Special Topics for the LPG Program

Overview

LPG stands for Linear Programming Graphs. This program can be used to enter and solve systems of linear inequalities in the two variables x and y. The solution to such a system is a convex set that is called the region of feasible solutions. In addition to the system of linear inequalities, you can enter an objective function that is a linear expression in x and y. The program can then find both the maximums and the minimums of the objective function over the entire region of feasible solutions.

The program has a few features that make it easy for a teacher to construct problems that have nice solutions in terms of integer coordinates. In fact, this program is a useful tool for experimenting with any region of feasible solutions. You can push lines around and define lines that go through any two points you touch with your mouse. In other words this program makes it easy to create nice regions of feasible solutions.

The program allows you to set the colors of all items and it can make graphs that are easily incorporated into any word processor.

There is a special tutorial help file that accompanies **LPG**. You can access this help file by pulling down the main menu item **Help** and then look for the submenu item **Tutorial For New Users...**. The tutorial has been designed so that you read it before you begin experimenting with the program for the first time. If you print the tutorial help file first, you can follow the instructions from your paper copy while you view the output on the screen.

Before you consult the on-line help you should first have read the examples contained in the tutorial. It is not necessary that you memorize or retain all the information in the tutorial. But you need to take the time to run the examples in the tutorial. After that, you will have a basic conceptual understanding of how the program operates so that all the concepts and most of the terminology contained in this help file should be more easily understood.

The following is a list of all the special topics in this help file.

Aspect Ratio
Bitmaps
Graph Translations
Grid Styles
MetaFiles
Objective Function Trace Mode
Redrawing
Saved File Contents
World Graph
Zooming

Aspect Ratio (special topic)

In the **World Graph Background and Axes Parameters** dialog box there is a checkbox as part of the complete Image information with the title **Apply True Aspect Ratio**. When this checkbox is checked and you resize the program window, the program will automatically compute its width. In fact, if you are manually resizing the window with a mouse you need only pull down on the bottom edge of the window. The program will automatically determine the window width. Moreover, the program will also automatically recalculate the X Maximum extent number while leaving the other three extent values unchanged. The new aspect ratio and extent value are based on the fixed Y extents that you set. The X Minimum extent is also left unchanged.

The program performs these computations automatically to give the program window the proper aspect ratio. Note that this only happens when the checkbox is checked and you resize the window. If you only check the checkbox without resizing the window then nothing special happens.

We recommend using the true aspect ratio if you are making images to put in a word processing document. When the true aspect ratio is set then circles and squares should appear perfectly round and square respectively.

If you do not use the true aspect ratio feature then depending on how you define the XY extents your circles may appear more like ellipses and what should be squares in say the lattice grid may look more like rectangles. This will be especially true if you are performing zooming by dragging a rectangle because this type of zoom usually destroys the aspect ratio. But you can follow such a zoom by a slight window resize operation and get back to a true aspect ratio. If you are trying to maintain a fixed graph window size then just minimize and then maximize the window.

When the program appears in its maximized default state the true aspect ratio is turned on and the program uses its maximal size to fill the vertical part of the screen. On a standard VGA display, if you use the true aspect ratio feature of the program then usually there will be a little horizontal part of your screen that goes unused, even when you make the window have the maximum height.

For those times when you wish to manually resize the program screen window, especially to give it a particular width, you should turn off the Apsect Ratio checkbox. Otherwise, the program will confine the window width to that which is required to make a true aspect ratio.

Bitmaps (special topic)

Although color bitmaps on the screen may look beautiful, if you work with bitmaps you will quickly learn this important fact about bitmaps. They suffer terrible distortion when they are stretched or compressed or resized in any way.

The source bitmap is always taken from the screen image, and if you make the destination bitmap a different size then the program must stretch the source bitmap to make it match the size of the destination. This creates distortion in the new image. It is best to always let the program automatically match the screen size or the printer size because the program will try to minimize any stretching of the image. If the image is still too small or too large then make the display screen program window larger or smaller and then try to rematch the screen size or the printer size.

