Using the Simplex Library (simplexlib.tns)

Forest W. Arnold

January 2019

Typeset in LaTeX.

Trademarks

1 Purpose

The functions in the simplex library,`simplexlib.tns`, implement the simplex method for solving linear programs. Given a linear function, called an **objective function** and a set of linear relations, called **constraints**, that limit the values of the independent variables within a certain range of values, the simplex method finds the values of the independent variables that optimize (minimize or maximize) the objective function. The simplex method is widely used in industry and government because of its ability to solve a large variety of complex problems in a reasonable amount of time.

The library can be installed and used with TI-Nspire CX CAS, TI-Nspire CAS Student Software, TI-Nspire CAS Teacher Software, and TI-Nspire CAS App for the iPad.

This library is designed solely for finding solutions for the types of linear programming problems commonly encountered in an introductory operations research textbook or a linear algebra textbook. There is no guarantee that the functions can be used for more general linear programs.

2 Restrictions/Limitations

1. The maximum number of variables is limited to 30. This number includes both objective function and constraint variables. The library has been tested with linear programs involving 21 variables.

2. Variables for the primary problem must be named $x1, x2, ..., xn$ and variables for the dual problem (described below) must be named $y1, y2, ..., yn$, where xn, yn is the highest numbered variable. These names are used by the software to determine the number of variables and their positions in the solution matrix.

3. Values must not be assigned to the named variables, since TI-Nspire replaces the names in the expressions for the objective function and the constraints with any values assigned to the variables.

4. Values in the solution matrix are scaled using the absolute value of the largest value in the matrix to prevent calculation errors that occur when both relatively large and small values are present.

5. The library functions require the CAS version of TI-Nspire.

3 Library History

Author: Forest W. Arnold
Date Created: 20 December 2018
Last Modified: 4 January 2019
Version: 1.0

4 Example Problem

The following objective function and associated set of constraints is used to demonstrate usage of the public library functions:

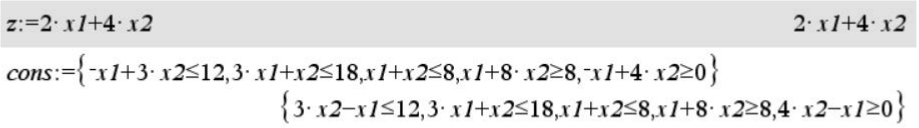
Objective function:

$$z = 2x1 + 4x2$$

Objective constraints:

$$\begin{aligned} -x1 + 3x2 &\leq 12 \\ 3x1 + x2 &\leq 18 \\ x1 + x2 &\leq 8 \\ x1 + 8x2 &\geq 8 \\ -x1 + 4x2 &\geq 0 \\ x1, x2 &\geq 0 \end{aligned}$$

The objective function and constraints are defined in a TI-Nspire calculator page as follows:



```
z:=2· x1+4· x2                                                    2· x1+4· x2
cons:={⁻x1+3· x2≤12,3· x1+x2≤18,x1+x2≤8,x1+8· x2≥8,⁻x1+4· x2≥0}
                  {3· x2−x1≤12,3· x1+x2≤18,x1+x2≤8,x1+8· x2≥8,4· x2−x1≥0}
```

Figure 1 shows plots of the objective function, the constraints, the feasible region (solution space), and the extreme points of the solution space for the example problem.

