Fourth International Derive TI-89/92 Conference
Liverpool John Moores University, July 12 – 15, 2000

Drawing Network Graphs with DERIVE 5

Peter Schofield
Trinity and All Saints (University of Leeds), UK
email: p.schofield@tasc.ac.uk

Introduction

Utilities

The Users Folder of DERIVE 5 now contains a Utility File NGRAPHS.MTH which provides utilities for drawing network graphs with DERIVE. Graphs can be drawn in the 2D-plot window from algebraic DRAW instructions using a specification for the coordinates of their vertices together with an edge set (or an adjacency matrix or an adjacency list). By combining this basic concept of a network graph with DERIVE's powerful algebraic and 2D-plotting facilities a surprisingly powerful set of network graph drawing tools can be produced.

As well as NGRAPHS.MTH, DERIVE's Users Folder also contains two more files:

- NGRAPHS-DMO.MTH - a demonstration file which demonstrates some of the graphs which can be drawn using NGRAPHS.MTH;
- NGRAPHS.DOC – document file (in basic text) listing all of the utilities of NGRAPHS.MTH and giving some examples of their use.

These files are regularly updated, with the updated versions replacing the old ones in the Users Folder of the next new issue of DERIVE. In this workshop I shall use the latest version of NGRAPHS (Version 3), which is on the accompanying floppy disc. You can use these files on floppy as they are, or you can replace the current files in DERIVE's Users Folder with the updated versions on the floppy disc.

On the Floppy Disc:
NGRAPHS.MTH and NGRAPHS-DMO.MTH will run in both DERIVE 4 and 5.
There are also two new DFW files: NGRS5-UTIL.DFW; NGRS5-DMO.DFW, which only run in DERIVE 5. There is also an updated document file NGRAPHS.DOC.
(These files can also be obtained by email from: p.schofield@tasc.ac.uk)

(In DERIVE 4 & 5

- File>Load>Utility> : Ngraphs.mth
- File>Load>Math> : Ngraphs4-dmo.mth)

In DERIVE 5

- Open> a\: Ngrs5-dmo.dfw

- (or Open> a\: Ngrs5-Util.dfw)

In DERIVE 5, this type of file saves all the (hidden) utilities of NGRAPHS.MTH. It can be used as a "stand alone" file.

Viewing graphs with NGRS5-DMO.DFW

Once DERIVE 5 has been properly customised, it is straightforward to run NGRAPHS with this file.

To View with NGRS5-DMO.DFW

- Open DERIVE 5 with factory default settings.
- Insert NGRAPHS floppy disc and, Open> a \: Ngrs5-dmo.dfw

NGRS5-DMO.DFW should now be listed in the Algebra Window.
(Scroll up to the beginning of the file.)

To Customise DERIVE 5 Select

- 2D-plot Window
- Options> Simplify Before Plotting (On)
- Options> Change Plot Colors (Off)
- Options> Display > Points : Connect (Yes), Large
- Options> Display > Grids (Off)
- Options> Display > Cross (Off)

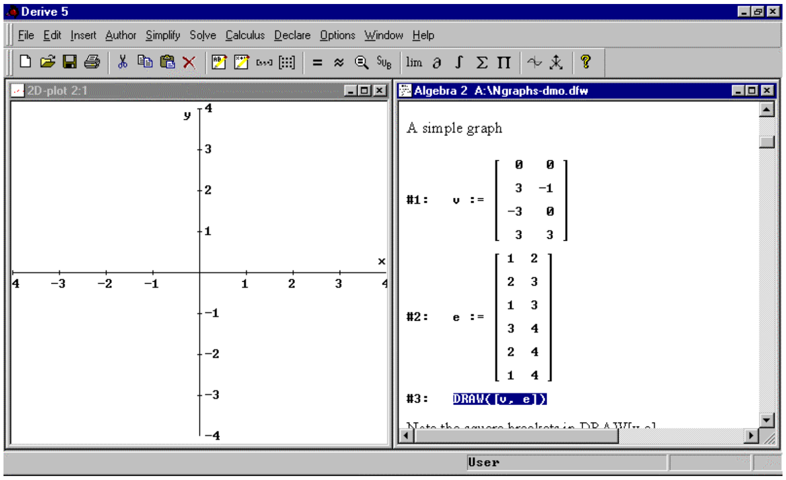
We shall also turn off Axes and Labels, but for our first examples it is instructive to leave these on.
Select: Window>Tile Vertically

Viewing a Simple Graph

A **network (di)graph** is a pair (V , E) where V is a non-empty set of **vertices** and E is a set of **edges**. Each edge $e \in E$ joins a pair 'v_1,v_2' of vertices, which can be directed (ordered) or undirected. Vertices joined by an edge are called **adjacent**.

