Prof. Dr. Ludwig Paditz
University of Applied Sciences,
Faculty of Information Technology and Mathematics
Friedrich-List-Platz 1, D-01069 Dresden, Germany
e-mail: paditz@informatik.htw-dresden.de

The Rank of a Matrix with Parameters and the Solution of a Linear System of Equations with Parameters

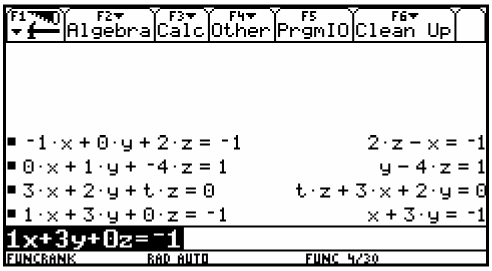
Abstract:
It seems, that sometimes the ref- and rref-functions does not work in a good manner, if we consider a matrix with parameters. This is a well-known problem of the TI-CAS-calculators updated with the newest OS version 3.10 too.

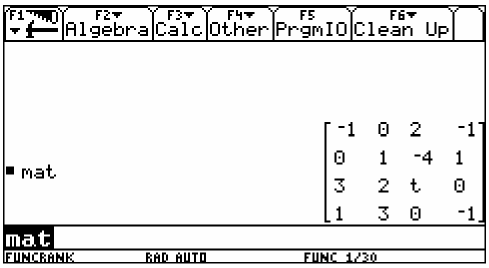
In this lecture we introduce a one-step-procedure to transform a matrix with parameters, the new created ***lineqsys(mat,i,k)*** *function, where* ***i*** *and* ***k*** *are the coordinates of the pivot.*
i *denotes the pivot-row and* ***k*** *denotes the pivot-column to pivoting the matrix* ***mat****.*
During the one-step-procedure we omit the old pivot-column, i.e. from step to step we get a smaller matrix for the considered problem. Finally we can see the behaviour of the solution of the considered system in the dependence on the parameters and we can see the solution too.

To get the rank of a matrix with parameters, we use a similar pivoting-procedure and omit the previous pivot-column and previous pivot-row after an exchange-step. Here we use the new created ***rank(mat,i,k)*** *function. The rank of a matrix is the possible number of exchange steps.*

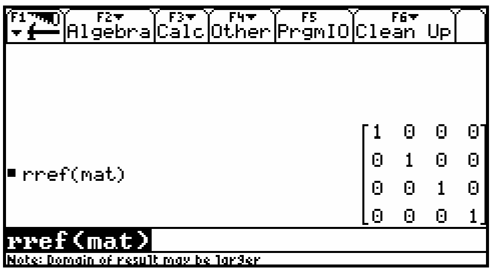
During the lecture several examples are given and demonstrated with the TI voyage200.

Let's start with the following linear system of equations with parameter t:



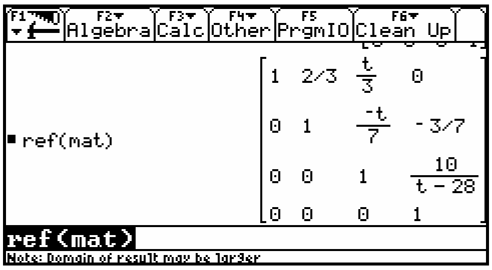
or


We try to solve this system with the **rref**-function and get:

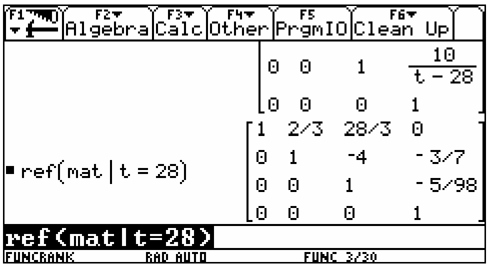


It seems no solution for all t.

With the **ref**-function we get more information:

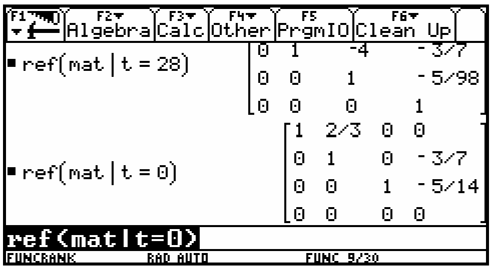


and



Again, we see: no solutions for all *t*.

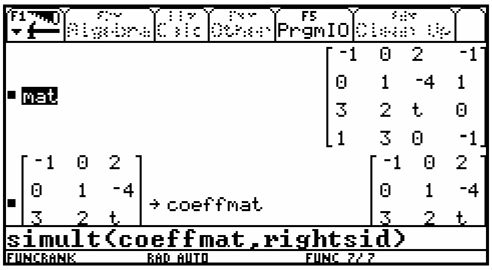
Now we choose *t* = 0 and get another result, a solution:

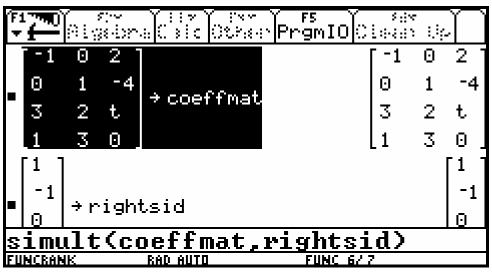


If *t* = 0 then we have the solution [-2/7, 3/7, 5/14] and otherwise (*t* not equal 0) no solution.
It seems, the **ref**- and **rref**-functions sometimes does not work in a good manner, if we consider a matrix with parameters.

