Rounding Errors and Stability of Algorithms

Rounding Errors

Let $N_{\min}, N_{\max}$ be the floating point numbers of smallest and largest magnitude in a finite precision system.

Rounding Errors

Let $N_{\min}, N_{\max}$ be the floating point numbers of smallest and largest magnitude in a finite precision system.

For any normalized number x (which is not necessarily a floating point number in the system) such that $N_{\min} \leq |x| \leq N_{\max}$,

$$fl(x) = x(1+\delta), \quad |\delta| \leq u \qquad (*)$$

This will be called the fundamental representation/rounding rule.

Rounding Errors

Let $N_{\min}, N_{\max}$ be the floating point numbers of smallest and largest magnitude in a finite precision system.

For any normalized number x (which is not necessarily a floating point number in the system) such that $N_{\min} \leq |x| \leq N_{\max}$,

$$fl(x) = x(1+\delta), \quad |\delta| \leq u \qquad (*)$$

This will be called the fundamental representation/rounding rule.

For normalized numbers floating point numbers x, y such that $N_{\min} \leq |x \; op \; y| \leq N_{\max}$,

$$fl(x \; op \; y) = (x \; op \; y)(1+\delta), \quad |\delta| \leq u \tag{1}$$

where '*op*' is any of the operations $+, -, /, \times$ and the unit roundoff $u \approx 10^{-16}$ in IEEE double precision.

Rounding Errors

Let $N_{\min}, N_{\max}$ be the floating point numbers of smallest and largest magnitude in a finite precision system.

For any normalized number x (which is not necessarily a floating point number in the system) such that $N_{\min} \leq |x| \leq N_{\max}$,

$$fl(x) = x(1+\delta), \quad |\delta| \leq u \qquad (*)$$

This will be called the fundamental representation/rounding rule.

For normalized numbers floating point numbers x, y such that $N_{\min} \leq |x \; op \; y| \leq N_{\max}$,

$$fl(x \; op \; y) = (x \; op \; y)(1+\delta), \quad |\delta| \leq u \qquad (1)$$

where 'op' is any of the operations $+, -, /, \times$ and the unit roundoff $u \approx 10^{-16}$ in IEEE double precision.

This is desirable as it implies that the maximum relative error in the operation is of the order of unit roundoff.

Rounding errors

Do similar statements hold for normalized numbers x and y that are not floating point numbers?

Rounding errors

Do similar statements hold for normalized numbers x and y that are not floating point numbers?

Suppose, $N_{\min} \leq |x|, |y|, |x \; op \; y| \leq N_{\max}$. In such a case, x and y need to be rounded before performing the operation and

$$fl(x \; op \; y) = fl(fl(x) \; op \; fl(y)) = ((x(1+\epsilon_1)) \; op \; (y(1+\epsilon_2)))(1+\epsilon_3),$$

where $|\epsilon_j| \leq u, \; j = 1, 2, 3.$

Rounding errors

Do similar statements hold for normalized numbers x and y that are not floating point numbers?

Suppose, $N_{\min} \leq |x|, |y|, |x \; op \; y| \leq N_{\max}$. In such a case, x and y need to be rounded before performing the operation and

$$fl(x \; op \; y) = fl(fl(x) \; op \; fl(y)) = ((x(1+\epsilon_1)) \; op \; (y(1+\epsilon_2)))(1+\epsilon_3),$$

where $|\epsilon_j| \leq u, \; j = 1, 2, 3.$

The answer depends on what 'op', x and y are.

Rounding errors

Do similar statements hold for normalized numbers x and y that are not floating point numbers?

Suppose, $N_{\min} \le |x|, |y|, |x \; op \; y| \le N_{\max}$. In such a case, x and y need to be rounded before performing the operation and

$$fl(x \; op \; y) = fl(fl(x) \; op \; fl(y)) = ((x(1+\epsilon_1)) \; op \; (y(1+\epsilon_2)))(1+\epsilon_3),$$

where $|\epsilon_j| \le u, \; j = 1, 2, 3.$

The answer depends on what 'op', x and y are.

