

Reivindicando o método de Gauss

M. Brozos Vázquez¹, A. B. Rodríguez Raposo²

¹Dpto. de Matemáticas, Universidade da Coruña, Escola Politécnica Superior, Ferrol,
mbrozos@udc.es

²Dpto. de Matemáticas, Universidade da Coruña, Escola Politécnica Superior, Ferrol,
abraposo@edu.xunta.es

Resumo

Nesta charla imos estudar o método de Gauss cos seguintes obxectivos:

- Entender a importancia teórica que ten o método no teorema de Rouché-Frobenius e na clasificación de sistemas de ecuacións lineares.
- Estudar a eficacia da diagonalización por eliminación gaussiana previa ó cálculo de determinantes.
- Comparar a súa eficiencia coa da regra de Cramer para resolver sistemas compatibles determinados, e coa do método do cálculo da inversa dunha matriz por menores.

Para realizar este estudio apoiarémonos en exemplos concretos e en cálculos realizados con Maxima.

PALABRAS CHAVE: Sistemas de ecuacións lineares, determinante dunha matriz, método de Gauss, regra de Cramer.

Abstract

The aim of this communication is to study the Gaussian elimination method with the following specific tasks:

- To understand the theoretical meaning of this method in relation with the Rouché-Frobenius theorem and the classification of linear equation systems.
- To study the effectiveness of the Gaussian elimination prior to the determinant computation.
- To compare the efficiency of the Cramer rule vs. the Gaussian elimination method when dealing with determinate systems, or the efficiency of the Gaussian elimination when computing the inverse of a matrix.

In order to illustrate the previous analysis, we provide examples computed with Maxima.

KEY WORDS: linear equation systems, determinant of a matrix, Gaussian elimination method, Cramer rule.

1. Introducción

O método de Gauss é o método máis utilizado para resolver sistemas de ecuacións lineares e outros problemas propios da álgebra linear. Sen embargo, dende o ámbito educativo, quizais non o estudiamos coa profundidade que se merece. Proba disto é que resulta moi frecuente que o alumnado de primeiros cursos de carreiras técnicas prefiran utilizar a regra de Cramer fronte ó método de Gauss, sen ter en conta a maior eficiencia computacional deste último ou a maior simplicidade dos cálculos a realizar, principalmente en sistemas de certo tamaño.

Comezaremos presentando un estudio somero do número de operacións necesarias para realizar o cálculo dun determinante por menores, e comparáremolo co número de operacións que supón o seu cálculo se aplicamos previamente o método de redución de Gauss. Para isto utilizaremos o concepto de complexidade dun algoritmo e presentaremos o estudio dende un punto de vista computacional. Unha vez feito este estudio, adentrarémonos na importancia teórica do método de Gauss e veremos como certos problemas da Álgebra linear, que tradicionalmente se estudian facendo uso do cálculo de determinantes por menores, poden ser resoltos unicamente coa aplicación do método de Gauss. Exemplos disto son a resolución de sistemas de ecuacións lineares, incluíndo o teorema de Rouché-Frobenius, ou o cálculo da matriz inversa.

2. Sobre a complexidade

Un algoritmo non é máis que unha colección de instrucións ordenadas que se deben seguir para resolver un problema determinado. Hai moitos problemas en Matemáticas que se resolven algorítmicamente, como por exemplo factorizar un número nos seus factores primos. Porén, moitas veces os algoritmos dos que dispoñemos non son útiles na práctica. Seguindo co problema da factorización, sabemos que temos un algoritmo infalible para levala a cabo. Sen embargo, non resulta ser eficiente: se nos dan un número de varios centos de cifras, levaríanos moitos anos atopar a súa factorización, incluso se podemos utilizar un ordenador realmente potente. Así, aínda que na teoría teñamos moitos algoritmos que resolven problemas, debemos saber dilucidar cales nos permitan resolver problemas na práctica.