Example 1

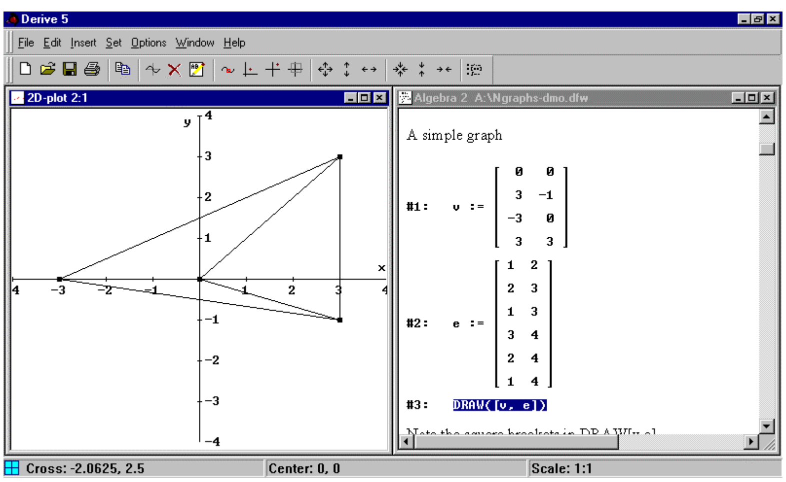
Select: the Algebra Window and use the scroll-bar to set up the following VDU display:



For this graph the coordinates of the vertices are stored, in sequence, as:
[0,0], [3,-1], [-3,0], [3,3]. To construct the edge vector, the position of each vertex in the sequence is taken as the label for each vertex. For example, [1,2] means join the vertex at [0,0] to the vertex at [3,-1].

In #3, notice the graph is stored as [v,e]. NGRAPHS will not **draw** the graph properly until it is instructed to do so using a DRAW[v,e] instruction. (Plotting [v,e] usually yields inconclusive results.)

Select 2D-plot Window and plot #3 to get:



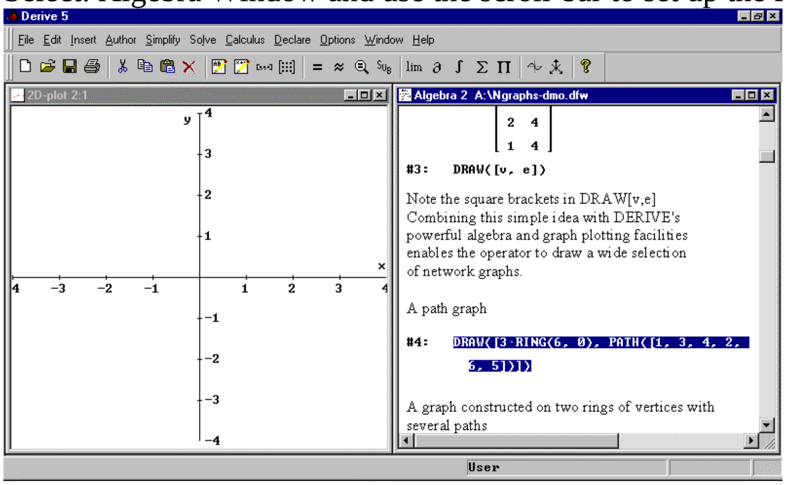
Remarks

1. This is one form of raw data that NGRAPHS uses for drawing graphs.
2. It is not the usual form that a human operator would use but, in the last analysis, it could be presented to NGRAPHS in this manner.
3. One way we would prefer to build up a graph drawing is by walks, paths, trails, cycles and/or circuits.