In the following

$O(u) \rightarrow$ *quantities whose absolute values are small multiples of unit roundoff* u;

$O(u^2) \rightarrow$ *quantities whose absolute values are small multiples of* u^2 *and can be ignored.*

Rounding Errors

If $op = \times$, or $op = /$,

$$fl(x \ op \ y) = (x \ op \ y)(1+\epsilon), \ |\epsilon| \le 3u + O(u^2) \tag{2}$$

Rounding Errors

If $op = \times$, or $op = /$,

$$fl(x\ op\ y) = (x\ op\ y)(1+\epsilon),\ |\epsilon| \leq 3u + O(u^2) \tag{2}$$

If $op = +$, or $op = -$,

$$fl(x\ op\ y) = (x\ op\ y)\left(1 + \left(\frac{x\epsilon_1}{x\ op\ y}\ op\ \frac{y\epsilon_2}{x\ op\ y}\right)\right), \tag{3}$$

where $|\epsilon_j| \leq 2u + O(u^2)$.

Rounding Errors

If $op = \times$, or $op = /$,

$$fl(x \; op \; y) = (x \; op \; y)(1 + \epsilon), \; |\epsilon| \leq 3u + O(u^2) \tag{2}$$

If $op = +$, or $op = -$,

$$fl(x \; op \; y) = (x \; op \; y)\left(1 + \left(\frac{x\epsilon_1}{x \; op \; y} \; op \; \frac{y\epsilon_2}{x \; op \; y}\right)\right), \tag{3}$$

where $|\epsilon_j| \leq 2u + O(u^2)$.

However $\left|\frac{x\epsilon_1}{x \; op \; y} \; op \; \frac{y\epsilon_2}{x \; op \; y}\right| \not\approx O(u)$ if $|x \; op \; y| \ll |x|$ or $|x \; op \; y| \ll |y|$.

Rounding Errors

If $op = \times$, or $op = /$,

$$fl(x\ op\ y) = (x\ op\ y)(1+\epsilon),\ |\epsilon| \leq 3u + O(u^2) \tag{2}$$

If $op = +$, or $op = -$,

$$fl(x\ op\ y) = (x\ op\ y)\left(1 + \left(\frac{x\epsilon_1}{x\ op\ y}\ op\ \frac{y\epsilon_2}{x\ op\ y}\right)\right), \tag{3}$$

where $|\epsilon_j| \leq 2u + O(u^2)$.

However $\left|\frac{x\epsilon_1}{x\ op\ y}\ op\ \frac{y\epsilon_2}{x\ op\ y}\right| \not\approx O(u)$ if $|x\ op\ y| \ll |x|$ or $|x\ op\ y| \ll |y|$.

This is called *catastrophic cancellation as it can result in sudden loss of accuracy.*

Catastrophic Cancellation

Example 1: In the finite precision system $(10, 3, -1, 2)$ with round to nearest rounding, $fl(100 - 99.95) = 0$.

Example 2: The computation $z = 1 - \sqrt{1 - x^2}$ is prone to catastrophic cancellation for $x \approx 0$. The errors magnify when the computed value is multiplied by a large number.

This may be avoided by using the equivalent formulation $x^2/(1 + \sqrt{1 - x^2})$.

x	$10^{20}(1 - \sqrt{1 - x^2})$	$10^{20}\left(x^2/(1 + \sqrt{1 - x^2})\right)$
$5.3105e - 007$	14100000	14101000
$4.7895e - 007$	11469000	11470000
$4.2684e - 007$	9114900	9109700
$3.7474e - 007$	7027700	7021400
$3.2263e - 007$	5206900	5204600
$2.7053e - 007$	3663700	3659200
$2.1842e - 007$	2387000	2385400
$1.6632e - 007$	1387800	1383000
$1.1421e - 007$	655030	652200
$6.2105e - 008$	199840	192850
$1e - 008$	11102	5000

Swamping $\rightarrow$ Catastrophic Cancellation

If $0 < fl(a) \ll fl(b)$, or $fl(b) \ll 0 < fl(a)$, then $fl(a+b) \approx fl(b)$. This is called *swamping.*