Para ser quen de distinguir os algoritmos eficientes, primeiro debemos saber que entendemos por ser eficiente. Supoñamos que temos un problema $P(n)$, que depende de un número $n \in \mathbb{N}$, por exemplo, o problema de factorizar un número nos seus factores primos. Cando deseñemos un algoritmo para resolver o noso problema, este algoritmo dependerá de n , que será o que chamaremos tamaño do problema. O noso algoritmo será eficiente se nos dá unha solución nun tempo razoablemente curto para un tamaño moi grande. Este tempo virá dado por unha función $T(n)$, que na práctica dependerá da implementación que fagamos do algoritmo, e do problema en particular que queiramos resolver. Así, factorizar o número 10^{100} resulta moi doado, a pesar de que se trata de un número moi grande. Co cal, para calcular a eficiencia dun algoritmo, o que faremos será supoñer que estamos no peor dos casos, é dicir, que sempre imos ter que realizar o máximo número de operacións necesarias. Este número máximo de operacións necesarias, que depende de n , é o que chamaremos complexidade do algoritmo. Deste xeito, cantas máis operacións requiramos no noso algoritmo, maior complexidade terá, e polo tanto maior será o seu tempo de execución. Cando falamos de operacións referímonos non só a operacións matemáticas, senón tamén a operacións lóxicas, asignación de valores das variables, comparacións e chamadas a elementos dunha matriz. As operacións matemáticas que consideraremos como elementais para o cálculo da complexidade son a suma, a resta, a multiplicación e a división, é dicir, as que están implementadas na máquina que imos utilizar.

Un caso de algoritmos que poden resultar extremadamente complexos son os algoritmos recursivos. Todos temos unha idea intuitiva do que significa a recursividade. Sen ir máis lonxe, os axiomas de Peano dan unha definición recursiva dos números naturais: defínese o primeiro elemento, e os demais obtéñense mediante unha función $f: \mathbb{N} \rightarrow \mathbb{N}$ tal que $f(n) = n + 1$, é dicir, di cal é o número seguinte a outro dado. Unha función recursiva segue unha idea similar: defínese o valor da función no primeiro elemento (ou nos primeiros elementos), e para definila nun elemento calquera recórrese ó elemento anterior. Por exemplo, a sucesión de Fibonacci é unha sucesión recursiva, xa que:

$$\begin{aligned} a_0 &= 1; & a_1 &= 1 \\ a_n &= a_{n-1} + a_{n-2}, & n &\geq 2 \end{aligned}$$

Outro exemplo de función recursiva é o cálculo do factorial de un número, posto que:

$$1! = 1$$
$$n! = n(n-1)!$$

O pseudocódigo para implementar o cálculo do factorial dun número pode ser:

```
funcion factorial (n)
se n = 1
  factorial(n) = 1
se non
  factorial(n) = n* factorial(n-1)
fin se
fin
```

Obsérvese que dentro da función se fai unha chamada á propia función, o cal é a clave da recursividade. Se analizamos a complexidade deste algoritmo, é doado ver que precisaremos da orde de $n-1$ operacións para realizar o cálculo que pretendemos.

3. O cálculo dun determinante por menores

Analicemos agora o algoritmo do cálculo dun determinate por menores, onde suporemos que temos unha función implementada que nos da o tamaño da matriz:

```
funcion detm(A) (desenvolvemos o determinate polos menores da 1ª columna)
n = tamaño(A)
se n = 1
  detm(A) = A
se non sum = 0
  para i=1,n (indica a fila)
    para k=1,i-1(calculamos a adxunta correspondente)
      para j=1,n-1
        B[k,j] = A[k,j+ 1]
      fin para
    fin para
    para k=i,n-1(calculamos a adxunta correspondente)
      para j=1,n-1
        B[k,j] = A[k+ 1,j+ 1]
      fin para
    fin para
    sum = sum + (-1)i+1 A[i,1]*detm(B)
  fin para
  detm(A) = sum
fin se
fin
```

Vemos que neste pseudocódigo existe unha chamada recursiva á función detm, que ademais está dentro dun bucle para de lonxitude n , co cal podemos concluír que a complexidade deste algoritmo é de orde aproximadamente $n!$. Se a matriz coa que operamos é de pequeno tamaño, non hai problema. Sen embargo, que sucede cando queremos calcular o determinante dunha matriz de tamaño 20x20? Pois que o número de operacións a realizar será da orde de 20!, e este xa comeza a ser un número francamente pouco manexable.