Contents

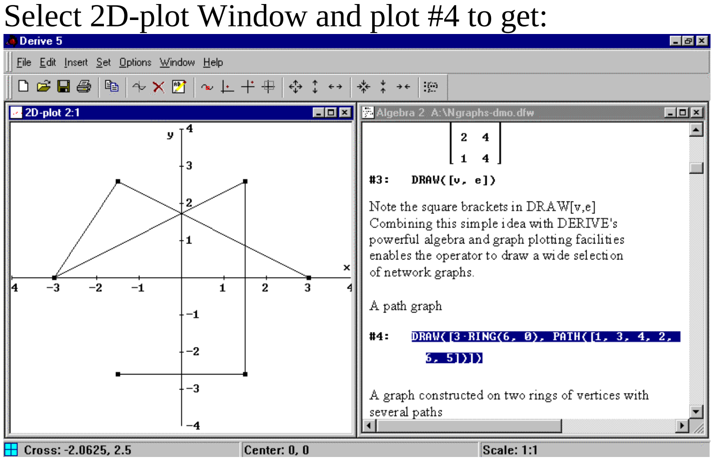
4. It will also be useful to have user-defined instructions for positioning sets of vertices in the 2D-plot Window.

Example 2
Select: Edit> Delete All Plots
Select: Algebra Window and use the scroll-bar to set up the following VDU display:



Highlight #4. Note the construction of the graph inside the square brackets.
3.RING(6,0) – constructs a ring of vertices on a circle of radius 3, with the first vertex making an angle (in radians) of 0 with the x-axis.
PATH [1,3,4,2,6,5] – constructs a path connecting vertices 1 and 5 (passing though each vertex in sequence).

Select 2D-plot Window and plot #4 to get:



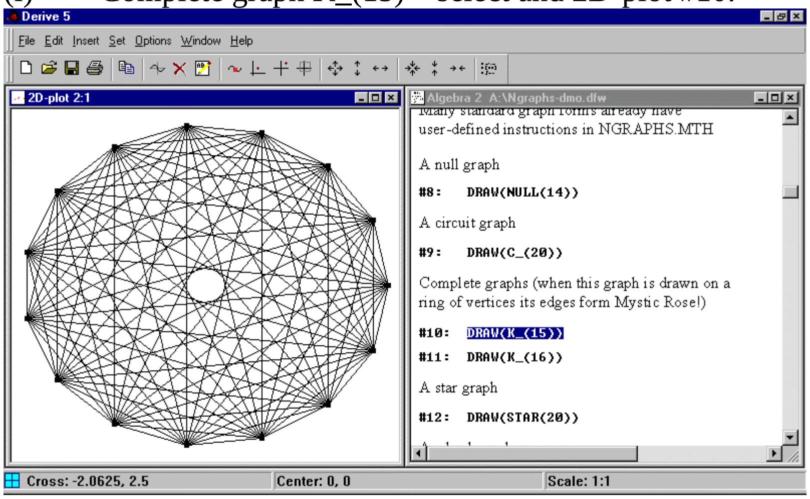
Remarks
(i) In DERIVE, the angle in RING can also be set to degree measure.
(ii) A selection of 'standard graphs' has already been defined in NGRAPHS.MTH. For example: Null, Circuit and Complete graphs; Star and Wheel graphs; Complete Bipartite

graphs; Platonic graphs; the Hypercube Graph q4; the Petersen graph. All these drawings can be viewed with
NGRS5-DMO.DFW.

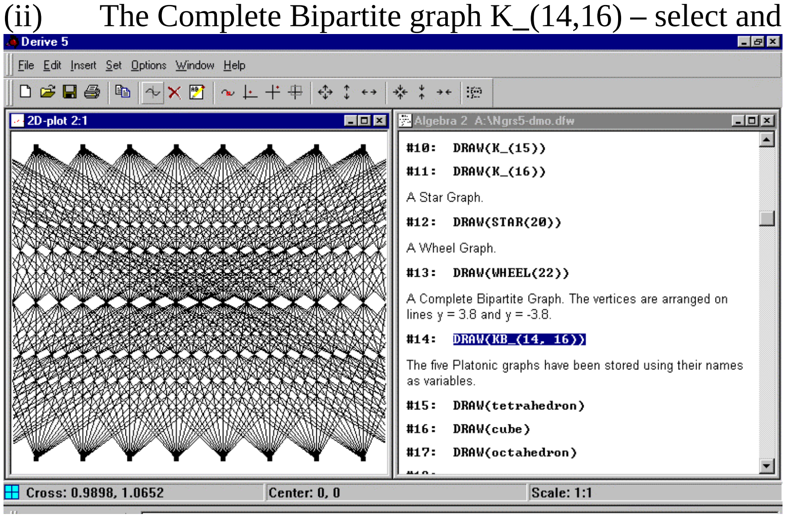
Example 3

At this stage, in the 2D-plot window **turn off** the axes and labels using Options>Display>Axes… Delete old graph.