Swamping $\rightarrow$ Catastrophic Cancellation

If $0 < fl(a) \ll fl(b)$, or $fl(b) \ll 0 < fl(a)$, then $fl(a + b) \approx fl(b)$. This is called *swamping.*

Swamping can lead to *catastrophic cancellation.*

Swamping $\to$ Catastrophic Cancellation

If $0 < fl(a) \ll fl(b)$, or $fl(b) \ll 0 < fl(a)$, then $fl(a+b) \approx fl(b)$. This is called *swamping.*

Swamping can lead to *catastrophic cancellation.*

Exercise: Perform GENP on

$$A = \begin{bmatrix} 0.1 & 10 & 9.985 \\ 1 & 0.05 & 0.15 \\ 1 & 0.04 & 0.19 \end{bmatrix}$$

in the finite precision system $(10, 3, -3, 2)$ with *round to nearest* as the rounding mode. Identify the computations where

(a) swamping occurs;

(b) catastrophic cancellation occurs;

Find L and U of the computed LU decomposition and compute $\|A - LU\|_1$. Repeat the calculations for GEPP.

Backward error Analysis and Stability of Algorithms

Some background

- In the early years of computing, the approach to understanding the accuracy of solution from algorithms involved bounding the rounding error at every stage of the computations.
- Apart from being practically very difficult in the presence of many computations, it was also not very successful as the risk of catastrophic cancellation which was not always possible to predict, loomed over the computations.
- Therefore, there was a general air of pessimism about the extent of errors in solutions computed via algorithms operating in a finite precision environment.
- In the early 1960s, a radically different approach to rounding error analysis was proposed by James Wilkinson.
- His idea was that instead of looking at errors at every stage of the computation, one should look at the computed answer from the algorithm as the exact answer of the same algorithm applied to perturbed data, the perturbations arising from a pushback of the errors in the computations back into data.

Backward Errors

- These relative perturbations to the data arising from the pushback are called the backward errors of the computed answer.
- If they are of the order of unit roundoff, then the algorithm is said to be backward stable or simply stable.
- Since working in finite precision environments and consequent rounding errors are inevitable for algorithms, the most that should be expected of them is that they are backward stable.
- After this, the accuracy of the computed solution depends upon the sensitivity of the problem to small changes in the data.
- The backward error analysis of the algorithm is combined with the sensitivity analysis of the problem to bound the relative errors in the solution.
- This approach separates the properties of the algorithm from those of the problem.
- To contrast with backward errors, the usual errors in computations are also referred to as forward errors.

Backward error analysis

James Hardy Wilkinson, FRS
(1919-1986)

Systematic analysis of backward errors and backward stability was introduced by James Wilkinson.

Posteriori versus priori backward error analysis of GE

- The analysis of backward stability of algorithms is an analysis of the backward errors in the computations.
- For Gaussian Elimination one such (posteriori) analysis was already undertaken *after* solving a system of equations and using the residual vector associated with the computed solution to construct perturbed systems of equations of which the computed solution is an exact solution.
- This approach depends on the computed solution and can only say whether the algorithm is backward stable with respect to the given problem.
- To know about the backward stability property of an algorithm in general before using it to solve a problem a *priori backward error analysis* is required.

A priori backward error analysis

- This analysis is made without actually using the algorithm to solve a problem.
- It involves estimating maximum possible backward errors that can arise in standard arithmetic operations as an initial step.
- These are used to estimate maximum possible backward errors in the computed answer arising from the collective backward errors in the computations.
- The algorithm is declared to be backward stable if these maximum possible values of backward errors are $O(u)$.
- As the analysis should hold for any problem that the algorithm is designed to solve, it cannot make assumptions about the input data and computed solution.
- However, sometimes these backward errors are guranteed to be of the order of unit roundoff only under certain conditions. In such cases the algorithm is *conditionally backward stable.*
- If no such conditions are required for backward stability, then the algorithm is *unconditionally backward stable.*

