

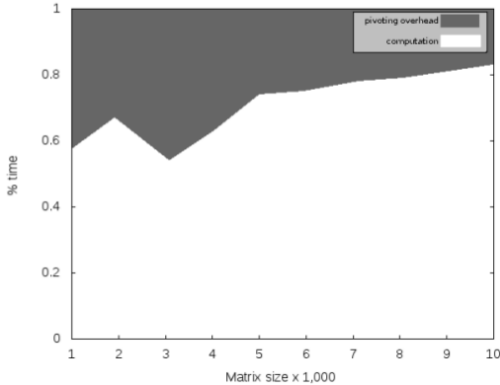
Mixed accelerated techniques for solving dense linear systems

Alexandros S. Papadakis, Efstratios Gallopoulos
 papadakis@ceid.upatras.gr, stratis@ceid.upatras.gr

Abstract. The rounding-error analysis of Gaussian elimination shows that the method is stable only when the elements of the matrix do not grow excessively in the course of the reduction. Usually such growth is prevented by interchanging rows and columns of the matrix so that the pivot element is acceptably large. In this paper firstly we introduce the Boosting LU factorization method based on a rank one modification. In the next we propose an efficient algorithm based on an existing algorithm that utilizes random transformation of the coefficient matrix to solve dense linear systems without pivoting. In the end we develop a mixed algorithm for solving dense linear systems.

Key words. dense linear algebra, Sherman-Morrison-Woodbery formula, random transformations, rank one modification, boosting, Gaussian elimination, LU factorization.

1. Introduction. It is known that for a general matrix, the solution based on a LU factorization, is stable only if some kind of pivoting strategy is utilized. However, pivoting can significantly complicate the algorithm, increase data movement, and reduce speed, particularly on high-performance computers. For example figure 1, which was taken from [2], shows the cost of pivoting, the percentage of time due to pivoting in LU factorization for several sizes of random matrices. We observe that pivoting can represent more than 40% of the global factorization time for small matrices and although the overhead decreases with the size of the matrix, it still represents 17% for a matrix of size 10000.



The fact that pivoting remains a bottleneck for linear system solutions is a motivation to present in this paper.

Let A be a real matrix of order n . The method of Gaussian elimination may be regarded as a technique for computing the LU decomposition of A into the product of a unit lower triangular matrix L and an upper triangular matrix U . Specifically, at the k th step of the reduction (for example k th pivot $<$ threshold), we have

$$A = \begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I \end{pmatrix} \begin{pmatrix} U_{11}^{(k)} & U_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix}$$

so to solve a system $Ax=b$, the form is

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

To find the solution we need to solve 2 systems of form

$$\begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (1)$$

and

$$\begin{pmatrix} U_{11}^{(k)} & U_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad (2)$$

The solutions y_1, y_2, x_1 can be found by solving triangular systems. To find x_2 solution we need to solve $A_{22}^{(k)} x_2 = y_2$. If we don't want to do pivoting to find x_2 , in bibliography [1],[3] there are 2 ideas that accomplish it. Both ideas try to solve a new system $Bz = y_2$ and then they correct the solution z to find the original solution x_2 . In the next sections we briefly mention the two existed algorithms and develop new. In the end we compare each algorithm under our experiment framework.

From now on when we refer in phrase "bad pivot", we mean a pivot that its absolute value is less than a constant selected threshold.

2. Stewart's algorithm. The idea is based on a rank one modification of the matrix $A_{22}^{(k)}$. Specifically the new matrix B and $A_{22}^{(k)}$ differ only in their (1,1)-elements. So B can be written in the form $B = A_{22}^{(k)} + \sigma e_1 e_1^T$, where e_1 is the first column of the identity matrix.

