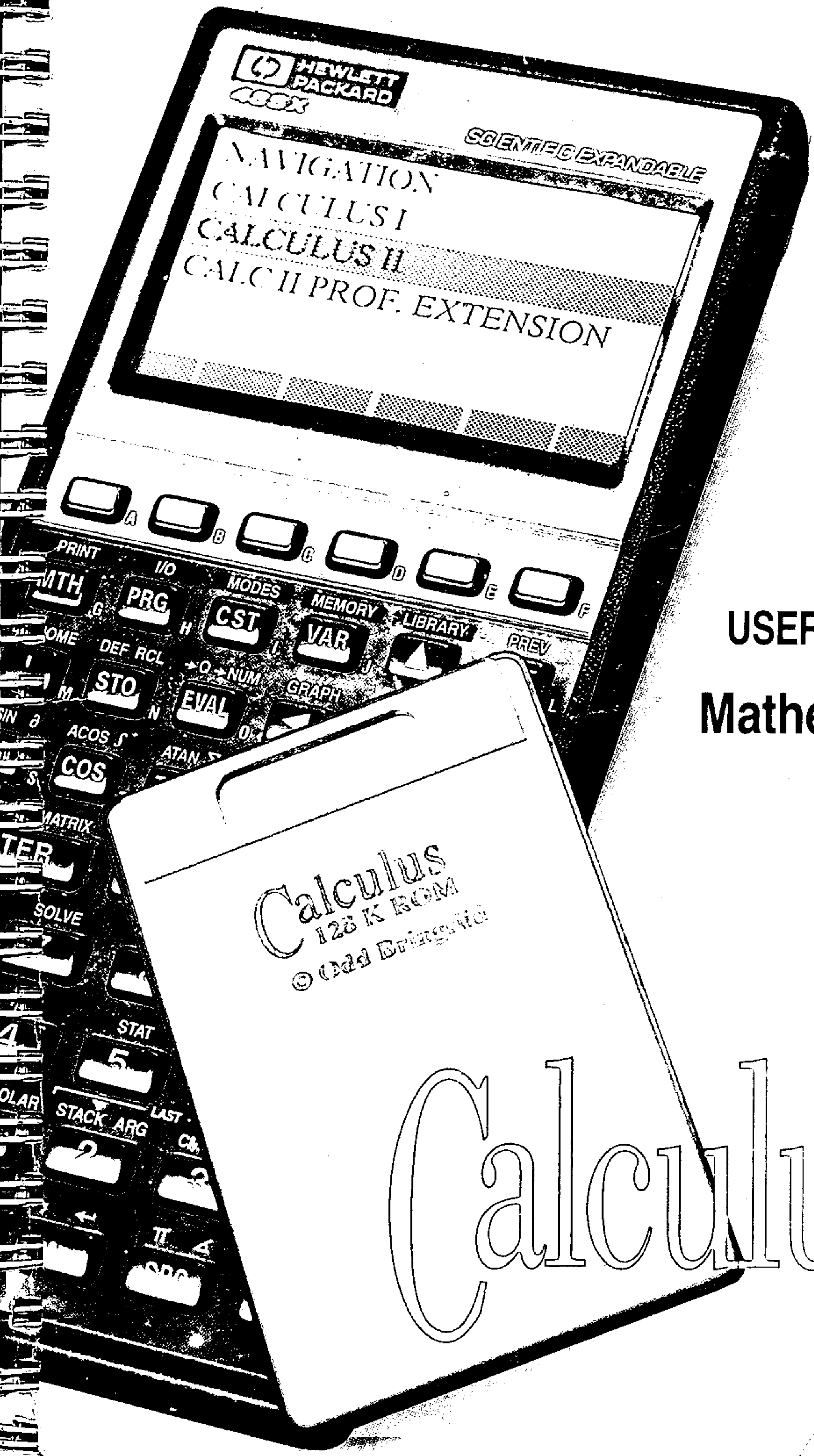
USER MANUAL

Mathematics II



Calculus

CALCULUS
MATHEMATICS II

VER 1.0

A mathematical program for calculators

Odd Bringslid
ISV
Postbox 1014   3601 Kongsberg
NORWAY

First edition August 1992

Contents

Laplace transforms 36

Probability 42

Statistics 57

Fourier series 75

Linear programming 81

1

Generaly

This is part II of CALCULUS mathematics. Together with part III this will represent a complete math pac for higher technical education.

As in part I a pedagogical interface is stressed. CALCULUS mathematics is a pedagogical tool in addition to a package for getting things calculated.

Hardware requirements

CALCULUS Math II runs under the calculator HP 48SX. The program card may be inserted into either of the two ports and Math I could be in the other port.

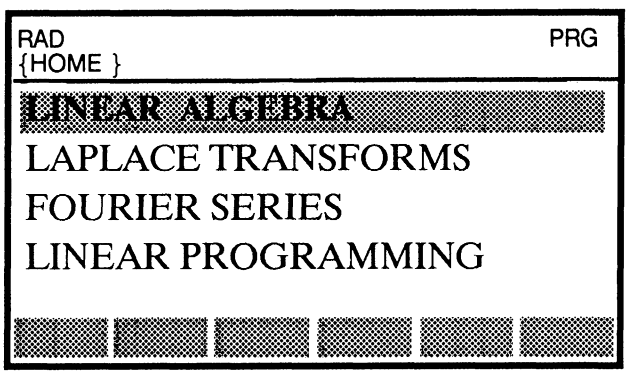
Starting up

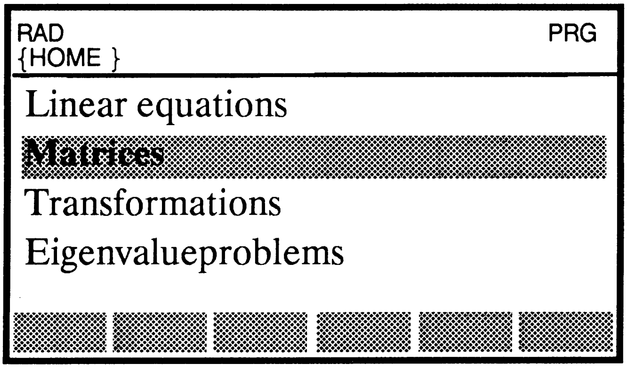
The LIBRARY menu will show up MAII. Pushing the MAII key will lead you into the main menu and then you simply push the START key.

User interface

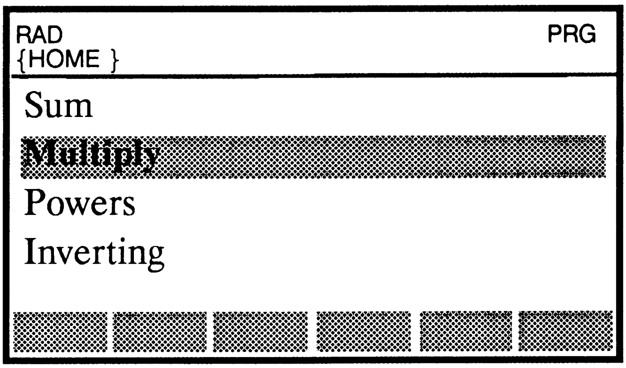
The CALCULUS meny system is easy to use. Using the arrow keys allows you to move the dark bar and select by pushing ENTER.

In the following example you will enter the submenu for LINEAR ALGEBRA and select Matrices/Multiply.





Under Matrices you will choose Multiply and you may multiply two symbolic matrices. The matrices are put into the SYMBOLIC MATRIX WRITER.



In the Matrix Writer you may delete, add, and echo from the stack (see manual for 48SX).