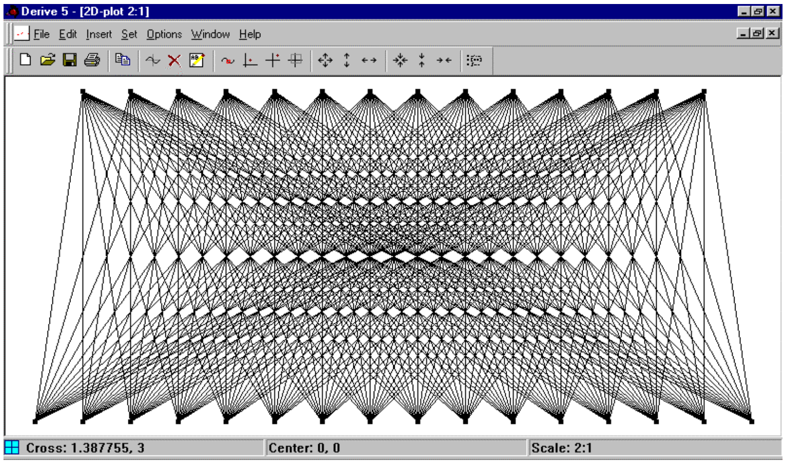
(i) Complete graph K_(15) – select and 2D-plot #10.



(ii) The Complete Bipartite graph K_(14,16) – select and 2D-plot #14.

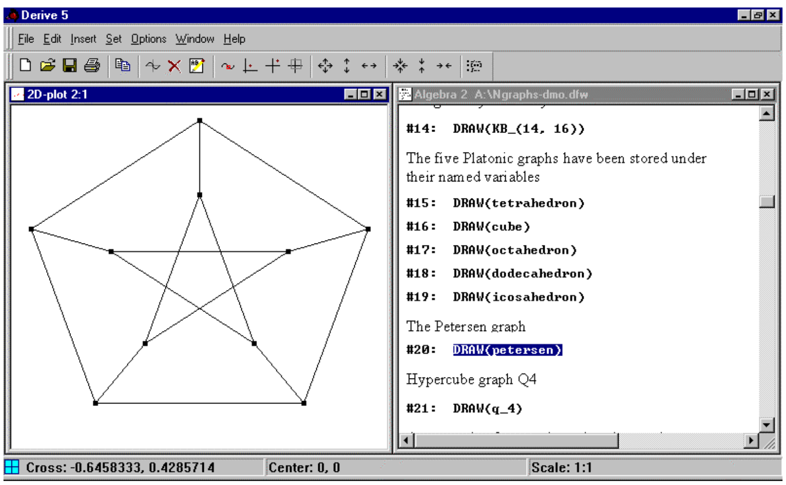


Note: the full display of this graph overlaps the reduced 2D-plot Window. Maximizing the window, and clicking once on the ‘zoom out horizontally’ button, can correct this as follows:



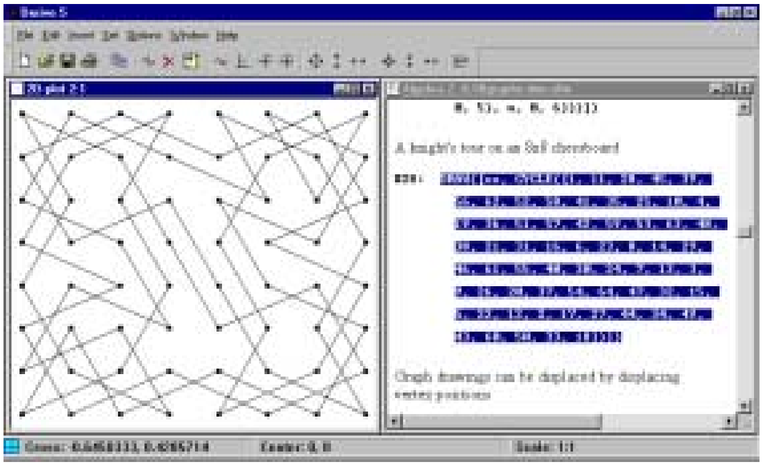
To return, select: Window>Tile Vertically and zoom in horizontally (once).