Esta cantidade desmesurada pódese reducir significativamente se calculamos o determinante dunha matriz triangular. É sabido que se $A=(a_{ij}), 1 \leq i, j \leq n$ é unha matriz triangular, tense que

$$\det(A) = \prod_{i=1, n} a_{ii}$$

o que reduce considerablemente o cálculo do determinante, xa que agora o algoritmo correspondente é:

```
funcion dett(A)
  n = tamaño(A)
  dett(A)=1
  para i=1,n
    dett(A) = A[i,i]*dett(A)
  fin para
fin
```

O cal nos dá un algoritmo de complexidade de orde $2n$. Entón o noso obxectivo debe ser transformar calquera matriz cadrada nunha matriz triangular, sen que estas transformacións alteren o valor do determinante nin aumenten excesivamente a complexidade do algoritmo. E aquí é onde entra en xogo o método de redución de Gauss.

4. Operacións elementais e cálculo do determinante

O determinante dunha matriz é unha aplicación multilinear

$$\det : (v_1, \dots, v_n) \in \mathbb{R}^n \times \dots \times \mathbb{R}^n \rightarrow \det(v_1, \dots, v_n) \in \mathbb{R},$$

onde os vectores v_1, \dots, v_n poden ser interpretados como as filas (ou as columnas) dunha matriz A . A este conxunto de vectores $\{v_1, \dots, v_n\}$ podemoslle aplicar varias transformacións que fan que a súa dependencia ou independencia linear permaneza invariante. A estas transformacións chamáremoslles *operacións elementais*:

- E1:** Intercambiar dous vectores de sitio.
- E2:** Substituír un vector pola suma del mesmo con outro do conxunto.
- E3:** Multiplicar un vector por un número distinto de cero.
- E4:** Realizar calquera combinación das operacións anteriores.

Se interpretamos os vectores $\{v_1, \dots, v_n\}$ como as filas dunha matriz A , as operacións **E1 - E4** pódense codificar mediante unha matriz que multiplica a A pola esquerda, e que depende do tipo de operación elemental que se realice. A esta matriz chamáremola *matriz elemental*. Para obter estas matrices elementais simplemente temos que considerar a matriz identidade de tamaño n e realizar a operación elemental correspondente. Así:

ME1: A matriz elemental que intercambia as filas i -ésima e j -ésima é a matriz identidade na que se intercambiaron as filas i e j . Chamáremoslle $E_{i,j}$.

ME2: A matriz elemental que substitúe a fila i -ésima pola suma dela coa fila j -ésima é o resultado de realizar esta operación na matriz identidade. Chamáremoslle E_{i+j} .

ME3: A matriz elemental que multiplica a fila i -ésima por un número $\alpha \in \mathbb{R}$ distinto de cero é a matriz identidade á que se lle aplica esta operación, é dicir, a matriz que ten uns na diagonal agás no lugar i,i , no que ten o número α , e o resto son ceros. A esta matriz chamámoslle $E_{\alpha i}$.

ME4: Para obter a matriz elemental correspondente a calquera combinación das anteriores, é suficiente con multiplicar de dereita a esquerda as matrices elementais correspondentes na orde na

que se deben realizar as operacións. En particular, se combinamos as operacións E3 e E2 (E3-2), obtemos unha matriz elemental que resulta de multiplicar a fila j -ésima por un número α e sumarlle á fila i -ésima, co que obtemos a matriz $E_{i+\alpha j}$.

Se queremos realizar transformacións elementais nunha matriz para convertila nunha matriz triangular, é suficiente con que apliquemos as transformacións E1 e E3-2 de maneira reiterada segundo o seguinte pseudocódigo:

```

funcion redgauss(A)
  n=tamaño(A)
  para i=1,n
    se A[i,i]=0 entón (buscamos un elemento da matriz non nulo que sirva de pivote)
      k=i
      facer
        k=k+1
      ata que A[k,i]≠0
      se i<k<n+1(se a columna ten algún elemento non nulo)
        para j=i,n (intercambiamos a fila i-ésima coa fila na que A[k,i]≠0)
          b=A[i,j]
          A[i,j]=A[k,j]
          A[k,j]=b
        fin para
      fin se
    fin se
  se A[i,i]≠0 entón
    para j=i+1,n (agora A[i,i]≠0, co que xa podemos facer a eliminación)
      piv=A[j,i]/A[i,i]
      para k=i,n (neste bucle facemos  $A = E_{i-\frac{A[j,i]}{A[i,i]}j}$ )
        A[j,k]=A[j,k]-piv*A[i,k]
      fin para
    fin para
  fin se
fin para
fin