Backward errors in standard arithmetic operations

$$\frac{|fl(x \; op \; y) - (x \; op \; y)|}{|x \; op \; y|} \rightarrow \text{ relative (forward) error in } (x \; op \; y).$$

Backward errors in standard arithmetic operations

$$\frac{|fl(x \; op \; y) - (x \; op \; y)|}{|x \; op \; y|} \rightarrow \text{ relative (forward) error in } (x \; op \; y).$$

In contrast if

$$fl(x \; op \; y) = \hat{x} \; op \; \hat{y}$$

for some $\hat{x}, \hat{y}$, then,

$$\frac{|\hat{x} - x|}{|x|}, \frac{|\hat{y} - y|}{|y|} \rightarrow \text{ relative backward errors in } (x \; op \; y).$$

Backward errors in standard arithmetic operations

$$\frac{|fl(x\ op\ y) - (x\ op\ y)|}{|x\ op\ y|} \rightarrow \text{ relative (forward) error in } (x\ op\ y).$$

In contrast if

$$fl(x\ op\ y) = \hat{x}\ op\ \hat{y}$$

for some $\hat{x}, \hat{y}$, then,

$$\frac{|\hat{x} - x|}{|x|}, \frac{|\hat{y} - y|}{|y|} \rightarrow \text{ relative backward errors in } (x\ op\ y).$$

The operation *op* is said to be *backward stable* if

$$\frac{|\hat{x} - x|}{|x|} \text{ and } \frac{|\hat{y} - y|}{|y|} \text{ are } O(u).$$

Backward errors in standard arithmetic operations

Since the relative representation error that arises when rounding normalized numbers with absolute values between $N_{\min}$ and $N_{\max}$ is very small, these may be ignored in the push back.

Backward errors in standard arithmetic operations

Since the relative representation error that arises when rounding normalized numbers with absolute values between N_{min} and N_{max} is very small, these may be ignored in the push back.

Therefore in backward error analysis it is assumed that all the standard arithmetic operations occur between floating point numbers.

Backward errors in standard arithmetic operations

Since the relative representation error that arises when rounding normalized numbers with absolute values between N_{min} and N_{max} is very small, these may be ignored in the push back.

Therefore in backward error analysis it is assumed that all the standard arithmetic operations occur between floating point numbers.

Theorem Let x, y be floating point numbers. Whenever *op* is any of the standard operations $+, -, /, \times$,

$$fl(x \; op \; y) = \hat{x} \; op \; \hat{y}$$

where $\hat{x} = x(1+\delta_1)$ and $\hat{y} = y(1+\delta_2)$ with $|\delta_j| \leq u$, for $j = 1, 2$.

Backward errors in standard arithmetic operations

Since the relative representation error that arises when rounding normalized numbers with absolute values between N_{min} and N_{max} is very small, these may be ignored in the push back.

Therefore in backward error analysis it is assumed that all the standard arithmetic operations occur between floating point numbers.

Theorem Let x, y be floating point numbers. Whenever *op* is any of the standard operations $+, -, /, \times$,

$$fl(x \; op \; y) = \hat{x} \; op \; \hat{y}$$

where $\hat{x} = x(1 + \delta_1)$ and $\hat{y} = y(1 + \delta_2)$ with $|\delta_j| \leq u$, for $j = 1, 2$.

Hence all the standard arithmetic operations between two floating point numbers are backward stable operations.

A fundamental result in backward error analysis

Theorem Let $w_i, i = 1, \ldots, n$, be floating point numbers. Then there exist $\gamma_i, i = 1, \ldots, n$, satisfying $|\gamma_i| \leq (n-1)u + O(u^2)$, such that

$$fl\left(\sum_{i=1}^{n} w_i\right) = \sum_{i=1}^{n} w_i(1+\gamma_i). \tag{4}$$

irrespective of the order of summation.

A fundamental result in backward error analysis

Theorem Let $w_i, i = 1, \ldots, n$, be floating point numbers. Then there exist $\gamma_i, i = 1, \ldots, n$, satisfying $|\gamma_i| \leq (n-1)u + O(u^2)$, such that

$$fl\left(\sum_{i=1}^{n} w_i\right) = \sum_{i=1}^{n} w_i(1+\gamma_i). \tag{4}$$

irrespective of the order of summation.