Although bitmaps can be useful, if you have choice we recommend using enhanced metafiles instead of bitmaps. Metafiles have the advantage of being able to be stretched or compressed without suffering the distortion that usually occurs when you try to stretch a bitmap.

Graph Translations (special topic)

In addition to zooming, sometimes you just need to make small horizontal and/or vertical translations with the current graph window. You can do this most easily by holding down the **CTRL** key on your keyboard and then left click the mouse near the graph center and then drag the mouse around in any direction. You should see the mouse cursor change to a hand and the graph should appear to move with the hand. Let up on both the mouse and the **CTRL** key on the keyboard to fix the current graph window position. This is sometimes a quick and easy way to make changes to the XY extents without bringing up the **World Graph Parameters** dialog box.

Grid Styles (special topic)

There are 2 distinct grid patterns that appear in order in the list below. The default grid has both XY-axes and the XY-Lattice Grid. The grid pattern that finally gets selected is a combination of the XY-axes, and lattice grid. Each of these two types can be selected in combination. See the **World Graph Background and Axes Parameters** dialog box.

No axes or grids
XY-Axes Only
XY-Axes and XY-Lattice Grid
XY-Lattice Grid Only

MetaFiles (special topic)

LPG can save images as enhanced metafiles. Since metafiles can be stretched without the same distortion that bitmaps suffer, it is generally recommended you save images as enhanced metafiles as opposed to bitmaps. However, bitmaps can be saved and used when the screen resolution matches or exceeds that which you require. Whenever you save an image as a metafile you are also given the chance to copy it to the Windows clipboard as a 32-bit enhanced metafile.

Objective Function Trace Mode (special topic)

This is one of two trace modes in the program that is intended to help you read graph coordinates and other information related to the graph.

This mode is used by first clicking the button in the tool bar that appears as 

. Then this button will turn white 

to indicate the mode is active.

Next, you can hold down the left mouse button and drag the mouse across the graph. Then look at the status line at the bottom of the window. It will show the coordinates of the point that is at the tip of the mouse cursor and it will also show the objective function value at that point.

You can click this button a second time to turn off the Objective Function trace mode.

Redrawing (special topic)

LPG has a button in the tool bar with the caption **Redraw**. The purpose of this button is to erase everything on the graphics screen except the current problem.

Redrawing may be desirable after performing certain operations. You can Redraw the current problem at any time.

Saved File Format (special topic)

At any time you can use the menu item under **File** to **Save Linear Programming Problem**. When you do this you will first see the standard file save dialog box. The current problem and all its associated parameters will be saved in an ASCII text file that you name. Such a file can have over 120 variable values, written with one value per line in the file.

The text file that you save can be edited with any word processor that can save a file as an ASCII text file. For example, the Notepad editor that comes with Windows will do this. After you have become familiar with how the program operates you can then perform the save function and then study the resulting text file. You can select certain lines from the text file the program creates to create your example text files.