```

A matriz A' que obtemos despois de realizar estas transformacións é unha matriz triangular superior, e o seu determinante calcularase mediante o produto dos elementos da diagonal. As operacións para obter esta nova matriz pódense escribir mediante as operacións $A' = E^1 \cdot \dots \cdot E^k \cdot A$, sendo as matrices E^i do tipo ME1 e ME3-2, é dicir, matrices de intercambio de filas e matrices que substitúen unha fila pola suma dela máis outra multiplicada por un número. Como $\det(E_{i,j}) = -1$ e $\det(E_{i+\alpha j}) = 1$ verificase a seguinte igualdade: $|\det(A)| = |\det(A')|$. Así, se facemos r cambios de filas (é dicir, hai r matrices do tipo $E_{i,j}$), teremos que $\det(A) = (-1)^r \det(A')$.

Este algoritmo de eliminación de Gauss ten unha orde de n^3 , e obtemos unha matriz triangular superior. Se agora lle aplicamos o algoritmo dett, obteremos unha complexidade da orde de $2n^3$, o cal é indubidablemente máis rápido que o cálculo dun determinante utilizando o desenvolvemento por menores para valores de n suficientemente grandes. Á hora de aplicar este método para obter o valor dun determinante, debemos ter en conta o número de veces que se realiza un cambio de filas, o cal se pode facer incluíndo no bucle de intercambio de filas un cambio de signo.

No Apéndice presentamos unha implementación do cálculo dun determinante por menores e do cálculo dun determinante utilizando o método de eliminación de Gauss no programa Maxima, así como unha táboa dos tempos de computación en función do tamaño da matriz.

5. Por que preferimos o método de Gauss?

Preferimos o método de Gauss, porque é máis rápido e máis sinxelo. Pero tamén o preferimos por outras razóns teóricas que expoñemos a continuación.

Supoñamos que temos un sistema compatible determinado $AX=B$. Nos cursos de bacharelato, e nos primeiros cursos das enxeñerías, é habitual explicar dous métodos que permiten resolver este sistema: o método de Gauss e a regra de Cramer.

Para resolver un sistema polo método de Cramer temos que calcular o determinante da matriz A , que será non nulo por ser o sistema compatible determinado. Para calcular o valor de cada incógnita obteremos o valor do determinante da matriz A na que substituímos a columna correspondente polo vector B de termos independentes. Multiplicando este valor polo inverso do determinante de A obtemos o valor da incógnita. Así, temos que calcular $n+1$ determinantes de orde n . Aínda que mnemotécnicamente a regra de Cramer é moi sinxela, por involucrar cálculos de determinantes resulta moi custosa computacionalmente. Sen embargo, para resolver un sistema polo método de Gauss, o que faremos será aplicar o método de redución de Gauss á matriz ampliada do sistema $(A|B)$, e unha vez que obteñamos unha matriz triangular superior seremos quen de despexar as incógnitas de maneira ascendente, o cal supón engadirlle ó método de eliminación de Gauss da orde de n^2 operacións, o cal non supón un incremento moi grande do custo computacional.

5.1 Teorema de Rouché-Frobenius

Á hora de resolver un sistema mediante a regra de Cramer, o primeiro que debemos facer é saber se o sistema é compatible determinado ou non, xa que se non o é non a podemos usar. Porén, o método de Gauss permítenos analizar a compatibilidade dun sistema sen máis que obter unha matriz escalonada. Incluso nos permite demostrar o teorema de Rouché-Frobenius sen necesidade de recurrir a conceptos de álgebra linear como o de dependencia e independencia linear, base dun espacio vectorial e dimensión. Recordemos o teorema de Rouché-Frobenius, e a continuación veremos como usamos o método de Gauss para dar unha proba.