Corollary Let $u_i, w_i, i = 1, \ldots, n$, be floating point numbers. Then there exist $\gamma_i, i = 1, \ldots, n$, satisfying $|\gamma_i| \leq nu + O(u^2)$, such that

$$fl\left(\sum_{i=1}^{n} u_i w_i\right) = \sum_{i=1}^{n} u_i w_i(1+\gamma_i). \tag{5}$$

irrespective of the order of summation.

Backward Error Analysis of Gaussian Elimination
&
Cholesky Decomposition Methods

Backward error analysis of backward and forward substitution

Some notation:

- For any two matrices $C = [c_{ij}]_{n\times m}$ and $F = [f_{ij}]_{n\times m}$, we write $C \leq F$ if $c_{ij} \leq f_{ij}$ for all $1 \leq i \leq n$, $1 \leq j \leq m$.
- For any $C = [c_{ij}]_{n\times m}$, $|C| = [|c_{ij}|]_{n\times m}$.

Backward error analysis of backward and forward substitution

Some notation:

- For any two matrices $C = [c_{ij}]_{n\times m}$ and $F = [f_{ij}]_{n\times m}$, we write $C \le F$ if $c_{ij} \le f_{ij}$ for all $1 \le i \le n$, $1 \le j \le m$.
- For any $C = [c_{ij}]_{n\times m}$, $|C| = [|c_{ij}|]_{n\times m}$.

Theorem Let G be any nonsingular lower (upper) triangular $n \times n$ matrix and b be any nonzero column vector of length n. If y_c be the computed solution of the system $Gy = b$ using any variant of forward (backward) substitution in floating point arithmetic, then y_c satisfies

$$(G + \delta G)y_c = b \tag{6}$$

where δG is lower (upper) triangular such that

$$|\delta G| \le 2nu|G| + O(u^2) \tag{7}$$

Backward error analysis of backward and forward substitution

Some notation:

- For any two matrices $C = [c_{ij}]_{n\times m}$ and $F = [f_{ij}]_{n\times m}$, we write $C \leq F$ if $c_{ij} \leq f_{ij}$ for all $1 \leq i \leq n$, $1 \leq j \leq m$.
- For any $C = [c_{ij}]_{n\times m}$, $|C| = [|c_{ij}|]_{n\times m}$.

Theorem Let G be any nonsingular lower (upper) triangular $n \times n$ matrix and b be any nonzero column vector of length n. If y_c be the computed solution of the system $Gy = b$ using any variant of forward (backward) substitution in floating point arithmetic, then y_c satisfies

$$(G + \delta G)y_c = b \tag{6}$$

where δG is lower (upper) triangular such that

$$|\delta G| \leq 2nu|G| + O(u^2) \tag{7}$$

Here (7) means $|\delta g_{ij}| \leq 2nu|g_{ij}| + O(u^2)$ for all $1 \leq i, j \leq n$ and $O(u^2)$ stands for a matrix whose entries are $O(u^2)$.

Backward error analysis of backward and forward substitution

Some notation:

- For any two matrices $C = [c_{ij}]_{n\times m}$ and $F = [f_{ij}]_{n\times m}$, we write $C \leq F$ if $c_{ij} \leq f_{ij}$ for all $1 \leq i \leq n$, $1 \leq j \leq m$.
- For any $C = [c_{ij}]_{n\times m}$, $|C| = [|c_{ij}|]_{n\times m}$.

Theorem Let G be any nonsingular lower (upper) triangular $n \times n$ matrix and b be any nonzero column vector of length n. If y_c be the computed solution of the system $Gy = b$ using any variant of forward (backward) substitution in floating point arithmetic, then y_c satisfies

$$(G + \delta G)y_c = b \tag{6}$$

where δG is lower (upper) triangular such that

$$|\delta G| \leq 2nu|G| + O(u^2) \tag{7}$$

Here (7) means $|\delta g_{ij}| \leq 2nu|g_{ij}| + O(u^2)$ for all $1 \leq i, j \leq n$ and $O(u^2)$ stands for a matrix whose entries are $O(u^2)$.