Then it follows from the well known Sherman-Morrison-Woodbery formula that

$$A_{22}^{(k)-1} = B^{-1} - \frac{B^{-1} e_1 e_1^T B^{-1}}{e_1^T B^{-1} e_1 - \sigma^{-1}}$$

So since we have found the solution of system $Bz = y_2$ we can find the original solution x_2 by

$$x = z - \frac{B^{-1}e_1e_1^T z}{e_1^T B^{-1}e_1 - \sigma^{-1}}$$

Now it follows the Stewart algorithm

- 1) perform LU since you have found kth pivot less than threshold
- 2) if there is not bad pivot, solve the system $LUx=b$ and return else solve the system

$$\begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

- 3) craft a new matrix $B = A_{22}^{(k)} + \sigma e_1 e_1^T$, σ is selected so the pivot is greater than threshold
- 4) solve the systems $Bz = y_2$ and $Bc_1 = e_1$ using Stewart algorithm
- 5) correct the solution z to find the original solution x_2 using the Sherman-Morrison-Woodbery formula
- 6) solve the system $U_{11}^{(k)}x_1 = y_1 - U_{12}^{(k)}x_2$
- 7) return $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ as solution

We observe that step 4 of Stewart's algorithm uses a recursion. Stewart in [1] agrees that if there are too much bad pivots his algorithm is not so good. In the next section, we present a slight modification of the above idea so that the algorithm doesn't need a recursion.

3. Boosting LU factorization. We introduce a new LU factorization method that avoid pivoting and utilizes the Stewart's idea.

It is known [4] that when we perform Gauss elimination on a matrix A to find L, U triangular matrices for which $A=LU$ for each step of elimination we have

step 1.

$$\begin{aligned} U_1 &= E_{N,1} \dots E_{3,1} E_{2,1} A, \\ L_1 &= E_{2,1}^{-1} E_{3,1}^{-1} \dots E_{N,1}^{-1} \end{aligned}$$

step 2.

$$\begin{aligned} U_2 &= E_{N,2} \dots E_{4,2} E_{3,2} L_1^{-1} A, \\ L_2 &= L_1 E_{3,2}^{-1} E_{4,2}^{-1} \dots E_{N,2}^{-1} \end{aligned}$$

...

step k.

$$\begin{aligned} U_k &= E_{N,k} \dots E_{k+2,k} E_{k+1,k} L_{k-1}^{-1} \dots L_2^{-1} L_1^{-1} A, \\ L_k &= L_1 L_2 \dots L_{k-1} E_{k+1,k}^{-1} E_{k+2,k}^{-1} \dots E_{N,k}^{-1} \end{aligned}$$

Where $E_{i,j}$ is the identity matrix in which the element (i, j) is the quotient of division of (i, j) and (j, j) elements of $L_{j-1}^{-1} \dots L_2^{-1} L_1^{-1} A$ matrix.

Then assuming that there isn't appeared any bad pivot, we have $U = U_N$ and $L = L_N$

Now let's suppose that at step $k+1$, the $k+1$ pivot is bad. Using the Stewart idea of a rank one modification, we have

step $k+1$.

$$\begin{aligned} \hat{U}_{k+1} &= E_{N,k+1} \dots E_{k+2,k+1} (L_k^{-1} \dots L_1^{-1} A + \sigma_{k+1} e_{k+1} e_{k+1}^T) \\ \hat{U}_{k+1} &= L_{k+1}^{-1} (L_k^{-1} \dots L_1^{-1} A + \sigma_{k+1} e_{k+1} e_{k+1}^T) \\ L_1 \dots L_{k+1} \hat{U}_{k+1} &= A + \sigma_{k+1} L_1 \dots L_k e_{k+1} e_{k+1}^T \end{aligned} \quad (1)$$

The term $\sigma_{k+1} L_1 \dots L_k e_{k+1} e_{k+1}^T$ is equal to $\sigma_{k+1} e_{k+1} e_{k+1}^T$ because of each L_j can be written as $I + u_j e_j$ where u_j is and e_j is and $1 \leq j \leq k$

So equation 1 is written as

$$\hat{L}_{k+1} \hat{U}_{k+1} = A + \sigma_{k+1} e_{k+1} e_{k+1}^T$$

As we can see, the product $\hat{L}_N \hat{U}_N$ is the LU factorization of the boosted matrix A . In the next, the above technique is referred as geboost factorization.

4. Parker's algorithm. The idea is to transform the matrix A randomly so that with probability 1, LU factorization without pivoting can be accomplished. Parker has proved [3] that there are special crafted random matrices (butterfly matrices) which when permultiply a matrix, the product matrix does not need pivoting on a LU factorization. Dongarra et al [2] demonstrate an efficiently method of storing butterfly matrices and computing the product.