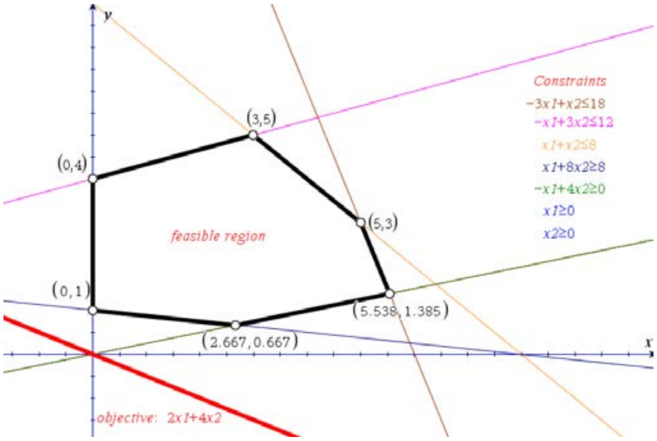


Figure 1: Graph of the Example Problem

5 Functions

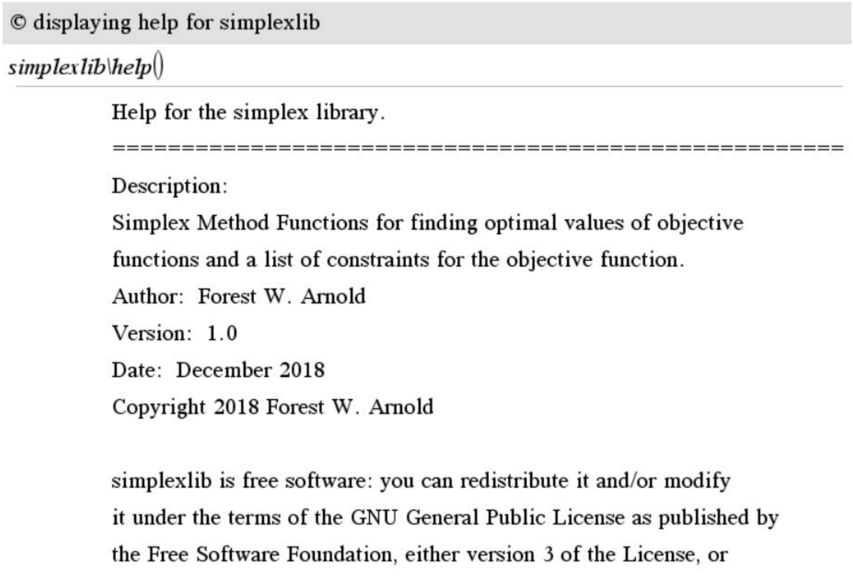
The following paragraphs describe each of the public functions in the library and present examples of their use. Although the examples are simple, they illustrate the way the functions are used for larger, more complex linear programs.

5.1 help()

Displays usage information for the functions in the library.

5.1.1 Example

Select the help function in the library pane to add it to a calculator page, then execute the function:



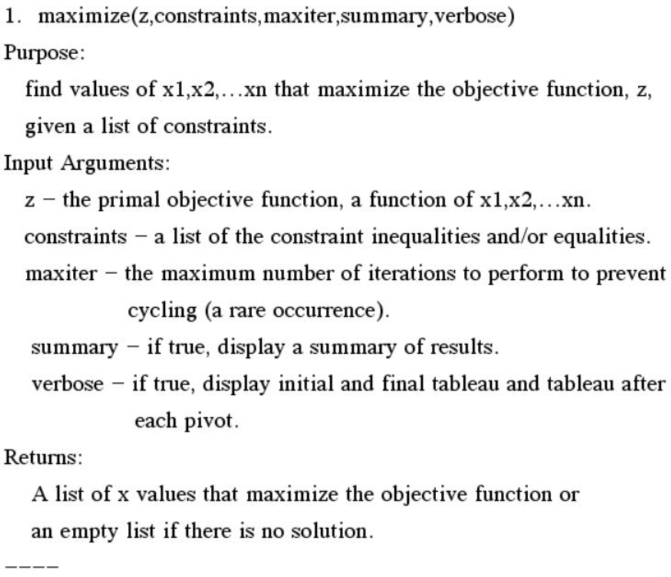
© displaying help for simplexlib

simplexlib\help()

```
Help for the simplex library.
==========================================================
Description:
Simplex Method Functions for finding optimal values of objective
functions and a list of constraints for the objective function.
Author:  Forest W. Arnold
Version:  1.0
Date:  December 2018
Copyright 2018 Forest W. Arnold

simplexlib is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
```

Scroll to display information about a library function:



```
1.  maximize(z,constraints,maxiter,summary,verbose)
Purpose:
  find values of x1,x2,...xn that maximize the objective function, z,
  given a list of constraints.
Input Arguments:
  z − the primal objective function, a function of x1,x2,...xn.
  constraints − a list of the constraint inequalities and/or equalities.
  maxiter − the maximum number of iterations to perform to prevent
              cycling (a rare occurrence).
   summary − if true, display a summary of results.
   verbose − if true, display initial and final tableau and tableau after
               each pivot.
Returns:
   A list of x values that maximize the objective function or
   an empty list if there is no solution.
----
```

5.2 maximize(z,constraints,maxiter,summary,verbose)

Given the expression defining an objective function, $f(x1, x2, ...xn)$, and a list of constraints, find the values of the independent variables, $x1, x2, ...xn$ that maximize the value of the objective function.

