the solution produced by Gaussian elimination.

In all our experiments we calculated the number of correct digits in the solution and in the residual. For a system with right-hand side b, exact solution x_0 and calculated solution x these numbers are given by $-{}^{10}\log(\|x - x_0\|/\|x_0\|)$ and $-{}^{10}\log(\|b - Ax\|/\|x_0\|)$ respectively. In fact the latter quotient between brackets should also be devided by $\|A\|$ for a homogeneous result. However, for all matrices in series a we have $\|A\|_2 = 1$ and for all other matrices the norms are of order n, so the omission of this factor is harmless.

Experiments on Timing

Our implementation of the Gauss-Jordan algorithm with row pivoting (column interchanges) GJPCF was compared with routines from LINPACK and with the NUMVEC implementation of Gaussian elimination, CCRPCF [4, 5]. The timing results are as in the following table.

Table 2. Overview of timing experiments

	$n=25$	50	100	200
LINPACK (SGEFA+SGESL)	0.0028	0.0107	0.0441	0.1965
CCRPCF	0.0014	0.0051	0.0232	0.1154
GJPCF	0.0014	0.0054	0.0256	0.1394

These results show that the Gauss-Jordan algorithm is rather efficiënt on a vector computer and that the processing time is competitive with Gaussian elimination for order up to 25.

As is well known, the number of floating-point operations equals, apart from lower order terms, n^3 for Gauss-Jordan and $(2/3)\,n^3$ for Gaussian elimination. The time needed for these algorithms, however, is not only determined by this order n^3 term, but also by a significant contribution of order n^2, needed for pivot search and interchanges and, on a vector machine, also for the start-up of the vector iterations. This contribution of order n^2 is (nearly) equal for CCRPCF and GJPCF, as also appears from this table.

Acknowledgements. We want to thank professors G.H. Golub, A. Ruhe and G.W. Stewart for mentioning to the second author the key reference [6] at the conference at Loen, Norway.

References

1. Businger, P.A.: Monitoring the numerical stability of Gaussian elimination. Numer. Math. **16**, 360–361 (1971)
2. Dongarra, J.J., Moler, C.B., Bunch, J.R., Stewart, G.W.: LINPACK User's guide. Philadelphia: SIAM 1979
3. Golub, G.H., Van Loan, C.F.: Matrix computations. Oxford: North Oxford Academic 1983
4. Hoffmann, W.: Solving linear systems on a vector computer. J. Comput. Appl. Math. **18**, 353–367 (1987)
5. Hoffmann, W., Lioen, W.M.: Chapter simultaneous linear equations. Report NM-R8614. In: NUMVEC FORTRAN Library Manual. Amsterdam: Centre for Mathematics and Computer Science 1986
6. Peters, G., Wilkinson, J.H.: On the stability of Gauss-Jordan elimination with pivoting. Commun. ACM **18**, 20–24 (1975)
7. Stewart, G.W.: Introduction to matrix computations. New York London: Academic Press 1973