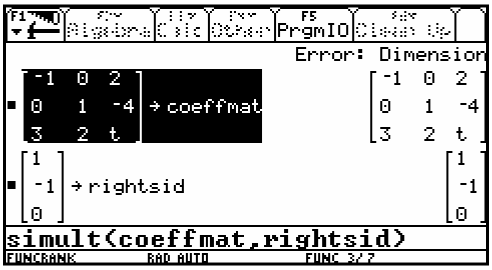
Now let's have a look on the well-known **simult**-function:
Consider the linear system of equations with 4 equations and 3 unknown variables and a parameter *t* in the following manner:

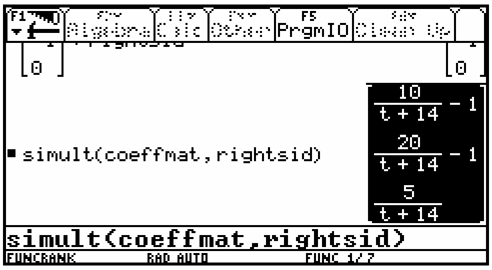
$$\begin{array}{rl} -1\,x + 0\,y + 2\,z - 1 = 0 & \text{(equation 1)} \\ 0\,x + 1\,y - 4\,z + 1 = 0 & \text{(equation 2)} \\ 3\,x + 2\,y + t * z + 0 = 0 & \text{(equation 3)} \\ 1\,x + 3\,y + 0\,z - 1 = 0 & \text{(equation 4)} \end{array}$$



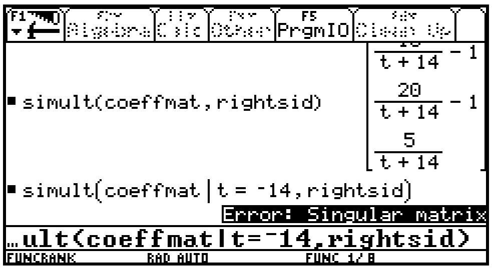


The **simult**-function can't solve the equation system with a singular matrix. Therefore we consider:





What happens if t = -14 ? In this case we get a singular matrix and can't decide exist a solution. Possible is a contradiction or many solutions.

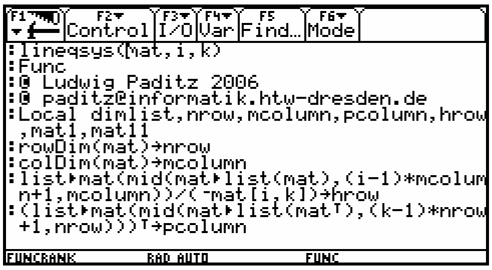


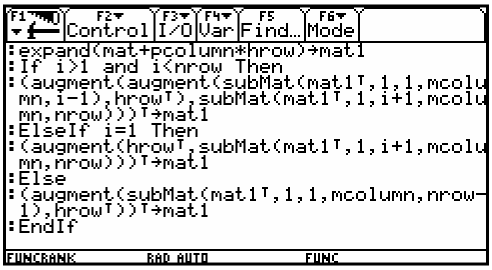
Thus the **simult**-function is not so helpful.

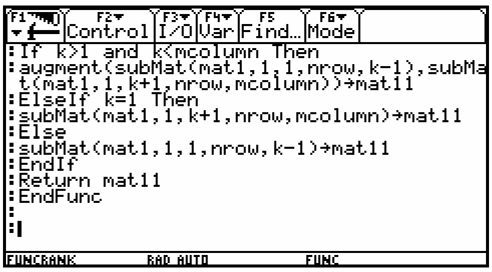
In this paper we introduce a one-step-procedure to transform a matrix (with parameters) in another one. Each of in such a manner created matrix is the matrix of an equivalent system, i.e. the system with the transformed matrix has the same behaviour concerning the solutions. It is clear, that we consider the augmented matrix with the coefficients and the right hand side of the linear system.
We hope, that the new commands will be included in one of the next updates of the OS of the here considered CAS-calculator.

We introduce a one-step-procedure to transform a matrix with parameters, the new created **lineqsys(mat,i,k)** function, where **i** and **k** are the coordinates of the pivot.
i is the pivot-row and **k** is the pivot-column to pivoting the matrix **mat**.
During the one-step-procedure we omit the old pivot-column, i.e. from step to step we get a smaller matrix for the considered problem.

Have a look in the program of the new created **lineqsys**-function:



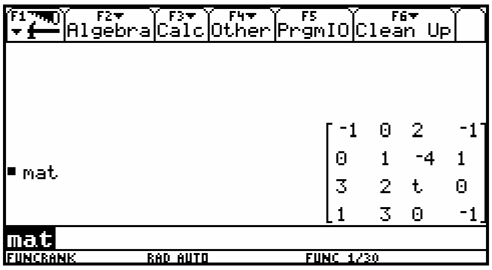


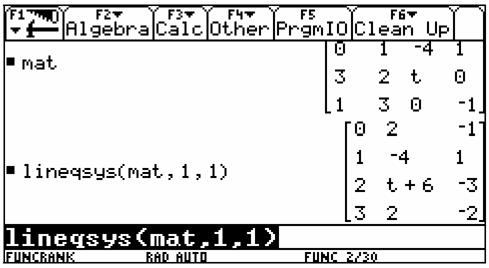


```
lineqsys(mat,i,k)
Func
© Ludwig Paditz 2006
© paditz@informatik.htw-dresden.de
Local dimlist,nrow,mcolumn,pcolumn,hrow,mat1,mat11
rowDim(mat)→nrow
colDim(mat)→mcolumn
list▶mat(mid(mat▶list(mat),(i-1)*mcolumn+1,mcolumn))/
(-mat[i,k])→hrow
(list▶mat(mid(mat▶list(matᵀ),(k-1)*nrow+1,nrow)))ᵀ→pcolumn
expand(mat+pcolumn*hrow)→mat1
If i>1 and i<nrow Then
(augment(augment(subMat(mat1ᵀ,1,1,mcolumn,i-1),hrowᵀ),
subMat(mat1ᵀ,1,i+1,mcolumn,nrow)))ᵀ→mat1
ElseIf i=1 Then
(augment(hrowᵀ,subMat(mat1ᵀ,1,i+1,mcolumn,nrow)))ᵀ→mat1
Else
(augment(subMat(mat1ᵀ,1,1,mcolumn,nrow-1),hrowᵀ))ᵀ→mat1
EndIf
If k>1 and k<mcolumn Then
augment(subMat(mat1,1,1,nrow,k-1),
subMat(mat1,1,k+1,nrow,mcolumn))→mat11
ElseIf k=1 Then
subMat(mat1,1,k+1,nrow,mcolumn)→mat11
Else
subMat(mat1,1,1,nrow,k-1)→mat11
EndIf
Return mat11
EndFunc
```