Teorema. Sexa $AX=B$ un sistema de m ecuacións lineares con n incógnitas. O sistema é

- incompatible se $\text{rango } A < \text{rango } (A|B)$,
- compatible se $\text{rango } A = \text{rango } (A|B)$, sendo
 - determinado se $\text{rango } A = \text{rango } (A|B) = n$,
 - indeterminado se $\text{rango } A = \text{rango } (A|B) < n$.

Para comprender este teorema debemos recurrir a un concepto fundamental da álgebra linear, que é o concepto de vectores linearmente independentes, que nos permite definir o rango dunha matriz. Definimos o rango dunha matriz como o número de filas linearmente independentes que ten, que coincide co número de columnas linearmente independentes. Obsérvese que o feito de que o rango por filas e o rango por columnas sexan o mesmo se pode demostrar sen utilizar o concepto de determinante (ver [1]). Agora, para demostrar o teorema, teremos en conta que o sistema non pode ter solución se o vector de termos independentes B non se pode expresar como combinación linear das columnas da matriz A , o cal quere dicir que o rango de A será menor que o rango de $(A|B)$. Se o rango de A é igual que o rango de $(A|B)$, entón o vector de termos independentes B é combinación

linear das columnas de A , e polo tanto o sistema ten solución. Esta combinación linear será única se as columnas da matriz son unha base do espazo xerado polas mesmas, é dicir, linearmente independentes, xa que esta propiedade caracteriza as bases dun espazo vectorial. Polo tanto, se non hai unha única solución, o rango non pode ser máximo, e se o rango é máximo só hai unha solución.

Este sinxelo razoamento resulta moitas veces difícil transmitirllo ós alumnos, e en moitos casos será imposible, porque ás veces nos atopamos coa necesidade de coñecer o teorema de Rouché-Frobenius antes de coñecer o concepto de rango dun conxunto de vectores. Este problema podemos evitalo dando unha definición alternativa do concepto de rango, na que tamén evitaremos utilizar o rango por menores dunha matriz, que involucra o cálculo de determinantes. Para definir o rango dunha matriz necesitaremos primeiro coñecer o concepto de matriz escalonada.

Definición Unha matriz A é escalonada se ten a seguinte forma:

$$\begin{pmatrix} * & * & * & * & * & * & * \\ 0 & * & * & * & * & * & * \\ 0 & 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

É dicir, é unha matriz triangular superior na que se as k primeiras entradas da fila i -ésima son cero, entón na fila $(i+1)$ -ésima polo menos as $k+1$ primeiras entradas tamén son cero. Dito de outro modo, en cada fila hai máis ceros nas primeiras entradas que na fila anterior.

Mediante o método de redución de Gauss podemos transformar calquera matriz nunha matriz escalonada, onde eventualmente as k últimas filas serán nulas. Isto quere dicir que as k últimas filas da matriz orixinal son combinación linear do resto, e que as $n-k$ primeiras filas son linearmente independentes. Así, poderemos definir o rango dunha matriz do seguinte modo:

Definición. O rango dunha matriz A é o número de filas non nulas que quedan despois de aplicar o método de redución de Gauss.

Nesta definición non estamos recurrido nin ó cálculo de determinantes, nin ó concepto de independencia linear.

Desta maneira o teorema de Rouché-Frobenius pode ser interpretado utilizando o método de Gauss do seguinte xeito:

- Que o rango de A sexa menor que o rango da matriz ampliada quere dicir que polo menos unha das filas da matriz A é nula, e esta fila non é nula na matriz ampliada, o cal significa que obtemos unha ecuación do tipo

$$0 = b_n, b_n \neq 0$$

é dicir, chegamos a un absurdo.

- Que o rango de A sexa igual ó rango da matriz ampliada quere dicir que podemos despexar as incógnitas de maneira ascendente. No caso en que o rango de ambas sexa igual ó número de incógnitas, a última ecuación será do tipo

$$a_{n,n}x_n = b_n, a_{n,n} \neq 0,$$

onde x_n é a última incógnita, e podemos calcular o seu valor. Unha vez coñecido este, despexaremos o resto. Nótese que debemos ter exactamente n filas non nulas, e neste caso a matriz

escalonada que obtemos ten os elementos da diagonal non nulos, o cal garante que se pode atopar a única solución do sistema.