(iii) The Petersen graph– select and 2D-plot #20.



Try viewing the drawings of some of the other ‘standard graphs’. The Platonic graphs are drawn using their stereographic projections.

<u>Example 4</u> Knight’s tour on an 8×8 chessboard – select and 2D-plot #28.

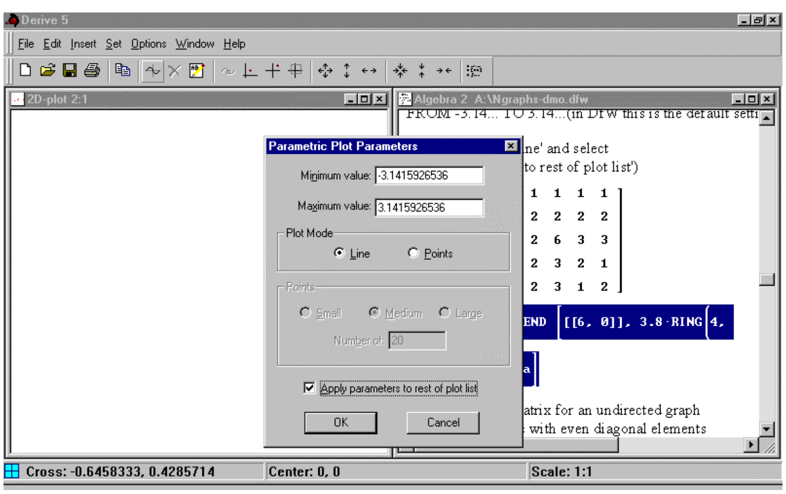


Note: this graph is drawn on an 8×8 grid of vertices:

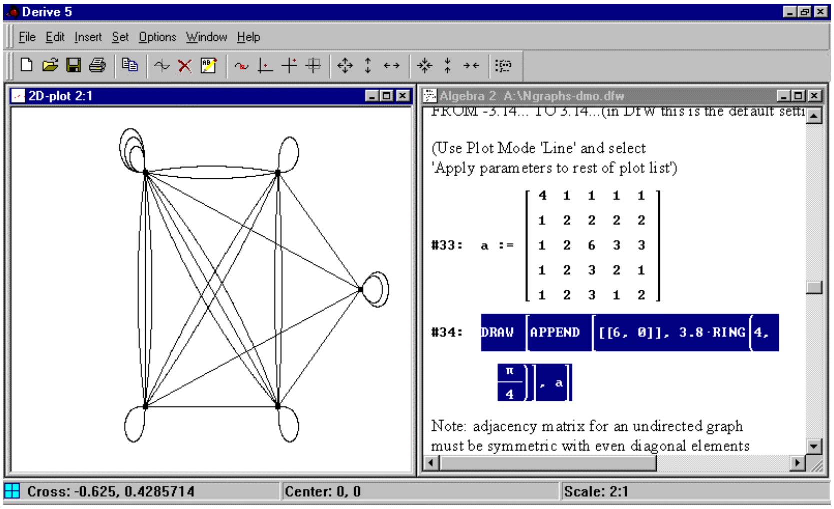
$$vv := V\_GRID\left(\begin{bmatrix} -3.8 & 3.8 \\ 3.8 & 3.8 \\ -3.8 & -3.8 \end{bmatrix}, 8, 8\right)$$

with an edge set formed by a single cycle:
CYCLE[1, 11, 28, 45, 39, 56, 62, 52, 58, 41, 35, 25, 10, 4, 19, 36, 51, 57, 42, 59, 53, 63, 48, 38, 21, 31, 16, 6, 23, 8, 14, 29, 46, 61, 55, 40, 30, 24, 7, 13, 3, 9, 26, 20, 37, 54, 64, 47, 32, 15, 5, 22, 12, 2, 17, 27, 44, 34, 49, 43, 60, 50, 33, 18]