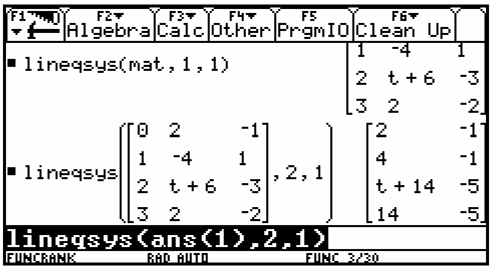
The idea for the new **lineqsys(mat,i,k)**-function and the new **rank(mat,i,k)**-function respectively is the exchange-algorithm by the Swiss Prof. Dr. math. Dr. h. c. Eduard Stiefel of the ETH Zurich (*1909, †1978).
Every step of the EX-algorithm (EX=exchange) works with a so-called **pivot** (pivot element) in the row ***i*** and the column ***k*** of the considered matrix ***mat***. The start-matrix ***mat*** is the augmented matrix ($\underline{\mathbf{A}}$,-$\underline{\mathbf{b}}$), if we want to solve the linear system $\underline{\mathbf{A}}*\underline{\mathbf{x}} = \underline{\mathbf{b}}$ or $\underline{\mathbf{A}}*\underline{\mathbf{x}} - \underline{\mathbf{b}} = \underline{\mathbf{o}}$. Here $\underline{\mathbf{o}}$ denotes the zero-vector and vector $\underline{\mathbf{x}}$ contains the unknown variables. Now we check the new **lineqsys(mat,i,k)** with voyage200:

Let's consider our example with the above considered equations 1 to 4:

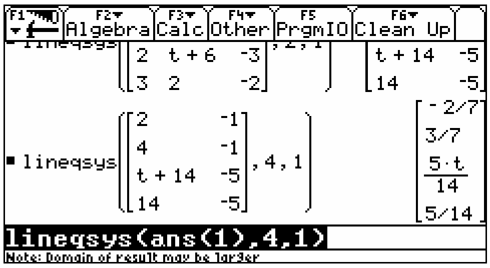




1st step



2nd step



3rd step

Thus we get immediately the result: If $t = 0$ then we have the solution [-2/7, 3/7, 5/14] and otherwise (t not equal 0) no solution. So we have a possibility to solve a system step by step.

The theoretical background is the following exchange-procedure, cp.

Paditz,Ludwig: Mathematische Modelle und wissenschaftlich-technische Anwendungen, Beispiele aus Schule und Studium mit dem grafikfähigen Symboltaschenrechner ClassPad 300,
Hrg. v. CASIO Europe GmbH im Bildungsverlag EINS, Norderstedt 2004 (1.Aufl.), 112 S.

ST	x	y	z	1
y_1	-1	0	2	-1
y_2	0	1	-4	1
y_3	3	2	t	0
y_4	1	3	0	-1
K	*	0	2	-1

T_1	y_1	y	z	1
x	*	0	2	-1
y_2	*	1	-4	1
y_3	*	2	t+6	-3
y_4	*	3	2	-2
K	*	*	4	-1

Austausch von Variablen in <u>y</u> und <u>w</u> zur Erzeugung äquivalenter Systeme:

① Wahl eines Pivots ungleich Null (z.B. a_{11})

② „Keller"-Zeile notieren (als Pivotzeile, geteilt durch das negative Pivot), kein Eintrag unter dem Pivot, hier „*" notieren.

System T_1 (y_1 mit x ausgetauscht):

③ $y_1 = 0$ (!Hilfsvariable), darunter kein Eintrag notwendig, hier „*" notieren.

④ Für alte Pivotzeile aus **ST** jetzt **K**-Zeile notieren. Restliche Elemente werden addiert mit Produkt aus darunter stehendem Element der **K**-Zeile und daneben stehendem Element der Pivotspalte.

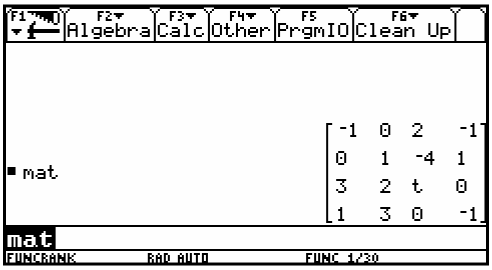
⑤ analog ① und ② (in T_1 Pivot und **K**-Zeile)

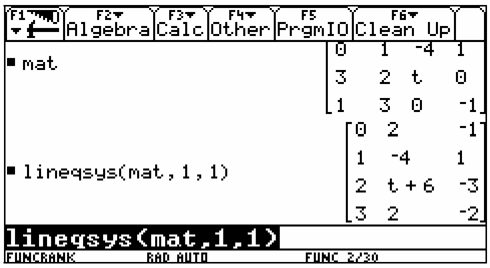
The first table shows the start table ST with the augmented matrix ***mat*** and the help-row **K** for computation the next table T1.

The pivot is -1 and we exchange x against $y1$. We don't fill the column $y1$ in T1, i.e. T1 is a shorter equivalent representation of the considered linear system. The EX-rules to get T1 we can summarize:

1. Choose a pivot not equal 0 in ST.
2. Compute the help-row K: Put * in the pivot column, the other elements of the pivot row are to be divided by the pivot with the opposite sign and put in K.
3. In T1 the pivot is replaced by its reciprocal value. The other elements of the pivot column are to be divided by the pivot. ($y1$=0, i.e. we omit rule 3., i.e. we have no new $y1$-column.)
4. The other elements of the pivot row are now the elements of the help-row K
   Elements in the remaining part of the new table or matrix are transformed by means of the old pivot column and help-row K in the following manner (triangle-rule): add to the considered element the product of the element of the help-row K and in the pivot column, where you have to go in vertical and horizontal direction respectively.
5. Similar to 1. and 2., now in T1: Choose the pivot in T1 and add the K-row in T1.

With the calculator:





1[st] step

Now we do the next EX-step with the same rules in the matrix T1 to get T2, and in T2 to get T3:

T_2	y_1	y_2	z	1
x	*	*	2	-1
y	*	*	4	-1
y_3	*	*	t+14	-5
y_4	*	*	14	-5
K	*	*	*	5/14

System T_2 (y_2 mit y ausgetauscht):

⑥ wie ③ und ④ (mit $y_2=0$)

⑦ wie ① und ② (in T_2 Pivot und K-Zeile festlegen)

T_3=ET	y_1	y_2	y_4	1
x	*	*	*	-2/7
y	*	*	*	3/7
y_3	*	*	*	-5+5(t+14)/14 =5t/14
z	*	*	*	5/14

System T_3 = ET (Endtabelle)

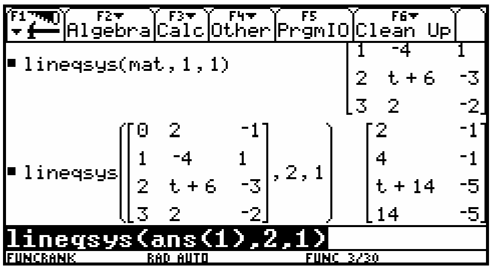
(y_4 mit z ausgetauscht):

Auswertung: Für $t=0$ ist $y_3=0$ erfüllt und die Lösung lautet:

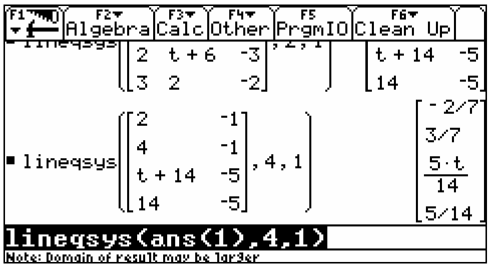
$x=-2/7,\ y=3/7,\ z=5/14.$

Für $t\neq 0$ ist $y_3=0$ nicht erfüllt, Widerspruch in y_3-Zeile, d.h. keine Lösung möglich.