Now it follows the Parker algorithm

- 1) compute butterfly matrices G, H
- 2) compute the product $B = G^T A H$
- 3) solve the system $Bz = G^T b$ with LU without pivoting
- 4) correct the solution z to find the original solution x , $x = Hz$

As it is mentioned in [2],[3] a butterfly matrix has size which is multiple of 2^d , where d is the depth of butterfly matrix. If the the size of matrix A is not multiple of 2^d , matrix A is augmented with additional 1's on the diagonal. This has the effect of additional overhead due to permultiply (step 2 of algorithm) which is applied to bigger matrices. Although Parker has proved that for $d = \log_2 N + 1$ the coefficient matrix which is the product of step 2 of algorithm does not need any pivoting during Gauss elimination, Dongarra et al [2] show that for $d = 2$ the algorithm is stable. From now on we consider $d = 2$ in the following sections.

5. Parker's algorithm when needed. Now we propose an efficient algorithm based on Parker's algorithm. If during the LU factorization, none pivot is less than

threshold, why to perform Parker's algorithm? It is better to perform Parker's algorithm when there is really need. So the butterfly matrices will be smaller and the product can be computed faster.

Let's suppose that we have done LU factorization of matrix A just to the kth pivot. So we need to solve two systems

$$\begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (1)$$

and

$$\begin{pmatrix} U_{11}^{(k)} & U_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad (2)$$

If we permultiply matrix $A_{22}^{(k)}$ by two butterfly matrices G,H, we have

$$\begin{pmatrix} I & 0 \\ 0 & G^T \end{pmatrix} \begin{pmatrix} U_{11}^{(k)} & U_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & H \end{pmatrix} = \begin{pmatrix} U_{11}^{(k)} & U_{12}^{(k)} H \\ 0 & G^T A_{22}^{(k)} H \end{pmatrix}$$

and our new systems are

$$\begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (1)$$

and

$$\begin{pmatrix} U_{11}^{(k)} & U_{12}^{(k)} H \\ 0 & G^T A_{22}^{(k)} H \end{pmatrix} \begin{pmatrix} x_1 \\ z \end{pmatrix} = \begin{pmatrix} y_1 \\ G^T y_2 \end{pmatrix} \quad (2)$$

Now it follows the plu algorithm

- 1) perform LU since you have found the kth pivot less than threshold
- 2) if there is not bad pivot, solve the system $LUx=b$ and return else solve the system

$$\begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

- 3) compute butterfly matrices G,H
- 4) compute the product $B = G^T A_{22}^{(k)} H$
- 5) solve the system $Bz = G^T y_2$ with LU without pivoting
- 6) correct the solution z to find the original solution x_2 , $x_2 = Hz$
- 7) solve the system $U_{11}^{(k)} x_1 = y_1 - U_{12}^{(k)} x_2$
- 8) return $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ as solution

Although at the first glance, plu seems better than parker algorithm, there are occasions that parker is better. First of all let's take the occasion which matrix size of A is of form $k2^d + 1$. Then if in matrix A is applied parker algorithm the augmentation that mentioned in previous section will cause the size of matrix A to be $k2^d + 1 + (2^d - 1)$. It is obvious then that the permultiply will cause a redoubtable overhead. This is the worst case of parker algorithm. Now if in matrix A is applied plu algorithm and the bad pivot appeared from $k2^d$ to 2, we

expect that plu solves the problem faster than parker does. In next let's take the occasion which the size of matrix A is multiple of 2^d , for example $k2^d$. If problem is going to be solved by parker algorithm, there will be not any augmentation which implies no additional overhead. Now in the case we don't choose the parker algorithm, the plu algorithm will be faster only if the bad pivot is between $k2^d - 2^d$ and 2. If bad pivot is between $k2^d - 1$ and $k2^d - 2^d + 1$, we expect plu to be worse than parker. The worst case for plu algorithm is when a bad pivot appeared at $k2^d - 2^d + 1$.