How much information you put into a given function text file is up to you. You don't have to put in all 120+. You can just put in as few as you need.

```
LP.Constraint[1].A = 1
LP.Constraint[1].B = 1
LP.Constraint[1].C = 12
LP.Constraint[1].Less Than = True
LP.Constraint[1].Active = True
LP.Constraint[2].A = 2
LP.Constraint[2].B = 1
LP.Constraint[2].C = 20
LP.Constraint[2].Less Than = True
LP.Constraint[2].Active = True
LP.Constraint[3].A = 1
LP.Constraint[3].B = 3
LP.Constraint[3].C = 30
LP.Constraint[3].Less Than = True
LP.Constraint[3].Active = True
LP.Constraint[4].A = 1
LP.Constraint[4].B = 1
LP.Constraint[4].C = 1
LP.Constraint[4].Less Than = True
LP.Constraint[4].Active = False
LP.Constraint[5].A = 1
LP.Constraint[5].B = 1
LP.Constraint[5].C = 1
LP.Constraint[5].Less Than = True
LP.Constraint[5].Active = False
LP.Constraint[6].A = 1
LP.Constraint[6].B = 1
LP.Constraint[6].C = 1
LP.Constraint[6].Less Than = True
LP.Constraint[6].Active = False
```

```
LP.Constraint[7].A = 1
LP.Constraint[7].B = 1
LP.Constraint[7].C = 1
LP.Constraint[7].Less Than = True
LP.Constraint[7].Active = False
LP.Constraint[8].A = 1
LP.Constraint[8].B = 1
LP.Constraint[8].C = 1
LP.Constraint[8].Less Than = True
LP.Constraint[8].Active = False
LP.Constraint[9].A = 1
LP.Constraint[9].B = 1
LP.Constraint[9].C = 1
LP.Constraint[9].Less Than = True
LP.Constraint[9].Active = False
LP.Constraint[10].A = 1
LP.Constraint[10].B = 1
LP.Constraint[10].C = 1
LP.Constraint[10].Less Than = True
LP.Constraint[10].Active = False
LP.Constraint[11].A = 1
LP.Constraint[11].B = 1
LP.Constraint[11].C = 1
LP.Constraint[11].Less Than = True
LP.Constraint[11].Active = False
LP.Constraint[12].A = 1
LP.Constraint[12].B = 1
LP.Constraint[12].C = 1
LP.Constraint[12].Less Than = True
LP.Constraint[12].Active = False
LP.Constraint[13].A = 1
LP.Constraint[13].B = 0
LP.Constraint[13].C = 0
LP.Constraint[13].Less Than = False
LP.Constraint[13].Active = True
LP.Constraint[14].A = 0
LP.Constraint[14].B = 1
LP.Constraint[14].C = 0
LP.Constraint[14].Less Than = False
LP.Constraint[14].Active = True
LP.ObjectiveFunction.A = 2
LP.ObjectiveFunction.B = 3
LP.Maximize Objective Function = True
LP.Minimize Objective Function = True
WorldGraph.WorldXMinimum = -2
WorldGraph.WorldXMaximum = 21.1
WorldGraph.WorldYMinimum = -1.5
WorldGraph.WorldYMaximum = 15
```

```
WorldGraph.WorldOriginX = 0
WorldGraph.WorldOriginY = 0
WorldGraph.WorldXScale = 1
WorldGraph.WorldYScale = 1
WorldGraph.PrintXScaleValues = TRUE
WorldGraph.PrintYScaleValues = TRUE
WorldGraph.ZeroBasedScales = TRUE
WorldGraph.ZeroBasedGrid = TRUE
WorldGraph.HorizontalGridSpacing = 1
WorldGraph.VerticalGridSpacing = 1
WorldGraph.GridIndex = 5
WorldGraph.TrueAspect = TRUE
WorldGraph.BorderThickness = 2
WorldGraph.BackgroundColor = White
WorldGraph.FeasibleInteriorColor = Lime
WorldGraph.LineBorderColor = Red
WorldGraph.LineLabelColor = Yellow
WorldGraph.ConnectedCurve = TRUE
WorldGraph.SignificantDigits = 5
WorldGraph.DecimalPlaces = 5
WorldGraph.Numeric Format = 0
WorldGraph.Scale SignificantDigits = 3
WorldGraph.Scale DecimalPlaces = 1
WorldGraph.Scale Numeric Format = 0
WorldGraph.PrintXYExtentValues = FALSE
WorldGraph.CurvePen.PenColor = Black
WorldGraph.CurvePen.PenWidth = 1
WorldGraph.XYAxesPen.PenColor = Blue
WorldGraph.XYAxesPen.PenWidth = 2
WorldGraph.HorzVertGridPen.PenColor = Aqua
WorldGraph.HorzVertGridPen.PenWidth = 1
WorldGraph.Feasible Brush Style = 0
WorldGraph.Font Name = MS Sans Serif
WorldGraph.Font Point Size = 8
WorldGraph.Font Color = Black
WorldGraph.Font Bold Style = FALSE
WorldGraph.Font Italic Style = FALSE
WorldGraph.Font Underline Style = FALSE
ALIGN.[2,1] @ North-East; Stretch = 1
ALIGN.[3,1] @ North-East; Stretch = 1
ALIGN.[3,2] @ North-East; Stretch = 1
ALIGN.[13,1] @ North-East; Stretch = 1
ALIGN.[13,2] @ North-East; Stretch = 1
ALIGN.[13,3] @ North-East; Stretch = 1
ALIGN.[14,1] @ North-East; Stretch = 1
ALIGN.[14,2] @ North-East; Stretch = 1
ALIGN.[14,3] @ North-East; Stretch = 1
ALIGN.[14,13] @ North-East; Stretch = 1
```