With the calculator:



2nd step



3rd step

Thus we get immediately the result: If t = 0 then we have the solution [-2/7, 3/7, 5/14] and otherwise (t not equal 0) no solution. So we have a possibility to solve a system step by step.

This one-step-procedure is helpful for the learning process of our students. It is an interactive work with the CAS-calculator. The decision on the choice of the pivot will be done by the student and the transformation of the matrix or table will be done by means of the tool. Thus we get in a good manner all cases of the parameter t, which we have to consider.

Remark:
We know that the number of EX-steps is the rank of the considered matrix. Thus we have:

$$\text{rank}(\underline{\mathbf{A}}) = 3 = \text{rank}(\underline{\mathbf{A}},-\underline{\mathbf{b}}) = \text{rank}(mat) \quad \text{for } t = 0$$

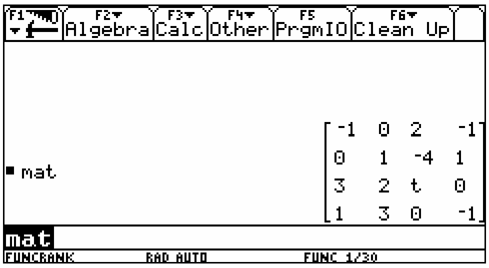
and

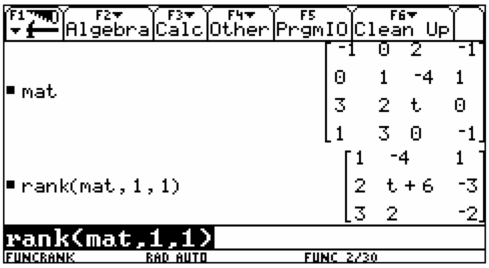
$$\text{rank}(\underline{\mathbf{A}}) = 3 < \text{rank}(\underline{\mathbf{A}},-\underline{\mathbf{b}}) = \text{rank}(mat) = 4 \quad \text{for } t \neq 0.$$

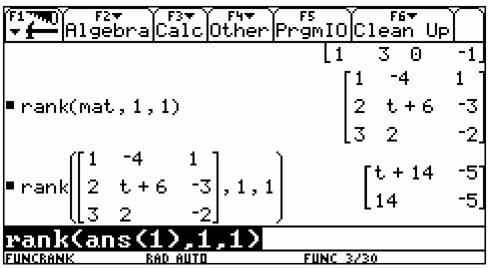
If only we want to get the rank of a matrix (and not the solution of the linear system, connected with the matrix), then we should use the new created **rank(mat,i,k)**-function, which omits step by step the old pivot column and old pivot row in the transformed matrix.

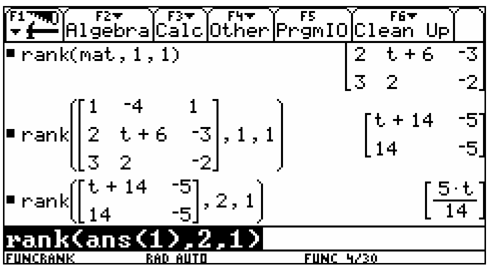
Thus we have a fast procedure to count the number of EX-steps to get the rank of a matrix in dependence on the values of the parameters.

Let us consider the example mentioned above:
In the screenshots you see that we have done three EX-steps.





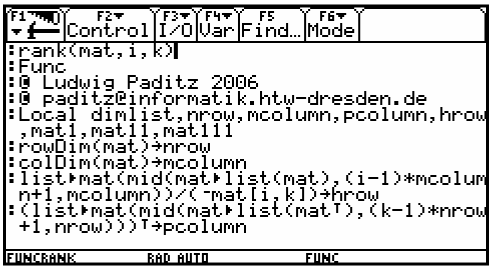


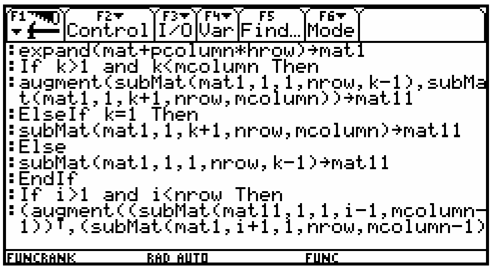


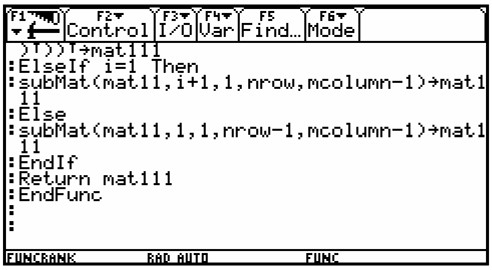
The theory of the pivoting-procedure says that the number of possible steps is the rank of the matrix, considered at the beginning.
The definition of the rank is the number of linear independent vectors in a matrix.
We could do 3 steps in every case and for t not equal 0 we can do formally 4 steps here.

This means for a linear system of equations with the augmented matrix ***mat***, cp. our example, that we have only in the case t=0 a solution of the considered system, otherwise the right side of the system is linear independent of the left side vectors of the considered system.