- Se o rango da matriz A é igual ó rango da matriz ampliada, pero menor que o número de incógnitas, a última ecuación significativa será da forma

$$a_{n,i}x_i + \dots + a_{n,n}x_n = b_n, a_{n,i} \neq 0,$$

o cal nos permite despxear a incógnita x_i en función das incógnitas x_{i+1}, \dots, x_n , e proceder coma no caso anterior, coa diferenza de que agora a solución que atopemos dependerá dos parámetros x_{i+1}, \dots, x_n .

Unha ventaxa de interpretar o teorema de Rouché-Frobenius desta maneira é que se aplicamos directamente o método de Gauss, xa sabemos de golpe se o sistema é compatible ou non. Pola contra, para aplicar a regra de Cramer necesitamos que o sistema sexa compatible determinado, o cal require unha análise previa do mesmo, levándonos a realizar da orde de $n!$ operacións para ver que o rango de A é n , e de non selo non resolver o sistema.

5.2 Cálculo da inversa dunha matriz

Outra das utilidades do método de Gauss é o cálculo da matriz inversa. Dada unha matriz cadrada A , a súa matriz inversa A^{-1} é outra matriz que cumpre que $A^{-1}A = A A^{-1} = I$, onde I é a matriz identidade. É sabido que unha matriz A ten inversa cando e só cando o seu determinante é non nulo, o cal é equivalente a dicir que despois de aplicar o método de redución de Gauss, a matriz A transformárase nunha matriz triangular superior, con todas as filas non nulas.

Esencialmente estúdanse dous métodos para calcular a inversa dunha matriz: ben calculando a matriz adxunta; ben aplicando a redución de Gauss, primeiro poñendo ceros nos elementos por debaixo da diagonal, e despois poñendo ceros nos elementos por riba da diagonal. O cálculo da matriz adxunta conleva o cálculo de n^2 determinantes de orde $n-1$, un por cada elemento da matriz, o que dispara o número de operacións a realizar.

Para obter a matriz inversa polo método de Gauss baseámonos en que as operacións por filas se codifican mediante as matrices elementais. Así, se despois de realizar r transformacións elementais se obtén a matriz identidade a partir dunha matriz cadrada, podémolo codificar como

$$E^1 \cdot \dots \cdot E^r \cdot A = I,$$

co que $E^1 \cdot \dots \cdot E^r = A^{-1}$. Para obter esta matriz A^{-1} simplemente debemos aplicar as operacións elementais á matriz identidade na mesma orde na que llas aplicamos á matriz A .

6. Conclusións

A principal conclusión que se obtén desta análise é que, a partir de certa orde, o método de Gauss resulta moito máis eficiente que calquera método que inclúa o cálculo de determinantes por menores. Moitos dos métodos que involucran o cálculo de determinantes, como a regra de Cramer, ou o cálculo do rango dun conxunto de vectores, obtéñense mediante regras moi sinxelas de enunciar. Sen embargo, comprobamos que estas regras sinxelas conducen a cálculos desmesurados en dimensións moi grandes. Vimos de ver que o método de Gauss proporciona unha ferramenta sinxela e válida para realizar operacións moi frecuentes en Álgebra linear. De feito, detrás de moitos programas de cálculo simbólico se esconde o método de Gauss para resolver certas operacións. Máis aínda, o método de redución de Gauss dá sustento formal a teorías clásicas de Álgebra linear. Estes feitos lévannos a facer unha reflexión sobre a nosa docencia: o método de Gauss, como continuación natural do método de redución visto para sistemas de dúas ecuacións con dúas incógnitas, resulta suficiente para adentrarse nas teorías básicas de Álgebra linear, e tal vez

deberíamos facer máis fincapé neste feito en niveis educativos medios.

7. Apéndice

Neste apéndice presentamos unha sinxela simulación para os dous métodos para o cálculo de determinantes que vimos de discutir. Recurrimos ó programa Maxima para realizar a simulación. Para os que non estean familiarizados co programa, a referencia [2] pode ser de grande axuda.