Input arguments:

z - the expression defining the objective function. The variables in the expression for the primal objective function must be labeled $x1, x2, ...xn$ or $y1, y2, ...yn$ for the dual objective function.

constraints - a list of the constraints. The variables in the primal constraints must be labeled $x1, x2, ...xn$ or $y1, y2, ...yn$ for the dual constraints. The constraints may be inequality constraints ($\leq, \geq$) or equality constraints ($=$).

maxiter - the maximum number of iterations to process. This value is a ”fail-safe” to prevent an endless number of iterations in case a cycle occurs. In general, cycles do not occur, but are possible. Most small problems require less than five iterations and larger problems (6 or more variables and 8 or more constraints) usually terminate after 10-15 iterations.

summary - if `true`, display a summary of the results, If `false`, omit displaying a summary.

verbose - if `true`, display the initial tableau, the tableau at each iteration, and the final tableau. If `false`, do not display the tableaus.

Returns:

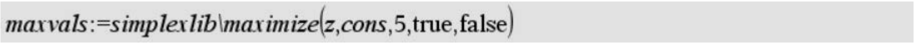
A list containing the variables and their values that maximize the objective function. Displays a message and returns an empty list if the feasible region (solution space) is empty or unbounded, or if the maximum number of iterations is exceeded.

Note:

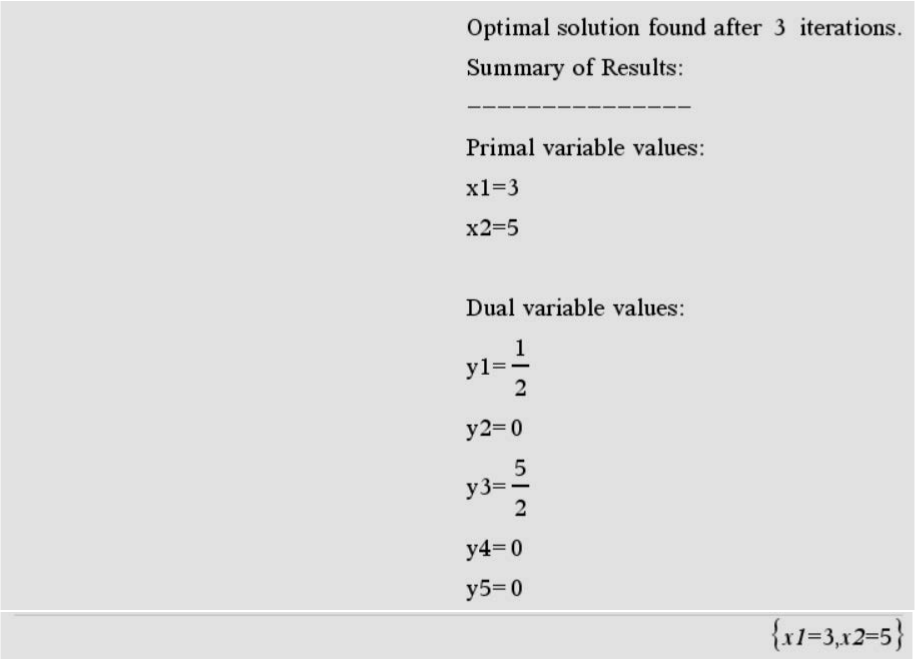
If a constraint variable is unrestricted, subtract an auxillary variable from the unrestricted variable and replace the unrestricted variable with the difference. For example, suppose there are only two variables in the linear program and a constraint is $x1 + 2x2 <= 4$ with $x2$ unrestricted, add the auxillary variable $x3$ to the list of variables and replace $x2$ with $x2 - x3$ in the constraints where $x2$ occurs. The new constraint for $x1 + 2x2 <= 4$ becomes $x1 + 2(x2 - x3) <= 4$.

