Stability of the Bruhat Decomposition

by

Omar Hasan Odeh

B.Sc., University of Victoria, 1994

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

We accept this thesis as conforming

to the required standard

[redacted]

Dr. D. Dale Olesky, Co-supervisor (Dept. of Computer Science)

[redacted]

Dr. P. van den Driessche, Co-supervisor (Dept. of Mathematics and Statistics)

[redacted]

Dr. A. Sourour, Outside Member (Dept. of Mathematics and Statistics)

University of Victoria

Co-supervisors: Dr. D. Dale Olesky
and Dr. P. van den Driessche

Abstract

For any nonsingular $n \times n$ matrix A there exists a decomposition $A = V\Pi U$, where Π is a unique permutation matrix and V,U are upper triangular matrices. When $\Pi^T V\Pi$ is lower triangular and U has unit diagonal, such a decomposition is called the left Bruhat decomposition of A. The Bruhat decomposition, known from the theory of linear algebraic groups, was discussed by Kolotilina and Yeremin in the context of solving large sparse linear systems. We give an algorithm for computing the left Bruhat decomposition and consider its application to solving non-sparse linear systems.

The numerical stability of an algorithm for computing a matrix factorization depends on the growth of elements in the derived matrices. For example, for Gaussian elimination with partial pivoting (GEPP), the growth of elements in the derived matrices is measured using the growth factor $\gamma_{GP} = \max\limits_{i,j,k}\left|a_{jk}^{(i)}\right| / \max\limits_{j,k}\left|a_{jk}\right|$, where $A^{(i)} = \left[a_{jk}^{(i)}\right]$ is the ith derived matrix. GEPP is well-known to be unstable (γ_{GP} is exponential in n) for classes of matrices given by Wilkinson and recently (from a practical application) by Foster. We show that the left Bruhat decomposition gives at most linear growth in n for these classes, demonstrating its stability.

In order to obtain a practical, general-purpose algorithm for solving non-sparse linear systems, we specify a partial pivoting strategy for the Bruhat decomposition (BDPP). Simple relationships are determined between the derived matrices, permutation matrices and the matrices of multipliers associated with BDPP (applied to matrix A) and those of GEPP (applied to $A^T\rho$, where ρ is the permutation matrix that reverses the order of the columns). All real matrices that give the maximum exponential growth factor of 2^{n-1} with GEPP were characterized by Higham and Higham. We show that BDPP gives a growth factor of at most 2 for these matrices. Thus BDPP is stable for matrices that give the maximum exponential growth factor with GEPP, as well as for the matrix of Foster. For linear systems in which GEPP is unstable, application of BDPP is a practical alternative.

We introduce a bipartite graph model for the Bruhat decomposition of a pattern, and we develop an algorithm for computing the patterns of the factors in the left Bruhat decomposition. This algorithm is shown to model the corresponding numerical algorithm.

Examiners:

[redacted]

Dr. D. Dale Olesky, Co-supervisor (Dept. of Computer Science)

[redacted]

Dr. P. van den Driessche, Co-supervisor (Dept. of Mathematics and Statistics)

[redacted]

Dr. A. Sourour, Outside Member (Dept. of Mathematics and Statistics)

Table of Contents

List of Tables

Chapter 1

Introduction

Basic definitions and notation from matrix theory that are used in the following chapters are presented. Matrix factorization techniques are discussed, in particular the LU factorization and Gaussian elimination with partial pivoting (GEPP), LPR decomposition, and the Bruhat decomposition. Computational aspects of triangular factorization are presented, and stability results for LU and GEPP are discussed. A brief historical review of matrix factorization in the context of solving systems of linear equations is given. The chapter concludes with an overview of the thesis.

1.1 Basic facts and notation

Before introducing the Bruhat decomposition and discussing known results, we give some terminology and notation. We start with basic definitions from linear algebra and matrix theory, and throughout consider square $n \times n$ matrices with real entries. The matrix entry in the *i*th row and the *j*th column of matrix *A* is denoted by a_{ij}, and frequently we write $A = \left[a_{ij}\right]$. For a description of the usual matrix operations and functions (e.g., addition, multiplication, transpose, inverse, determinant), the reader can refer to [H&K] or [H&J].

A matrix *U* is *upper triangular* if $u_{ij} = 0$ whenever *i>j*, and similarly a matrix *L* is *lower triangular* if $l_{ij} = 0$ whenever *i<j*. A triangular matrix with all diagonal entries equal to 1 is called a *unit* triangular matrix and is referred to as a *normalized* triangular matrix. Note that the product of upper (lower) triangular matrices is an upper (lower) triangular matrix. Moreover, if all matrices in the product are normalized, the resulting matrix is normalized. A special case of a triangular matrix is a diagonal matrix. Matrix *D* is *diagonal* if $d_{ij} = 0$ whenever $i \neq j$. The identity matrix, denoted *I*, is the normalized diagonal matrix. Clearly the determinant of a triangular matrix is the product of its diagonal entries.

A *permutation matrix* is a matrix with entries from $\{0,1\}$ with exactly one 1 in every row and every column; note that its inverse is equal to its transpose. Associated with an $n \times n$ permutation matrix Π is a permutation function $\pi : \{1,2,\ldots,n\} \to \{1,2,\ldots,n\}$, defined as $\pi(i) = j$ if $\pi_{ij} = 1$. The set of *inversions* associated with *i*, denoted *inv*(*i*), is defined as $inv(i) = \{j : j > i \text{ and } \pi(i) > \pi(j)\}$, and the *length of* π (or Π) is defined as $l(\Pi) = l(\pi) = \sum_{i=1}^{n} |inv(i)|$, where $|inv(i)|$ denotes the cardinality of the set *inv*(*i*).

We denote by ρ the permutation matrix that reverses the order of the rows of A in the product ρA. Thus

$$\rho = \begin{bmatrix} 0 & & & 1 \\ & & \iddots & \\ & 1 & & \\ 1 & & & 0 \end{bmatrix}. \tag{1.1.1}$$

Clearly the permutation function associated with the $n \times n$ permutation matrix ρ has the maximum length among all permutation functions defined on the set $\{1,2,\ldots,n\}$, namely $l(\rho) = n(n-1)/2$. Note that for any lower triangular matrix L, $\rho L \rho$ is upper triangular, and for any upper triangular matrix U, $\rho U \rho$ is lower triangular.

We consider *nonsingular matrices*, which are characterized in the following well known result, see e.g. [Wat., Thm. 1.3.3].

Theorem 1.1.1. Let A be an $n \times n$ matrix. Then the following are equivalent.

1. A is nonsingular.
2. The determinant of A, denoted $\det(A)$, is nonzero.
3. The inverse of A, denoted A^{-1}, exists.
4. There is no nonzero vector x such that $Ax = 0$.
5. The rows of A are linearly independent.
6. The columns of A are linearly independent.
7. The rank of A is n (i.e., A has full rank).

Let $I, J \subseteq \{1,2,\dots,n\}$. We denote by $A[I,J]$ the submatrix of A consisting of the entries in A in rows indexed by I and columns indexed by J, and we write $A[I]$ whenever $I=J$. A *leading principal submatrix* of A is the submatrix $A[I]$, where $I = \{1,2,\cdots,k\}$ and $1 \le k \le n-1$. For any two $n \times n$ matrices A and B, we write $A<B$ if $a_{ij} < b_{ij}$ for all $1 \le i, j \le n$. The matrix $|A|$ has entries $|a_{ij}|$.

1.2 Triangular matrix factorizations and solution of linear systems

Often matrices and vectors are used to represent a system of linear equations as $Ax = b$, where A is the $n \times n$ coefficient matrix, $x \in \mathbf{R}^n$ is a vector of variables, and $b \in \mathbf{R}^n$ is the vector of constants. The system has a unique solution if and only if the coefficient matrix A is nonsingular. A well known method for solving this system is the reduction of matrix A to upper triangular form using a sequence of *elementary row operations*. These operations are of three types:

1. Interchange two rows
2. Multiply a row by a nonzero constant
3. Add a multiple of one row to another row .

Elementary column operations are defined similarly. If a type 3 operation intentionally introduces a 0 in a certain location, the operation is referred to as an *elimination* operation.

We consider three matrix factorizations that can be used to solve a system of linear equations. The most common is the LU decomposition, which can be computed by the Gaussian elimination algorithm. Although less common, the LPR decomposition and the Bruhat decomposition can also be used to solve linear systems. All three factorizations employ elementary column or row operations.

1.2.1 Gaussian elimination and its variants

The *Gaussian elimination* algorithm uses repeated applications of type 3 elementary row operations to reduce the coefficient matrix A (provided A satisfies the conditions of Theorem 1.2.1 below) to upper triangular form. This is called forward elimination. Once the matrix is reduced to upper triangular form, the solution of the linear system can be obtained by applying the back substitution algorithm [G&V, Algorithm 3.1.4].

Each of the n-1 steps of the forward elimination of Gaussian elimination can be accomplished by pre-multiplication by a unit lower triangular matrix $L^{(i)}$ [G&V, p. 96]. That is, $L^{(n-1)} \cdots L^{(2)} L^{(1)} A = U$, where U is an upper triangular matrix, and thus Gaussian elimination results in a factorization of the matrix A as $A = LU$, where L is a unit lower triangular matrix. This is known as the *LU decomposition* of A. The next theorem is known as the LU decomposition theorem. For a proof, see for example [G&V, Thm. 3.2.1].

Theorem 1.2.1. Let A be an $n \times n$ matrix whose leading principal submatrices are all nonsingular. Then A can be decomposed uniquely into a product $A = LU$, where L is a lower triangular matrix, U is an upper triangular matrix and either L or U is normalized.

Note that the converse of Theorem 1.2.1 is true only if A is nonsingular. That is, if A is a nonsingular matrix and has an LU decomposition, then the leading principal submatrices of *A* are nonsingular. Clearly not every nonsingular matrix has an LU decomposition. However, Theorem 1.2.2 [Dat., Cor. 5.2.1] states that some permutation of the rows of *A* results in a matrix that has an LU decomposition.

Theorem 1.2.2. Let *A* be an $n \times n$ nonsingular matrix. Then there exists a permutation matrix *P* such that *PA* has an LU decomposition.

Gaussian elimination with partial pivoting (GEPP) [G&V, Algorithm 3.4.3] is an algorithm that uses type 1 and type 3 elementary row operations to compute a permutation matrix *P* and the LU decomposition of *PA*, where *A* is nonsingular. GEPP is the most commonly used algorithm for solving systems of linear equations. In the next section we discuss the computational aspects of Gaussian elimination and GEPP that attribute to the popularity of GEPP in solving linear systems. Another variant of the Gaussian elimination algorithm is Gaussian elimination with complete pivoting. This algorithm uses type 1 elementary row and column operations as well as type 3 elementary row operations. Complete pivoting is rarely used since GEPP is more efficient and, for practical problems, almost always determines the solution as accurately as can be determined by Gaussian elimination with complete pivoting.

1.2.2 LPR decomposition

The *LPR decomposition* is a decomposition of matrix *A* into a product $A = LPR$, where *L* is a lower triangular matrix, *P* is a permutation matrix, and *R* is an upper triangular matrix.

Because of the similarity to the LU decomposition, the LPR decomposition is referred to in some literature, see e.g. [K&Y], as the generalized Gauss decomposition. Theorem 1.2.3 below shows that any nonsingular matrix has such a factorization. For a proof, see e.g. [Els., Thm. 1] .

Theorem 1.2.3. Let A be an $n \times n$ nonsingular matrix. Then A can be decomposed into a product $A = LPR$, where L is a unit lower triangular matrix, P is a unique permutation matrix, and R is an upper triangular matrix.

Note that no additional condition on the matrix A is needed in order for it to be decomposed as a product $A = LPR$. The reason for this is the presence of the permutation matrix in the product. An algorithm for computing a specific LPR decomposition is given in the proof of Theorem 1 in [Els.]. Note that in general an LPR decomposition is not unique for a given matrix A. For example, if $A = L\rho R$, where L is a unit lower triangular matrix and $R = \rho L \rho$, then $A = L\rho R = L\rho(\rho L \rho) = L^2 \rho I$. Hence A has two different LPR decompositions, but the permutation matrix remains unique (See Sections 2.1 and 2.2). Clearly the LU decomposition is a special case of the LPR decomposition with the permutation matrix being the identity matrix. Although the LPR decomposition is not practical for solving linear systems, it is still of interest from a theoretical point of view. Further discussion of the LPR decomposition is presented in Section 2.2.

1.2.3 Bruhat decomposition

The *Bruhat decomposition* of a nonsingular matrix A is a factorization of A as a product $A = V\Pi U$, where V and U are upper triangular matrices and Π is a unique permutation matrix (see e.g. [Gri., App.] and [K&Y]). The matrix Π is referred to as the *Bruhat permutation* of A. Theorem 1.2.4 states the existence of the Bruhat decomposition for any nonsingular matrix. For a proof, see e.g. [Gri., Prop. 11].

Theorem 1.2.4. Let A be an $n \times n$ nonsingular matrix. Then A can be decomposed into a product $A = V\Pi U$, where V is an upper triangular matrix, Π is a unique permutation matrix, and U is a unit upper triangular matrix.

Note that the factors U and V of the Bruhat decomposition are not generally unique, as shown in the following example.

Example 1.2.5.

Consider

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}.$$

Then A has the following two Bruhat decompositions:

$$\begin{aligned}\begin{bmatrix}1&2&3\\1&1&2\\1&1&1\end{bmatrix}&=\begin{bmatrix}1&1&1\\0&1&1\\0&0&1\end{bmatrix}\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}\begin{bmatrix}1&1&1\\0&1&1\\0&0&1\end{bmatrix}\\&=\begin{bmatrix}1&0&1\\0&1&1\\0&0&1\end{bmatrix}\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}\begin{bmatrix}1&1&1\\0&1&2\\0&0&1\end{bmatrix}.\end{aligned}$$

In Chapter 2 we state a condition on the factor V that gives a unique Bruhat decomposition and give an algorithm for computing the Bruhat decomposition of a nonsingular matrix A. The Bruhat decomposition can be used to solve a system of linear equations $Ax = b$ as follows:

1. Compute the Bruhat decomposition of A so that the system can be written as

   $$V\Pi Ux = b$$

2. Use back substitution to solve for y in

   $$Vy = b$$

3. Use back substitution to solve for x in

   $$Ux = \Pi^T y.$$

In the following chapters we focus on the Bruhat decomposition and develop a practical algorithm for solving a linear system with a nonsingular coefficient matrix.

We remark that if A is an $n\times n$ singular matrix, then there exists a decomposition $A = V\Pi U$, where V,U are upper triangular matrices and Π is a unique subpermutation matrix (i.e., each row and column of the matrix Π with entries from $\{0,1\}$ has at most one nonzero entry). The uniqueness of Π follows from a rank argument, see e.g. [G&G, p.

220] and Theorem 2.1.3. Both matrices U and V can be chosen to be nonsingular, as the following example illustrates.

Example 1.2.6.