For a proof see pp. 159-161 of Fundamentals of Matrix Computations, 2nd Edition.

Backward error analysis of backward and forward substitution

Exercise Given any two matrices $C = [c_{ij}]_{n\times m}$ and $F = [f_{ij}]_{n\times m}$, show that

1. $|C| \leq |F| \Rightarrow \|C\|_p \leq \|F\|_p$ where $p = 1, F, \infty$.
2. $\|C\|_p = \||C|\|_p$ for $p = 1, F, \infty$.

Backward error analysis of backward and forward substitution

Exercise Given any two matrices $C = [c_{ij}]_{n \times m}$ and $F = [f_{ij}]_{n \times m}$, show that

1. $|C| \leq |F| \Rightarrow \|C\|_p \leq \|F\|_p$ where $p = 1, F, \infty$.
2. $\|C\|_p = \||C|\|_p$ for $p = 1, F, \infty$.

Corollary Let G be any nonsingular lower (upper) triangular $n \times n$ matrix and b be any nonzero column vector of length n. If y_c be the computed solution of the system $Gy = b$ using any variant of forward (backward) substitution in floating point arithmetic, then y_c satisfies

$$(G + \delta G)y_c = b$$

where δG is lower (upper) triangular such that

$$\|\delta G\|_\infty \leq 2nu\|G\|_\infty + O(u^2) \tag{8}$$

Backward error analysis of backward and forward substitution

Exercise Given any two matrices $C = [c_{ij}]_{n\times m}$ and $F = [f_{ij}]_{n\times m}$, show that

1. $|C| \leq |F| \Rightarrow \|C\|_p \leq \|F\|_p$ where $p = 1, F, \infty$.
2. $\|C\|_p = \||C|\|_p$ for $p = 1, F, \infty$.

Corollary Let G be any nonsingular lower (upper) triangular $n \times n$ matrix and b be any nonzero column vector of length n. If y_c be the computed solution of the system $Gy = b$ using any variant of forward (backward) substitution in floating point arithmetic, then y_c satisfies

$$(G + \delta G)y_c = b$$

where δG is lower (upper) triangular such that

$$\|\delta G\|_\infty \leq 2nu\|G\|_\infty + O(u^2) \tag{8}$$

Therefore backward and forward substitution processes are unconditionally backward stable.

Backward error analysis of Gaussian Elimination

Theorem Let A be an $n \times n$ matrix. Let L_c and U_c be the LU factors of A computed via Gaussian Elimination in floating point arithmetic. If no zero pivots were encountered in the process, then

$$A + \delta A = L_c U_c$$

where

$$|\delta A| \leq 2nu|L_c||U_c| + O(u^2) \tag{9}$$

and consequently,

$$\|\delta A\|_\infty \leq 2nu\|L_c\|_\infty \|U_c\|_\infty + O(u^2) \tag{10}$$

Backward error analysis of Gaussian Elimination

Theorem Let A be an $n \times n$ matrix. Let L_c and U_c be the LU factors of A computed via Gaussian Elimination in floating point arithmetic. If no zero pivots were encountered in the process, then

$$A + \delta A = L_c U_c$$

where

$$|\delta A| \leq 2nu|L_c||U_c| + O(u^2) \tag{9}$$

and consequently,

$$\|\delta A\|_\infty \leq 2nu\|L_c\|_\infty \|U_c\|_\infty + O(u^2) \tag{10}$$

Further if x_c be the computed solution of $Ax = b$, $b \neq 0$, obtained by solving lower and upper triangular systems with L_c and U_c as coefficient matrices via forward and backward substitution respectively, then

$$(A + E)x_c = b$$

where

$$|E| \leq 6nu|L_c||U_c| + O(u^2) \tag{11}$$

and consequently,

$$\|E\|_\infty \leq 6nu\|L_c\|_\infty \|U_c\|_\infty + O(u^2) \tag{12}$$

Backward error analysis of Gaussian Elimination

Therefore the process is backward stable if

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty}$$

is small.