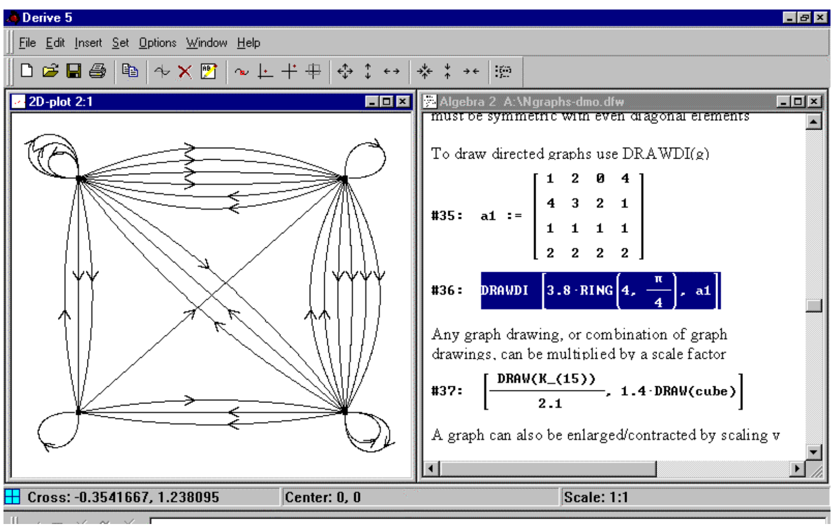
Example 5 Drawing a graph with multiple edges from an adjacency matrix.
Select: Edit> Delete All Plots.
Select: Algebra Window and use the scroll-bar to highlight #34:
Select: 2D-plot Window and plot, and in the Parameter Plot Window click the mouse on the ‘Apply parameters to rest of plot list’ box to obtain the following VDU display:



Select: OK and zoom out horizontally (once)to obtain:



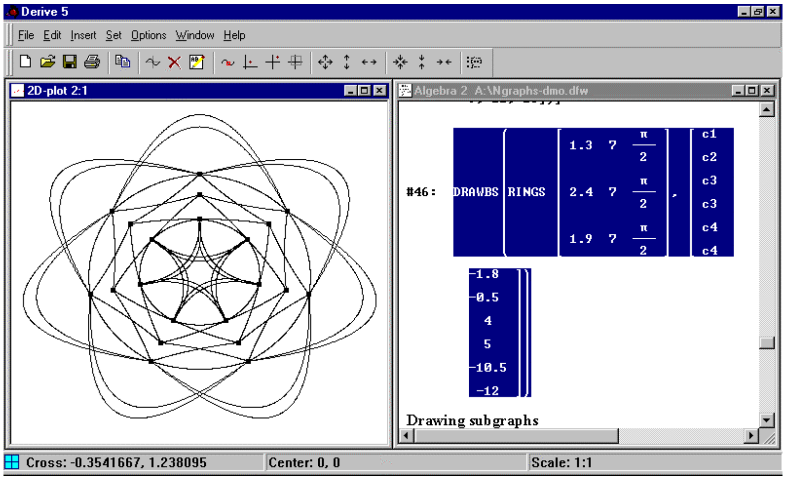
Example 6 Drawing directed graphs and graphs with bow edges.
(i) Zoom in horizontally and select: Edit> Delete All Plots.
Select: Algebra Window and use the scroll-bar to highlight #36:
Select: 2D-plot Window and plot, and in the Parameter Plot Window click on OK to obtain the following VDU display:



Note: this digraph has been drawn from the data in matrix **a1** using the instruction: DRAWDI[3.8RING(4,π/4),a1]. Compare the entries of **a1** with the directed edges and loops of the graph drawing.

(ii) Select: Edit> Delete All Plots.
Select: Algebra Window and use the scroll-bar to highlight #46:

Select: 2D-plot Window and plot, and in the Parameter Plot Window click on OK to obtain the following VDU display:



This is the graph of a **21 bead Mandala**. It is one of the most ambitious graphs I have attempted so far. To draw a graph g with a 'bow factor' use DRAWB(g,b) (or DRAWDIB(g,b)). However , this graph uses DRAWBS(v,ebs), where v consists of three rings of vertices and ebs consists of a number of edge specifications together with corresponding bow factors.
Remark: Take a little time to browse through the rest of NGRS5-DMO.DFW. It provides examples of most of the utilities, graphs and techniques of NGRAPHS.MTH.