Consider the singular matrix

$$A=\begin{bmatrix}1 & 1 & 2\\ 2 & -1 & 1\\ 1 & -1 & 0\end{bmatrix}=\begin{bmatrix}\otimes & 2 & 1\\ 0 & 1 & 2\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 1 & 0 & 0\end{bmatrix}\begin{bmatrix}1 & -1 & 0\\ 0 & 1 & 1\\ 0 & 0 & 1\end{bmatrix},$$

where $\otimes$ is any real number.

1.3 Computational aspects of triangular factorization

Triangular systems of linear equations are of particular interest since such systems can be solved inexpensively using back substitution or forward substitution. Although most systems are not triangular, a given system can be solved by factoring the coefficient matrix into a product of triangular matrices and possibly a permutation matrix as discussed in Section 1.2. The solution is then obtained by solving two triangular systems.

Numerical analysis is concerned with the development of numerical algorithms that are accurate, efficient, and easy to implement. The efficiency of a numerical algorithm depends on the choice of the platform of computation. The *flop* (floating point operation) count is frequently used to measure the efficiency of a numerical algorithm. We define a flop as one multiplicative ($/$ or $*$) operation [Hig., p. 3]. Note that in some of the literature, see e.g. [Wat.], a flop is defined as one multiplicative operation and one additive operation ($-$ or $+$).

There are several factors that should be considered when choosing and implementing a numerical algorithm to solve a linear system $Ax = b$. These include:

1. Hardware and software considerations
2. Stability of an algorithm
3. Properties of A, including its condition number.

The efficiency of a numerical algorithm depends on the architecture of the computing hardware used. Parallel computing has a great influence on the implementation of numerical algorithms. Some algorithms run faster when programmed on parallel computers, and thus measuring the efficiency of an algorithm using the flop count is no longer sufficient. However, not all numerical algorithms can be implemented in parallel. For further details on the effect of computer architecture on numerical computation, see e.g. [Wat., Section 1.10] and [G&V, Ch. 6].

The choice of data structures for object representation is crucial in implementing a numerical algorithm. For example, in sparse matrix factorization 30-40% of the total time is spent on index processing , see e.g. [K&Y] and [G&L]. Thus it is important to develop data structures with fast access time and to use storage efficiently.

Another aspect of numerical computation is the notion of *stability.* Algorithm $\mathcal{A}$ for solving $Ax = b$ is stable if the computed solution $\hat{x}$ satisfies $(A+E)\hat{x} = b + \delta b$, where E, the perturbation of A, and δb, the perturbation of b, are small. The size of a perturbation is determined by the relative error using norms, e.g. $\|E\|/\|A\|$ and $\|\delta b\|/\|b\|$, or by the magnitude of the entries of the matrix E and the vector δb. The perturbations in A and b are usually obtained from a *backward error analysis* of $\mathcal{A}$, see e.g. [Hig.] or [Wil.63]. Suppose Algorithm $\mathcal{A}$ uses elementary (row or column) operations to reduce matrix A to upper triangular form in n-1 steps. Denote by $A^{(k)} = \left[a_{ij}^{(k)}\right]$ the derived matrix after k elimination steps of Algorithm $\mathcal{A}$. The *unit roundoff error*

$$\mu = \begin{cases} \frac{1}{2}\beta^{1-t} & \text{for rounded arithmetic} \\ \beta^{1-t} & \text{for chopped arithmetic} \end{cases}$$

is a function of β, the base of the floating point number system in use, and t, the number of digits used to represent numbers in the system, see e.g. [G&V, p. 62].

Wilkinson [Wil.63] showed, using backward error analysis, that if Gaussian elimination (without pivoting) is applied to solve a system $Ax = b$, then the computed solution $\hat{x}$ satisfies $(A+E)\hat{x} = b$ with $|E| \le n\mu\left(3|A| + 5|\hat{L}||\hat{U}|\right) + O(\mu^2)$, where $\hat{L}\hat{U}$ is the computed LU factorization of A using Gaussian elimination. For a derivation of this result, see e.g. [G&V, Thm. 3.3.2]. The bound can be written using a matrix norm as $\|E\|_\infty \le n\mu\left(3\|A\|_\infty + 5\|\hat{L}\|_\infty\|\hat{U}\|_\infty\right) + O(\mu^2)$, where $\|A\|_\infty = \max_{1\le i\le n}\sum_{j=1}^{n}|a_{ij}|$ is the $\infty-norm$ of A. Clearly $\|\hat{L}\|_\infty$ or $\|\hat{U}\|_\infty$ can be arbitrarily large, as can be shown by

$$A = \begin{bmatrix} \varepsilon & 1 \\ 1 & 0 \end{bmatrix},$$

with $\|\hat{U}\|_\infty = \max\{|1/\varepsilon|, 1+|\varepsilon|\}$ and $\|\hat{L}\|_\infty = 1 + |1/\varepsilon|$. By letting ε be arbitrarily small, $\|\hat{L}\|_\infty$ and $\|\hat{U}\|_\infty$ become arbitrarily large. Thus Gaussian elimination without pivoting may be unstable.

Wilkinson showed that the above bound on $\|E\|_\infty$ for Gaussian elimination also holds for GEPP . However, GEPP guarantees that the multipliers used in the elimination process have magnitudes less than or equal to 1, and this restricts the magnitudes of entries of the derived matrices .Thus $\|\hat{L}\|_\infty \le n$ and it can be shown (see, e.g. [G&V]) that $\|E_\infty\| \le c\mu n^3 \gamma_{GP} \|A\|_\infty + O(\mu^2)$, where c is a constant and γ_{GP} is the *growth factor* for Gaussian elimination with partial pivoting, defined as $\gamma_{GP} = \max_{i,j,k} |a_{jk}^{(i)}| / \max_{j,k} |a_{jk}|$. Thus the stability of GEPP is essentially determined by the magnitude of γ_{GP}. Wilkinson introduced an $n \times n$ matrix that achieves the largest possible growth factor of 2^{n-1} when GEPP is applied. All real matrices with $\gamma_{GP} = 2^{n-1}$ were characterized by Higham and Higham [H&H]. Wilkinson also noted that matrices with large γ_{GP} do not seem to arise in practical applications. However, recently Foster [Fos.] and Wright [Wri.] discussed classes of matrices arising from practical problems that have exponential growth factors with GEPP. Since matrices with exponential growth factor rarely arise in practical applications, GEPP is generally the algorithm of choice for solving a linear system. A tight upper bound for the growth of elements in Gaussian elimination with complete pivoting is still unknown; the best known bound, proposed by Wilkinson [Wil.63, p. 97], is $n^{1/2}(2^1 3^{1/2} 4^{1/3} \cdots n^{1/(n-1)})$. Although this bound grows much slower than 2^{n-1}, the extra cost associated with the complete pivoting strategy makes GEPP a more efficient algorithm and Gaussian elimination with complete pivoting is seldom used in practice.

The stability results for Bruhat decomposition and Bruhat decomposition with partial pivoting (BDPP) are discussed in Chapter 3.

The attributes of A affect the choice of an algorithm. For example, if A is positive definite, the algorithm of choice is the Cholesky factorization algorithm since it is always stable and faster than Gaussian elimination. Other relevant attributes of A include sparsity, diagonal dominance, symmetry, etc. For a description of the effect of such attributes on the choice of algorithm, see e.g. [G&V] or [Wat.].

One of the most significant attributes of A with regard to solving a linear system is its condition number. Problem $\mathcal{P}$ is said to be *ill-conditioned* if the exact solution to $\mathcal{P}$ changes greatly with small changes in the data defining $\mathcal{P}$, otherwise, $\mathcal{P}$ is said to be *well-conditioned.* Associated with a nonsingular matrix A is a quantity called the *condition number* of A, denoted by $\kappa(A)$ and defined by

$$\kappa(A) = \|A\| \left\|A^{-1}\right\|.$$

In general, if the condition number of A is very large, the problem of solving $Ax = b$ is ill-conditioned, and we say that A is ill-conditioned. It is clear from the definition that the condition is a measure of how sensitive a problem is to small perturbations. The condition of a problem is very relevant in interpreting the results of a numerical computation. Finite precision arithmetic, unlike real arithmetic, uses a finite number of digits to store the data, and thus the data may be perturbed at the storage or input phase.

1.4 Historical Remarks

The earliest mention of solutions of simultaneous linear equations (systems of linear equations) is attributed to the ancient Greeks as early as 275 BC [Smi., p. 432]. The subject was greatly extended by the ancient Chinese between 206 BC - 220 AD. The Chinese used elementary column operations for solving small systems of linear equations. The largest documented problem solved by the Chinese involves six variables. The method they used is identical to the Gaussian elimination algorithm with the exception that the Chinese used only integer and fractional arithmetic and the operations were performed on columns not on rows, see e.g. [Smi.] and [Y&S]. The same method was also used in Japan for solving linear systems of equations.

C. F. Gauss (1777-1855) was the first to arrive at an algorithm for solving a general system of n linear equations in n unknowns. He developed the algorithm, known today as the Gaussian elimination algorithm, in 1795 in his work on the least squares problem; see e.g. [Gol., pp. 212-224] for details of the discovery. Gauss was also the first to give conditions for the existence of a unique solution to the problem.

With the development of computers in the twentieth century, more work was done on numerical methods for solving systems of linear equations. The first work on the stability of Gaussian elimination is attributed to Goldstine and von Neumann in two papers published in 1947 and 1951 ([G&vN.47] and [G&vN.51]). They used a forward error analysis to obtain their results. Backward error analysis was developed by Wilkinson in the 1950's and 1960's and was used to obtain the perturbation results in Section 1.3 for Gaussian elimination and GEPP. It is documented that backward error analysis was developed by Wilkinson, however the idea had appeared in the papers by Goldstine and von Neumann and also in a paper by Turing in 1948 [Hig.].

Bruhat decomposition is known to have its origin from the theory of linear algebraic groups. The decomposition was introduced by Bruhat in 1954 in work on representation theory [Suz.]. The matrix form of the Bruhat decomposition was considered for sparse matrix computation by Kolotilina and Yeremin in 1987 [K&Y]. In fact, they used the Bruhat decomposition for solving large sparse systems of linear equations and showed that the Bruhat decomposition can give better sparsity results than GEPP. In 1982, Grigor'ev [Gri.] used the matrix version of the Bruhat decomposition as a theoretical algebraic tool, whereas Kolotilina and Yeremin used the decomposition as a practical alternative to GEPP for numerical matrix computation. In the following chapters, we deal with the decomposition solely from a matrix analysis point of view.

1.5 Thesis overview

Chapter 2 describes the Bruhat decomposition for nonsingular matrices. We prove the uniqueness of the Bruhat permutation and of the triangular factors in the left Bruhat decomposition (Section 2.1). We also present a relationship between the left Bruhat decomposition and the LU factorization, and discuss the relationship between the Bruhat decomposition and the LPR decomposition (Section 2.2). We give and illustrate an algorithm for computing the left Bruhat decomposition (Algorithm 2.3.1), and complete the proof of the existence and uniqueness theorem (Theorem 2.1.4) for the left Bruhat decomposition. We conclude the chapter by deriving the left Bruhat decomposition for a rank one perturbation of a diagonal matrix, which gives alternative proofs for the inverse and determinant formulas for a rank one perturbation of a diagonal matrix.

Chapter 3 discusses the stability of the Bruhat decomposition and compares stability results to those for LU and GEPP. We use the relationship between the left Bruhat decomposition and the LU factorization from Chapter 2 to obtain stability results similar to those for the LU decomposition presented in Section 1.3. We discuss the stability of the left Bruhat decomposition for classes of matrices that give exponential growth factor for GEPP. In particular, we discuss two classes of matrices; a class introduced by Wilkinson [Wil.65] that gives the maximum exponential growth factor for

GEPP, and a class of matrices given by Foster [Fos.] that arises in a practical application. We show that the Bruhat decomposition is stable for these classes of matrices. We present a pivoting strategy for Bruhat decomposition (BDPP) that ensures that the magnitudes of multipliers is less than or equal to 1. We also obtain a relationship between the factors of BDPP and GEPP, which is used to derive stability results for BDPP similar to those for GEPP presented in Section 1.3. Finally, we show that BDPP is stable for any real $n \times n$ matrix that gives the maximum possible growth factor for GEPP, namely 2^{n-1}, this class of matrices was characterized by Higham and Higham [H&H]. We also compare growth factor results for GEPP, BDPP, and the Bruhat decomposition and give examples that illustrate other aspects of the growth factor for BDPP.

Chapter 4 deals with a sparsity analysis for the Bruhat decomposition of a pattern when accidental numerical cancellation is ignored. Firstly, we give the necessary definitions and notation from graph theory and combinatorial matrix theory that are needed for the sparsity analysis. We develop an algorithm for computing the patterns of the factors in the left Bruhat decomposition (Algorithm 4.2.1), and show that this algorithm is consistent with the numerical algorithm given in Chapter 2.

Chapter 2

Description of the Bruhat decomposition

In this chapter, a detailed description of the Bruhat decomposition of a nonsingular matrix is given. Uniqueness results for the Bruhat decomposition are proved, and relationships between the Bruhat decomposition and the LU (Corollary 2.1.5) and LPR factorizations (Section 2.2) are established. Knowing the Bruhat decomposition of a matrix A, either the LU or an LPR factorization of a permutation of A can be derived, and vice versa. An algorithm for computing the left Bruhat decomposition of a matrix is developed and illustrated. This algorithm is the analog of an algorithm given in [K&Y, Section 2] for the right Bruhat decomposition.

2.1 Uniqueness results

The focus of this section is on the Bruhat decomposition as defined in Section 1.2.3. Some properties of the Bruhat decomposition are investigated and some uniqueness results are presented. A relationship between the Bruhat decomposition and the LU decomposition is also derived.

We start by proving the uniqueness of the Bruhat permutation. A different uniqueness proof that uses algebraic techniques can be found in [Gri., App.]. Our proof, however, uses properties of permutation matrices. We need the following lemma about permutation matrices, see e.g. [G&G, p. 221].

Lemma 2.1.1. Let P and Q be $n \times n$ permutation matrices. Then $P = Q$ if and only if $\operatorname{rank} P[S_i, S_j] = \operatorname{rank} Q[S_i, S_j]$ for all $1 \leq i \leq n-1$ and $1 \leq j \leq n$, where $S_k = \{1, 2, \cdots, k\}$.

Proof

If $P = Q$, then clearly $\operatorname{rank} P[S_i, S_j] = \operatorname{rank} Q[S_i, S_j]$ for $1 \leq i \leq n-1$ and $1 \leq j \leq n$. Conversely, suppose that $\operatorname{rank} P[S_i, S_j] = \operatorname{rank} Q[S_i, S_j]$ for all $1 \leq i \leq n-1$ and $1 \leq j \leq n$. Clearly $P[S_1, S_n] = Q[S_1, S_n]$ and $P[S_n, S_1] = Q[S_n, S_1]$. Suppose that $P \neq Q$. Then let $r \geq 2$ be the least integer such that $P[S_r, S_n] \neq Q[S_r, S_n]$ and let $t \geq 2$ be the least integer such that $p_{rt} \neq q_{rt}$. Without loss of generality suppose that $p_{rt} = 1$ (thus

$q_{rt} = 0$). Then $p_{rj} = q_{rj} = 0$ for $1 \le j \le t-1$ and $p_{it} = q_{it} = 0$ for $1 \le i \le r-1$, implying that rank $P[S_r, S_t] =$ rank $Q[S_r, S_t] + 1$. This contradiction implies that $P = Q$. ■

An immediate consequence is the following, which can be proved by applying Lemma 2.1.1 to the permutation matrices ρP and ρQ.