Backward error analysis of Gaussian Elimination

Therefore the process is backward stable if

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty}$$

is small.

In GENP, the presence of small pivots can make $\|L_c\|_\infty$ very large. Therefore, GENP is unstable.

Backward error analysis of Gaussian Elimination

Therefore the process is backward stable if

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty}$$

is small.

In GENP, the presence of small pivots can make $\|L_c\|_\infty$ very large. Therefore, GENP is unstable.

In GEPP and GECP, $\|L_c\|_\infty \leq n$. Therefore,

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty} \leq n \frac{\|U_c\|_\infty}{\|A\|_\infty}.$$

Backward error analysis of Gaussian Elimination

Therefore the process is backward stable if

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty}$$

is small.

In GENP, the presence of small pivots can make $\|L_c\|_\infty$ very large. Therefore, GENP is unstable.

In GEPP and GECP, $\|L_c\|_\infty \leq n$. Therefore,

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty} \leq n \frac{\|U_c\|_\infty}{\|A\|_\infty}.$$

For GEPP it is observed that for almost all matrices in practice, $\frac{\|U_c\|_\infty}{\|A\|_\infty} \approx \sqrt{n}$.

Backward error analysis of Gaussian Elimination

Therefore the process is backward stable if

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty}$$

is small.

In GENP, the presence of small pivots can make $\|L_c\|_\infty$ very large. Therefore, GENP is unstable.

In GEPP and GECP, $\|L_c\|_\infty \leq n$. Therefore,

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty} \leq n \frac{\|U_c\|_\infty}{\|A\|_\infty}.$$

For GEPP it is observed that for almost all matrices in practice, $\frac{\|U_c\|_\infty}{\|A\|_\infty} \approx \sqrt{n}$.

However, $\frac{\|U_c\|_\infty}{\|A\|_\infty} = \frac{2^{n-1}}{n}$ when A is the $n \times n$ Wilkinson matrix in GEPP.

Backward error analysis of Gaussian Elimination

Therefore the process is backward stable if

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty}$$

is small.

In GENP, the presence of small pivots can make $\|L_c\|_\infty$ very large. Therefore, GENP is unstable.

In GEPP and GECP, $\|L_c\|_\infty \leq n$. Therefore,

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty} \leq n \frac{\|U_c\|_\infty}{\|A\|_\infty}.$$

For GEPP it is observed that for almost all matrices in practice, $\frac{\|U_c\|_\infty}{\|A\|_\infty} \approx \sqrt{n}$.

However, $\frac{\|U_c\|_\infty}{\|A\|_\infty} = \frac{2^{n-1}}{n}$ when A is the $n \times n$ Wilkinson matrix in GEPP.

Therefore GEPP is conditionally backward stable as stabiity is subject to $\frac{\|U_c\|_\infty}{\|A\|_\infty}$ being small.

Backward error analysis of Gaussian Elimination

Therefore the process is backward stable if

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty}$$

is small.

In GENP, the presence of small pivots can make $\|L_c\|_\infty$ very large. Therefore, GENP is unstable.

In GEPP and GECP, $\|L_c\|_\infty \leq n$. Therefore,

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty} \leq n \frac{\|U_c\|_\infty}{\|A\|_\infty}.$$

For GEPP it is observed that for almost all matrices in practice, $\frac{\|U_c\|_\infty}{\|A\|_\infty} \approx \sqrt{n}$.

However, $\frac{\|U_c\|_\infty}{\|A\|_\infty} = \frac{2^{n-1}}{n}$ when A is the $n \times n$ Wilkinson matrix in GEPP.

Therefore GEPP is conditionally backward stable as stabiity is subject to $\frac{\|U_c\|_\infty}{\|A\|_\infty}$ being small.

For GECP it can be proved that $\frac{\|U_c\|_\infty}{\|A\|_\infty} = O(n)$.

Backward error analysis of Gaussian Elimination

Therefore the process is backward stable if

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty}$$

is small.