The input Editor

If you select Linear equations under LINEAR ALGEBRA you will enter the editor for input (input screen).

```
RAD                                PRG
{HOME }
              A*X=B
:PartAns Y/N: J
: B {B1....}:   {1 2 3}
: X {x y...}:    {x y z}
```

Here the input data can be modified and deleted and you can move around by using the arrow keys.

The cursor is placed right behind :PartAns Y/N: and here you enter Y if you want intermediate results. The arrow keys are used to get right behind :B {B1...}: and here you enter the right side vector of the system.

If you have done a mistake you may alter your input by using the delete keys on the calculator keyboard. You will not be able to continue before the data are correctly put in.

In the input screen there is often information about the problem you are going to solve (formulaes etc). Remember the ' ' in algebraics and separation of several data on the same input line by using blanks (space).

Echoing from the stack

If an expression or a value laying on the stack is going to be used, then the EDIT/↑STK/ECHO/ENTER sequence will load data into the input screen. Be sure to place the cursor correctly

Calculation finished

When a calculation is finished CALCULUS will either show up intermediate results by using the VIEW routine (intrinsic MAII) or return directly to the menu. In the last case you will need to use the →STK key to see the result laying on the stack.

Moving up and down in the menu

You can move downwards in the menu system by scrolling the dark bar and pressing ENTER. If you need to move upwards the UPDIR key will help. At any time you may HALT CALCULUS and use the calculator independent of CALCULUS by pressing the →STK key. CONT will get you back to the menu system.

STAT and MATR menues

On the menu line at the bottom the choices STAT and MATR are possible. Here you will have access to some routines regardless of your current menu position.

STAT:

- NORM — Normal distribution
- INVN — Inverse normal distribution
- UΣDAT — Sample mean, st. dev., median
- KΣDAT — Class table
- ΣDAT — Discrete table, two columns

MATR:

- ADD — Add symbolic matrices
- MULT — Multiply
- INV — Invert
- TRN — Transpose
- DET — Determinant

Leaving CALCULUS

Pushing the EXIT key will leave CALCULUS.

Intermediate results

In the input screen you may choose PartAns Y/N. Choosing N the result will be laying on the stack and you have to use →STK to see the answer. Choosing Y, different pages of intermediate results will show up or more than one result is laying on the stack.

The degree of details in the partial answers is somewhat different, but some of the results covers "the whole answer". In every case this will give the user a good help.

Different parts of an answer may be found on different pages and the page number can bee seen (use arrow up/down).

When PartAns Y(es) is chosen all numbers will show up with two figures behind comma. If a more accurate answer is necessary, you will have to look on the stack and perhaps use the N FIX option.

Flag status and CST menu

The flag status and CST menu you have before going into CALCULUS will be restored when you leave by pushing EXIT.

2

Linear algebra

The subject linear algebra covers linear eqautions with solution also for singular systems, matrix manipulation (symbolic), eigenvalue problems included systems of linear differential equations, linear transformations in two and three dimensions and vector spaces.

Linear equations (Gauss method)

Linear equations with symbolic parameters are handled. The equations have to be ordered to reckognize the coefficient matrix and the right side. The equations are given in the form:

$$\{\{\mathbf{A}\}\}*\{\{\mathbf{X}\}\}=\{\{\mathbf{B}\}\}$$

A is the coefficient matrix , **X** a column vector for the unknowns and **B** the right side column vector. Symbolic coefficients are possible.

If Det(**A**) = 0 (determinant) the system will be singular (self contradictory or indefinite). This is stated as "Self contradictory" or the solution will given in terms of one ore more of the unknowns (indefinite). Example:

{x,y,z} = {x,2*x-1,x-4}

The value of x is arbitrary so there is an infinite number of solutions.

If the system is underdetermined (too few equations), the solution will be given in the idefinite form. If the system is overdetermined (too many equations) the solution will be given in the indefinite form if the equations are lineary dependent or as "Self contradictory" if they are lineary independent.

The solution algorithm is the Gauss elimination. If PartAns Y(es) is selected, the different stages in the process will be given as matrices on the stack which may be viewed by using the MATW option (LIBRARY). The coefficient matrix and the right side vector are assembled in one matrix (**B** is the rightmost column).

Interface:

```
RAD                                  PRG
{HOME }
              A*X=B
: PartAns Y/N:   Y
:     B {B1...}: {7 6 0}
:      X {x y...}: {x1 x2 x3 x4 x5}
```

The symbolic matrix writer will now appear. The following matrix is put into it:

$$\begin{matrix} 2 & -1 & 3 & 2 & -1 \\ 1 & 2 & 1 & -1 & 1 \\ 1 & -4 & -1 & 3 & -1 \end{matrix}$$

The example solves the system:

$$2x_1 - x_2 + 3x_3 + 2x_4 - x_5 = 7$$
$$x_1 + 2x_2 + x_3 - x_4 + x_5 = 6$$
$$x_1 - 4x_2 - x_3 + 3x_4 - x_5 = 0$$

The system is indefinite (too few eqautions)) and the solution is given in terms of x_5 and x_4.

Matrix calculations

Some operations on symbolic matrices are done (not covered by the 48SX intrinsic functions). The matrices are put into the SYMBOLIC MATRIX WRITER and the matrix is put on the stack by pushing ENTER.

Addition
Both matices are put into the matrix writer and added. An error message is given for wrong dimension.

Multiplication
Both matrices are put into the matrix writer and multiplied. An error message is given for wrong dimensions. The first matrix has to have the same number of columns as the second has rows. Be aware of the order of the matrices.

Inverting
The matix is put into the matrix writer and inverted. An error message is given if its not quadratic.

Determinant
The determinant of a symbolic matrix is calculated. The matrix is put into the matrix writer. An error message is given if its not quadratic.

Rank of a matrix

The rank of a marix is calculated. This routine may be used for testing linear independency of rowvectors. The routine makes the matrix upper traingular and PartAns Y gives the different stages of the process.

Trace

The trace of a square matrix is calculated. An error message is given if the matrix is not quadratic.

Orthogonal matrix

This routine is testing whether the matrix is orthogonal i.e. the inverse is equal to the transpose. An error message is given for wrong dimension (must be quadratic). The answer is logic 0 or 1. May be used to investigate if rowvectors are orthogonal i.e. is an orthogonal basis of a vector space.

Transpose matrix

The transpose of a symbolic matrix is calculated.

Symmetric

Investigates whether a matrix is symmetric or not. Logic 0 or 1.

Linear transformations

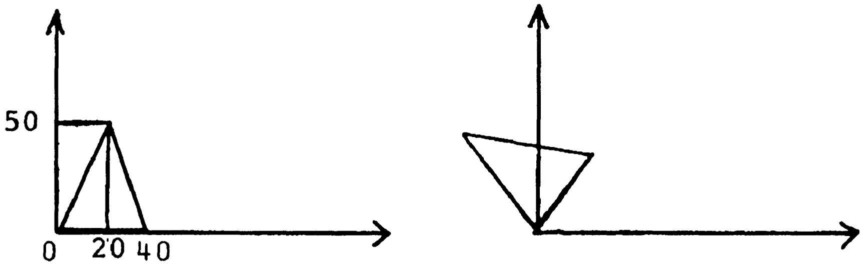
Linear transformations include coordinate transformations in the plane and in the three dimensional space. The transformations are rotation, translation and scaling. The point to be transformed is given relative a rectangular coordinate system.

Mixed transformations (concatinating) is possible. The order of the transformations is important if rotation is one of the them.