Corollary 2.1.2. Let P and Q be $n \times n$ permutation matrices. Then $P = Q$ if and only if rank $P[T_{n-i}, S_j] =$ rank $Q[T_{n-i}, S_j]$ for all $1 \le i \le n-1$ and $1 \le j \le n$, where $T_{n-i} = \{i+1, i+2, \cdots, n\}$ and $S_j = \{1, 2, \cdots, j\}$.

Note that Lemma 2.1.1 and Corollary 2.1.2 imply that the rank equalities hold for all $1 \le i \le n$. This follows from the fact that both P and Q are permutation matrices, thus they have full rank.

Theorem 2.1.3. (Uniqueness of the Bruhat permutation). Let A be an $n \times n$ nonsingular matrix. Let $A = V\Pi U$ be a Bruhat decomposition of A. Then the Bruhat permutation matrix Π is unique.

Proof

Partition the matrices A, V, Π, and U conformally so that

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} \\ 0 & V_{22} \end{bmatrix} \begin{bmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{21} & \Pi_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix},$$

where A_{11} is an $r \times j$ submatrix, V_{11} is an $r \times r$ submatrix, Π_{11} is an $r \times j$ submatrix, and U_{11} is a $j \times j$ submatrix, with $1 \le r \le n-1$ and $1 \le j \le n$. Thus $A_{21} = V_{22}\Pi_{21}U_{11}$. Since A is nonsingular, both of V_{22} and U_{11} must be nonsingular, and thus rank A_{21} = rank Π_{21}. Suppose that $A = V_1 \Pi U_1 = V_2 P U_2$ are two different Bruhat decompositions of A. Using the same partitioning as above for both decompositions, we must have rank A_{21} = rank Π_{21} = rank P_{21}, where A_{21} is an $(n-r)\times j$ submatrix of A. By applying Corollary 2.1.2, $\Pi = P$. ■

Theorem 2.1.3 shows that if matrices A and B have the same Bruhat permutation, then rank $A[T_{n-i}, S_j]$ = rank $B[T_{n-i}, S_j]$, where T_{n-i} and S_j are as defined in Corollary 2.1.2.

A Bruhat decomposition $A = V\Pi U$ is called a *right Bruhat decomposition* if the matrix $\Pi U \Pi^T$ is lower triangular. Similarly a decomposition is called a *left Bruhat decomposition* if the matrix $\Pi^T V \Pi$ is lower triangular. If $\Pi = \rho$ as defined in (1.1.1), then a Bruhat decomposition is both a left and a right Bruhat decomposition. Note that the definition of a left or a right Bruhat decomposition is a pattern property (see Lemma 2.2.3). Recall that Theorem 1.2.4 (the Bruhat decomposition theorem) states the uniqueness of only the Bruhat permutation. The next theorem, however, guarantees the uniqueness of the decomposition, provided it is a left or a right Bruhat decomposition.

Theorem 2.1.4. Let A be an $n \times n$ nonsingular matrix. Then A can be decomposed into a product $A = V\Pi U$, where V is an upper triangular matrix, Π is a unique permutation matrix, and U is a unit upper triangular matrix. If, in addition, the decomposition is a right or a left Bruhat decomposition, then the decomposition is unique.

Proof

We leave the existence proof for Section 2.3 where we introduce an algorithm for computing the Bruhat decomposition. For the uniqueness, suppose $A = V\Pi U$ is a left Bruhat decomposition. Then $\Pi^T A = \Pi^T V\Pi U = LU$, where $L = \Pi^T V\Pi$ is a lower triangular matrix and U is a unit upper triangular matrix. From the uniqueness of the LU decomposition (Theorem 1.2.1) of $\Pi^T A$ and the uniqueness of the Bruhat permutation, we have uniqueness for the left Bruhat decomposition. Similarly the right Bruhat decomposition is unique, see also Theorem 2.1.6. ■

The above proof gives the following relationship between the left Bruhat decomposition and the LU decomposition.

Corollary 2.1.5. Let A be an $n \times n$ nonsingular matrix and $A = V\Pi U$ be its left Bruhat decomposition. Then $\Pi^T A = (\Pi^T V\Pi)U$ is the LU decomposition of $\Pi^T A$ with U normalized.

Note that, unlike the LU decomposition, the Bruhat decomposition needs to be either a left or a right Bruhat decomposition in order to have uniqueness. The two decompositions in Example 1.2.5 satisfy the normalization condition, but the second decomposition is the (unique) left Bruhat decomposition of the matrix A.

The left and right Bruhat decompositions are related in the sense that the left Bruhat decomposition of a nonsingular matrix A can be obtained from the right Bruhat decomposition of $\rho A^T \rho$. The next theorem shows this relationship.

Theorem 2.1.6. Suppose A is an $n \times n$ nonsingular matrix. Let $\rho A^T \rho = V\Pi U$ be the left Bruhat decomposition of $\rho A^T \rho$. Then the right Bruhat decomposition of A is $A = \widetilde{V}\widetilde{\Pi}\widetilde{U}$, where $\widetilde{V} = \rho U^T \rho$, $\widetilde{\Pi} = \rho\Pi^T \rho$, and $\widetilde{U} = \rho V^T \rho$.

Proof

Consider the left Bruhat decomposition of $\rho A^T \rho$, namely $\rho A^T \rho = V\Pi U$. Thus $A = \rho U^T(\rho\rho)\Pi^T(\rho\rho)V^T\rho$, where $\rho\rho = I$. Clearly both of $\rho U^T \rho$ and $\rho V^T \rho$ are upper triangular matrices, and $\rho U^T \rho$ is normalized. The matrix $\rho\Pi^T \rho$ is a permutation matrix. Thus $A = \widetilde{V}\widetilde{\Pi}\widetilde{U}$, where $\widetilde{V} = \rho U^T \rho$, $\widetilde{\Pi} = \rho\Pi^T \rho$ and $\widetilde{U} = \rho V^T \rho$, is a Bruhat decomposition of A. To show this is the right Bruhat decomposition, consider the matrix $\widetilde{\Pi}\widetilde{U}\widetilde{\Pi}^T = \rho\Pi^T\rho\rho V^T\rho\rho\Pi\rho = \rho\Pi^T V^T \Pi\rho = \rho L^T \rho$, where $L^T = \Pi^T V^T \Pi$ is upper triangular. Thus $\widetilde{\Pi}\widetilde{U}\widetilde{\Pi}^T$ is lower triangular and the decomposition is the right Bruhat decomposition. ■

Matrix A is *persymmetric* if $A = \rho A^T \rho$. Suppose $A = V\Pi U$ is the left Bruhat decomposition of a persymmetric matrix A. Then by Theorem 2.1.6 and the uniqueness of the Bruhat permutation, $\Pi = \rho\Pi^T \rho$ and the decomposition is both the left and the right Bruhat decomposition.

We remark that the Bruhat decomposition of A is sometimes written as $A = V\Pi DU$, where both V and U are unit upper triangular matrices, Π is the Bruhat permutation of A, and D is a diagonal matrix, see e.g. [K&Y]. The following result shows that the diagonal matrix D is unique.

Theorem 2.1.7. Let A be an $n \times n$ nonsingular matrix, and let $A = V\Pi DU$ be a Bruhat decomposition of A, where V and U are unit upper triangular matrices, Π is the Bruhat permutation of A and D is a diagonal matrix. Then the matrix D is unique.

Proof

Let $A = V_0\Pi D_0U_0 = V_1\Pi D_1U_1$ be two different Bruhat decompositions of A. Thus $\Pi^T(V_1^{-1}V_0)\Pi D_0 = D_1(U_1U_0^{-1})$, and since $U_1U_0^{-1}$ is a unit upper triangular matrix, the matrix $\Pi^T(V_1^{-1}V_0)\Pi$ must be an upper triangular matrix. Moreover, since $V_1^{-1}V_0$ is a unit upper triangular matrix and is similar to $\Pi^T(V_1^{-1}V_0)\Pi$, the latter must also be a unit upper triangular matrix. The diagonal entries in the matrix $D_1(U_1U_0^{-1})$ are precisely those of the matrix D_1 in the same order. Similarly, the diagonal entries in the matrix $\Pi^T(V_1^{-1}V_0)\Pi D_0$ are precisely those of the matrix D_0 in the same order, and thus $D_0 = D_1$. ■

2.2 Relation to the LPR decomposition

We now investigate the relationship between the Bruhat decomposition and the LPR decomposition. Suppose $A = LPR$ is an LPR decomposition of an $n \times n$ nonsingular matrix A. Then $\rho A = (\rho L \rho)(\rho P)R$ is a Bruhat decomposition of ρA. Note that the Bruhat permutation of ρA is ρP, where P is the permutation matrix from an LPR decomposition of A. Thus if $P = \rho$, then the Bruhat permutation of ρA is I, which indicates that the matrix ρA is an upper triangular matrix. Finally, the Bruhat decomposition of ρA as derived above is a left Bruhat decomposition if $P^T LP$ is a lower triangular matrix. Similarly, the decomposition of ρA is a right Bruhat decomposition if PRP^T is an upper triangular matrix. The following lemma is used to obtain conditions on an LPR decomposition of A in order for the Bruhat decomposition of ρA, as derived above, to be the left or right decomposition.

Lemma 2.2.1. Let P be an $n \times n$ permutation matrix with $p_{i,k_i} = 1$ for $i = 1,2,\ldots,n$, and let L be an $n \times n$ nonsingular lower triangular matrix. Then there exists a nonsingular lower triangular matrix $\widetilde{L}$ such that $P^T LP = \widetilde{L}$ if and only if $l_{ij} = 0$ whenever $k_i < k_j$.

Proof

If $P^T LP = \widetilde{L}$, then $l_{ij} = \widetilde{l}_{k_i,k_j} = 0$ whenever $k_i < k_j$. For the converse, define $\tilde{L} = \left[\tilde{l}_{pq}\right]$, where $\widetilde{l}_{pq} = l_{\phi(p),\phi(q)}$ and ϕ is defined by $\phi(k_i) = i$, so that $p_{\phi(k_i)k_i} = 1$. Thus $\widetilde{L}$ is a nonsingular lower triangular matrix. Then $(LP)_{iq} = l_{i,\phi(q)} = l_{\phi(k_i),\phi(q)} = \widetilde{l}_{k_i,q} = (P\widetilde{L})_{iq}$, and thus $LP = P\widetilde{L}$ or equivalently $P^T LP = \widetilde{L}$. ■

With this lemma and the result of applying it to the nonsingular upper triangular matrix L^T, we obtain the following.

Theorem 2.2.2. Suppose $A = LPR$ is an LPR decomposition of the nonsingular $n \times n$ matrix A with $p_{i,k_i} = 1$ for $i = 1,2,\ldots,n$. Then $\rho A = (\rho L \rho)(\rho P)R$ is a Bruhat decomposition of ρA. Moreover, it is the left Bruhat decomposition if and only if $l_{ij} = 0$ whenever $k_i < k_j$, and the right Bruhat decomposition if and only if $r_{k_i,k_j} = 0$ whenever $j<i$.

Thus, the relationship between the Bruhat decomposition and the LPR decomposition is extended. Theorem 2.2.2 gives conditions on an LPR decomposition of A in order for the derived Bruhat decomposition of ρA to be the left or the right Bruhat decomposition.

The following lemma gives a relationship between two LPR decompositions of an $n \times n$ nonsingular matrix A.

Lemma 2.2.3. Let P be an $n \times n$ permutation matrix with $p_{i,k_i} = 1$ for $i = 1,2,\ldots,n$, and let L be an $n \times n$ nonsingular lower triangular matrix. Then there exists a nonsingular upper triangular matrix U such that $LP = PU$ if and only if $l_{ij} = 0$ whenever $k_i > k_j$.

Proof

Let $LP = PU$. Thus $L = PUP^T$ and

$$\begin{aligned} l_{ij} &= \sum_{t=1}^{n}\sum_{s=1}^{n} p_{is}u_{st}p_{jt} \\ &= u_{k_i,k_j}. \end{aligned}$$

Thus $l_{ij} = 0$ whenever $k_i > k_j$ since U is an upper triangular matrix. For the converse, define $U = \left[u_{pq}\right]$, where $u_{pq} = l_{\phi(p),\phi(q)}$ and ϕ is as defined in the proof of Lemma 2.2.1. Thus U is a nonsingular upper triangular matrix. Then $\left(LP\right)_{iq} = l_{i,\phi(q)} = l_{\phi(k_i),\phi(q)} = u_{k_i,q} = \left(PU\right)_{iq}$. Thus $LP = PU$. ■

Recall that an LPR decomposition of a nonsingular matrix A is not in general unique. However, Lemma 2.2.3 gives more insight. Suppose $A = L_0PR_0 = L_1PR_1$ are two different LPR decompositions of A. Then $P(R_0R_1^{-1}) = (L_0^{-1}L_1)P$, and from Lemma 2.2.3, $\left(R_0R_1^{-1}\right)_{ij} = \left(L_0^{-1}L_1\right)_{\phi(i),\phi(j)}$ and $\left(L_0^{-1}L_1\right)_{ij} = 0$ whenever $k_i > k_j$.