Of course the constraints and objective function coefficients and the World dimensions are the most important variables and normally you would include at least these five variables with every function that you save.

When the program is initialized it sets up default values for all the variables. If you subsequently load an example from a text file, only those variables that appear in the text file will change. All the other remaining variables just keep their current values until you manually change them by bringing up one of the program's many dialog boxes.

You might also note that anytime you save or open such a text file, the program extends the items under **File** to show the six most recently used files. If you select any filename from this list you can open it immediately without having to go through a normal file open operation. So if you create your own examples, be sure to look for this time-saving feature at the bottom of the **File** menu items.

Saved Function Notes:

The values to the right of each equal sign generally fall into one of four categories.

1. An integer value that is used as a counter or an index.
2. A floating point real number value.
3. A boolean value that must be either TRUE or FALSE.
4. A string value.

Boolean values are the simplest since they can take on only two values. Floating point real values are the next simplest and have some kind of meaning in the context of the world graph. Real values can be in scientific notation and some values can appear as if they were integers whereas they are really floating point values. Although integer values are simple, they usually represent some type of coding field. String values are generally names of things. See the special notes at the end of these descriptions for more information regarding various coding fields and string values.

These four data type categories can be further broken down with the following lists of actual values. The items listed under each category are in alphabetical order.

INTEGER VALUES

==========

.BorderThickness
.CurvePen.PenWidth
.DecimalPlaces
.Font Point Size
.GridIndex
.HorzVertGridPen.PenWidth
.Numeric Format
.Rectangle Brush Style
.Scale DecimalPlaces
.Scale Numeric Format
.Scale SignificantDigits
.Shell Brush Style
.SignificantDigits
.XYAxesPen.PenWidth

FLOATING POINT REAL VALUES

==========

- .HorizontalGridSpacing
- .Integral Ending Value
- .VerticalGridSpacing
- .WorldOriginX
- .WorldOriginY
- .WorldXMaximum
- .WorldXMinimum
- .WorldXScale
- .WorldYMaximum
- .WorldYMinimum
- .WorldYScale

BOOLEAN VALUES

==========

- .ConnectedCurve
- .Font Bold Style
- .Font Italic Style
- .Font Underline Style
- .PrintXScaleValues
- .PrintXYExtentValues
- .PrintYScaleValues
- .TrueAspect
- .ZeroBasedGrid
- .ZeroBasedScales

STRING VALUES

==========

- .BackgroundColor
- .CurvePen.PenColor
- .Font Color
- .Font Name
- .HorzVertGridPen.PenColor
- .RectBorderColor
- .RectInteriorColor
- .XYAxesPen.PenColor

Special Explanations

A .GridIndex integer value uses the following codes:
- 0 = No XY, No Lattice, No Polar
- 1 = XY Axes Only
- 4 = Lattice Grid Only
- 5 = XY Axes and Lattice Grid

A .BorderThickness value acts like any other pen width value, except that when it is 0 it means no border will be drawn.