2 D transformations (two dimensions)

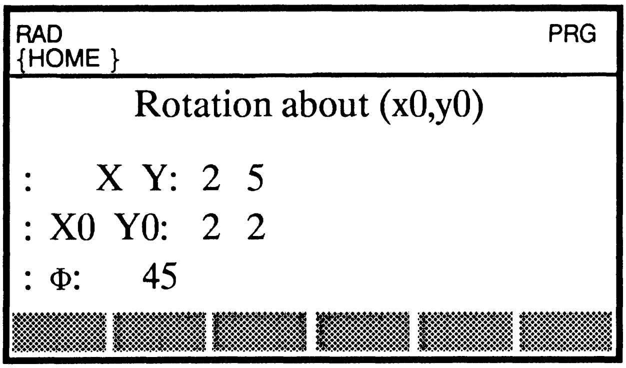
Rotation

The rotation angle must be given in degrees and the transformed point is given as components of a list (to allow symbols). The rotation is counterclockwise for positive angles about an arbitrary point.



2.1 Rotation of a triangle about origo

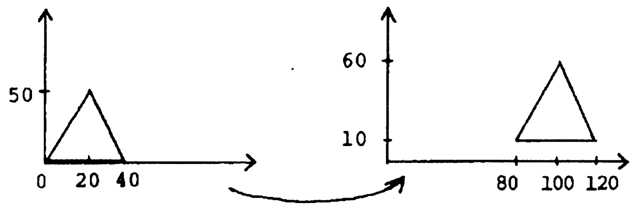
Interface:



The point (2,5) is rotated about (2,2) an angle 45°. To rotate a triangle all three points have to be transformed.

Translation

This is a pure translation of a point, the coordinates are given an addition.



2.2 Translation of a triangle

Interface:

RAD
{HOME }
PRG

$$Xt = X + Tx \quad Yt = Y + Ty$$

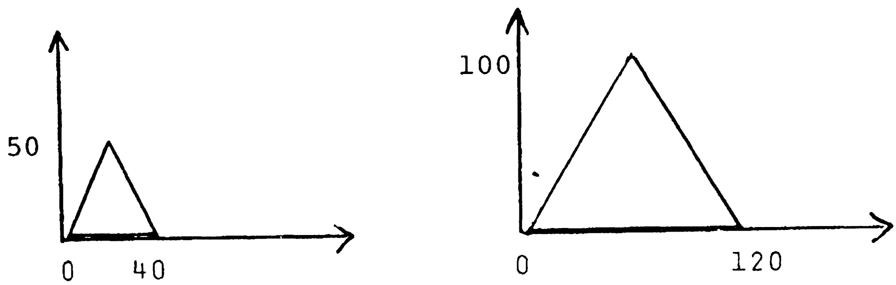
:X Y: 2 5

:Tx Ty: 3 6

The example moves the point (2,5) to (2,5) + (3,6) = (5,11)

Scaling

The coordinates are multiplied by a factor. For a geometric figure where the points are scaled, this will give a smaller or bigger figure. If the X and Y coordinates are scaled differently this will alter the shape of the geometric figure.



2.3 Enlarging of a triangle

Interface:

```
RAD                                PRG
{HOME }
        Xt=X*Sx  Yt=Y*Sy

:X Y:        2 5
:Sx Sy:      3 6
```

The example multiplies 2 with 3 and 5 with 6 and the point (2,5) is moved.

Concatinating (mixed transformations)

Concatinating means a mixture of several transformations.

The order of the transformations is important, given in a list as {R S T} (rotation, scaling and translation).

Interface:

```
RAD                                   PRG
{HOME }
:       X Y: 2  5
:      Sx  Sy:1  1
:      Tx Ty:2  6
:Θ X0 Y0 {S T R}:  45 2 2 {R T S}
```

In the example the point (2,5) is rotated clockwise 45^{o} about the point (2,2) first, then a translation of 2 in the x-direction and 6 in the y-direction. There is no scaling, indicated by 1 1 for the scaling factors.

> *Rem. No translation gives $Tx = Ty = 0$ and no rotation gives $\Theta = 0$.*

3 D transformations (three dimensions)

Translation

The interface is the same as 2D translation, with one extra coordinate and one extra translation.

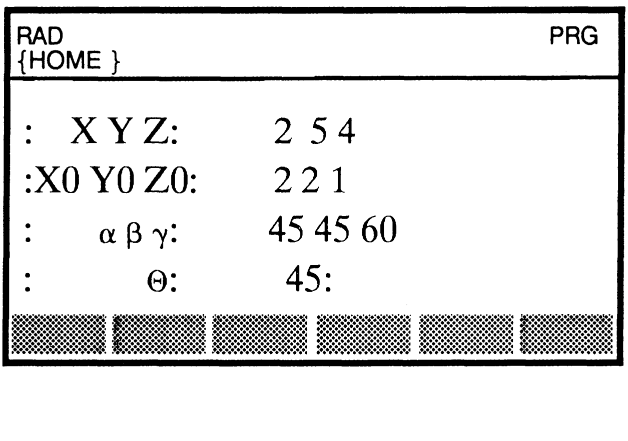
Scaling

The interface is the same as 2D scaling, with one extra coordinate and one extra scaling.

Rotation

3D rotation is somewhat more complicated than in two dimensions. The rotation axes has to be specified, i.e angles relative the coordinate axes and a point.

Interface:

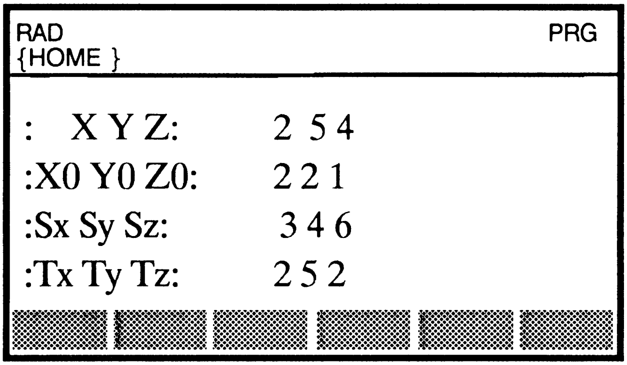


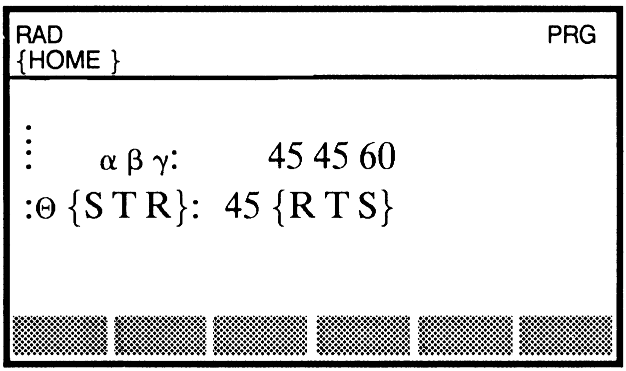
The example rotates the point (2,5,4) about an axes through the point (2,2,1) and with angles relative the x-, y-, and z-axes equal to 45°, 45° and 60°.

Concatinating (mixed)

The same interface as in the 2D case, but the rotation axes now has to be specified.

Interface:





In the example the point (2,5,4) is rotated about the given axes and then the specified translation and scaling is carried out.

Eigenvalueproblems

Here you may find the eigenvalues and the eigenvectors of a matrix, diagonalize a matrix and solve a system of linear differential equations.

Eigenvalues

The eigenvalues of a matrix are determined by the equation $\mathbf{A}*\mathbf{X}=\lambda*\mathbf{X}$, where **A** is the matrix and **X** a column vector (eigenvector). λ is called the eigenvalue.

Interface:

RAD PRG
{HOME }

$$A*X=\lambda*X$$
$$\mathrm{Det}(A-\lambda*I)=0$$

:PartAns Y/N: Y

Now the matrix has to be specified, and you may put it into the symbolic matrix writer:

$$\begin{matrix} 4 & 6 & 6 \\ 1 & 3 & 2 \\ -1 & -5 & -2 \end{matrix}$$

The example finds the eigenvalues of the matrix:

$$\begin{bmatrix} 4 & 6 & 6 \\ 1 & 3 & 2 \\ -1 & -5 & -2 \end{bmatrix}$$

The matrix has an eigenvalue with multiplicity 2 ($\lambda = 2$).