2.3 An algorithm for computing the Bruhat decomposition

The left Bruhat decomposition can be computed by post-multiplication of A by n-1 nonsingular matrices $U^{(i)}$, whose entries are chosen so as to introduce zeros into the matrix product. Let $A^{(0)} = A$ and $A^{(i)} = A^{(i-1)}U^{(i)}$, $1 \le i \le n-1$, so that $A^{(i)} = AU^{(1)}U^{(2)}\cdots U^{(i)}$. If t_1 is the maximum row index such that $a_{t_1,1} \ne 0$(i.e., $a_{t_1,1}$ is the pivot entry in the first column), then

$$U^{(1)} = \begin{bmatrix} 1 & -\frac{a_{t_1,2}}{a_{t_1,1}} & -\frac{a_{t_1,3}}{a_{t_1,1}} & \cdots & -\frac{a_{t_1,n}}{a_{t_1,1}} \\ & 1 & & & \\ & & \ddots & \mathbf{0} & \\ & \mathbf{0} & & 1 & \\ & & & & 1 \end{bmatrix}$$

and

$$A^{(1)} = \begin{bmatrix} a_{11} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ \vdots & & & \vdots \\ a_{t_1-1,1} & a_{t_1-1,2}^{(1)} & \cdots & a_{t_1-1,n}^{(1)} \\ a_{t_1,1} & 0 & \cdots & 0 \\ 0 & a_{t_1+1,2} & \cdots & a_{t_1+1,n} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2} & \cdots & a_{nn} \end{bmatrix} = [a_{jk}^{(1)}].$$

Note that $U^{(i)}$ can be written compactly as $U^{(i)} = I - e^{(i)}\left(m^{(i)}\right)^T$, where

$$m_j^{(i)} = \begin{cases} \dfrac{a_{t_i,j}^{(i-1)}}{a_{t_i,i}^{(i-1)}} & \text{, for } i+1 \le j \le n \\ 0 & \text{, otherwise,} \end{cases}$$

$$e_j^{(i)} = \begin{cases} 1 & , i = j \\ 0 & \text{, otherwise,} \end{cases}$$

and t_i is the maximum row index such that $a_{t_i,i}^{(i-1)} \neq 0$. For the second elimination step, if t_2 is the maximum row index such that $a_{t_2,2}^{(1)} \neq 0$, then $U^{(2)} = I - e^{(2)}\left(m^{(2)}\right)^T$. Note that t_2 may be less than or greater than t_1, either may be equal to *n*, but $t_2 \neq t_1$. If $t_1 < t_2$, then

$$A^{(2)} = \begin{bmatrix} a_{11} & a_{12}^{(1)} & a_{13}^{(2)} & \cdots & \cdots & a_{1n}^{(2)} \\ \vdots & \vdots & \vdots & & & \vdots \\ a_{t_1-1,1} & a_{t_1-1,2}^{(1)} & a_{t_1-1,3}^{(2)} & \cdots & \cdots & a_{t_1-1,n}^{(2)} \\ a_{t_1,1} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & a_{t_1+1,2} & a_{t_1+1,3}^{(2)} & \cdots & \cdots & a_{t_1+1,n}^{(2)} \\ \vdots & \vdots & \vdots & & & \vdots \\ 0 & a_{t_2-1,2} & a_{t_2-1,3}^{(2)} & \cdots & \cdots & a_{t_2-1,n}^{(2)} \\ 0 & a_{t_2,2} & 0 & \cdots & \cdots & 0 \\ 0 & 0 & a_{t_2+1,3} & \cdots & \cdots & a_{t_2+1,n} \\ \vdots & \vdots & \vdots & & & \vdots \\ 0 & 0 & a_{n3} & \cdots & \cdots & a_{nn} \end{bmatrix} = \left[a_{jk}^{(2)}\right].$$

After *n*-1 elimination steps, $A^{(n-1)} = AU^{(1)}U^{(2)}\cdots U^{(n-1)}$. Let $a_{t_n,n}^{(n-1)}$ denote the sole nonzero entry in column *n* of $A^{(n-1)}$, and $\Pi = \left[\pi_{jk}\right]$be the permutation matrix with $\pi_{t_k,k} = 1$, for $1 \le k \le n$. Then

$$V\Pi = A^{(n-1)}, \tag{2.3.1}$$

and letting $U^{-1} = U^{(1)}U^{(2)}\cdots U^{(n-1)}$ gives $A = V\Pi U$.

The following algorithm determines the factors of this decomposition.

Algorithm 2.3.1: Left Bruhat Decomposition

Input: Nonsingular $n \times n$ matrix A

Output: The matrices V, Π, and U, where the left Bruhat decomposition is $A = V\Pi U$

Initialization: $U = I$

```
for $i = 1$ to $n$
    $j = \max\{p \mid a_{pi} \neq 0\}$
    $\pi_{ji} = 1, \pi_{li} = 0$ for $l \neq j$
    $v_{tj} = a_{ti}$ for $1 \le t \le n$
    for $k = i+1$ to $n$
        $m = \frac{a_{jk}}{a_{ji}}$
        $u_{ik} = m$
        for $l = 1$ to $j-1$
            $a_{lk} = a_{lk} - m a_{li}$
        $a_{jk} = 0$
```

Theorem 2.3.2. Let A be an $n \times n$ nonsingular matrix. Then Algorithm 2.3.1 determines the left Bruhat decomposition of A.

Proof

We show that the matrices V and U are upper triangular, and then show that $\Pi^T V \Pi$ is lower triangular. Let π be the permutation function associated with the permutation matrix Π. From (2.3.1), $V = A^{(n-1)}\Pi^T$, and thus for any fixed q, $v_{iq} = a^{(n-1)}_{i,\pi(q)}$. If $q = \max\left\{p : a^{(n-1)}_{p,\pi(q)} \neq 0\right\}$, then $a^{(n-1)}_{i,\pi(q)} = 0$ for $i>q$, hence V is upper triangular. Also by (2.3.1), $(\Pi^T V \Pi)_{rj} = (\Pi^T A^{(n-1)})_{rj} = a^{(n-1)}_{\pi^{-1}(r),j}$. But $a^{(n-1)}_{\pi^{-1}(r),r} \neq 0$ and $a^{(n-1)}_{\pi^{-1}(r),j} = 0$ for $j > r$ as these entries are eliminated in the rth step of the algorithm. Hence $\Pi^T V \Pi$ is lower triangular and the algorithm gives the left Bruhat decomposition of A. ■

Thus Theorem 2.3.2 combined with the uniqueness results from Section 2.1 completes the proof of Theorem 2.1.4.

In Algorithm 2.3.1, the matrices U and V are explicitly stored. The algorithm can be modified so that the entries of both matrices overwrite the entries of A. Then, the only additional storage needed is for an n-dimensional vector that stores the Bruhat permutation, and which is used to access entries in V and U. This is the same technique used for some implementations of GEPP, where rows of matrix A are not physically interchanged, see e.g. [Dat., Alg. 5.2.2].

The time complexity of Algorithm 2.3.1 as shown by the flop count as a function of n is $\sum_{i=2}^{n}(n-i)\pi^{-1}(i)$, where π is the permutation function associated with Π. The

l-loop requires $\pi^{-1}(i)$ flops, and is executed n-i times inside the k-loop. The worst case complexity is $\frac{n^3}{3}+O(n^2)$ when $\Pi=\rho$ (giving $l(\Pi)$ its maximum value), A is full, and accidental cancellation is ignored, in which case the algorithm has to eliminate $n(n-1)/2$ entries. On the other hand, if Π is obtained from the identity by interchanging any two rows (i.e., $\pi^{-1}(i)=i$ for all but two values of i), then the complexity is $\frac{n^3}{6}+O(n^2)$. This shows the dependence of execution time on the Bruhat permutation.

In general, computation of the left Bruhat decomposition of A requires $\frac{n^3}{3}+O(n^2)$ flops, which is the same as that for Gaussian elimination. Algorithm 2.3.1 also requires the extra cost of $O(n^2)$ search operations. However, the Bruhat decomposition exists for any nonsingular matrix. (In comparison, Gaussian elimination requires that matrix A satisfies the conditions of Theorem 1.2.1.)

We illustrate Algorithm 2.3.1 by the following.

Example 2.3.3.

Let

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

In the first step of the Bruhat decomposition, a_{41} is the pivot entry; thus $\pi(4)=1$ and

$$A^{(1)} = A\begin{bmatrix} 1 & 0 & -3 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 6 & 1 & 2 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & 2 & -1 & 2 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 & 1 \end{bmatrix}.$$

In the second step $a_{52}^{(1)}$ is the pivot entry. Thus

$$A^{(2)} = A^{(1)}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 6 & 1 & 2 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & 2 & 1 & 3 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The final result of the decomposition is

$$A = \begin{bmatrix} 0.5 & -2 & 6 & 2 & 0 \\ 0 & 0.5 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & -3 & -1 & 0 \\ 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0.5 & 1.5 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Matrix A in Example 2.3.3 above does not have an LU decomposition, but there exists a permutation matrix P such that PA has an LU decomposition. Note that such a permutation matrix is Π^T from the left Bruhat decomposition. Note that neither P nor Π can be determined from the zero-nonzero pattern of the matrix A, i.e., they depend on the numeric computation as well as the zero-nonzero pattern of A. In particular, $a_{51} = 0$ and $a_{41} \neq 0$ imply that $\pi(4) = 1$, but $a_{24}^{(3)} \neq 0$ implies that $\pi(2) = 4$ even though $a_{24} = 0$. (See the third step of Example 4.2.2).

2.4 Bruhat decomposition of a rank one perturbation of a diagonal matrix

In this section, the Bruhat decomposition of a rank one perturbation of a diagonal matrix is presented. Some properties of the decomposition and the original matrix are investigated. We write $D = diag(d_1, d_2, \ldots, d_n)$ to denote the $n \times n$ diagonal matrix D with $d_{ii} = d_i$. For nonzero $u, v \in \mathbf{R}^n$, the matrix $D + uv^T$ is a rank one perturbation of a diagonal matrix. The next theorem gives the Bruhat decomposition of such a matrix.

Theorem 2.4.1. Let $u, v \in \mathbf{R}^n$ with $u_n v_1 \neq 0$, and $D = diag(d_1, d_2, \ldots, d_n)$ be a nonsingular matrix. If $D + uv^T$ is nonsingular, then $D + uv^T = V\Pi U$ is its left Bruhat decomposition, where

$$V = \begin{bmatrix} c & x^T & a \\ & D_1 & y \\ 0 & & u_n v_1 \end{bmatrix}, \ \Pi = \begin{bmatrix} 0 & & 1 \\ & I_{n-2} & \\ 1 & & 0 \end{bmatrix}, \text{ and } U = \begin{bmatrix} 1 & z^T & (d_n + u_n v_n)/u_n v_1 \\ & I_{n-2} & w \\ 0 & & 1 \end{bmatrix},$$

with $c = \frac{-d_1 d_n}{u_n v_1}\left(u^T D^{-1} v + 1\right)$, $a = d_1 + u_1 v_1$, $D_1 = diag(d_2, d_3, \ldots, d_{n-1})$, and $x, y, z \in \mathbf{R}^{n-2}$

with $x_j = \frac{-d_1 v_j}{v_1}$, $y_j = u_j v_1$, $z_j = \frac{v_j}{v_1}$, and $w_j = \frac{-u_j d_n}{u_n d_j}$ for $j = 2, 3, \ldots, n-1$.

The decomposition can be verified by matrix multiplication. As observed in [K&Y], the decomposition above is both the left and the right Bruhat decomposition with $\Pi = \Pi^T$, and both matrices U and V have the same zero-nonzero pattern, namely

$$\mathbf{R}=\begin{bmatrix} * & \otimes & \otimes & \cdots & \otimes \\ & * & & 0 & \otimes \\ & & \ddots & & \vdots \\ & 0 & & * & \otimes \\ & & & & * \end{bmatrix},$$

where $\otimes$ is any real number and $*$ is any nonzero real number.

It is easy to verify that the set of nonsingular upper triangular matrices having the pattern $\mathbf{R}$ is a subgroup under matrix multiplication of the set of nonsingular upper triangular matrices. Thus the left (and right) Bruhat decomposition of $R_1\left(D+uv^T\right)R_2$ is $(R_1V)\Pi(UR_2)$, where R_1 and R_2 have the pattern $\mathbf{R}$ above and R_2 is normalized. That is, the pattern of the decomposition is invariant under multiplication on the left or right by a matrix with pattern $\mathbf{R}$. With U and V as in Theorem 2.4.1,

$$U^{-1}=\begin{bmatrix} 1 & -z^T & z^T w-(d_n+u_nv_n)/u_nv_1 \\ & I_{n-2} & -w \\ 0 & & 1 \end{bmatrix},$$

$$V^{-1}=\begin{bmatrix} c^{-1} & -c^{-1}D_1^{-1}x^T & c^{-1}(x^T D_1^{-1}y-a)/u_nv_1 \\ & D_1^{-1} & -D_1^{-1}y/u_nv_1 \\ 0 & & 1/u_nv_1 \end{bmatrix},$$

and $U^{-1}\Pi V^{-1}$ is both the left and the right Bruhat decomposition of $\left(D+uv^T\right)^{-1}$. The Bruhat decompositions of $D+uv^T$ and $\left(D+uv^T\right)^{-1}$ give an alternate proof of known formulas for $\det\left(D+uv^T\right)$ and $\left(D+uv^T\right)^{-1}$, see e.g. [Dat., p. 239].

Chapter 3

Numerical stability and a pivoting strategy

In this chapter, the stability of the Bruhat decomposition is discussed and compared with stability results for LU and GEPP. Stability results of the left Bruhat decomposition for classes of matrices that give exponential γ_{GP} are discussed, and Bruhat decomposition is shown to be stable for classes of matrices given by Wilkinson and by Foster. A pivoting strategy for Bruhat decomposition (BDPP) is developed and a relationship between the factors of BDPP and GEPP is given. BDPP is shown to give a growth factor of at most 2 for any real $n \times n$ matrix that gives the maximum growth for GEPP, namely $\gamma_{GP} = 2^{n-1}$.

3.1 Bruhat decomposition of matrices with large γ_{GP}

In this section we consider the application of the left Bruhat decomposition to matrices for which GEPP gives large growth factor. Recall from Section 1.3 that if Gaussian elimination (without pivoting) is applied to solve a system $Ax = b$, then the computed solution $\hat{x}$ satisfies $(A+E)\hat{x} = b$ with $\|E\|_\infty \le n\mu\left(3\|A\|_\infty + 5\|\hat{L}\|_\infty \|\hat{U}\|_\infty\right) + O(\mu^2)$, where $\hat{L}\hat{U}$ is the computed LU factorization of A using Gaussian elimination. In fact, if the left Bruhat decomposition (Algorithm 2.3.1) is used to solve a system $Ax = b$, then the computed solution $\tilde{x}$ also satisfies an equation of the same form, namely $(A+\tilde{E})\tilde{x} = b$, with

$$\|E\|_\infty \le n\mu\left(3\|A\|_\infty + 5\|\tilde{V}\|_\infty \|\tilde{U}\|_\infty\right) + O(\mu^2). \tag{3.1.1}$$

To see this, let $\tilde{V}\Pi\tilde{U}$ be the computed left Bruhat decomposition of A. Using the relationship between the left Bruhat and the LU decompositions of A (Corollary 2.1.5), we obtain

$$\|\tilde{E}\|_\infty \le n\mu\left(3\|\Pi^T A\|_\infty + 5\|\Pi^T \tilde{V}\Pi\|_\infty \|\tilde{U}\|_\infty\right) + O(\mu^2). \tag{3.1.2}$$

Using the facts that $\|\Pi^T \tilde{V}\Pi\|_\infty = \|\tilde{V}\|_\infty$ and $\|\Pi^T A\|_\infty = \|A\|_\infty$, we obtain (3.1.1), which is similar to the bound for Gaussian elimination. Note that $\|\tilde{V}\|_\infty$ or $\|\tilde{U}\|_\infty$ can be unbounded, as can be shown by the matrix

$$A = \begin{bmatrix} 1 & 1 \\ \varepsilon & 1 \end{bmatrix}$$

for arbitrarily small ε.

It is important to note that growth in both triangular factors U and V of the Bruhat decomposition affects the computation of the solution of a linear system. We define the *growth factor* for Bruhat decomposition as

$\gamma_B = \max\{\gamma_U, \gamma_V\}$, where $\gamma_U = \dfrac{\max\limits_{i,j,k} \left|u_{jk}^{(i)}\right|}{\max\limits_{j,k} \left|a_{jk}\right|}$ and $\gamma_V = \dfrac{\max\limits_{i,j,k} \left|a_{jk}^{(i)}\right|}{\max\limits_{j,k} \left|a_{jk}\right|}$. Note that γ_U and γ_V represent the growth in U and V, respectively. Note that $\|V\|_\infty \le n\gamma_V \|A\|_\infty \le n\gamma_B \|A\|_\infty$, and the bound for $\|U\|_\infty$ is similar, and thus the perturbation in A depends on γ_B (see (3.1.1)).