Let us consider the program-text of the new created rank-function:








```
rank(mat,i,k)
Func
© Ludwig Paditz 2006
© paditz@informatik.htw-dresden.de
Local dimlist,nrow,mcolumn,pcolumn,hrow,mat1,mat11,mat111
rowDim(mat)→nrow
colDim(mat)→mcolumn
list▶mat(mid(mat▶list(mat),(i-1)*mcolumn+1,mcolumn))/
(-mat[i,k])→hrow
(list▶mat(mid(mat▶list(matᵀ),(k-1)*nrow+1,nrow)))ᵀ→pcolumn
expand(mat+pcolumn*hrow)→mat1
If k>1 and k<mcolumn Then
augment(subMat(mat1,1,1,nrow,k-
1),subMat(mat1,1,k+1,nrow,mcolumn))→mat11
ElseIf k=1 Then
subMat(mat1,1,k+1,nrow,mcolumn)→mat11
Else
subMat(mat1,1,1,nrow,k-1)→mat11
EndIf
If i>1 and i<nrow Then
(augment((subMat(mat11,1,1,i-1,mcolumn-1))ᵀ,
(subMat(mat11,i+1,1,nrow,mcolumn-1))ᵀ))ᵀ→mat111
ElseIf i=1 Then
subMat(mat11,i+1,1,nrow,mcolumn-1)→mat111
Else
subMat(mat11,1,1,nrow-1,mcolumn-1)→mat111
```

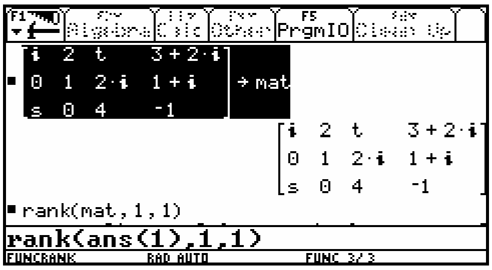
```
EndIf
Return mat111
EndFunc
```

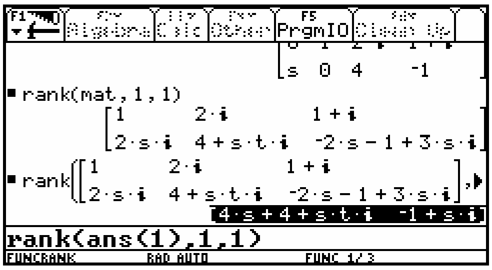
As another example, let us consider the three linear equations with the three unknowns x, y and z and the two parameters s and t, cp.

Paditz, Ludwig: Rechnen und graphische Darstellungen mit komplexen Zahlen, Anwendungsbeispiele aus Schule und Studium für den ALGEBRA FX 2.0, 1. Aufl., 2001, Hrg. v. CASIO Europe GmbH Norderstedt:

$$\mathbf{j}\ x + 2\,y + t * z = 3 + 2\,\mathbf{j} \qquad \text{(equation 1)}$$
$$0\,x + 1\,y + 2\,\mathbf{j}\,z = 1 + \mathbf{j} \qquad \text{(equation 2)}$$
$$s * x + 0\,y + 4\,z = -1 \qquad \text{(equation 3)}$$

For which parameters ***s*** and ***t*** we have a unique solution or infinite many solutions or no solution? In the case of solution, what is the result in the complex plane? At first we consider the rank of the augmented matrix of the given linear system in dependence on the parameters s and t. Two EX-steps are possible in every case and the 3rd EX-step only for **4 * *s* + *s* * *t* * j + 4 ≠ 0 :**





By the way, the pivot for the 3rd EX-step is similar to the determinant of the coefficient matrix of our considered system.

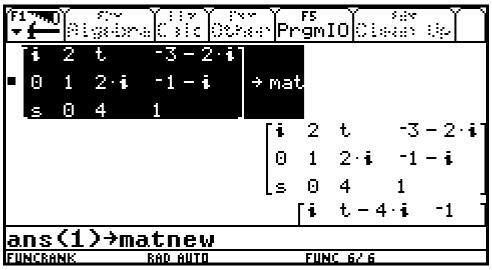
Now we conclude from the row- and column-reduced matrix after two EX-steps:

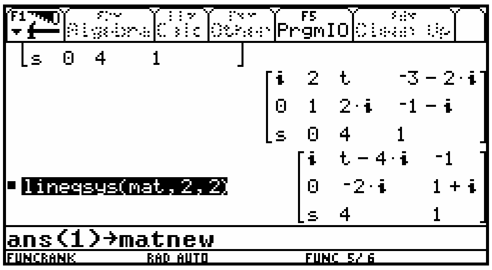
rank(**A**) = rank(**A**,**b**) = 2, if and only if **4 * *s* + *s* * *t* * j +4** = **0, *s* = -j,**
i.e. ***s*=-j** and ***t*=-4+4j** and det(**A**)=0. We have infinite many solutions.

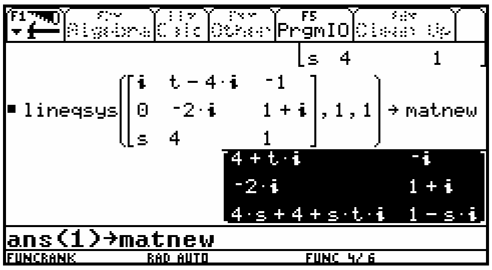
rank(**A**) = 2 < rank(**A**,**b**) = 3, if and only if **4 * *s* + *s* * *t* * j + 4** = **0 , *s* ≠ -j,** i.e. ***s*≠-j** and ***t* ≠ -4+4j** and det(**A**) = 0. We have no solution.

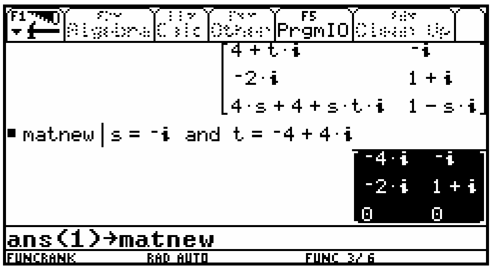
rank(**A**) = rank(**A**,**b**) = 3, if and only if **4 * *s* + *s* * *t* * j + 4 ≠ 0**, i.e. det(**A**) ≠ 0 and ***s* = -j** and ***t* ≠ -4 + 4j** or ***s* ≠ -j** and ***t* = -4 + 4j**.
Here we have a unique solution.