Eigenvectors

A matrix has infinite many eigenvectors because the system of equations that determines the vectors are indefinite. The eigenvectors are given in terms of arbitrary parameters. A set of eigenvectors will normaly be linear independent even if the eigenvalues have multiplicity greater than 1. But this is not always the case.

Interface:

```
RAD                              PRG
{HOME }
            A*X=λ*X
            Finds X
:        λ:  2
:PartAns Y/N:  Y
```

The matrix now has to be put into the matrix writer. We use the same matrix as in the determination of eigenvalues. By choosing PartAns Y the indefinite system of equations that determines the eigenvectors will be given. We see that only two of the eigenvectors are lineary independent (only one arbitrary parameter c).

Diagonalization

For a matrix **A** we can write:

$$\mathbf{D} = \mathbf{K}^{-1} * \mathbf{A} * \mathbf{K}$$

Here **K** is a matrix composed of the eigenvectors of **A** which have to be lineary independent. **D** is a diagonal matrix with the eigenvalues on the diagonal. If the eigenvectors are lineary dependent (as in the example of eigenvectors), then the matrix cannot be diagonalized (not diagonalizable).

Interface:

The matrix has to be put into the matrix writer:

-4	5	5
-5	6	5
-5	5	6

The output is the matrices **K** and **D**. The matrix $\mathbf{K}^{-1}$ may be found by inverting **K**.

> *Rem. If intermediate results are wanted, you may look at the problems of finding eigenvalues and eigenvectors separately.*

System of differential equations

Here a set of linear, homogenous differential equations with constant coefficients are solved by using the method of diagonalization.

Interface:

```
RAD                                PRG
{HOME }
        dX/dt= A*X

:X={x,y...}:  {X Y Z}
:          t:  t
```

$$\begin{matrix} -4 & 5 & 5 \\ -5 & 6 & 5 \\ -5 & 5 & 6 \end{matrix}$$

The following system is solved:

$$\begin{aligned} dx/dt &= -4x + 5y + 5z \\ dy/dt &= -5x + 6y + 5z \\ dz/dt &= -5x + 5y + 6z \end{aligned}$$

The output contains the constants C1, C2 and C3 and the independent variable is t.

Rem. If intermediate results are wanted, you may look at the problems of finding eigenvalues and eigenvectors separately.

Under Info some information about the solving strategy is given.

Vector spaces

A vector space is a collection of vectors relative a basis where certain operations on them are defined. A basis is a set of linear independent vectors from the space. In an orthogonal basis the vectors are mutualy orthogonal (inner product equals zero).

Basis?

This routine examines whether a set of vectors in the space is linear independent. The vectors are put into the matrix writer as rows and the output is logic 0 or 1.

Interface:

4	6	6
1	3	2
-1	-5	-2

The example examines whether the vectors in $\mathbf{R}_3$ {4 6 6}, {1 3 2} and {-1 -5 -2} are lineary independent and then form a basis in $\mathbf{R}_3$ (three dimensional vector space).

Norm

Here the length or absolute value is calculated. The input vector is $\{v_1\ v_2\ v_3 ....\}$ and the output is a number or an expression if the vector is symbolic.

Norming

A vector is transformed into an unit vector
$e = V/NORM(V). V = \{v_1\ v_2 ....\}$.

Scalar product (inner product)

The scalar product of two vectors is calculated. Symbolic vectors are possible.

Orthogonalization

An orthogonal basis is calculated with an arbitrary basis as a starting point using the Gram-Schmidt process.

Interface:

4	6	6
1	3	2
-1	-5	-2

The basis b1 = [4 6 6], b2 = [1 3 2] and b3 = [-1 -5 -2] is given in $\mathbf{R}_3$. The basis is not orthogonal, but the routine makes it orthogonal.

Rem. Symbolic vectors are not possible

Orthogonal?

The routine examines whether a matrix is orthogonal. If the row vectors building up the matrix form an orthogonal basis, then the matrix is orthogonal.

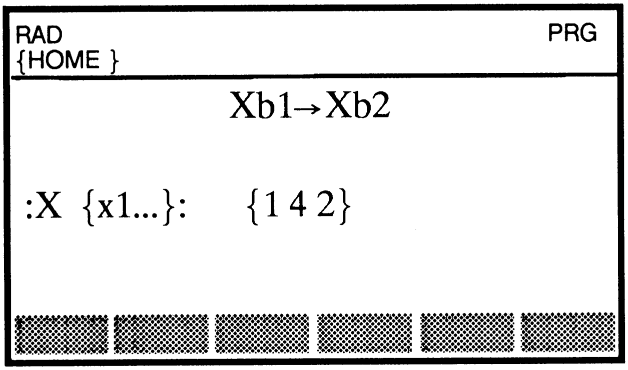
Orthonorming

The routine is norming an orthogonal basis.

Vector in new basis

Given a vector V_{B1}, i.e. relative a basis B1. A new vector relative a basis B2 is calculated.

Interface:



$$\begin{matrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 2 & 5 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 3 \\ 4 & 1 & 2 \\ 2 & 6 & 5 \end{matrix}$$

The vector {1 4 2} relative the basis { [1 2 3], [3 1 2], [2 2 5]} is transformed to the new basis {[113],[4 1 2],[2 6 5]}.

Transformation matrix in new basis

A matrix defines a linear transformation in a vector space relative the "natural" basis. This routine calculates a new transformation matrix relative a new basis. The "natural" basis is {[1 0 0],[0 1 0],[0 0 1]} in $\mathbf{R}_3$.

Interface:

1	1	0
0	1	1
1	0	1

1	1	1
0	1	1
0	0	1

The transformation matrix {{1 1 0}{0 1 1}{1 0 1}} in natural basis defines the transformation:

$$L(x_1,x_2,x_3) = (x_1 + x_2, x_2 + x_3, x_3 + x_1)$$

The example calculates the new transformation matrix relative basis {[1 1 1],[0 1 1],[0 0 1]}. Symbolic elements are possible in the matrices.

3

Laplace transforms

Laplace transforms are used for solving differential equations and can, contrary to other methods, deal with functions f(t) that are discontinous in the equation

$$a*y'' + b*y' + c*y = f(t)$$

Discontinous f(t) may be composed by using the Unit Step function u(t-a) defined as:

u(t-a): IF t a < THEN 0 ELSE IF t a > then 1 END END

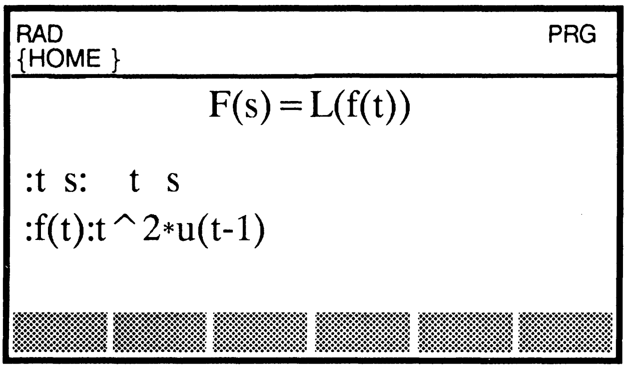
This function is not implemented in CALCULUS in other ways than as a symbol, and the user has to make a program to define it for evaluation.

Laplace transform

The Laplace transform of the following functions may be found:

- f(t) = t ^ n, n > -1
- f(t) = Sin(a*t), a arbitrary
- f(t) = Cos(a*t), a arbitrary
- g(t) = f(t)*e^{at}, a arbitrary
- g(t) = f(t)*u(t-a), a ≥ 0
- g(t) = f(t)*u(t-a)*e^{bt}, a ≥ 0 b arbitrary
- h(t) = g(t)*t
- Linear combinations of theese functions

Interface:



The example calculates the Laplace transform of f(t) = t ^ 2*u(t-1).