5.2.1 Example 1

Maximize the objective function subject to the constraints defined above for the example problem. Display the summary results.



```
maxvals:=simplexlib\maximize(z,cons,5,true,false)
```



```
Optimal solution found after 3 iterations.
Summary of Results:
---------------
Primal variable values:
x1=3
x2=5

Dual variable values:
y1=1/2
y2=0
y3=5/2
y4=0
y5=0
```

$\{x1=3, x2=5\}$

5.3 minimize(z,constraints,maxiter,summary,verbose)

Given the expression defining an objective function, $f(x1,x2,...xn)$, and a list of constraints, find the values of the independent variables, $x1,x2,...xn$ that minimize the value of the objective function.

Input arguments:

> z - the expression defining the objective function. The variables in the primal objective function must be labeled $x1,x2,...xn$ or $y1,y2,...yn$ for the dual objective function.
>
> constraints - a list of the constraints. The variables in the primal constraints must be labeled $x1,x2,...xn$ or $y1,y2,...yn$ for the dual constraints. The constraints may be inequality constraints $(\leq,\geq)$ or equality constraints $(=)$.
>
> maxiter - the maximum number of iterations to process. This value is a ”failsafe” to prevent an endless number of iterations in case a cycle occurs. In general, cycles do not usually occur, but are possible. Most small problems require less than five iterations and larger problems (6 or more variables and 8 or more constraints) usually terminate after 10-15 iterations.
>
> summary - if `true`, display a summary of the results, If `false`, omit displaying a summary.

verbose - if `true`, display the initial tableau, the tableau at each iteration, and the final tableau. If `false`, do not display the tableaus.

Returns:

A list containing the variables and their values that minimize the objective function. Displays a message and returns an empty list if the feasible region (solution space) is empty or unbounded, or if the maximum number of iterations is exceeded.

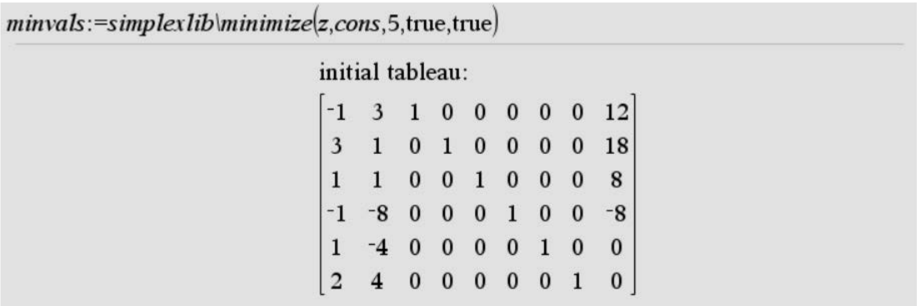
Notes:

1. Since maximizing the negative of an objective function minimizes the objective function, this function multiplies `z` by -1 and calls `maximize()` to find the values of the variables that minimize the objective function.

2. If a constraint variable is unrestricted, subtract an auxillary variable from the unrestricted variable and replace the unrestricted variable with the difference. For example, suppose there are only two variables in the linear program and a constraint is $x1+2x2<=4$ with $x2$ unrestricted, add the auxillary variable $x3$ to the list of variables and replace $x2$ with $x2-x3$ in the constraints where $x2$ occurs. The new constraint for $x1+2x2<=4$ becomes $x1+2(x2-x3)<=4$.

5.3.1 Example 2

Minimize the objective function subject to the constraints defined above for the example problem. Display the tableaus and the summary results.



minvals:=*simplexlib\minimize*(*z*,*cons*,5,true,true)

initial tableau:

$$\begin{bmatrix} -1 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 12 \\ 3 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 18 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 8 \\ -1 & -8 & 0 & 0 & 0 & 1 & 0 & 0 & -8 \\ 1 & -4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