We now consider application of Algorithm 2.3.1 to matrices that give exponential growth when GEPP is applied. The classical $n \times n$ example due to Wilkinson[Wil.65] has $\gamma_{GP} = 2^{n-1}$. The Bruhat decomposition, on the other hand, gives growth factor equal to 2. The following example demonstrates this fact.

Example 3.1.1.

The left Bruhat decomposition of the 5×5 Wilkinson matrix W_5 is

$$W_5=\begin{bmatrix}1&0&0&0&1\\-1&1&0&0&1\\-1&-1&1&0&1\\-1&-1&-1&1&1\\-1&-1&-1&-1&1\end{bmatrix}=\begin{bmatrix}2&-1&-1/2&-1/4&1\\0&2&0&0&-1\\0&0&2&0&-1\\0&0&0&2&-1\\0&0&0&0&-1\end{bmatrix}\begin{bmatrix}0&0&0&0&1\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\1&0&0&0&0\end{bmatrix}\begin{bmatrix}1&1&1&1&1\\0&1&1/2&1/2&0\\0&0&1&1/2&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}.$$

In general, application of Algorithm 2.3.1 to the $n\times n$ Wilkinson matrix W_n gives $\gamma_U=1,\ \gamma_V=2,$ and thus $\gamma_B=2.$

Recently, Foster[Fos.] discussed a class of $n\times n$ matrices that arises in the numerical solution of Volterra integral equations and that for GEPP has growth factor close to the maximal value of 2^{n-1}. We consider the numerical solution of the following boundary value problem given by Foster.

Suppose positive constants L, k, and C, and a function $g(t)$, $0\le t\le L$, are given. We wish to find a numerical approximation to the solution $x(s)$ of the problem

$$x(s)-\int_0^s kx(t)dt-x(L)/C=\int_0^s g(t)dt\ .$$

Foster used the trapezoidal rule to approximate the integrals and obtained the linear system $Ax=b$, where the $n\times n$ matrix A is given below with $\alpha=kh/2$, $h=L/(n-1)$ and $\beta=-1/C$:

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & \beta \\ -\alpha & 1-\alpha & 0 & \cdots & 0 & \beta \\ -\alpha & -2\alpha & 1-\alpha & \ddots & \vdots & \vdots \\ \vdots & \vdots & & \ddots & 0 & \\ -\alpha & -2\alpha & \cdots & -2\alpha & 1-\alpha & \beta \\ -\alpha & -2\alpha & \cdots & -2\alpha & -2\alpha & 1+\beta-\alpha \end{bmatrix}. \tag{3.1.4}$$

It is easily seen that if $|\alpha| \leq 1/3$, then no pivoting is required when GEPP is applied. Application of GEPP to the matrix A above with n=61, L=40, C=6, and k=1 gives a growth factor equal to 1.28×10^{17} (see Foster [Fos., p. 1360]). The left Bruhat decomposition of the above matrix A is $A=V\Pi U$, where

$$V = \begin{bmatrix} d & -2 & \frac{-2(1-\alpha)}{1+\alpha} & \cdots & \frac{-2(1-\alpha)^{n-3}}{(1+\alpha)^{n-3}} & 1 \\ & 1+\alpha & 0 & \cdots & 0 & -\alpha \\ & & 1+\alpha & \ddots & \vdots & \vdots \\ & & & \ddots & 0 & \\ & \mathbf{0} & & & 1+\alpha & -\alpha \\ & & & & & -\alpha \end{bmatrix} \text{ with } d = \beta + \frac{1+\beta-\alpha}{\alpha} - 2\sum_{i=1}^{n-2}\left(\frac{1-\alpha}{1+\alpha}\right)^i,$$

$$\Pi = \begin{bmatrix} 0 & & 1 \\ & I_{n-2} & \\ 1 & & 0 \end{bmatrix}, \text{ and } U = \begin{bmatrix} 1 & 2 & 2 & \cdots & 2 & \frac{1+\beta-\alpha}{-\alpha} \\ & 1 & \frac{2\alpha}{1+\alpha} & \cdots & \frac{2\alpha}{1+\alpha} & \frac{\alpha-1}{\alpha+1} \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & 1 & \frac{2\alpha}{1+\alpha} & \frac{\alpha-1}{\alpha+1} \\ & \mathbf{0} & & & 1 & \frac{\alpha-1}{\alpha+1} \\ & & & & & 1 \end{bmatrix},$$

from which we obtain the following.

Theorem 3.1.2. Let A be the $n \times n$ matrix given by (3.1.4) with $\alpha = kL/2(n-1)$. If $|\alpha| \leq 1/3$ and $\max_{1 \leq i,j \leq n} |a_{ij}| = 1$, then γ_B is linear in n.

Proof

Application of Algorithm 2.3.1 shows that $\max_{i,j,k} |a_{jk}^{(i)}| \leq \max_{i,j} |v_{jk}|$. The maximum absolute value of an entry in V is $\max\{2, |d|\}$ where for large n

$$\begin{aligned}|d| &\leq |\beta| + \left|\frac{1+\beta-\alpha}{\alpha}\right| + 2(n-2)\\ &= |\beta| + \left|\frac{2(n-1)(1+\beta)}{kL} - 1\right| + 2(n-2).\end{aligned}$$

That is, $|d|$ grows linearly with n and $\gamma_V = \max\{2, |d|\}$. Similarly $\gamma_U = \max\left\{2, \left|\frac{1+\beta-\alpha}{\alpha}\right|\right\}$ is linear in n for large n. Thus γ_B is linear in n. ■

Note that the zero-nonzero pattern of V is **R** as defined in Section 2.4. This pattern occurs in V whenever A is a rank one perturbation of an upper triangular matrix .

3.2 A pivoting strategy for Bruhat decomposition

Bruhat decomposition is a good alternative to GEPP for the matrices in Section 3.1 when the latter gives exponentially large growth factors. However, for some matrices both GEPP and Bruhat decomposition give exponential growth. For example, the block matrix given by Wright [Wri., equations (10) and (12)] has an exponential growth factor when GEPP is applied, and application of Algorithm 2.3.1 to Wright's equation (10) also gives exponential growth. There are also examples of matrices for which Bruhat decomposition gives exponential growth and GEPP gives constant growth; one such example is ρW_n, see the comments after Example 3.2.2.

We now present a pivoting strategy for Bruhat decomposition that, like the use of partial pivoting with Gaussian elimination, keeps the magnitudes of the multipliers bounded by 1 and usually results in a stable computation. The decomposition is computed by post-multiplication of A by n-1 pairs of nonsingular matrices $P^{(i)}U^{(i)}$ for $i=1,2,\ldots,n-1$, where $P^{(i)}$ is a permutation matrix and $U^{(i)}$ is chosen to introduce zeros into the matrix product. Let $A^{(0)}=A$ and $A^{(i)}=A^{(i-1)}P^{(i)}U^{(i)}=\left[a_{jk}^{(i)}\right]$ so that $A^{(i)} = AP^{(1)}U^{(1)}P^{(2)}U^{(2)}\cdots P^{(i)}U^{(i)}$. At the ith step of the decomposition, $P^{(i)}$ is chosen to interchange columns i and c of $A^{(i-1)}$, where c is such that $\max\limits_{i\le t\le n}\left|a_{n-i+1,t}^{(i-1)}\right|=\left|a_{n-i+1,c}^{(i-1)}\right|$.

Then $U^{(i)}$ is chosen so that $a^{(i)}_{n-i+1,r} = 0$, for

$r = i+1, i+2, \ldots, n$. That is, letting $A^{(i-1)}P^{(i)} = \left[\tilde{a}^{(i-1)}_{jk}\right]$, then $U^{(i)} = I - e^{(i)}\left(m^{(i)}\right)^T$, where

$$m^{(i)}_j = \begin{cases} \dfrac{\tilde{a}^{(i-1)}_{n-i+1,j}}{\tilde{a}^{(i-1)}_{n-i+1,i}} & \text{, for } i+1 \le j \le n \\ 0 & \text{, otherwise.} \end{cases}$$

Note that $U^{(i)}$ differs from I only in its ith row. Thus, for example,

$$U^{(1)} = \begin{bmatrix} 1 & -\dfrac{\tilde{a}_{n2}}{\tilde{a}_{n1}} & -\dfrac{\tilde{a}_{n3}}{\tilde{a}_{n1}} & \cdots & -\dfrac{\tilde{a}_{nn}}{\tilde{a}_{n1}} \\ & 1 & & & \\ & & \ddots & 0 & \\ & 0 & & 1 & \\ & & & & 1 \end{bmatrix} \text{and } A^{(1)} = \begin{bmatrix} a^{(1)}_{11} & a^{(1)}_{12} & \cdots & a^{(1)}_{1n} \\ \vdots & \vdots & & \vdots \\ & & & \\ a^{(1)}_{n-1,1} & a^{(1)}_{n-1,2} & \cdots & a^{(1)}_{n-1,n} \\ a^{(1)}_{n1} & 0 & \cdots & 0 \end{bmatrix}.$$

After n-1 steps,

$$\begin{aligned} A^{(n-1)} &= AP^{(1)}U^{(1)}P^{(2)}U^{(2)}\cdots P^{(n-1)}U^{(n-1)} \\ &= \begin{bmatrix} a^{(1)}_{11} & a^{(2)}_{12} & \cdots & a^{(n-2)}_{1,n-2} & a^{(n-1)}_{1,n-1} & a^{(n-1)}_{1n} \\ a^{(1)}_{21} & a^{(2)}_{22} & \cdots & a^{(n-2)}_{2,n-2} & a^{(n-1)}_{2,n-1} & \\ a^{(1)}_{31} & a^{(2)}_{32} & \cdots & a^{(n-2)}_{3,n-2} & & \\ \vdots & \cdot^{\cdot^{\cdot}} & & & \mathbf{0} & \\ a^{(1)}_{n1} & & & & & \end{bmatrix} \\ &= V\rho, \end{aligned}$$

where V is an upper triangular matrix and ρ is the permutation matrix given in (1.1.1).

The following algorithm essentially determines the factors of the above decomposition

$$A = V\rho(U^{(n-1)})^{-1}P^{(n-1)}(U^{(n-2)})^{-1}P^{(n-2)}\cdots(U^{(1)})^{-1}P^{(1)},$$

where we note that $(P^{(i)})^{-1} = P^{(i)}$ for $i = 1, 2, \dots, n-1$. The one-dimensional array P has $P(i)=c$ if $P^{(i)}$ interchanges columns i and c of $A^{(i-1)}$. The ith row of the upper triangular matrix $(U^{(i)})^{-1}$ is stored in the ith row of an $n \times n$ matrix U. The reduced matrices $A^{(i)}$ overwrite A and the function *swap(i,c)* is used to interchange columns i and c of A.

Algorithm 3.2.1: Bruhat Decomposition with Partial Pivoting (BDPP)

Input: Nonsingular $n \times n$ matrix A

Output: The essential components of the factors V, $(U^{(i)})^{-1}$ and $P^{(i)}$ of the Bruhat decomposition with partial pivoting of A.

Initialization: $U = I$, $P(j) = j$ for $j = 1, 2, \dots, n-1$

```
for $j = n$ to $2$
    $i = n - j + 1$
    find $c : \max_{i \le t \le n} |a_{jt}| = |a_{jc}|$
    if $c > i$ then
        $swap(i,c)$
        $P(i) = c$
    $v_{tj} = a_{ti}$ for $1 \le t \le j$
    for $k = i+1$ to $n$
        $m = \frac{a_{jk}}{a_{ji}}$
        $u_{ik} = m$
        for $l = 1$ to $j-1$
            $a_{lk} = a_{lk} - m a_{li}$
        $a_{jk} = 0$
$v_{11} = a_{1n}$
```

We illustrate Algorithm 3.2.1 on the following matrix.

Example 3.2.2.

Consider the matrix

$$\rho W_5 = \begin{bmatrix} -1 & -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

In the first step of Algorithm 3.2.1, no swapping is necessary. Thus $P(1) = 1$,

$$U^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \text{and } (\rho W_5)^{(1)} = \begin{bmatrix} -1 & -1 & -1 & -1 & 2 \\ -1 & -1 & -1 & 1 & 2 \\ -1 & -1 & 1 & 0 & 2 \\ -1 & 1 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

In the second step of the algorithm, $P(2) = 5$ and

$$U^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -0.5 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \text{ Thus } (\rho W_5)^{(2)} = \begin{bmatrix} -1 & 2 & -1 & -1 & -2 \\ -1 & 2 & -1 & 1 & -2 \\ -1 & 2 & 1 & 0 & -2 \\ -1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The final result of the decomposition is

$$V\rho = (\rho W_5)^{(4)} = \begin{bmatrix} -1 & 2 & -2 & -2 & -2 \\ -1 & 2 & -2 & -2 & 0 \\ -1 & 2 & -2 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \; U = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0.5 \\ 0 & 0 & 1 & 0 & 0.5 \\ 0 & 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and}$$

$P = [1 \quad 5 \quad 5 \quad 5]$.

In general, application of Algorithm 3.2.1 to ρW_n gives V, U, and P having similar patterns to those for ρW_5, and thus $\gamma_{BP} = 2$, where we define

$\gamma_{BP} = \max_{i,j,k} \left|a_{jk}^{(i)}\right| \Big/ \max_{j,k} \left|a_{jk}\right|$ to be the *growth factor* for BDPP. Note that the definition for γ_{BP} is a special case of γ_B when the growth in U is bounded by 1. For ρW_n, it can be shown that $\gamma_{GP} = 2$ and $\gamma_B = 2^{n-1}$. The transpose of the Wilkinson matrix, W_n^T, is another matrix that has an exponential growth factor ($\gamma_B = 2^{n-1}$) when Algorithm 2.3.1 is applied, and a constant growth factor ($\gamma_{BP} = 4$) when Algorithm 3.2.1 is applied.

Note the similarity between BDPP and GEPP. In fact the two algorithms are equivalent in the sense that if GEPP is applied to a matrix A, then the permutation matrices, matrices of multipliers, and derived matrices, $A^{(i)}$, can be obtained from those of BDPP of ρA^T. The next theorem shows this equivalence.

Theorem 3.2.3. Let A be an $n \times n$ nonsingular matrix. Suppose that $L^{(n-1)}\overline{P}^{(n-1)}L^{(n-2)}\overline{P}^{(n-2)} \ldots L^{(1)}\overline{P}^{(1)}A = \overline{U}$ is the result of applying GEPP to A, where $L^{(i)}$ is the lower triangular matrix of multipliers and $\overline{P}^{(i)}$ is the permutation matrix associated with the ith step of GEPP. Suppose also that

$\rho A^T P^{(1)} U^{(1)} P^{(2)} U^{(2)} \dots P^{(n-1)} U^{(n-1)} = V\rho$ is the result of applying BDPP to $\rho A^T = B$.