Rem. If CALCULUS cannot find the Laplace transform an error message is given (the transform does not exist or its not implemented)

Inverse Laplace transform:

The inverse transform is calculated. The types of functions which can be inverted are the transforms of the functions listed on page 37.

Interface:

RAD PRG
{HOME }

f(t) = InvL(F(s))

:s t: s t
:F(s): $(1\text{-}e^{-\pi s})/(s \wedge 2+1)$

The example calculates the inverse transform of
$F(s) = (1\text{-}e^{-\pi s})/(s^2 + 1)$.

Inverse L Partial fractions

If the denominator of F(s) is of second degree and may be factorized in first degree factors, or of a higher degree than 2, the denominator has to be split into partial fractions.

Intermediate results (Partial Answers) is possible to show the splitting into partial fractions.

> *Rem. If the transformation does not exist the error message "does not exist" is given. If the expression is too complicated the message "not rational" may appear. The expression may then be split up.*

Interface:

```
RAD                                   PRG
{HOME }
            F(s) = P(s)/Q(s)
            InvL(F(s)) = f(t)
:      t s:  t   s
:PartAns Y/N:Y
```

```
RAD                                   PRG
{HOME }
              Input cont
:Numerator P: 'Exp(-π*s)'
:Denominator Q:     's^2-1'
```

In the example $F(s) = e^{-\pi s}/(s^2-1)$ is split into partial fractions and then transformed. The shift $e^{-\pi s}$ will be taken care of before the splitting into partial fractions.

Differential equations (initial value problem)

Laplace transforms are suitable for solving initial value problems, in particular when the "right hand side function" is discontinous.

The answer is given in the form Y(s) = P(s)/Q(s)/R(s) which has to be transformed into P(s)/((Q(s)*R(s))) before the routine for partial fractions is used to solve the problem.

Interface:

```
RAD                                  PRG
{HOME }
           ay" + by' + cy = f(t)
          y(0) = y0 y' (0) = Dy0
:a b c y0 Dy0  :    1 3 2 0 1
:          f(t)  t:  'SIN(t)'  t
```

The equation y" + 3y' + 2y = Sin(t) with initial conditions y(0) = 0 and y' (0) = 1 is transformed.

4

Probability

In this chapter of probability theory we will look at unlike discrete probability distributions in addition to the normal distribution which is continous. For the discrete distributions both the cummulative probability and the point probability may be calculated.

For the discrete distributions and in connection with pure combinatorial calculations, we have distinquished between with and without replacement.

> *Rem. Probabilities must be less then or equal to 1 and greater than or equal to 0.*

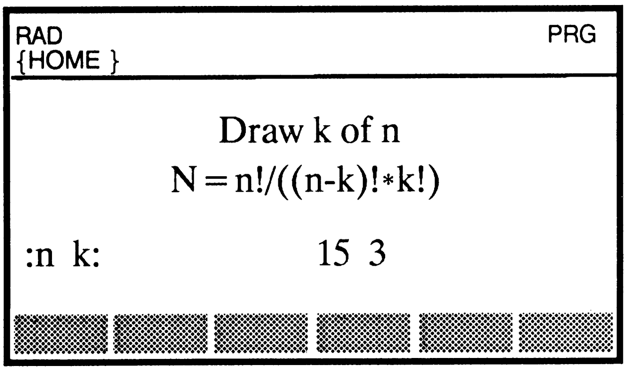
Without replacement

Without replacemnet means that we dont put the drawn element back again.

Combinations, not ordered

This routine calculates the number of possibilities to draw k elements of total n without replacement and without regard to order.

Interface:

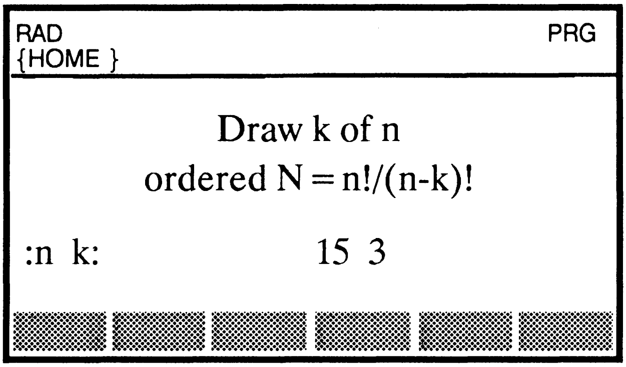


The example calculates the number og possibilities to draw 3 elements form total 15 elements without regard to order.

Combinations, ordered

If the order is important you will have to use this routine. The same elements in different orders will then be to separate events.

Interface:



The example calculates the number of combinations when drawing 3 elements from 15 with regard to order.

Hypergeometric distribution

Here the probability of drawing exactly k X'es from a population of n when a elements are drawn at a time without replacement is calculated. The probability for the X to be drawn is p.

Interface:

```
RAD                         PRG
{HOME }
          k X'es of n
        draw a P(X) = p
: n  a:    20  8
:p  k:     0.6  3
```

The probability that 3 elements have the mark X when drawing 8 elements of total 20 is calculated. The probability of X to occur is 0.6. If $k > a$ or $p > 1$ the probability is 0.

The example may be "drawing" individuals from a population of 20 where 12 is women ($p = 12/20 = 0.6$). The probability that of 8 "drawn" individuals 3 is women is calculated.

Hypergeometric distribution function

The cummulative probability is calculated, i.e. the probability that maximum k elements are drawn. This is the sum of the probabilities of $k = 0, k = 1, k = 2$ and $k = 3$.

Interface:

```
RAD                                PRG
{HOME }
            max k X'es of n
            draw a P(X) = p
: n  a:      20  8
:p  k:      0.6  3
```

The example calculates the probability that 3 elements is drawn with the mark X (p = 0.6) from a total of 20 by drawing 4 at a time or 1 by 1 without replacement.

With replacement

Here the elements are replaced by drawing so that the probability is the same every time an element is drawn (unconditional drawing).

Combinations, unordered
This routine calculates the number of combinations of drawing k elements from n without replacement, without regard to order.

Interface:

```
RAD                                   PRG
{HOME }
            Draw k of n
      N = (n + k-1)!/((n-1)!*k!)

:n  k:              15  3
```

Here the probability of drawing 3 elements of total 15 is calculated. Order is indifferent.

Combinations, ordered

If the order is critical, this routine has to be used. The same elements in different orders are two separate events.

Interface:

```
RAD                                   PRG
{HOME }
            Draw k of n
          ordered N = n ^ k

:n  k:              15  3
```

The example calculates the number of possibilities with the same figures as in the previous example, but now with regard to order.

Binomial distribution
The routine calculates the probability of drawing exactly k elements with the mark X of total n, where the probability of X itself is p. Independent trials (with replacement).

Interface:

```
RAD                                   PRG
{HOME }
              k X'es of n
               P(X) = p
: n p:    10  0.6
:   k:    3
```

The probability of drawing 3 elements with the mark X when X has the probability of 0.6 is calculated. The number of independent trials is 10. $p > 1$ gives an error message.

The example may be the production of glasses where the probability of first assortment is 0.6. If 20 glasses are produced, the example calculates the probability that 3 glasses are first assortment.

Binomial distribution function

The cummulative probability is calculated, i.e. the sum of the probabilities for k = 0, k = 1, k = 2 and k = 3 if k = 3.

Interface:

```
RAD                                PRG
{HOME }
-----------------------------------------
          max k X'es of n
              P(X) = p
: n p:   10  0.6
:   k:   3
```

The example calculates the probability that maximum 3 elements have the mark X (p = 0.6) in 10 independent trials.