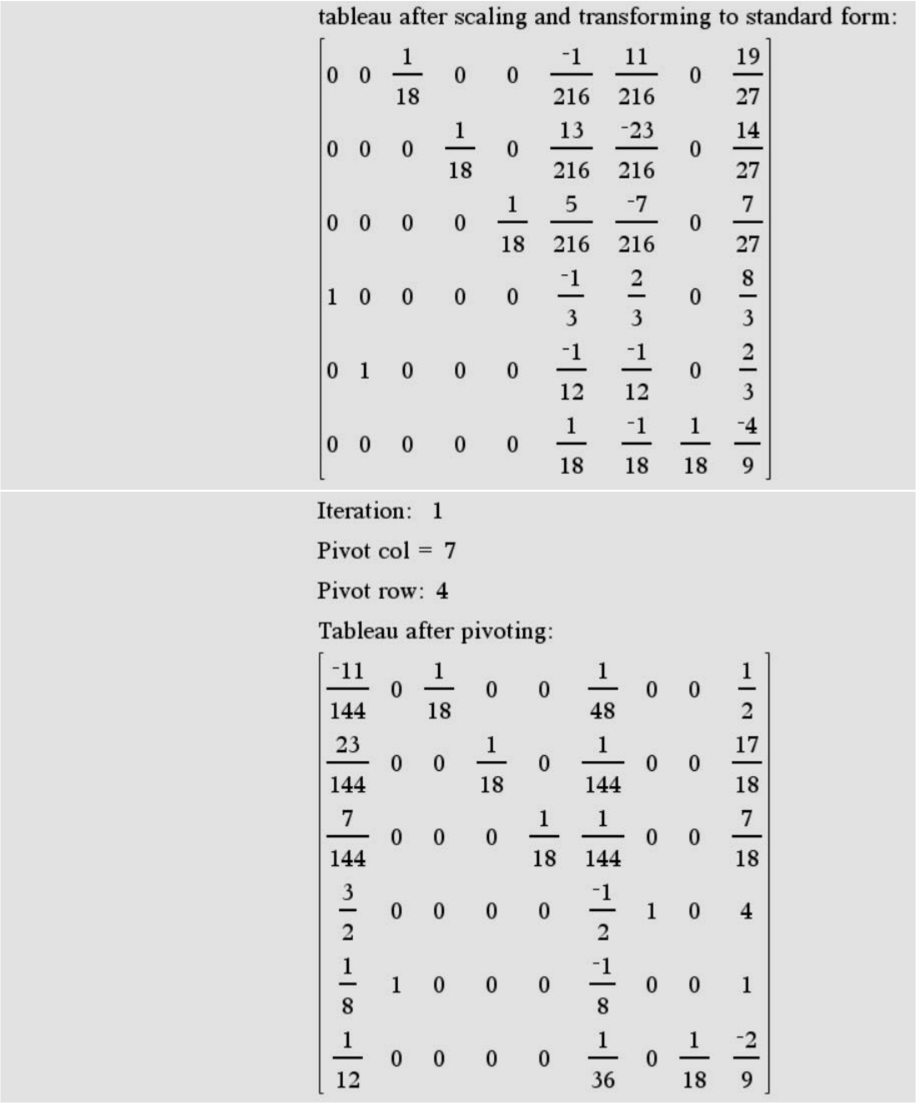


tableau after scaling and transforming to standard form:

$$\begin{bmatrix} 0 & 0 & \frac{1}{18} & 0 & 0 & \frac{-1}{216} & \frac{11}{216} & 0 & \frac{19}{27} \\ 0 & 0 & 0 & \frac{1}{18} & 0 & \frac{13}{216} & \frac{-23}{216} & 0 & \frac{14}{27} \\ 0 & 0 & 0 & 0 & \frac{1}{18} & \frac{5}{216} & \frac{-7}{216} & 0 & \frac{7}{27} \\ 1 & 0 & 0 & 0 & 0 & \frac{-1}{3} & \frac{2}{3} & 0 & \frac{8}{3} \\ 0 & 1 & 0 & 0 & 0 & \frac{-1}{12} & \frac{-1}{12} & 0 & \frac{2}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & \frac{-1}{18} & \frac{1}{18} & \frac{-4}{9} \end{bmatrix}$$