Then $P^{(i)} = \overline{P}^{(i)}$, $U^{(i)} = (L^{(i)})^T$, and $\rho\left(A^{(i)}\right)^T = B^{(i)}$ for $1 \le i \le n-1$.

Proof

The proof is by induction. For i=1, consider the first step of GEPP. Let $\max_{1\le j\le n} |a_{j1}| = |a_{t1}|$.

Thus the effect of $\overline{P}^{(1)}$ is to interchange rows 1 and t. Letting $\overline{P}^{(1)}A = \left[\tilde{a}_{jk}\right]$, then

$L^{(1)} = I - m^{(1)}(e^{(1)})^T$, where $m_j^{(1)} = \dfrac{\tilde{a}_{j1}}{\tilde{a}_{11}}$ is the jth entry of the vector $m^{(1)}$ for $2 \le j \le n$

and $m_1^{(1)} = 0$. Thus $A^{(1)} = L^{(1)} P^{(1)} A$. Now consider the first step of BDPP applied to

$B = \rho A^T$. Note that $b_{n-p+1,j} = \sum_{k=1}^{n} \rho_{n-p+1,k} a_{jk} = a_{jp}$ for $1 \le j, p \le n$. Thus

$\max_{1\le j\le n} |b_{nj}| = \max_{1\le j\le n} |a_{j1}| = |a_{t1}| = |b_{nt}|$, and the effect of $P^{(1)}$ is to interchange columns 1 and

t, so that $P^{(1)} = (\overline{P}^{(1)})^T = \overline{P}^{(1)}$. Let $BP^{(1)} = \left[\tilde{b}_{jk}\right]$, and note that $BP^{(1)} = \rho\left(\overline{P}^{(1)} A\right)^T$.

Now $U^{(1)} = I - e^{(1)}(x^{(1)})^T$, where $x_j^{(1)} = \dfrac{\tilde{b}_{nj}}{\tilde{b}_{n1}} = \dfrac{\tilde{a}_{j1}}{\tilde{a}_{11}} = m_j^{(1)}$ for $2 \le j \le n$ and

$x_1^{(1)} = m_1^{(1)} = 0$. Thus $U^{(1)} = (L^{(1)})^T$, hence $B^{(1)} = BP^{(1)}U^{(1)} = \rho\left(L^{(1)}\overline{P}^{(1)}A\right)^T = \rho\left(A^{(1)}\right)^T$.

Thus the statement is true for i=1.

Suppose that the theorem is true for all i such that $1 \le i \le s < n-1$, and consider the (*s*+1)st step of GEPP. Let $\max_{s+1\le j\le n}\left|a_{j,s+1}^{(s)}\right| = \left|a_{r,s+1}^{(s)}\right|$. Thus the effect of $\overline{P}^{(s+1)}$ is to interchange rows (*s*+1) and *r*. Letting $\overline{P}^{(s+1)}A^{(s)} = \left[\tilde{a}_{jk}^{(s)}\right]$, then $L^{(s+1)} = I - m^{(s+1)}(e^{(s+1)})^T$, where $m_j^{(s+1)} = \dfrac{\tilde{a}_{j,s+1}^{(s)}}{\tilde{a}_{s+1,s+1}^{(s)}}$ for $s+2\le j\le n$ and $m_j^{(s+1)} = 0$ for $1\le j\le s+1$. Thus $A^{(s+1)} = L^{(s+1)}\overline{P}^{(s+1)}A^{(s)}$. Now consider the (*s*+1)st step of BDPP applied to $B=\rho A^T$. By the induction hypothesis, $B^{(s)} = \rho\left(A^{(s)}\right)^T$, and consequently $\max_{s+1\le j\le n}\left|b_{n-s,j}^{(s)}\right| = \max_{s+1\le j\le n}\left|a_{j,s+1}^{(s)}\right| = \left|a_{r,s+1}^{(s)}\right| = \left|b_{n-s,r}^{(s)}\right|$. Thus the effect of $P^{(s+1)}$ is to interchange columns (*s*+1) and *r*, so that $P^{(s+1)} = (\overline{P}^{(s+1)})^T = \overline{P}^{(s+1)}$. Let $B^{(s)}P^{(s+1)} = \left[\tilde{b}_{jk}^{(s)}\right]$, and note that $B^{(s)}P^{(s+1)} = \rho\left(\overline{P}^{(s+1)}A^{(s)}\right)^T$. Now $U^{(s+1)} = I - e^{(s+1)}(x^{(s+1)})^T$, where $x_j^{(s+1)} = \dfrac{\tilde{b}_{n-s,j}^{(s)}}{\tilde{b}_{n-s,s+1}^{(s)}} = \dfrac{\tilde{a}_{j,s+1}^{(s)}}{\tilde{a}_{s+1,s+1}^{(s)}} = m_j^{(s+1)}$ for $s+2\le j\le n$ and $x_j^{(s+1)} = m_j^{(s+1)} = 0$ for $1\le j\le s+1$. Thus $U^{(s+1)} = (L^{(s+1)})^T$, and $B^{(s+1)} = BP^{(s+1)}U^{(s+1)} = \rho\left(L^{(s+1)}\overline{P}^{(s+1)}A\right)^T = \rho\left(A^{(s+1)}\right)^T$, completing the proof. ■

Remark 3.2.4. Consider GEPP applied to $A^T\rho$. From Theorem 3.2.3, this is equivalent to application of BDPP to $\rho(A^T\rho)^T = A$.

An immediate consequence of Theorem 3.2.3 is the following, which shows that

Bruhat decomposition with partial pivoting on A determines the left Bruhat decomposition of a column permutation of A. This result is analogous to a well known result for GEPP .

Corollary 3.2.5. Suppose A is an $n \times n$ nonsingular matrix and let $AP^{(1)}U^{(1)}P^{(2)}U^{(2)}\cdots P^{(n-1)}U^{(n-1)} = V\rho$ be the result of BDPP applied to A. Then there exist a permutation matrix P and an upper triangular matrix U such that $AP = V\rho U$.

Proof

Let $L^{(n-1)}\overline{P}^{(n-1)}L^{(n-2)}\overline{P}^{(n-2)}\cdots L^{(1)}\overline{P}^{(1)}A^T\rho = L^{-1}\overline{P}A^T\rho = \overline{U}$ be the result of applying GEPP to $A^T\rho$ (see, e.g., [Dat., p. 123] and [Ste., p.125]). Thus

$$\begin{aligned}\overline{U}^T &= \rho A\overline{P}^T(L^{-1})^T \\ &= \rho A\overline{P}^{(1)}(L^{(1)})^T\cdots\overline{P}^{(n-1)}(L^{(n-1)})^T \\ &= \rho AP^{(1)}U^{(1)}\cdots P^{(n-1)}U^{(n-1)}, \text{ by Remark 3.2.4.}\end{aligned}$$

Hence $A\overline{P}^T(L^{-1})^T = V\rho$, which implies that $A\overline{P}^T = V\rho L^T$, giving the required result with $P = \overline{P}^T$ and $U = L^T$. ■

We summarize the relationship between GEPP and BDPP in the following Theorem, where the growth factor result (*ii*) follows from the relation between the derived matrices in Theorem 3.2.3.

Theorem 3.2.6. Suppose A is an $n \times n$ nonsingular matrix. If the result of applying GEPP to A is $\overline{P}A = L\overline{U}$, and the result of applying BDPP to ρA^T is $\rho A^T P = V\rho U$, then (*i*) $\overline{P} = P^T$, $L = U^T$, and $\overline{U} = \rho V^T \rho$,

(*ii*) γ_{GP} for A equals γ_{BP} for ρA^T.

Finally, Theorem 3.2.3 also implies that if Bruhat decomposition with partial pivoting (Algorithm 3.2.1) is used to solve a system $Ax = b$, then the computed solution $\hat{x}$ satisfies $(A + \hat{E})\hat{x} = b$, with

$$\left\|\hat{E}\right\|_\infty \le c\mu n^3 \gamma_{BP} \left\|A^T\right\|_\infty + O(\mu^2), \tag{3.2.1}$$

where γ_{BP} denotes the growth factor for A. This can be shown by considering the computed solution $\hat{x}$ to $(A^T\rho)x = b$ using GEPP, which was shown in Section 1.3 to satisfy $(A^T\rho + E)\hat{x} = b$ with

$$\|E\|_\infty \le c\mu n^3 \gamma_{BP} \left\|A^T\rho\right\|_\infty + O(\mu^2),$$

where γ_{GP} denotes the growth factor for $A^T\rho$. The bound (3.2.1) follows using Remark 3.2.4 and Theorem 3.2.6 (*ii*), and the fact that $\left\|A^T\rho\right\|_\infty = \left\|A^T\right\|_\infty$.

3.3 Stability of BDPP for matrices with large γ_{GP}

We now show that Bruhat decomposition with partial pivoting is stable for every $n \times n$ real matrix that has $\gamma_{GP} = 2^{n-1}$. The following theorem due to Higham and Higham characterizes such matrices.

Theorem 3.3.1. [H&H, Theorem 2.2]. All real $n \times n$ matrices for which $\gamma_{GP} = 2^{n-1}$ are of the form $A = DM\begin{bmatrix} T & \vdots & \theta d \\ 0 & \vdots & \end{bmatrix}$, where $D = diag(\pm 1)$, M is unit lower triangular with $m_{ij} = -1$ for $i > j$, $T = [t_{ij}]$ is a nonsingular upper triangular matrix of order n-1, $d = [1\ 2\ 4\ \cdots\ 2^{n-1}]^T$, and θ is a scalar such that $\theta = |a_{1n}| = \max_{i,j}|a_{ij}|$.

The general form of a 5×5 matrix with $D=I$ having $\gamma_{GP} = 2^4$ is

$$A = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} & \theta \\ -t_{11} & t_{22} - t_{12} & t_{23} - t_{13} & t_{24} - t_{14} & \theta \\ -t_{11} & -(t_{22} + t_{12}) & t_{33} - (t_{23} + t_{13}) & t_{34} - (t_{24} + t_{14}) & \theta \\ -t_{11} & -(t_{22} + t_{12}) & -(t_{33} + t_{23} + t_{13}) & t_{44} - (t_{34} + t_{24} + t_{14}) & \theta \\ -t_{11} & -(t_{22} + t_{12}) & -(t_{33} + t_{23} + t_{13}) & -(t_{44} + t_{34} + t_{24} + t_{14}) & \theta \end{bmatrix} \qquad (3.3.1)$$

We show in Theorem 3.3.3 that application of BDPP to this class of matrices gives a growth factor of at most 2.

Theorem 3.3.2. Let *A* be an $n \times n$ nonsingular matrix for which $\gamma_{GP} = 2^{n-1}$. Then application of BDPP to *A* gives $\gamma_{BP} \le 2$.

Proof

As *A* is assumed to have $\gamma_{GP} = 2^{n-1}$, matrix *A* must be of the form given in Theorem 3.3.1. At the first step of BDPP on *A*, if $\theta = \max_{1 \le q \le n-1} |a_{nq}| = |a_{n1}| = |t_{11}|$, then no interchange is performed; however, if $\theta > \max_{1 \le q \le n-1} |a_{nq}|$ or $\theta = \max_{2 \le q \le n-1} |a_{nq}| = |a_{nk}|$ with $2 \le k \le n-1$, then $P^{(1)}$ interchanges columns 1 and *n*. (Note that this includes a tie-breaking strategy for BDPP.) After one step of Algorithm 3.2.1, $\max_{j,k} \left|a_{jk}^{(1)}\right| \Big/ \max_{j,k} \left|a_{jk}\right| \le 2$, and the resulting matrix $A^{(1)}$ can be partitioned as $A^{(1)} = \begin{bmatrix} z & \vdots & H \\ & \vdots & 0 \end{bmatrix}$, where *z* is either column 1 or column *n* of *A*, and *H* is an $(n-1) \times (n-1)$ upper Hessenberg matrix with $h_{i,n-1} = 0$ for $i = 2, \ldots, n-1$. Thus further steps require only column permutations (but no eliminations). Thus $\gamma_{BP} \le 2$. ■

Table 3.3.3 summarizes growth factor results for GEPP, BDPP, and Algorithm 2.3.1 applied to the matrix W_n of Example 3.1.1 and related matrices.

Table 3.3.3. Growth factor results.

	γ_{GP}	γ_{BP}	γ_{B}
W_n	2^{n-1}	2	2
ρW_n	2	2	2^{n-1}
W_n^T	2	4	2^{n-1}
ρW_n^T	2	2^{n-1}	2^{n-1}
$\rho W_n^T \rho$	2	2	2
$\rho W_n \rho$	2	2	2^{n-1}
$W_n \rho$	4	2	2
$W_n^T \rho$	2	2	2

Note that for a general $n \times n$ nonsingular matrix, Theorem 3.2.6 shows that the upper bound for the growth factor for BDPP is the same as that for GEPP, namely 2^{n-1}. Thus, Theorem 3.3.1 gives a characterization of all real matrices achieving this bound, namely ρA^T where A is given by Theorem 3.3.1. The matrix ρW_n^T belongs to this class as shown in the table above.

We illustrate other aspects of the growth factor for BDPP in the following examples.

Example 3.3.4.

Consider the matrix

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

Matrix A has $\gamma_B = 3/2$ and $\gamma_{BP} = 2$. Thus it is possible for γ_{BP} to be larger than γ_B. This happens because of the nature of the pivoting strategy, which is not designed to minimize the growth factor.

Example 3.3.5.

Consider the matrix

$$A = \begin{bmatrix} \varepsilon & 1 & 1 & 1 & 20 \\ -\varepsilon & -3 & 1 & 1 & 20 \\ -\varepsilon & 1 & 0 & 0 & 20 \\ -\varepsilon & 1 & -6 & -2 & 20 \\ -\varepsilon & 1 & -6 & -10 & 20 \end{bmatrix},$$

which is the matrix of (3.3.1) with $t_{11} = \varepsilon,\ t_{12} = t_{13} = t_{14} = 1,\ t_{22} = -2,\ t_{23} = t_{24} = 2$, $t_{33} = t_{34} = 3$, $t_{44} = 4$, and $\theta = 20$. For matrix A above $\gamma_{GP} = 2^4$, $\gamma_{BP} = 1$, and $\gamma_B = \max\{2, |20/\varepsilon|\}$. For arbitrarily small ε, γ_B is unbounded. Thus, unlike BDPP, the left Bruhat decomposition (without pivoting) is not stable for all matrices that give the maximum exponential growth factor with GEPP.

As a final remark, we note that other pivoting strategies for Bruhat decomposition were considered. In particular, we considered pivoting strategies that preserve the Bruhat permutation. One such strategy is to find a permutation matrix $P^{(i)}$ whose effect is to interchange the row containing the pivot entry r_i (see Section 2.3) with row *c*, where $\max\limits_{1\le t\le r_i}\left|a_{ti}^{(i-1)}\right| = \left|a_{ci}^{(i-1)}\right|$, so that the pivot entry is the largest entry in column *i*. The next step is to eliminate all entries to the right of $\tilde{a}_{r_i,i}^{(i-1)}$, where $P^{(i)}A^{(i-1)} = \left[\tilde{a}_{jk}^{(i-1)}\right]$. The disadvantage of this strategy is that it does not restrict the magnitudes of the multipliers. It is still possible to have an unbounded multiplier as shown by

$$A = \begin{bmatrix} \varepsilon & 1 \\ -\varepsilon & 1 \end{bmatrix},$$

for arbitrarily small ε. For both GEPP and BDPP, the magnitude of each multiplier is less than or equal to 1, resulting in a growth factor less than or equal to 2^{n-1}. To control the magnitudes of multipliers, a pivoting strategy must take into account the magnitudes of entries that are to be eliminated. The strategy described above does not take these into account, thus resulting in the possibility of an unbounded growth factor.