Negative binomial distribution

This distribution gives the probability of k failures before the rth success in a series of independent trials each of which the probability of success is p.

Interface:

```
RAD                                PRG
{HOME }
        k  not X'es before
        X rth time P(X)=p
: r p:   10  0.25
:   k:   15
```

The probability of 15 failures before the 10th success when the probability of success is 0.25 is calculated.

The example may be the drawing of cards and the calculation of the probability of drawing 15 cards that are not clubs before the 10th club.

Negative binomial distribution function

This routine calculates probability of maximum k failures before the rth success.

Interface:

```
RAD                                  PRG
{HOME }
        max k not X'es before
         X rth time P(X) = p
: r  p:    5   0.4
:    k:    4
```

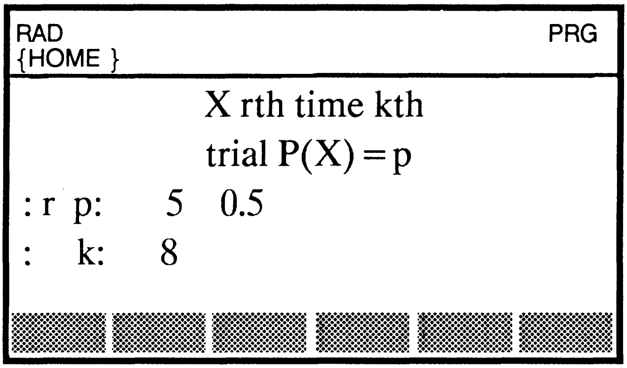
The probability of maximum 4 failures before the 5th success is calculated. Probability of success is 0.4.

The example may be the drawing of balls from a hat that contains 40% white balls. The probability of finding 5 not white balls before drawing maximum 4 white balls is calculated.

Pascal distribution

The probability of the rth success in kth trial in a series of independent trials is calculated.

Interface:



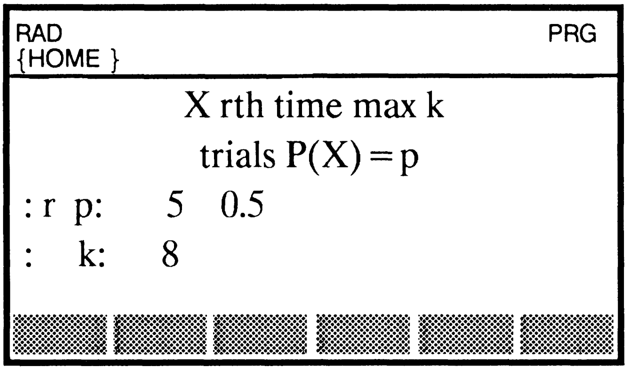
The probability of finding the mark X 5th time in the 8th trial is calculated.

> *Rem. The geometric distribution is a special case with r = 1.*

Pascal distribution function

The probability of the rth success in maximum k trials is calculated. The probability of success is p.

Interface:



The example calculates the probability of finding the mark X the 5th time in maximum 8 trials (Tossing a fair coin we find the probability of finding the 5th head in maximum 8 trials).

Normal distribution

This is a continous distribution and only cummulative probabilities are calculated.

Interface:

RAD PRG
{HOME }

Normal distribution
param. μ and σ gives $P(X \leq x)$
:μ σ: 0 1
: x: 1

The probability that a random variable is less than or equal to 1 is calculated. The mean and the standard deviation is 0 and 1.

Rem. $P(a \leq x \leq b) = P(x \leq b) - P(x \leq a)$ *and* $P(x > a) = 1 - P(x \leq a)$

Poisson distribution

The Poisson distribution is used as a model when we are interested in events within intervals of time or other variables.

Interface:

```
RAD                                  PRG
{HOME }
            Poisson distr.
        mean μ  and  P(X=k)

:μ k:   4 5
```

The example calculates the probability that a random variable X is exactly 5 when the mean is 4.

Poisson distribution function

Interface:

```
RAD                                  PRG
{HOME }
            Poisson distr.
        mean μ  and  P(X≤k)

:μ k:   4 5
```

The probability that X is less then or equal to 5 is calculated, the mean is 4.

Info

Here information about probability and some distributions is given.

Binomial coefficients

Binomial coefficients Bnk = n!/((n-k)!*k!) are calculated from k = 0 to k = n and put in a list.

Interface:

```
RAD                              PRG
{HOME }
         Binomial coeff.
        Bnk = n!/((n-k)!*k!)
             k = 0...n

:n:      5
```

The example calculates {Bn0 Bn1 Bn2 Bn3 Bn4 Bn5}.

5

Statistics

We will focus on some statistical methods and description of samples. Within description of samples we will use discrete tables and class tables (discrete and class statistics). You can convert from class statistics to discrete statistics by using the mean value of the intervals as the discrete value.

Statistical methods are represented by confidence intervals and hypothesis testing for distributions. The "best" fit for the normal distribution uses the method of least squares.

The normal distribution, kjisquare distribution and student-t distribution are included and its possible to find both the probability and the value of the random variable for given probability.

Distributions

Normal distribution

The normal distribution gives $p(X \leq x)$ for given x-value. The mean μ and standard deviation σ have to be known.

Interface:

```
RAD                                   PRG
{HOME }
          Normal distribution
   param. μ  and σ gives P(X≤x)
:μ σ:  0  1
:  x:  1
```

$P(X \leq 1)$ for $\mu = 0$ and $\sigma = 1$ is calculated.

Inverse normal distribution

The routine finds the value of the random variable x with given probability p, μ and σ are known.

Interface:

```
RAD                                   PRG
{HOME }
          Normal distribution
        μ σ gives  x, P(X≤x)=p
:μ σ:  0  1
:  p:  0.6
```

The value of x with $P(X \leq x) = 0.6$, $\mu = 0$ and $\sigma = 1$ is calculated.

Kjisquare distribution

The kjisquare distribution is used to find confidence intervals and in connection with fitting a distribution to a sample.

Interface:

```
RAD                                   PRG
{HOME }
           Kjisquare distr.
      degrees of freed. K P(X≤x)

:K  x: 3  5.6
```

The example calculates $P(X \leq 5.6)$ where X is kjisquaredistributed with 3 degrees of freedom.

Inverse kjisquare

The value of x for given probability is calculated.

Interface:

```
RAD                                PRG
{HOME }
           Kjisquare distr.
   degrees of freed. K P(X≤x)=p

: K  p: 3  0.85
```

The example calculates the value of X so that $P(X \leq x) = 0.85$ with 3 degrees of freedom.

Studen-t distribution

This distribution is used to find confidence intervals in CALCULUS.

Interface:

```
RAD                                   PRG
{HOME }
           Student-t distr.
    degrees of freed. K P(X≤x)

:K  x: 3  7.5
```

$P(X \leq 7.5)$ with 3 degrees of freedom is calculated.

Inverse student-t

This routine calculates the value of the random variable x.

Interface:

```
RAD                                   PRG
{HOME }
           Student-t distr.
    degrees of freed. K P(X≤x)=p

:K  p: 3  0.85
```

The value of x is calculated so that $P(X \leq x) = 0.85$ with 3 degrees of freedom.

Confidence intervals

Confidence intervals in connection with the normal distribution are calculated for the mean μ and variance σ^2.

> *Rem. Mean value and standard deviation for a sample may be calculated under this menu. Theese values may be used as point estimates for the parameters in the distribution function.*

Confidence interval for the mean μ, given value of σ.
For known σ we may use the normal dsitribution to find the value c so that $F(c) = P(x \leq c) = 1/2(\gamma + 1)$ with confidence level γ. The interval is given in the form $[a \leq \mu \leq b]$.