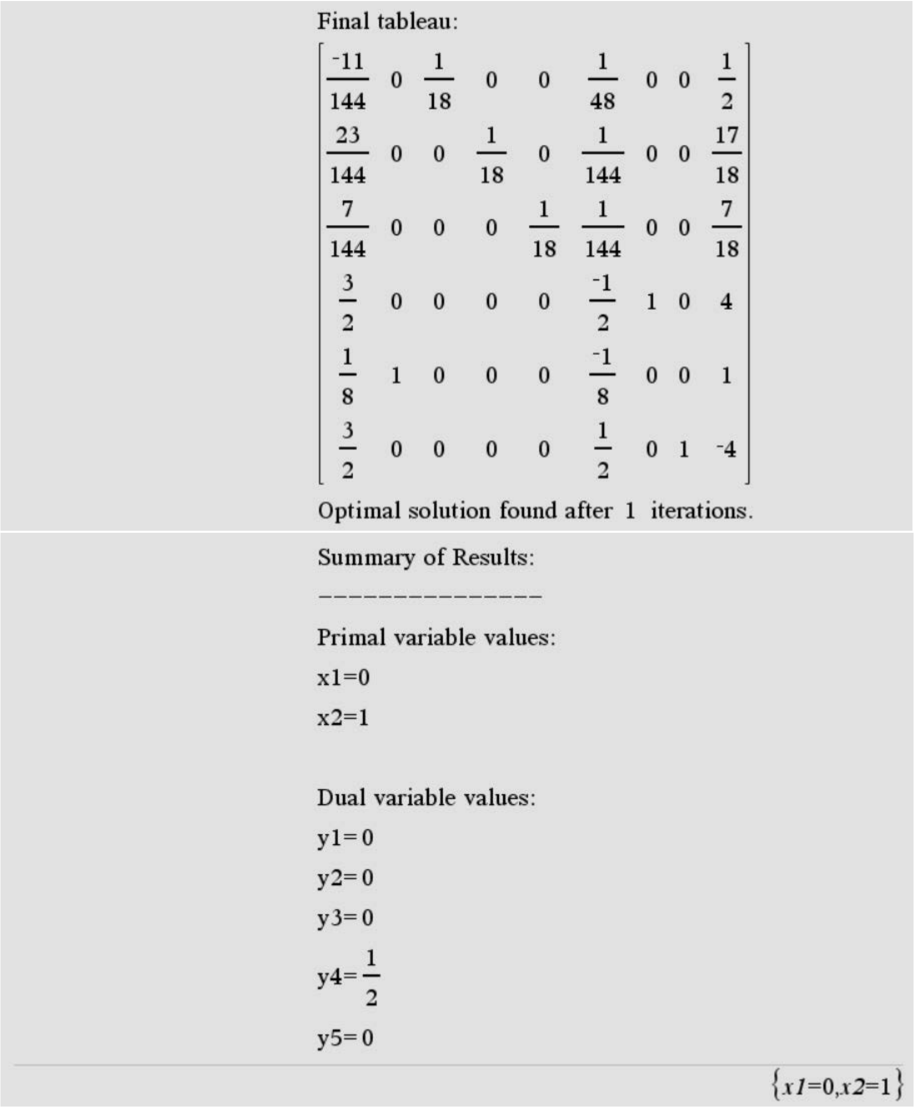
Iteration: 1

Pivot col = 7

Pivot row: 4

Tableau after pivoting:

$$\begin{bmatrix} \frac{-11}{144} & 0 & \frac{1}{18} & 0 & 0 & \frac{1}{48} & 0 & 0 & \frac{1}{2} \\ \frac{23}{144} & 0 & 0 & \frac{1}{18} & 0 & \frac{1}{144} & 0 & 0 & \frac{17}{18} \\ \frac{7}{144} & 0 & 0 & 0 & \frac{1}{18} & \frac{1}{144} & 0 & 0 & \frac{7}{18} \\ \frac{3}{2} & 0 & 0 & 0 & 0 & \frac{-1}{2} & 1 & 0 & 4 \\ \frac{1}{8} & 1 & 0 & 0 & 0 & \frac{-1}{8} & 0 & 0 & 1 \\ \frac{1}{12} & 0 & 0 & 0 & 0 & \frac{1}{36} & 0 & \frac{1}{18} & \frac{-2}{9} \end{bmatrix}$$



Final tableau:

$$\begin{bmatrix} \frac{-11}{144} & 0 & \frac{1}{18} & 0 & 0 & \frac{1}{48} & 0 & 0 & \frac{1}{2} \\ \frac{23}{144} & 0 & 0 & \frac{1}{18} & 0 & \frac{1}{144} & 0 & 0 & \frac{17}{18} \\ \frac{7}{144} & 0 & 0 & 0 & \frac{1}{18} & \frac{1}{144} & 0 & 0 & \frac{7}{18} \\ \frac{3}{2} & 0 & 0 & 0 & 0 & \frac{-1}{2} & 1 & 0 & 4 \\ \frac{1}{8} & 1 & 0 & 0 & 0 & \frac{-1}{8} & 0 & 0 & 1 \\ \frac{3}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 1 & -4 \end{bmatrix}$$

Optimal solution found after 1 iterations.