6. Mixed algorithm. Now we are ready to develop a new algorithm which is a mix of Stewart and Parker idea. The idea is to perform NUM times the Stewart's algorithm for the NUM bad pivots and if a NUM+1 bad pivot is appeared to perform the Parker's algorithm. There is a trade-off between small and big values of NUM. For small value implies low cost of Sherman-Morrison-Woodbery formula but high cost of possibly augmentation. For big value holds the inverse. In section 8, it is shown the performance of various values of NUM. Below we describe the idea of mixed algorithm.

First we perform geboost factorization of matrix A with parameter NUM. As a result we have $\hat{L}, \hat{U}, E, \sigma, k$ for which hold that $\hat{L} \hat{U} = A + \sigma EE^T$ and k is the NUMth bad pivot.

Then we need to solve the following two systems

$$\begin{pmatrix} \hat{L}_{11}^{(k)} & 0 \\ \hat{L}_{21}^{(k)} & I \end{pmatrix} \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} = \begin{pmatrix} b_1 & E_1 \\ b_2 & E_2 \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} \hat{U}_{11}^{(k)} & \hat{U}_{12}^{(k)} \\ 0 & \hat{A}_{22}^{(k)} \end{pmatrix} \begin{pmatrix} z_1 & c_1 \\ z_2 & c_2 \end{pmatrix} = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \quad (2)$$

System 1 is solved normally. To solve system 2 we observe that we need to solve the following 2 sub-systems

$$\hat{A}_{22}^{(k)} (z_2 \ c_2) = (y_{21} \ y_{22}) \quad (3)$$

$$\hat{U}_{11}^{(k)} (z_1 \ c_1) = (y_{11} \ y_{12}) - \hat{U}_{12}^{(k)} (z_2 \ c_2) \quad (4)$$

Sub-system 3 is solved by Parker's algorithm and sub-system 4 is solved normally. In the end we find the real solution $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ by correcting the solution $\begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ using the Sherman-Morrison-Woodbery formula.

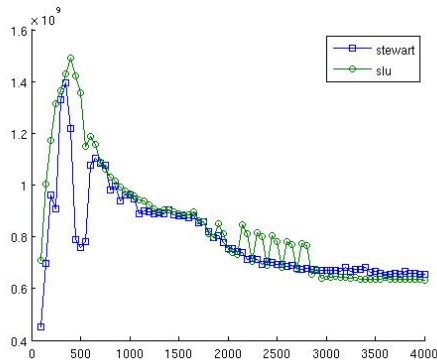
7. Experiment Framework. In our experiments, a critical point to compare each algorithm was the need of matrices that during Gauss elimination they appear enough bad pivots and in selected positions. So we created a framework which a matrix during Gauss elimination appear enough virtual bad pivots and in selected positions. Below we explain the framework.

FW1 model assume the kth pivot as virtual bad pivot if k is divided perfectly by a constant selected value

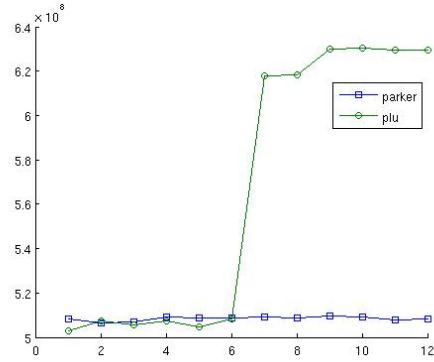
(CSV). FW2 model assume the kth pivot as virtual bad pivot if k is equal to CSV. For example a matrix of size 1000x1000 with the constant selected value to be 100, during a Gauss elimination process, in case FW1 model it appears 10 virtual bad pivots at positions {1000,900,800,...,100} and in the case FW2 model it appears 1 virtual bad pivot at position {100}.

8. Results. We implement all algorithms that mentions in this work in matlab. Performance experiments were run using a Intel Core i5-2500 CPU @ 3.30GHz. The measure of comparison is the count floating point operations per second (flops) which is estimated as $\frac{(2N-1)N^2}{t}$ where N is the matrix size and t the execution time.