The interval is calculated from a sample [[xi]] and the mean value has to calculated in advance by using the menu option: Mean value.

Interface:

```
RAD                                    PRG
{HOME }
        CONF μ, given σ  Mean
        x̄ sample [[xn]] Level γ

: σ  γ: 1.12  0.90
: x n: 3 20
```

The example calculates the confidence interval for the mean in the normal distribution, based on a sample [[xn]] with mean 3 and confidence level 90%. The number of values in the sample is 20 and the normal distribution has the standard deviation 1.12.

Confidence interval for the mean μ, unknown σ

If σ is not known the estimate s for standard deviation from the sample is used. To find the value of c so that $F(c) = 1/2(\gamma + 1)$, the student-t distribution is used..

The interval is calculated from a sample [[xi]] with n values and the mean and the standard deviation of the sample has to be calculated in advance.

Interface:

```
RAD                                   PRG
{HOME }
        CONF μ,  Mean/St.dev.
        x̄ ,s samp.  [[xn]] Level γ

: γ s:  0.90  1.2
: x n: 3 20
```

The confidence interval for the mean is calculated based on a sample with 20 values, standard deviation 1.2, mean 3 and with confidence level 90%.

Confidence interval for variance, μ is unknown

The standard deviation of the sample has to be calculated first (separate menu option). The calculation is based on the kji-square distribution.

Interface:

RAD PRG
{HOME }

CONF σ^2, St.dev. s
sample [[xn]] Level γ

: γ s: 0.90 1.2
: n: 20

The example calculates the confidence interval for σ^2 based on a sample with standard deviation 1.2, confidence level 90% and 20 values in the sample.

Sample $\overline{x}$, s, n and median

The mean value, standard deviation, number of values and the median are calculated. The table is stored as UΣDAT.

Interface:

RAD PRG
{HOME }

St.dev. s and mean $\overline{x}$
[[x1 x2 x...]]

: [[x1 x2...]]: [[2 3 7]]

The mean value, standard deviation, number of values and the median for the sample [[2 8 7]] are calculated.

Fitting

By using a sample of values, the "best" fit for the normal distribution is calculated, i.e. estimates for the mean μ and standard deviation σ is calculated. Hypothesis testing for assumed distribution is done (kjisquare goodness of Fit).

The sample has to be given as a class statistic with given frequences (se description of samples).

In order to calculate estimates for the the distribution parameters a discrete statistic has to be stored as ΣDAT (see separate menu option).

Normal distribution
A "best" fit based on the least squares is calculated. The sample has to be stored as a class statistic (KΣDAT) in advance.

When KΣDAT is stored by using a separate menu option, there is no more input data necessary. Estimates for μ and σ will be calculated.

Hypothesis normal distribution

The class table is stored by using the separate menu option. The separation in different classes is done with minimum 5 values in each class. Upper and lower limit has to be ∞ and $-\infty$ and you can achieve this by using big numbers as the lower and upper limit for the class intervals.

For the calculation of mean and standard deviation as estimates for the parameters the discrete statistic has to be stored with known frequencies (ΣDAT).

Interface:

```
RAD                                PRG
{HOME }
           Normald. n values
           Level α Num.est. r
: μ σ:        360    26
:α n r:       0.05 100 2
```

The example is testing whether the sample KΣDAT may be fitted to a normal distribution with significance level 5%. μ and σ are estimates and r, number of estimates, is 2. Number of values in the sample is 100.

The Kjisquare distribution is used and the figure $kij0^2$ is tested against the theoretical value c, $P(X \leq c) = 1-\alpha$, where α is the significance level. If $kji0^2 \leq c$, the hypothesis is not rejected.

Hypothesis binomial distribution

The class statistic is stored and the parameter p is, if necessary, estimated as $p = \mu/n$.

Interface:

```
RAD                                PRG
{HOME }
          Binomiald. n values
          Level α Num.est. r
: p α:        0.5     0.05
:n r:        50 1
```

The example is testing whether the sample may be fitted to a binomial distribution with significance level 5%. p is estimated (0.5) and the value of r is 1. Number of values is 50.

Hypothesis Poisson distribution
The class statistic is stored and the parameter μ (mean) is, if necessary, estimated.

Interface:

```
RAD                                PRG
{HOME }
           Poissond. n values
           Level α Num.est. r
: μ α:         56     0.05
:n r:       100  1
```

The example is testing whether the sample may be fitted to a Poisson distribution with significance level 5%. μ is estimated and $r = 1$. Number of values in the sample is 100.

Class tabel (class statistic)
A class table is a double list {{Ik}{fk}} where the first list is the limits of the class intervals and the other list is the number of values in the separate intervals.

Interface:

```
RAD                                  PRG
{HOME }
              Class table

:{{Ik}{fk}}:  {{3 4 6 7}{2 5 4}}
```

Interval limits are 3,4,6 og 7 and number of values in the intervals are 2,3 and 4.

Mean value and standard deviation based on frequency table

If the data is stored as a frequency table, we cannot use the ordinary sample routine to find the mean etc. The data is stored in ΣDAT. Use F for frequency table.

Storing discrete table

The sample is stored as ΣDAT. The data is put directly into the matrix writer where the first column contains the values and the second column the frequencies.

Description of samples

In this menu some calculations on discrete data and class tables from samples are done. This includes mean values, standard deviation, relative frequencies, histograms and frequency polygons.

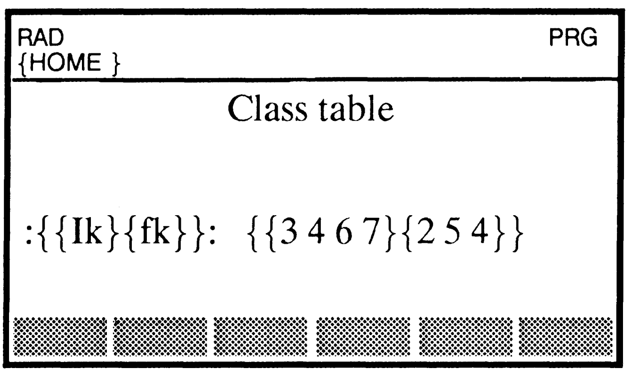
Discrete table ΣDAT

Here the values are stored in the first column and the frequencies in the second. In two variable statistics the second column will be the data for the second variable.

Classes KΣDAT

A class table is a double list {{Ik}fk}} where the first list is the limits of the intervals and the other the numbers in each interval.

Interface:



The limits of the intervals are 3,4,6 and 7 and the number of values is 2,3 and 4. The table is stored as KΣDAT for further uses.

Cummulative table.

The routine is calculating a table from a discrete table (ΣDAT). The new table will include relative frequencies, (column 4) cummulative frequencies (column 3) and cummulative relative frequencies (column 5).

KΣDAT→ΣDAT

This routine transforms a class statistics to a discrete statistics by using the mean value of each interval as the representative value. The table is stored as ΣDAT. Input data is KΣDAT which is stored in advance.

ΣDAT $\bar{x}$ og s

The mean value and the standard deviation is calculated based on a frequency table (F) or a two variable statistic table (D).

Interface:

```
RAD                         PRG
{HOME }
       F is frequency tab.
        D is two var. stat.

: F or D:  D
```

The sample has to be stored in ΣDAT and the choice D will give the mean and standard deviation for each variable in a two variable statistics.

> *Rem. Under menu confidence intervals and STAT menu the mean and standard deviation for simple samples are calculated (one dimensional tables).*

Histogram KΣDAT
Creates a histogram based on the class table KΣDAT which has to be stored in advance.

Frequency polygon KΣDAT

Creates a cummulative frequency polygon from KΣDAT.

Linear regression and correlation

A straight line is fitted by the use of least squares from the table ΣDAT. The line will be given as $Y = aX + b$, where X is the data in the second column in ΣDAT.