Chapter 4

Bipartite graph model of the Bruhat decomposition

A bipartite graph model for the Bruhat decomposition of a pattern is discussed in this chapter. This analysis is carried out ignoring accidental cancellation. Definitions and notation from graph theory and combinatorial matrix theory that are needed for the sparsity analysis are introduced. An algorithm for computing the patterns of the factors in the left Bruhat decomposition (Algorithm 4.2.1) is developed and illustrated. This algorithm is shown to model the numerical algorithm (Algorithm 2.3.1) in Section 2.3.

4.1 Introduction

We start this section by introducing some terminology and notation from graph theory and combinatorial matrix theory that is needed for the following analysis. For further details see e.g. [McH.], [Har.], [Pis.], and [G&L].

A (undirected) *graph* $G=(V(G),E(G))$ consists of a pair of sets, $V(G)$ and $E(G)$, where $V(G)$ is a set of vertices and $E(G)$ is a set of unordered pairs of distinct vertices of $V(G)$ called *edges*. We denote by uv the edge between vertices u and v in G. If uv is an edge in G, then we say that the vertices u and v are *adjacent*. The *neighborhood* of vertex $v \in V(G)$ is the set of vertices adjacent to v and is denoted by N_v, thus $N_v = \{u: uv \in E(G)\}$. If $u,v \in V(G)$ and $uv \notin E(G)$, we use the notation $G+uv$ to denote the graph obtained from G by adding the edge uv. Similarly, if $uv \in E(G)$, then G-uv is the graph obtained from G by removing the edge uv. A graph $H=(V(H),E(H))$ is called a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. If $S \subseteq V(G)$, then the subgraph $H=(S,E(H))$, where $E(H)=\{uv:\ uv \in E(G) \text{ and } u,v \in S\}$, is called the *subgraph of G induced by S.*

A graph G is a *bipartite graph* (bigraph) if there exists a bipartition of $V(G)$ such that no two vertices in the same partition are adjacent. Let G be a bipartite graph with bipartition $V(G) = V_1 \cup V_2$. We say that G is a *complete bipartite graph* if every vertex in V_1 is adjacent to every vertex in V_2. The *complement* of G, denoted $\overline{G}$, is a bipartite graph with $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{uv:\ u \in V_1 \text{ and } v \in V_2 \text{ and } uv \notin E(G)\}$. A *uv-path* in G, or simply a *path*, is a sequence of vertices $u = u_1, u_2, u_3, \cdots, u_n = v$ such that $u_i u_{i+1} \in E(G)$ for $1 \le i \le n-1$.

Let A be an $n \times n$ matrix. The *zero-nonzero pattern* (or simply pattern) of A is an $n \times n$ array A with 0 and * entries such that A has a * entry exactly when the corresponding entry in A is nonzero. If matrix A has pattern A, we write $A \in$ A. Let A and B be two $n \times n$ patterns. Pattern addition and multiplication are carried out similarly to matrix addition and multiplication under the following assumptions, which imply that accidental cancellation is ignored:

(Addition) $*+*=*+0=0+*=*$, and $0+0=0$;

(Multiplication) $0 \cdot 0 = * \cdot 0 = 0 \cdot * = 0$, and $* \cdot * = *$.

We illustrate pattern operations in the following example.

Example 4.1.1.

Consider the patterns

$$\mathrm{A} = \begin{bmatrix} * & * & 0 \\ 0 & * & 0 \\ * & 0 & * \end{bmatrix}, \quad \mathrm{B} = \begin{bmatrix} 0 & * & 0 \\ 0 & * & * \\ * & * & * \end{bmatrix}.$$

Then

$$\mathrm{A}+\mathrm{B}=\begin{bmatrix} * & * & 0 \\ 0 & * & * \\ * & * & * \end{bmatrix} \text{ and } \mathrm{AB}=\begin{bmatrix} 0 & * & * \\ 0 & * & * \\ * & * & * \end{bmatrix}.$$

An $n \times n$ pattern A is *combinatorially singular* if and only if every matrix $A \in \mathrm{A}$ is singular. An $n \times n$ pattern A is sometimes modelled by a bigraph $B(\mathrm{A}) = (V(B), E(B))$ with bipartition $V(B) = R \cup C$, where $R = \{r_j : 1 \le j \le n\}$ and $C = \{c_j : 1 \le j \le n\}$, and $r_i c_j \in E(B)$ if and only if the *(i,j)* entry in A is nonzero. For an illustration with *n=5*, see the beginning of Example 4.2.2.

Kolotilina and Yeremin discussed some aspects of the sparsity of the factors of the Bruhat decomposition. Theorem 1 in [K&Y, p. 424] states that the length of the Bruhat permutation $l(\pi)$ is an upper bound on the number of nonzero off diagonal entries in the factor *U* of the right Bruhat decomposition. They also studied effects of permuting the original matrix in order to reduce fill-ins in the upper triangular factor *U*. In a subsequent paper [Kol.] , Kolotilina studied the locations of some zero entries in the triangular factors of the right Bruhat decomposition. The focus of the paper was on obtaining upper and lower bounds on the length of the Bruhat permutation. Kolotilina obtained tight upper and lower bounds on the length of the Bruhat permutation and demonstrated these bounds for Hessenberg matrices and block triangular matrices. Neither paper gave a detailed study of the patterns of the triangular factors of the Bruhat decomposition, which is the aim of this chapter.

4.2 Computation of the patterns of factors in the Bruhat decomposition

We consider the following problem: For an $n \times n$ pattern A that is not combinatorially singular, determine (ignoring accidental cancellation) the Bruhat permutation associated with a nonsingular matrix $A \in$A, and the patterns of the factors U and V in the left Bruhat decomposition of A. Sparsity analysis for the LU decomposition is often modelled by means of directed graphs. However, Golumbic [Gol.] and George, Liu and Ng [GL&N] used bigraphs for their sparsity analyses.

To solve the above problem, we develop an algorithm that computes the patterns of the factors U and V and the Bruhat permutation by performing symbolic elimination on the bigraph of A. For the symbolic elimination, the rules of pattern addition and multiplication as given in Section 4.1 are used.

Algorithm 4.2.1.

Input: An $n \times n$ pattern A that is not combinatorially singular.

Output: Bigraphs of the patterns of V and U, and the matrix Π, in the left Bruhat decomposition of nonsingular $A \in$A (ignoring accidental cancellation).

Initialization: set $B_0 = B(\mathrm{A})$.

```
        set $W_0 = (V(B_0), \{r_i c_i : 1 \le i \le n\})$
        set $colour(v) = 0$ for every $v \in V(B_0)$
for $i = 1$ to $n$
    $j = \max\{p \mid c_i r_p \in E(B_{i-1})\}$
    $\pi_{ji} = 1,\ \pi_{li} = 0$ for $l \ne j$
    Let $H_i$ be the subgraph induced by $\{c_i, r_j\} \cup N_{c_i} \cup N_{r_j} - \{x : colour(x) = 1\}$
    $colour(c_i) = 1$
    $colour(r_j) = 1$
    $B_i = B_{i-1} + E(\overline{H}_i) - \{r_j v \in E(B_{i-1}) : v \ne c_i \text{ and } colour(v) = 0\}$
    $W_i = W_{i-1} + \{r_i v : r_j v \in E(B_{i-1}) \text{ and } r_j v \notin E(B_i)\}$
Relabel column vertices of $V(B_n)$ such that $c_i \to c_{\pi^{-1}(i)}$ for $1 \le i \le n$
```

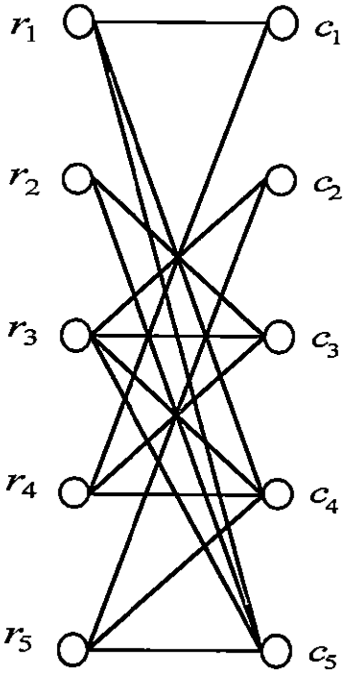
Algorithm 4.2.1 stores the pattern of V in the bigraph B_n and the pattern of U in the bigraph W_n. We illustrate Algorithm 4.2.1 in the following example. In the bigraphs, vertices with colour 0 are unfilled and vertices with colour 1 are filled. An edge of $\overline{H}_i$ that is added to B_{i-1} is indicated by a thick line in B_i.

Example 4.2.2.

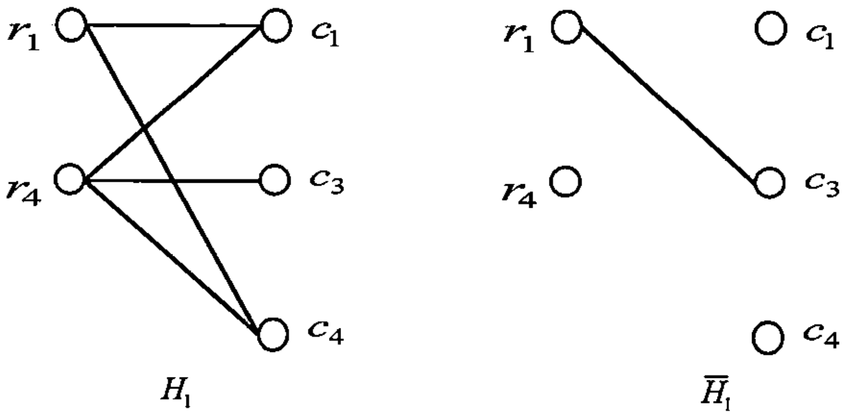
Consider the pattern

$$A = \begin{bmatrix} * & 0 & 0 & * & * \\ 0 & 0 & * & 0 & * \\ 0 & * & * & * & * \\ * & 0 & * & * & 0 \\ 0 & * & 0 & * & * \end{bmatrix}$$

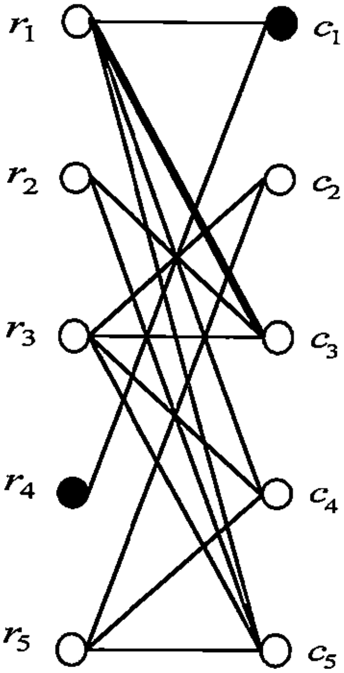
The bigraph $B(\mathrm{A}) = B_0$ is



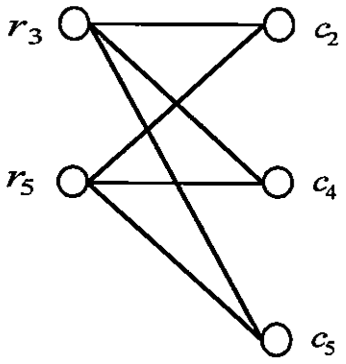
In the first step of Algorithm 4.2.1, *p=4* is the largest index such that $c_1 r_p \in E(B_0)$; thus $\pi_{41} = 1$ and H_1, $\overline{H_1}$ are as follows.



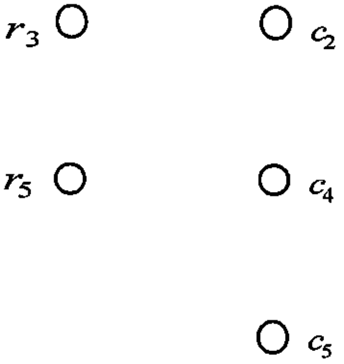
The bigraph B_1 is



and $W_1 = W_0 + \{r_1c_3, r_1c_4\}$. In the second step, $p=5$ is the largest index such that $c_2r_p \in E(B_1)$; thus $\pi_{52} = 1$ and H_2, $\overline{H_2}$ are as follows.

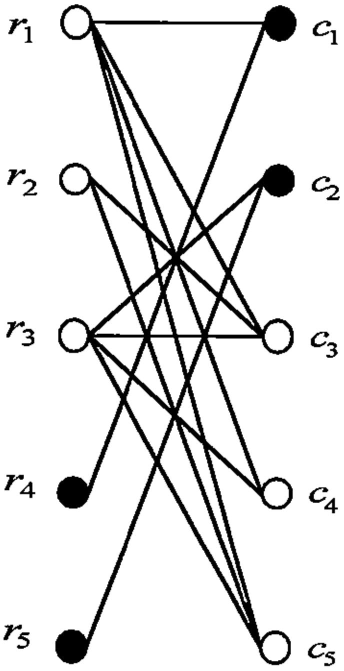


H_2

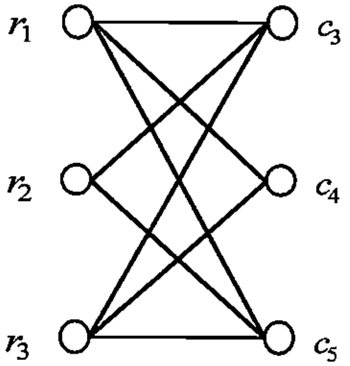


$\overline{H}_2$

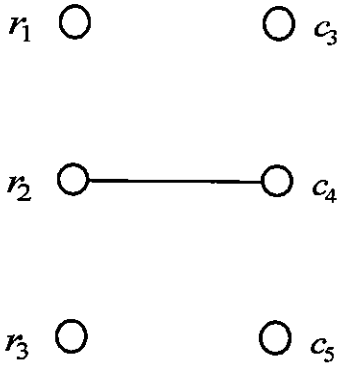
The bigraph B_2 is



and $W_2 = W_1 + \{r_2c_4, r_2c_5\}$. In the third step, *p=3* is the largest index such that $c_3r_p \in E(B_2)$; thus $\pi_{33} = 1$ and H_3, $\overline{H}_3$ are as follows.