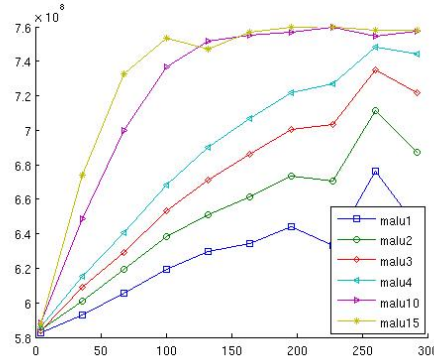
Figure 2 compares Stewart’s algorithm based on recursion (blue line) and boosting LU factorization (green line).



Next Figure compares Parker’s algorithm and plu algorithm. X axis represent the matrix size which pure Parker’s algorithm called. Matrix size was 2063 and CSV was [2061 2060 2057 2056 2053 2052 1003 1002 503 502 93 92]. FW2 model is used.



Next figure compares mixed algorithm for various values of NUM. X axis represent the CSV which takes values [4:32:300]. Matrix size was 2063. FW1 model is used.



Now it’s follow experiments to measure the accuracy of each procedure described in the previous sections. Table 1 presents relative residuals comparisons of linear systems solved using the mentioned algorithms: STW (stewart’s algorithm), SLU (stewart’s algorithm based on geboost factorization), PLU (efficient parker’s algorithm), PRK (parker’s algorithm), MLU4 (mixed accelerated LU using NUM=4)

All matrices are of size 512x512, either belonging to the Matlab function gallery or generated using Matlab function rand.

For all test matrices, we suppose that the exact solution is $x = [1 \ 1 \ \dots \ 1]$ and we set the right-hand side $b = Ax$.

Matrix	STW	SLU	PLU	PRK	MLU4
condex	2e-12	1e-12	1e-12	1e-12	2e-12
fiedler	5e-8	NAN	6e-7	4e-8	NAN
toeppe	5e-12	5e-12	5e-12	5e-12	5e-12
randcorr	3e-14	2e-14	4e-14	3e-14	3e-14
orthog	8e-14	3e+2	2e-9	1e-8	8e-9
prolate	4e-1	2e-13	9e-2	2e-8	1e-7
hadamard	8e-11	4e-11	1-11	1-11	4-11
rand	4e-8	2e-10	5e-8	3e-10	2e-10

9. Conclusion and Future Work. In this work we studied algorithms that solve dense linear systems using LU factorization without pivoting. First, we introduced the Boosting LU factorization method based on a Stewart's idea of a rank one modification. Next, we proposed an efficient algorithm based on Parker's algorithm that utilizes random transformation of the coefficient matrix to solve dense linear systems without pivoting. Then we developed a mixed algorithm for solving dense linear systems which compines Stewart's and Parker's algorithm. In the end we compared each algorithm under our experiment framework, and we observed that in some occasions our suggested algorithms is much better than that of literature.

As part of future work would be the implementation of mentioned algorithms in GPU, for example using the MAGMA framework [6] and straight comparison them with already implimented algorithms like Parker's algorithm [2]. Also a more theoretical work would be to find the optimal NUM for the mixed algorithm. From our research we know that NUM is associated clearly by problem's matrix, for example during Gauss elimination when bad pivots is appeared and when the size of sub-matrix for which will be applied Parker's algorithm is multiple of 2^d .

Source code for all the algorithms presented here is available at the project Web site [5]. Stewart's and Parker's

algorithms were evidently not publicly available from their respective authors, which are essentials to evaluate their performance relative to other competing algorithms. The current authors have strived to implement them as efficiently as they could, paying attention to every detail by following the description of the algorithms available in the published literature. By making the source code for these as well as the other implemented algorithms publicly available, the authors provide the scientific community with an infrastructure to facilitate further studies and comparisons on an objective basis without having to re-implement them.

References

- [1] Modifying Pivot Elements in Gaussian Elimination, G.W.Stewart
- [2] Accelerating linear system solutions using randomization techniques, M.Baboulin,J.Dongarra,J.Herrmann,S.Tomov
- [3] Random Butterfly Transformations with Applications in Computational Linear Algebra, D. Stott Parker
- [4] Introduction to Linear Algebra, Gilbert Strang
- [5] <http://students.ceid.upatras.gr/~papadakis/malu>
- [6] <http://icl.cs.utk.edu/magma/software/index.html>