The correlation coefficient is a measure of the goodness of the fit and a value between -0.7 and 0.7 is a good fit.

Rem. ΣDAT is now a two variable statistics table and not a simple statistics frequency table.

6

Fourier series

Under Fourier series in symbolic form one can find the Fourier series of polynomials up to 2nd degree and a copule of other possibilites. The numeric series for a given number of terms may also be found. The input function may be bifurcated with different expressions in two different intervals.

The Fourier series are generaly given as:

$$f(x) = a0 + \sum_{n=1}^{\infty} an*Cos(2*\pi*x*n/T) + \sum_{n=1}^{\infty} b_n*Sin(2*\pi*x*n/T)$$

T is the period and the coefficients are given:

$$a0 = 1/T \int_{-T/2}^{T/2} f(x)dx$$

$$an = 2/T \int_{-T/2}^{T/2} f(x)*Cos(2*\pi*x*n/T)dx$$

$$bn = 2/T \int_{-T/2}^{T/2} f(x)*Sin(2*\pi*x*n/T)dx$$

CALCULUS is able to find the series of the following types of functions:

- $f(x) = kx + b$
- $f(x) = kx^2$
- $f(x) = k\text{Sin}(ax)$
- $f(x) = k\text{Cos}(ax)$
- $f(x) = ke^{ax}$

Fourier series, symbolic form

The expressions for a0, an and bn are found for given f(x). The series itself must be set up by the user. The function f(x) may be given as two expressions in two intervals:

$$f(x) = \begin{cases} f1, \; a < x < b \\ f2, \; c < x < d \end{cases}$$

The functions are given as {f a b} where a and b are defining the interval or bifurcated as {f1 a b f2 c d}.

Rem. If the function is split in more than two intervals, CALCULUS may be used on two and two (or two plus one) intervals.

Interface:

```
RAD                                PRG
{HOME }
      f(t)=a0+Σan*COS(ω*t)+
      Σbn*SIN(ω*t)  ω=2*π*n/T
:  t T: t  '2*π'
:PartAns Y/N:  Y
```

```
RAD                                PRG
{HOME }
            Input cont


:{f a b ...}: {-1 '-π' 0 1 0 'π'}
```

The example calculates the Fourier coefficients of

$$f(t) = \begin{cases} -1 & -\pi < t < 0 \\ 1 & 0 < t < \pi \end{cases} \qquad T = 2\pi$$

The answer is given with intermediate results (indefinite integrals are given).

Fourier series numeric results

A specified number of terms are calculated, but in terms of the independent variable. The input is the same as in symbolic form, but the number of terms has to be included and also the start and stop of the summation index. The integration is numeric and intermediate results are not given.

Interface:

```
RAD                                   PRG
{HOME }
   f(t) = a0 + Σ(m,n)an*COS(ω*t) +
   Σ(m,n)bn*SIN(ω*t)  ω = 2*π*n/T

:   t T: t  '2*π'
```

```
RAD                                   PRG
{HOME }
               Input cont

:      m n: 0 2
:{f a b ...}: {-1 '-π' 0 1 0 'π'}
```

The first three terms are calculated. The terms n = 0 and n = 2 are 0.

Rem. The accuracy of the integration is dependent of the choice for N FIX on the calculator

Half range expansions

In many situations there is a practical need to use Fourier series in connection with functions that are given merely on some definite interval. They may be done periodic by an extension with the period as the double interval. The extension may be even or odd by choice (E/O).

Interface:

```
RAD                                   PRG
{HOME }
      f(t) = a0 + Σan*COS(ω*t) +
      Σbn*SIN(ω*t)  ω = 2*π*n/T
:  t T: t  '2*π'
:PartAns Y/N:  Y
```

```
RAD                                   PRG
{HOME }
              Input cont

:{f a b ...}: {-1 '-π' 0 }
:     O/E:        O
```

Here $f(t) = -1$, $-\pi < t < 0$ is given. An odd extension is marked by O.

7

Linear programming

The maximum or minimum of a linear function in several varaibles are calculated. The constraints are given as inequalities ("less then"). The routine does not handle degeneracy or solutions constrained to be natural numbers.

> *Rem. If such constraints are given and a decimal number is the answer, one cannot simply round off to nearest natural number. This will not always give the optimal solution.*

The algorithm used is the simplex method and it finds only the minimum of an object function. Any problem can be written as a minimum problem. If a maximum value of f(x) is going to be found, one may simply change the sign and find the minimum of -f(x).

The constraints are defining the constraint matrix A. They are assumed to be of the "less then" type, but if you have got them as "greater then" type you may simply multiply both sides with -1 and change the inequality sign.

If we are going to maximize $f(x) = x1 + 2x2$ and one of the constraints are $x1 - 2x2 > 2$, then minimize $-x1 - 2x2$ with the constraint $x1 + 2x2 < -2$.

All independent variables are assumed to be positive or zero.

> *Rem. The independent variables have the symbol xi regardless of the symbols used in a given problem.*

Interface:

```
RAD                              PRG
{HOME }
        min f(X) = C*X   A*X ≤ B
            A  is the last input
:B[ ]:    [ 8 5]
:C[ ]:       [ -25 -7 -24]
```

A:

3	1	5
5	1	3

The example solves the problem of finding the maximum of

$$f(x1,x2,x3) = 25x1 + 7x2 + 24x3 \text{ under the constraints}$$

$$\begin{gathered} 3x1 + x2 + 5x3 \leq 8 \\ 5x1 + x2 + 3x3 \leq 5 \\ 0 \leq x1,x2,x3 \end{gathered}$$

C vector is the object function f, B vector is the right side of the constraints and A is the constraint matrix (left side). The solution is given as -39.5 which means that the maximum is 39.5.

8

Numerical solution diffequations

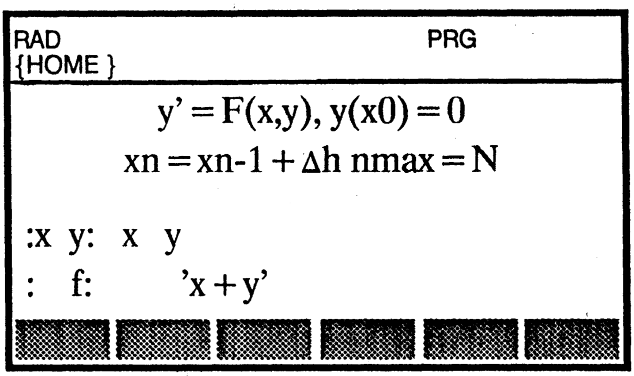
First order differential equations may be written

$$y' = F(x,y),\ y(x0) = y0$$

F(x,y) is a function in two variables x and y where y is dependent of x. The solution is y = f(x). The value of y at a given point x0 has to be known.

The method used is the 4th order Runge-Kutta (RK4). This will give a fair good solution for "non-stiff" problems.

Interface:



```
RAD                                   PRG
{HOME }
              Input cont
: x0  y0:  0    0
:Δh  N:     0.2  5
```

In the example the solution of $y' = x + y$, $y(0) = 0$ is tabulated.

College and University Mathematics

- Linear Algebra and Matrices
- Transformations
- Linear Equations Systems
- Eigenvalues and eigenvectors
- Linear Programming
- Laplace Transforms
- Fourier Series
- Statistics

With the symbolic
MATRIX WRITER

Odd Bringslid International Software Vendor
Box 1014 KNP
3601 Kongsberg
NORWAY
Tel. +47-3-73 88 40. Fax. +47-3-73 86 33

Distributor:
BB Marketing ANS. Ensjøveien 12 B, 0655 Oslo, Norway.
Tel. +47-2-67 11 57. Fax. +47-2-67 11 37