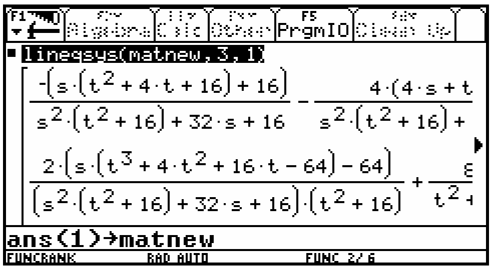
Thus we get in a quick manner by the help of the new **rank**-function all information on the considered complex system. Now we want to state the result by the help of the **lineqsys**-function:

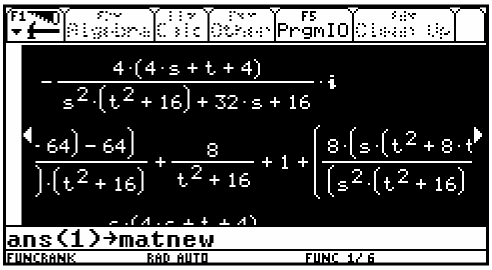










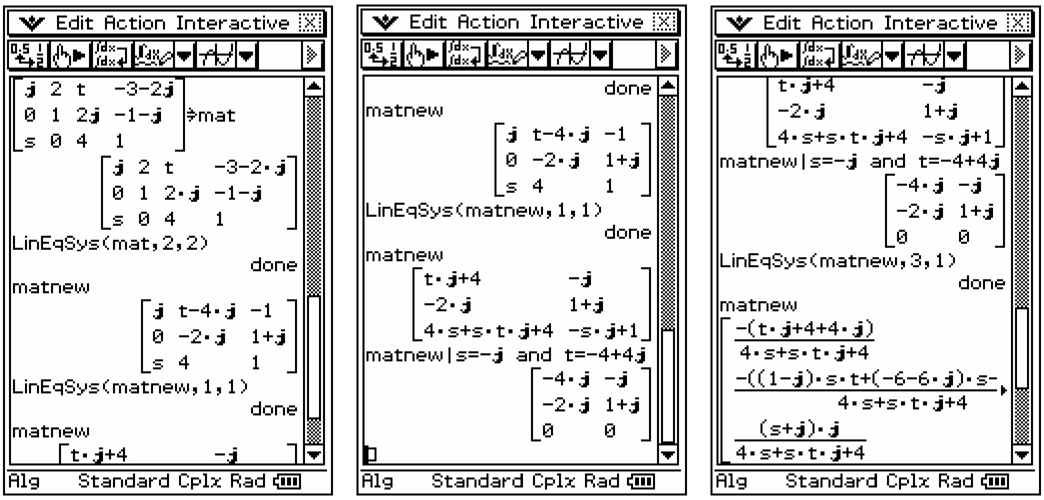


In the case ***s*=-j** and ***t*=-4+4j** we get ***x* = -4j*c*-j, *y* = -2j*c*+1+j, *z* = *c*,** where *c* is an arbitrary complex number.
In the case det(**A**) ≠ 0 we get the unique solution:
x = -(*t**j + 4 + 4*j) / (4**s* + *s* * *t* *j + 4),
y = (((-1+j)* *t* + 6 + 6*j)* *s* + 4 + 6*j) / (4* *s* + *s* * *t* *j + 4),
z = (*s* + j)*j / (4* *s* + *s* * *t* *j + 4),
where the voyage200 gives a representation in the arithmetic form: real-part + **j***imag-part.

The last screen-shots show that the TI-calculator will simplify the result without an additional command. Thus we get here a very large expanded representation of the solution and the smaller form, cp. the written solution above, we will not get from the TI-calculator.
To simplify results without additional commands is the didactical conception of the TI-calculators in difference to the CASIO-calculators, e.g. ClassPad300 PLUS.

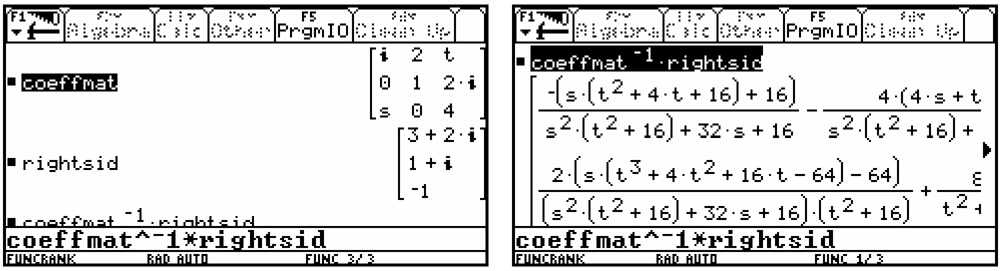
The didactical conception of the CASIO-calculators is not to simplify. Thus the user has the possibility to see step by step the transformation of the results, if he uses additional the simplify-function.

Now we want to state the result by the help of the **LinEqSys**-function of ClassPad300 PLUS:

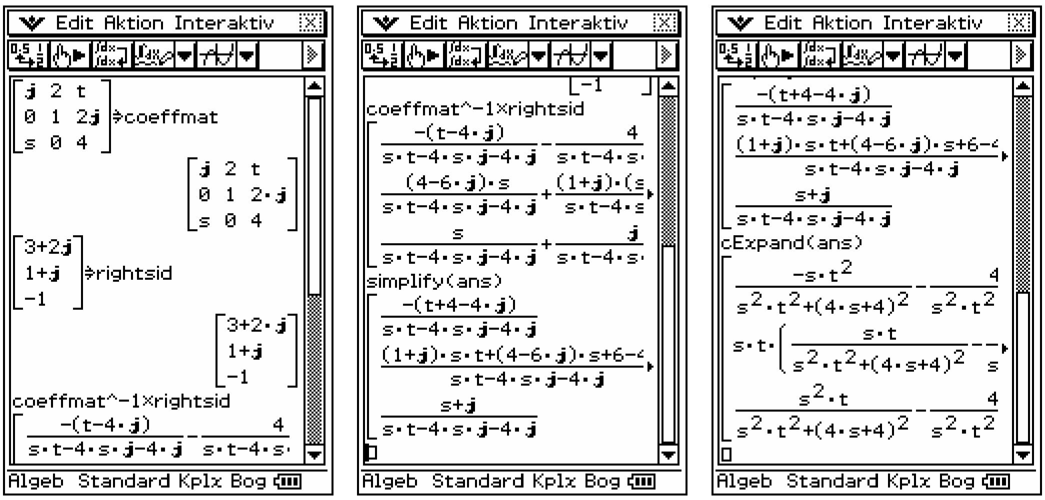


In the case det(**A**) ≠ 0 we see the unique solution in the last screen-shot:
x = -(*t**j + 4 + 4*j) / (4**s* + *s* * *t* *j + 4),
y = (((-1+j)* *t* + 6 + 6*j)* *s* + 4 + 6*j) / (4* *s* + *s* * *t* *j + 4),
z = (*s* + j)*j / (4* *s* + *s* * *t* *j + 4).