In GENP, the presence of small pivots can make $\|L_c\|_\infty$ very large. Therefore, GENP is unstable.

In GEPP and GECP, $\|L_c\|_\infty \leq n$. Therefore,

$$\frac{\|L_c\|_\infty \|U_c\|_\infty}{\|A\|_\infty} \leq n \frac{\|U_c\|_\infty}{\|A\|_\infty}.$$

For GEPP it is observed that for almost all matrices in practice, $\frac{\|U_c\|_\infty}{\|A\|_\infty} \approx \sqrt{n}$.

However, $\frac{\|U_c\|_\infty}{\|A\|_\infty} = \frac{2^{n-1}}{n}$ when A is the $n \times n$ Wilkinson matrix in GEPP.

Therefore GEPP is conditionally backward stable as stabiity is subject to $\frac{\|U_c\|_\infty}{\|A\|_\infty}$ *being small.*

For GECP it can be proved that $\frac{\|U_c\|_\infty}{\|A\|_\infty} = O(n)$.

Hence GECP is unconditionally backward stable.

GEPP: The algorithm of choice for solving square systems

However despite the verdict of conditional backward stability on GEPP, it remains the algorithm of choice when it comes to direct methods for solving square system of equations that are not too large and the coefficient matrix is not positive definite and not sparse, that is most of its entries are nonzero.

This is because it is cheaper than GECP and it has excellent stability properties for almost all matrices.

Therefore the MATLAB command $\mathtt{A} \setminus \mathtt{b}$ runs GEPP to solve $Ax = b$ when A is not too large, not sparse and not positive definite.

A complete understanding of the reasons for the backward stability of GEPP in almost all cases is an open problem.

Backward error analysis of Cholesky factorization

Theorem Let A be an $n \times n$ positive definite matrix and G_c be the Cholesky factor computed in floating point arithmetic via some version of Cholesky factorization algorithm. Then

$$A + \delta A = G_c^T G_c$$

where $|\delta A| \leq 2nu|G_c^T||G_c| + O(u^2)$.

Backward error analysis of Cholesky factorization

Theorem Let A be an $n \times n$ positive definite matrix and G_c be the Cholesky factor computed in floating point arithmetic via some version of Cholesky factorization algorithm. Then

$$A + \delta A = G_c^T G_c$$

where $|\delta A| \leq 2nu|G_c^T||G_c| + O(u^2)$.

Consequently,

$$\frac{\|\delta A\|_F}{\|A\|_F} \leq \frac{2n^{3/2}u}{1 - 2n^{3/2}u} + O(u^2).$$

Backward error analysis of Cholesky factorization

Theorem Let A be an $n \times n$ positive definite matrix and G_c be the Cholesky factor computed in floating point arithmetic via some version of Cholesky factorization algorithm. Then

$$A + \delta A = G_c^T G_c$$

where $|\delta A| \leq 2nu|G_c^T||G_c| + O(u^2)$.

Consequently,

$$\frac{\|\delta A\|_F}{\|A\|_F} \leq \frac{2n^{3/2}u}{1 - 2n^{3/2}u} + O(u^2).$$

Thus, the algorithms for finding a Cholesky factor of a positive definite matrix and using it to solve a system of equations is unconditionally backward stable.

Backward error analysis of Cholesky factorization

Theorem Let A be an $n \times n$ positive definite matrix and G_c be the Cholesky factor computed in floating point arithmetic via some version of Cholesky factorization algorithm. Then

$$A + \delta A = G_c^T G_c$$

where $|\delta A| \leq 2nu|G_c^T||G_c| + O(u^2)$.

Consequently,

$$\frac{\|\delta A\|_F}{\|A\|_F} \leq \frac{2n^{3/2}u}{1 - 2n^{3/2}u} + O(u^2).$$

Thus, the algorithms for finding a Cholesky factor of a positive definite matrix and using it to solve a system of equations is unconditionally backward stable.

Consequently, the solutions of a system of equations with a positive definite coefficient matrix is unconditionaly backward stable.