A continuación presentamos o código para a función `det`, que calcula o determinante dunha matriz por menores:

```
det(matriz):=block([i,n:length(matriz)],
  if n=1 then matriz[1,1]
  else sum(matriz[1,i]*(-1)^(1+i)*det(submatrix(1,matriz,i)),i,1,n)
)$
```

Como se observa, a función `det` é unha función definida recursivamente e ten como único argumento a matriz á que lle queremos calcular o determinante.

Para o cálculo do determinante mediante diagonalización gaussiana da matriz, introducimos a seguinte función auxiliar `reducciongauss`, que realiza a eliminación gaussiana na primeira columna da matriz, intercambiando filas co correspondente cambio de signo se o elemento (1,1) é cero. A continuación a función `detgauss`, que ten como único argumento unha matriz cadrada, dá como resultado o determinante da mesma:

```
reducciongauss(matriz):=block([i,j,l,k:1,n:length(matriz),aux,piv],
  while matriz[k,1]=0 and k<n do k:k+1,
  if k#1 then for l:1 thru n
    do (aux:matriz[k,l], matriz[k,l]:-matriz[1,l], matriz[1,l]:aux),
  if matriz[1,1]#0 then
    for i:2 thru n do (piv:matriz[i,1]/matriz[1,1],
      for j:1 thru n do matriz[i,j]:(matriz[i,j]-piv*matriz[1,j])),
  matriz
)$
```

```
detgauss(matriz):=block([auxmatriz,i,m],
  m:length(matriz),
  if m=1 then matriz[1,1]
  else (auxmatriz:reducciongauss(matriz),
    auxmatriz[1,1]*detgauss(submatrix(1,auxmatriz,1)))
)$
```

Para ver como actúan estas funcións, amosamos un exemplo sinxelo:

1.- Primeiro compilamos as funcións anteriores e xeramos unha matriz pseudoaleatoria de tamaño 9x9 coa función `creamatriz` definida a continuación:

```
(%i4) creamatriz(n):= block([gen],gen[i,j]:=(-1)^(random(i+j))*random(10*n),
genmatrix(gen,n,n))$
```

```
(%i5) matriz:creamatriz(9);
```

```
Calculamos o det:
(%i6) det(matri
(%i7) time(%)
(%o5) (%o6) -541148537375803696 -33 52
(%o7) [13.65]
De seguido calculamos o detgauss:
(%i8) detgauss
(%o8) -541148537375803696
(%i9) time(%)
(%o9) [0.01]
```

rda en executar a

Comprobamos así nun exemplo concreto como o determinante da matriz é calculado máis rapidamente coa función detgauss que coa función det. Porén, non imos sacar conclusións dun exemplo concreto, así que tomamos 5 matrices cadradas de ordes que varían de 1 a 10 que foron xeradas coa función creamatriz e calculamos os determinantes e os tempos, obtendo a táboa seguinte:

Táboa comparativa

(M=tempo no cálculo por menores, G=tempo no cálculo por eliminación gaussiana)

	M	G	M	G	M	G	M	G	M	G	Media M	Media G
1	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
2	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
3	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
4	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
5	0,000	0,000	0,000	0,016	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,003
6	0,047	0,000	0,047	0,015	0,032	0,000	0,063	0,000	0,047	0,000	0,047	0,003
7	0,327	0,016	0,327	0,000	0,297	0,000	0,343	0,000	0,343	0,000	0,327	0,003
8	2,543	0,000	2,543	0,016	2,543	0,016	2,573	0,000	2,621	0,000	2,565	0,006
9	22,931	0,000	23,290	0,016	23,290	0,000	23,105	0,000	23,353	0,016	23,194	0,006
10	231,070	0,015	232,722	0,015	231,600	0,015	232,721	0,016	232,863	0,015	232,195	0,015

Obsérvase que o tempo para matrices de tamaño ata 5x5 non é superior para a función det (columnas coa lenda **M**), pero a partir de orde 6 os tempos medran considerablemente máis axiña

para det que para detgauss (columnas coa lenda **G**). Isto faise evidente nas dúas columnas da dereita que representan a media dos valores das simulacións.

8. Bibliografía

[1] Juan de Burgos Román, *Álgebra lineal*, McGrawHill, 1993.

[2] Mario Rodríguez Riotorto, *Primeros pasos en Maxima*, www.telefonica.net/web2/biomates/.