A .Numeric Format integer value uses the following codes:
- 0 = General Format
- 1 = Scientific Format
- 2 = Fixed Point Format

A .Font Name string value should match the name of an installed font

A Brush Style integer value uses the following codes:
- 0 = Solid Brush
- 2 = Horizontal
- 3 = Vertical
- 4 = Backward Diagonal \
- 5 = Forward Diagonal /
- 6 = Cross Hatched +
- 7 = Cross Hatched -

Note that the value 1 is not used and is illegal!

A string name of any color value must come from the following list.
- Aqua
- Black
- Blue
- Fuchsia
- Gray
- Green
- Lime
- Maroon
- Navy
- Olive
- Purple
- Red
- Silver
- Teal
- White
- Yellow

Some lines have the form **ALIGN.[m,n]=Z; Stretch = n** where **Z** is one of the following directions. The stretch value **n** acts as a multiplier of the standard label distance from the point. So this kind of a line actually stores two values that go togther. The stretch value should be an integer in the range 1-5 while the direction comes from the following list.

East
North-East
North
North-West
West
South-West
South
South-East

World Graph (special topic)

What is called the World Graph is really the background over which your graph gets drawn. You define your own world by giving the four values called X Minimum, X Maximum, Y Minimum, and Y Maximum. These four values determine the rectangular portion of the XY-plane that is seen on your screen display and are called the XY extent values. By default these numerical values are not displayed at the four edges of the graph, but you can turn on their display if you want. The world graph fills the entire program window, whatever the window size or the XY extent values.

There are additional World Graph parameters that also determine what your graph background looks like. Your graph background can have any one of 16 colors. Your graph can also be surrounded by a rectangular border or not. All the other parameters determine what axes and grids get drawn and how they get drawn. See also **World Graph Background and Axes Parameters** dialog box.

Zooming (special topic)

Zooming with **LPG** is especially simple.

To zoom Out, make sure the Zoom Button in the tool bar shows the zoom mode is Out and then point the mouse at the center point of interest and double left-click the mouse.

To zoom In you have two choices. You can also make sure the zoom mode is In and then point the mouse at the center point of interest and double left-click the mouse. An alternative is to use the right mouse button in conjunction with the Control key on your keyboard to drag a surrounding rectangle about the region of interest and then let go of the mouse and the Control key.

You can change the zoom mode in either of two ways. You can just click on the Zoom Button to change its mode, or you can right-click the mouse over the graph. In either case the new active Zoom mode should appear on the face of the Zoom Button. If you right-click the mouse be sure to hold the mouse absolutely still. Otherwise the program may think you are trying to draw a zoom rectangle with the mouse and you will end up with an empty-looking graph. To recover from this situtation, right-click the Zoom Button to bring up the **World Graph Parameters** dialog box and the click the button to restore the default graph extents.

This program also performs a special type of scaling operation when you zoom very far out in a graph. This means the program will not draw too large a number of either axes tic marks or grid lines. Instead, it will automatically skip drawing some tic marks or grid lines so that neither becomes too numerous for the display. For tic marks in particular however, the program will show the correct numeric values associated with those marks that it does draw. The grid marks should be easy to read in all cases expect when you give XY extents that are extremely and unusually large compared to your axes and grid spacing values.

Basically all a zoom operation does is reset the XY extent numbers for the World Graph.

In addition to zooming, sometimes you just need to make small horizontal and/or vertical translations with the current graph window. You can do this most easily by holding down the **CTRL** key on your keyboard and then left click the mouse near the graph center and then drag the mouse around in any direction. You should see the mouse cursor change to a hand and the graph should appear to move with the hand. Let up on both the mouse and the **CTRL** key on the keyboard to fix the current graph window position.