Summary of Results:

----------------

Primal variable values:

x1=0

x2=1

Dual variable values:

y1=0

y2=0

y3=0

$y4=\frac{1}{2}$

y5=0

$\{x1=0, x2=1\}$

5.4 evaluate(z,solutions)

Calculate the optimal value of the objective function *z* using the list of variables and values returned by `maximize()` or `minimize()`.

Input arguments:

z - the expression defining the objective function.

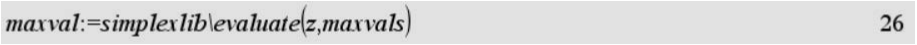
solutions - A list containing the variables and values that minimize or maximize the objective function.

Returns:

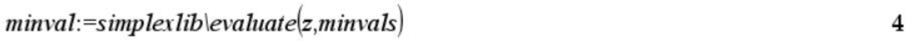
A real scalar value resulting from substituting the solution values for the variables in z, the objective function.

5.4.1 Example 3

Use the list of variables and values returned from `maximize()` and `minimize()` to find the maximum and minimum values of the objective function z:



```
maxval:=simplexlib\evaluate(z,maxvals)                    26
```



```
minval:=simplexlib\evaluate(z,minvals)                     4
```

5.5 dual(z,constraints)

Find the expression for the dual objective function and dual constraints associated with the primal objective function z and the primal constraints.

Input arguments:

z - the expression for the primal objective function.

constraints - the constraints for the primal objective function.

Returns:

A matrix containing the expression for the dual objective function followed by the dual constraints. The dual objective function is the row 1 entry and each constraint is an entry in the following rows.

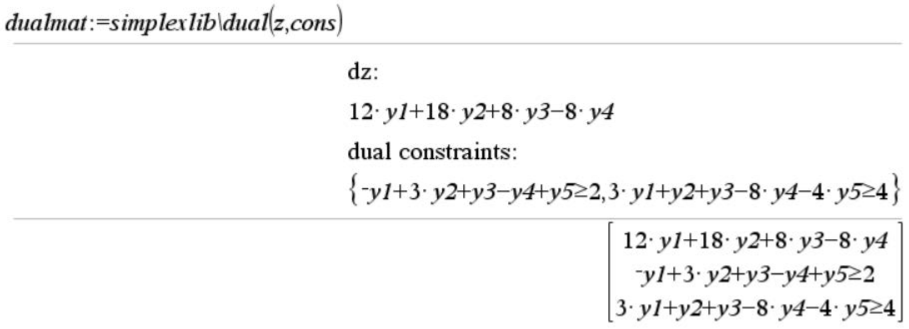
Note:

1. The input arguments must be the expression for the primal objective function and primal constraints; i.e., the variables must be labeled $x1, x2, ...xn$. This function does not process the dual of the dual.

2. The variables for the dual objective and dual constraints are labeled $y1, y2, ...yn$, where n is the nth variable. Do not assign values to these variables since TI-Nspire CAS will automatically replace the labels with their assigned values in the returned matrix.

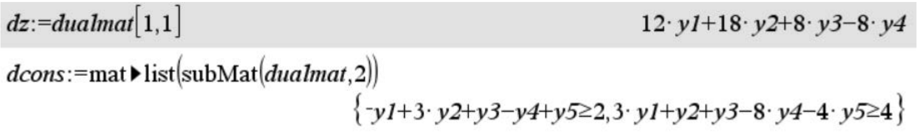
5.5.1 Example 4

Calculate the dual and dual constraints associated with the primal and primal constraints:



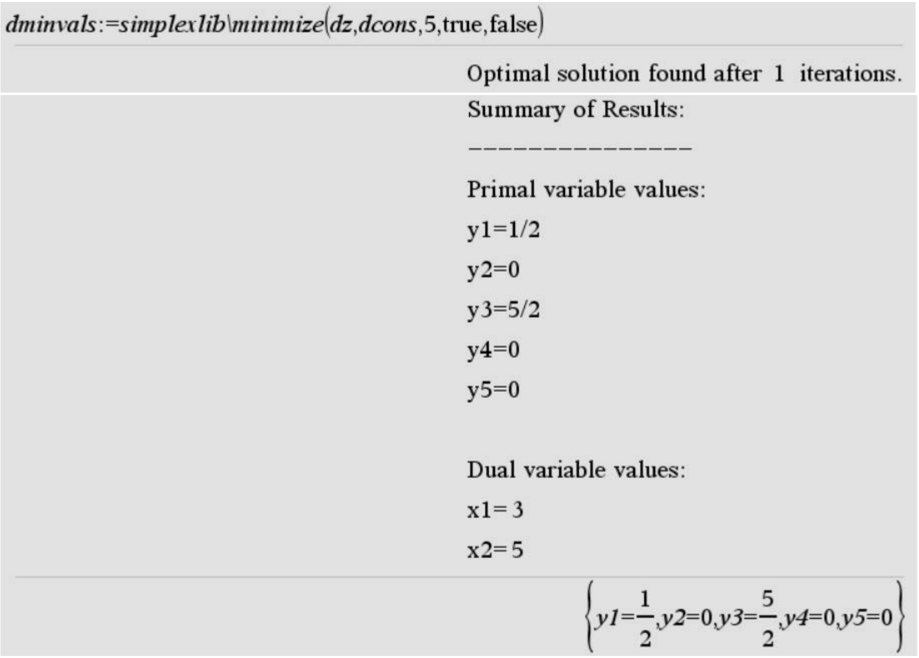
```
dualmat:=simplexlib\dual(z,cons)
                                   dz:
                                   12· y1+18· y2+8· y3−8· y4
                                   dual constraints:
                                   {⁻y1+3· y2+y3−y4+y5≥2,3· y1+y2+y3−8· y4−4· y5≥4}
                                                 [ 12· y1+18· y2+8· y3−8· y4 ]
                                                 [  ⁻y1+3· y2+y3−y4+y5≥2     ]
                                                 [3· y1+y2+y3−8· y4−4· y5≥4  ]
```

Extract the dual and dual constraints from the matrix returned from `dual()`:



```
dz:=dualmat[1,1]                                12· y1+18· y2+8· y3−8· y4
dcons:=mat▸list(subMat(dualmat,2))
                                   {⁻y1+3· y2+y3−y4+y5≥2,3· y1+y2+y3−8· y4−4· y5≥4}
```

Find the values of variables that minimize the value of the dual:

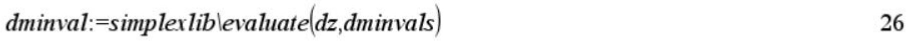


```
dminvals:=simplexlib\minimize(dz,dcons,5,true,false)
                                   Optimal solution found after 1  iterations.
                                   Summary of Results:
                                   ----------------
                                   Primal variable values:
                                   y1=1/2
                                   y2=0
                                   y3=5/2
                                   y4=0
                                   y5=0

                                   Dual variable values:
                                   x1= 3
                                   x2= 5
```

$$\left\{y1=\frac{1}{2},y2=0,y3=\frac{5}{2},y4=0,y5=0\right\}$$

Calculate the minimum value of the dual and verify that the minimum value of the dual is equal to the maximum value of the primal, which equals 26:



```
dminval:=simplexlib\evaluate(dz,dminvals)                                26
```

5.6 standardize(constraints)

Convert a list of inequality and/or equality constraints to standard form ($\leq$ constraints).

Input arguments:

> constraints - a list of constraints containing $<=, >=$ and/or $=$ constraints.

Returns:

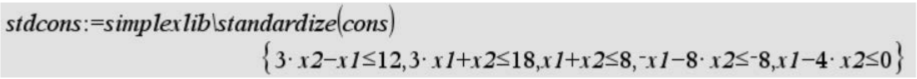
> The list of constraints in standard form. All constraints are converted to $<=$ inequalities.

Note:

> This function is provided as a convenience for manually using the Simplex method. This function is called internally by the `maximize()` and `minimize()` functions, so it does **not** need to be used in conjunction with those functions.

5.6.1 Example 5

Convert the primal constraints to standard form:



```
stdcons:=simplexlib\standardize(cons)
                          {3· x2−x1≤12,3· x1+x2≤18,x1+x2≤8,¯x1−8· x2≤¯8,x1−4· x2≤0}
```