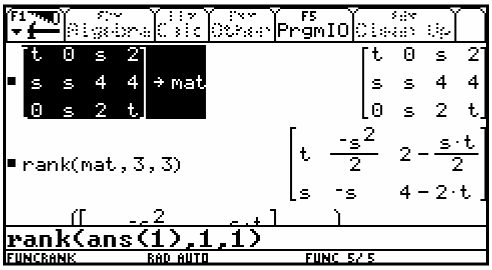
Now we try to compute the unique solution by the help of the inverse matrix:

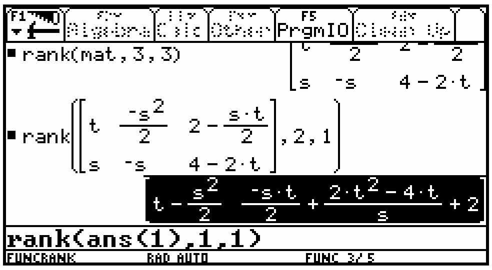


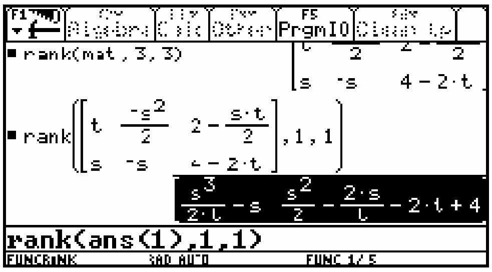
In comparision with the ClassPad300PLUS we get the following result, which we can transform step by step, if we wish another representation:



Another application of the new created functions is the following real linear problem with two real parameters. The augmented matrix *mat* you see in the screenshot:





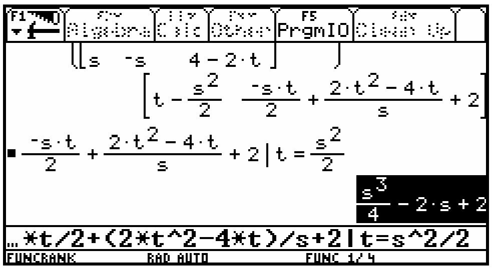


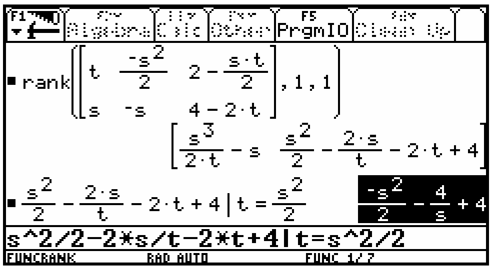
In the case $s = t = 0$ we get rank($\underline{\mathbf{A}}$) = 1 < rank($\underline{\mathbf{A}}$,$\underline{\mathbf{b}}$) = 2 (1 EX-step).

In the case $s \neq 0$ or $t \neq 0$ we can do a 2^{nd} EX-step and thus get with **rank(ans(1),2,1)** and **rank(ans(1),1,1)** respectively:
rank($\underline{\mathbf{A}}$) = 2 for $s \neq 0$ and $t = s$^2/2 and in the other case
rank($\underline{\mathbf{A}}$) = 2 for $t \neq 0$ and $s = 0$ or $t = s$^2/2.

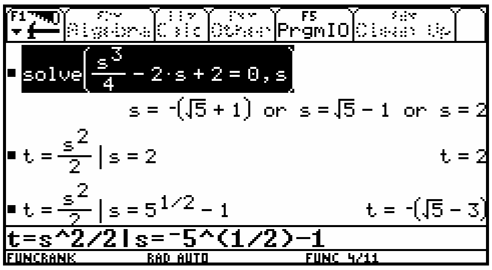
Finally we can establish that a 3^{rd} EX-step exist in the case $s \neq 0$ and $t \neq s$^2/2 and thus rank($\underline{\mathbf{A}}$) = 3 = rank($\underline{\mathbf{A}}$,$\underline{\mathbf{b}}$).

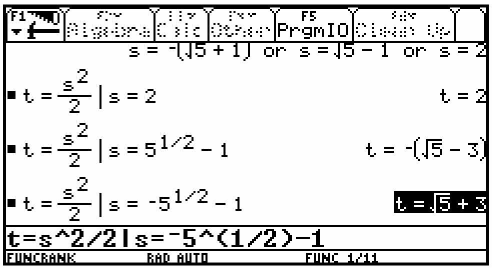
Now we discuss the open question on the rank($\underline{\mathbf{A}}$,$\underline{\mathbf{b}}$) in the case rank($\underline{\mathbf{A}}$) = 2. Again have a look in the screenshots after 2^{nd} EX-step:





We have to consider the 2^{nd} column in the last matrix.
In the case $s \neq 0$, $t = s$^2/2 we consider s^3/4 – 2s + 2.
In the case $t \neq 0$ we have to discuss s^2/2-2s/t – 2t + 4.





Thus in case $s \neq 0$, $t = s$^2/2 we get rank($\underline{\mathbf{A}}$) = 2 = rank($\underline{\mathbf{A}}$,$\underline{\mathbf{b}}$) for the following pairs of parameters:

$(s, t) = (2, 2)$ or $(s, t) = (-\sqrt{5}-1, \sqrt{5}+3)$ or $(s, t) = (\sqrt{5}-1, -\sqrt{5}+3)$.

Furthermore in case $s \neq 0$, $t = s$^2/2 we get rank(**A**) =2 < rank(**A**,**b**) = 3 for $s \neq 0$, $t = s$^2/2 and s ^3 – 8 s + 8 ≠ 0.

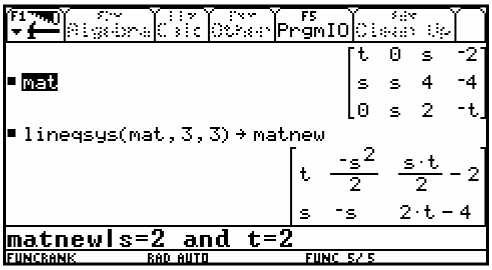
Now in the case $t \neq 0$ (rank(**A**) = 2) we consider s^2/2-2s/t – 2t + 4.
At first with $s = 0$ we get

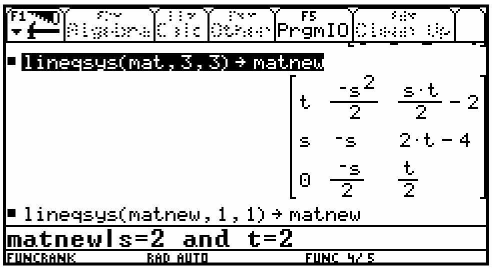
rank(**A**,**b**) = 2 for $t = 2$ and rank(**A**,**b**) = 3 for $t \neq 2$.