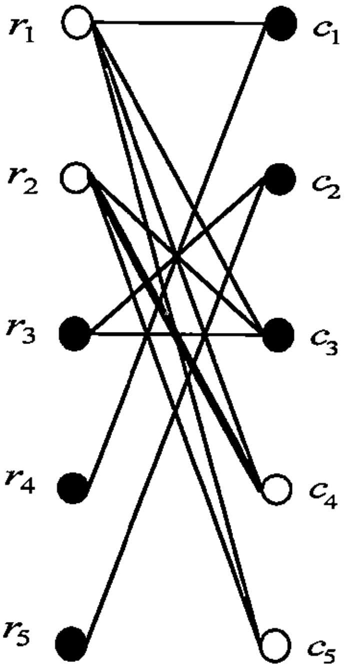


H_3

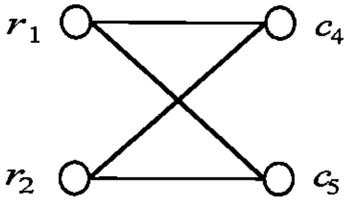


$\overline{H}_3$

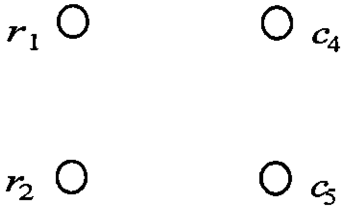
The bigraph B_3 is



and $W_3 = W_2 + \{r_3c_4, r_3c_5\}$. In the fourth step, p=2 is the largest index such that $c_4r_p \in E(B_3)$; thus $\pi_{24} = 1$ and H_4, $\overline{H_4}$ are as follows.

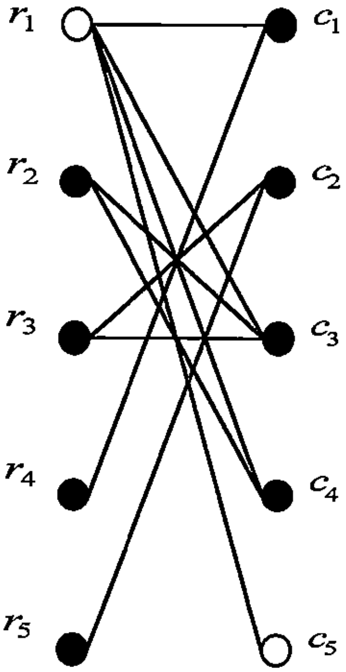


H_4

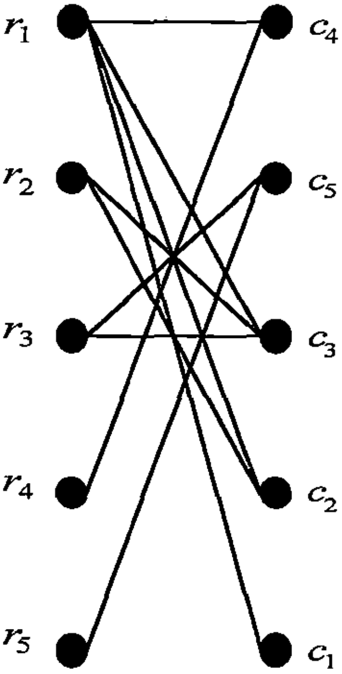


$\overline{H}_4$

The bigraph B_4 is



and $W_4 = W_3 + r_4c_5$. In the final step, the bigraph B_5 is identical to B_4 except that r_1 and c_5 are filled, and $W_5 = W_4$. The relabelled B_5 is as follows.



Thus

$$Pattern(V)=\begin{bmatrix} * & * & * & * & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & 0 & * \\ 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & * \end{bmatrix},\ Pattern(U)=\begin{bmatrix} * & 0 & * & * & 0 \\ 0 & * & 0 & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix}, \text{ and}$$

$$\Pi=\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Matrix A in Example 2.3.3 has pattern A in Example 4.2.2. We note that the patterns of V, Π, and U as computed in Example 4.2.1 are identical to the patterns of the computed factors in the left Bruhat decomposition of A in Example 2.3.3.

We now justify that Algorithm 4.2.1 models the left Bruhat decomposition (Algorithm 2.3.1) assuming no accidental cancellation. The search operation for finding a pivot entry and constructing Π is identical in both algorithms, hence it suffices to show the consistency of the elimination operations. This is shown in the following theorem where the notation from Section 2.3 is used.

Theorem 4.2.3. Let A be an $n \times n$ pattern that is not combinatorially singular. Suppose $A \in$A is nonsingular and let $A^{(i)} = AU^{(1)}U^{(2)} \cdots U^{(i)}$ be the ith derived matrix in the left Bruhat decomposition of A. Then, assuming no accidental cancellation, for $1 \le i \le n-1$ the bigraph of the pattern of $A^{(i)}$ is B_i as defined in Algorithm 4.2.1.

Proof

The proof is by induction. For i=1, consider the first step of the left Bruhat decomposition. Let t_1 be the largest index such that $a_{t_1,1} \neq 0$, and let $U^{(1)} = I - e^{(1)}(m^{(1)})^T$. Thus for $j \ge 2$, $m_j^{(1)} \neq 0$ if and only if $a_{t_1,j} \neq 0$, or equivalently $r_{t_1}c_j \in E(B_0)$. Let $j \ge 2$ be an index such that $m_j^{(1)} \neq 0$, and denote by $A_j^{(i)}$ the jth column of $A^{(i)}$. Then $A_j^{(1)} = A_j - m_j^{(1)} A_1$ and $a_{t_1,j}^{(1)} = 0$. Assuming no accidental cancellation, $a_{kj}^{(1)} \neq 0$ if and only if $a_{kj} \neq 0$ or $a_{k1} \neq 0$, for $1 \le k \le t_1 - 1$. If $a_{kj} \neq 0$ then

$r_k c_j \in E(B_0)$, hence $r_k c_j \in E(B_1)$. On the other hand, if $a_{kj} = 0$ and $a_{k1} \neq 0$, then since $\{r_k, c_1, r_{t_1}, c_j\} \subseteq V(H_1)$ and $r_k c_j \notin E(H_1)$, we must have $r_k c_j \in E(\overline{H}_1)$, hence $r_k c_j \in E(B_1)$. Thus $r_k c_j \in E(B_1)$ if and only if $a_{kj}^{(1)} \neq 0$, hence the bigraph of $A^{(1)}$ is B_1.

Suppose the theorem is true for all *i* such that $1 \le i \le s < n-1$, and consider the (*s*+1)st step of the left Bruhat decomposition. Let t_{s+1} be the largest index such that $a^{(s)}_{t_{s+1},s+1} \neq 0$, and let $U^{(s+1)} = I - e^{(s+1)}(m^{(s+1)})^T$. Thus for $j \ge s+2$, $m_j^{(s+1)} \neq 0$ if and only if $a^{(s)}_{t_{s+1},j} \neq 0$, or equivalently $r_{t_{s+1}} c_j \in E(B_s)$. Let $j \ge s+2$ be an index such that $m_j^{(s+1)} \neq 0$. Then $A_j^{(s+1)} = A_j^{(s)} - m_j^{(s+1)} A_{s+1}^{(s)}$ and $a^{(s+1)}_{t_{s+1},j} = 0$. Assuming no accidental cancellation, $a_{kj}^{(s+1)} \neq 0$ if and only if $a_{kj}^{(s)} \neq 0$ or $a_{k,s+1}^{(s)} \neq 0$, for $1 \le k \le t_{s+1} - 1$. If $a_{kj}^{(s)} \neq 0$ then $r_k c_j \in E(B_s)$, hence $r_k c_j \in E(B_{s+1})$. On the other hand, if $a_{kj}^{(s)} = 0$ and $a_{k,s+1}^{(s)} \neq 0$, then since $\{r_k, c_{s+1}, r_{t_{s+1}}, c_j\} \subseteq V(H_{s+1})$ and $r_k c_j \notin E(H_{s+1})$, we must have $r_k c_j \in E(\overline{H}_{s+1})$, hence $r_k c_j \in E(B_{s+1})$. Thus $r_k c_j \in E(B_{s+1})$ if and only if $a_{kj}^{(s+1)} \neq 0$, hence the bigraph of $A^{(s+1)}$ is B_{s+1}, completing the proof. ■

Theorem 4.2.3, also shows that W_i is the bigraph of the sum of the patterns of the matrices of multipliers $U^{(1)}, U^{(2)}, \cdots, U^{(i)}$.

The symbolic elimination operation in Algorithm 4.2.1 can be interpreted in terms of a path condition on H_i for $1 \le i \le n-1$.

Theorem 4.2.4. Let A be an $n \times n$ pattern that is not combinatorially singular. Suppose $A \in \mathrm{A}$ is nonsingular. Using the notation of Algorithm 4.2.1 and ignoring accidental cancellation, suppose for some *i,j* that $\pi_{ji} = 1$. Let $s \neq j$ and $t > i$ be two integers such that $r_s, c_t \in V(H_i)$. If c_t is reachable by a path from r_s in H_i, then $r_s c_t \in E(B_i)$. Moreover, if *k*>*i* and $r_j c_k \in E(H_i)$, then $r_i c_k \in E(W_i)$.

Proof

Note that $c_t \in V(H_i)$ if and only if $r_j c_t \in E(B_{i-1})$, or equivalently the (*j,t*) position in $A^{(i-1)}$ is nonzero, which implies that the multiplier for column *t* is nonzero (see Theorem 4.2.3). Similarly $r_s \in V(H_i)$ if and only if $r_s c_i \in E(B_{i-1})$, which implies that the (*s,i*) position in $A^{(i-1)}$ is nonzero. Thus r_s, c_i, r_j, c_t is an $r_s c_t$-path in H_i. Since a nonzero multiple of column *i* is added to column *t* to obtain the pattern of $A^{(i)}$, we must have a * in the (*s,t*) position of the pattern of $A^{(i)}$, thus $r_s c_t \in E(B_i)$. Finally $r_j c_k \in E(H_i)$ implies that the multiplier in column *k* in the *i*th step is nonzero, hence $r_i c_k \in E(W_i)$. ■

Note that if $r_s, c_t \in V(H_i)$, then there is always an $r_s c_t$-path in H_i, namely r_s, c_i, r_j, c_t. This is true since $c_t \in N_{r_j}$ and $r_s \in N_{c_i}$ and $r_j c_i \in E(H_i)$. Thus, if $r_s c_t \notin E(H_i)$, then $r_s c_t \in E(\overline{H_i})$, which justifies adding the edges of $\overline{H_i}$ to B_{i-1} to obtain B_i (after removing the edges associated with multipliers).

Finally, we remark that accidental cancellation can affect the computed factors significantly. For example, a rank one perturbation of a diagonal matrix (see Section 2.4) can have a full pattern (no zero entries). Application of Algorithm 4.2.1 to such a matrix gives ρ as the Bruhat permutation and the patterns of U and V are full upper triangular, which does not agree with the numerical results in Section 2.4 (where the matrices U and V have pattern $\mathbf{R}$).

BIBLIOGRAPHY

[Dat.] B. Datta, *Numerical Linear Algebra and Applications*, Brooks/Cole Publishing Company, Toronto, 1995.

[Els.] L. Elsner, *On some algebraic problems in connection with general eigenvalue algorithms*, Linear Algebra Appl. 48 (1982), 123-138.

[Fos.] L. V. Foster, *Gaussian elimination with partial pivoting can fail in practice*, SIAM J. Matrix Anal. Appl. 15 (1994), 1354-1362.

[G&L] A. George and J. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall Inc., New Jersey, 1981.

[GL&N] A. George, J. Liu and E. Ng, *Row-ordering schemes for sparse Givens transformations. I. Bipartite graph model*, Linear Algebra Appl. 61 (1984), 55-81.

[G&G] I. Gohberg and S. Goldberg, *Finite dimensional Weiner-Hopf equations and factorizations of matrices*, Linear Algebra Appl. 48 (1982), 219-236.

[Gol.] H. Goldstine, *A History of Numerical Analysis from the 16th Through the 19th Century*, Springer-Verlag Inc., New York, 1977.

[G&vN.47] H. Goldstine and J. von Neumann, *Numerical inverting of matrices of high order*, Bull. Amer. Math. Soc. 53 (1947), 1021-1099.

[G&vN.51] H. Goldstine and J. von Neumann, *Numerical inverting of matrices of high order II*, Proc. Amer. Math. Soc. 2 (1951), 188-202.

[G&V] G. H. Golub and C. F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1989.

[Gol.] M. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[Gri.] D. Yu. Grigor'ev, *Additive complexity in direct computation*, Theoret. Comput. Sci. 19 (1982), 39-67.

[Har.] F. Harary, *Graph Theory*, Addison-Wesley Publishing Co. Inc., Don Mills, 1969.

[Hig.] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, 1996.

[H&H] N. J. Higham and D.J. Higham, *Large growth factors in Gaussian elimination with pivoting*, SIAM J. Matrix Anal. Appl. 10 (1989), 155-164.

[H&K] K. Hoffman and R. Kunz, *Linear Algebra*, Prentice-Hall Inc., New Jersey, 1971.

[H&J] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge Univ. Press, Cambridge, 1985.

[Kol.] L. Yu. Kolotilina, *On the sparsity of factors of the Bruhat decomposition of nonsingular matrices*, Zap. Nauchn. Sem. S.-Petersburg. Otdel. Mat. Inst. Steklov. (POMI) 202 (1992), 5-17 (in Russian). (MR 94m:15008)

[K&Y] L. Yu. Kolotilina and A. Yu. Yeremin, *Bruhat decomposition and solution of sparse linear algebraic systems*, Sov. J. Numer. Anal. Math. Modelling 2 (1987), 421-436.

[McH.] J. McHugh, *Algorithmic Graph Theory*, Prentice-Hall Inc., New Jersey, 1990.

[Pis.] S. Pissanetsky, *Sparse Matrix Technology*, Academic Press, New York, 1984.

[Smi.] D. Smith, *History of Mathematics Volume II*, Ginn and Company, Toronto, 1953.

[Ste.] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[Suz.] M. Suzuki, *Group Theory I*, Springer-Verlag Inc., New York, 1982.

[Wat.] D. S. Watkins, *Fundamentals of Matrix Computations*, J. Wiley, New York, 1991.

[Wil.63] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall Inc., New Jersey, 1963.

[Wil.65] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

[Wri.] J. Wright, *A collection of problems for which Gaussian elimination with partial pivoting is unstable*, SIAM J. Sci. Comput. 14 (1993), 231-xxx

[Y&S] L. Yuan and D. Shiran, *Chinese Mathematics a Concise History*, Oxford University Press, New York, 1987.

VITA

Surname: Odeh
Given Names: Omar Hasan
Place of Birth: Amman, Jordan
Date of Birth: May 26, 1972
Deceased: December 30, 1996

Educational Institutions Attended:

University of Victoria	1991-1996

Degrees Awarded:

B.Sc., University of Victoria	1995

Honours and Awards:

A winner of the student paper competition, Tenth Annual Pacific Northwest Numerical Analysis Seminar, Vancouver, B.C., 1996.

Publications:

O. H. Odeh, D.D. Olesky and P. van den Driessche, Bruhat Decomposition and Numerical Stability, to appear in the SIAM Journal of Matrix Analysis and Applications. (accepted December 27, 1996)

PARTIAL COPYRIGHT LICENSE

Title of Thesis:

Stability of the Bruhat Decomposition

Author: [redacted]

Omar Hasan Odeh

April 28, 1997
Date