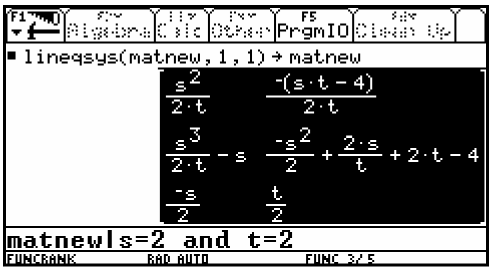
Now with $t = s$^2/2 ≠ 0 we get in the 2nd column of the last matrix:
s^2/2-2s/t – 2t + 4 = – (s^3 – 8*s + 8) / (2 s) and have the same situation considered already above with s^3 – 8*s + 8.

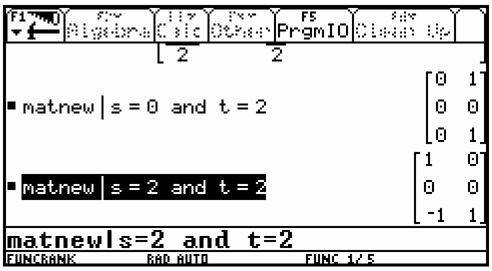
Summary:
The discussion on the rank shows:
A unique solution: only for $s \neq 0$ and $t \neq s$^2/2.
Infinite many solutions: only for $(s, t) = (0, 2)$ or $(s, t) = (2, 2)$ or
$(s, t) = (-\sqrt{5}-1, \sqrt{5}+3)$ or $(s, t) = (\sqrt{5}-1, -\sqrt{5}+3)$.

By the help of the **lineqsys**-function we get the following results:



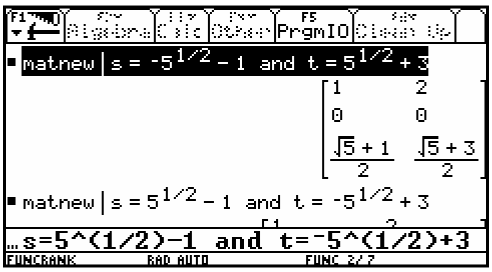


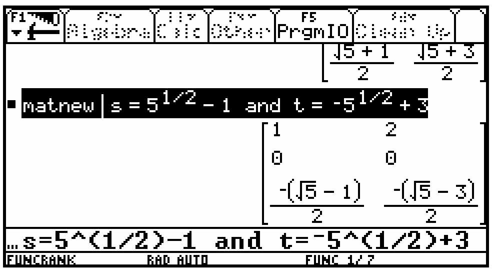




$(s, t) = (0, 2)$: $x = z = 1$, $y = c$, where c an arbitrary real number.

$(s, t) = (2, 2)$: $x = y = c$, $z = 1 – c$, where c an arbitrary real number.

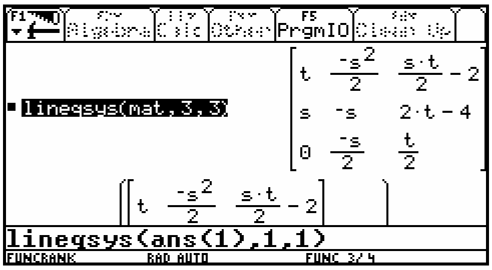


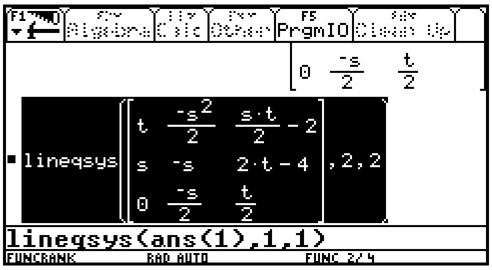


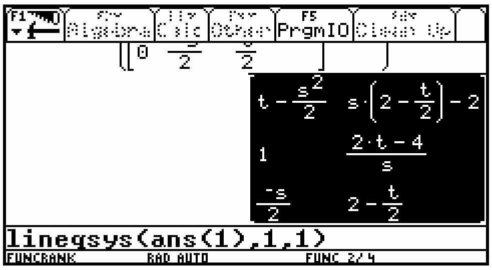
$(s, t) = (-\sqrt{5}-1, \sqrt{5}+3)$: $x = c + 2$, $y = c$, $z = (\sqrt{5}+1)*c/2 + (\sqrt{5}+3)/2$, c arbitrary real number.

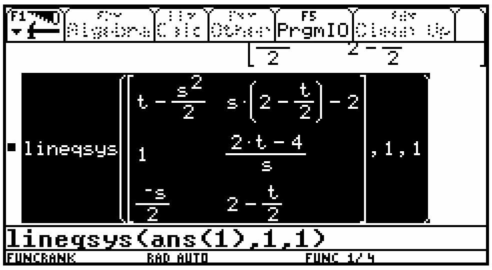
$(s, t) = (\sqrt{5}-1, -\sqrt{5}+3)$: $x = c + 2$, $y = c$, $z = (-\sqrt{5}+1)*c/2 + (-\sqrt{5}+3)/2$, c arbitrary real number.

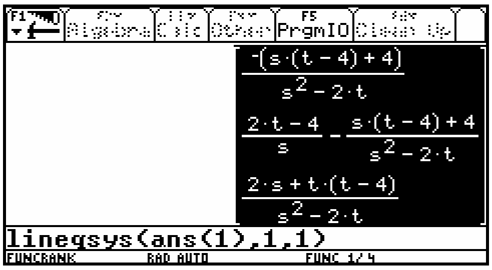
Finally the unique solution you see in the next screenshots:











This examples show, how to work with the useful tool, to solve linear systems with parameters or to get the rank of the matrix or augmented matrix. Thus it is good work in algebra with voyage200.

Further References:

Brauch, Wolfgang; Dreyer, Hans-Joachim; Haacke, Wolfhart: Mathematik für Ingenieure
11. Aufl., Teubner 2006, ISBN 3-8351-0073-4

Labuch, Dieter: Aufgaben zur Linearen Algebra, 3. neubearb. Aufl., 1998, Teubner-Verl. Stuttgart, ISBN 3-519-00240-X

Stiefel, Eduard: Einführung in die numerische Mathematik, 5.Aufl., Teubner 1976,
ISBN 3-519-12039-9

Stiefel, Eduard: An introduction to numerical mathematics, Academic Press, N.Y. 1963.

Paditz, Ludwig: Solving Problems in Algebra and Analysis with CAS-Calculator,
Schriftenreihe des Collegiums Europaeum Jenense CEJ 2006 (in print)