2. Manipulating and Transforming Graph Drawings in the 2D-plot Window

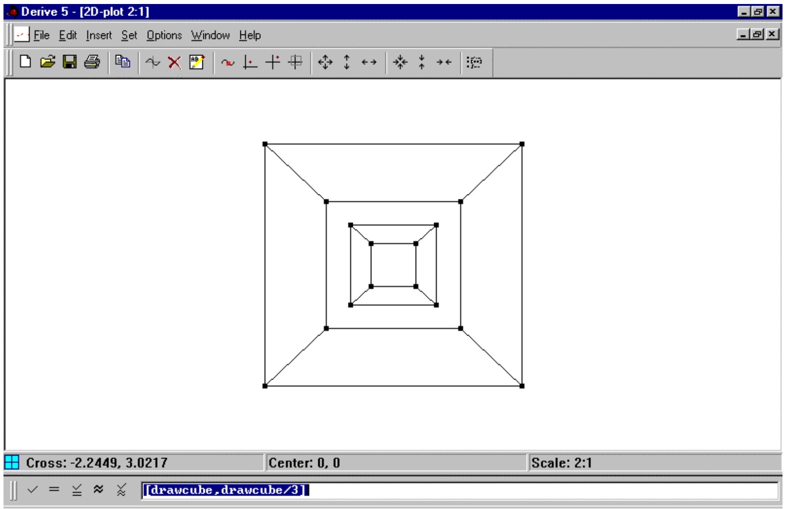
The graph drawings generated by NGRAPHS.MTH can be manipulated by 2D-linear transformations in a straightforward manner. This is useful for scaling and positioning graph drawings in the 2D-plot Window.

2.1 Working with NGRAPHS.MTH in a maximized 2D-plot Window

Select: Algebra Window and Maximize
Close> A:\NGRS5.DMO.DFW
Open> a:\ NGRS5.UTIL.DFW
(This DFW file lists all of the current utilities of NGRAPHS. These have also been stored (hidden) in this file, and they are now available to use!)
Select: 2D-plot Window and zoom out horizontally (once)
(Reset the customized settings of Section 1.0.)

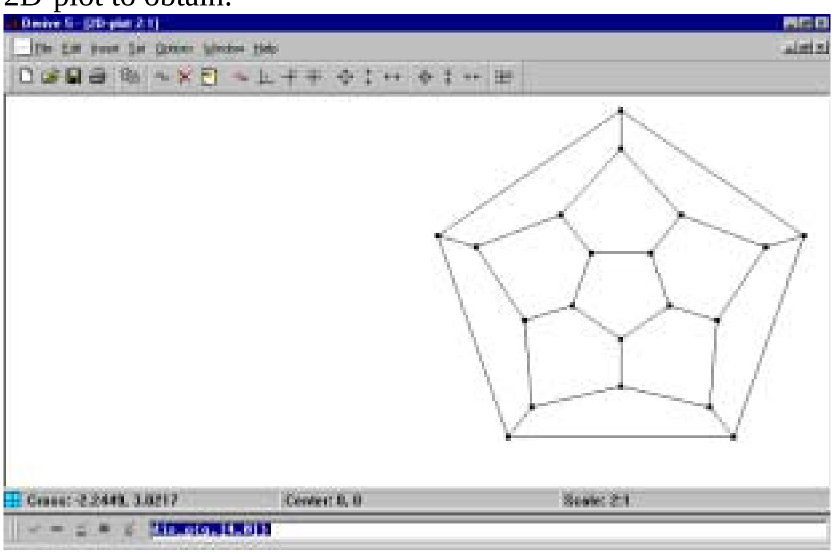
Example 7 Scaling a cube.

Note that the Entry Expression Toolbar is available to use with the maximized 2D-plot Window. Click on the Entry Expression Toolbar and type:
[drawcube, drawcube/3] ↵
2D-plot to obtain:

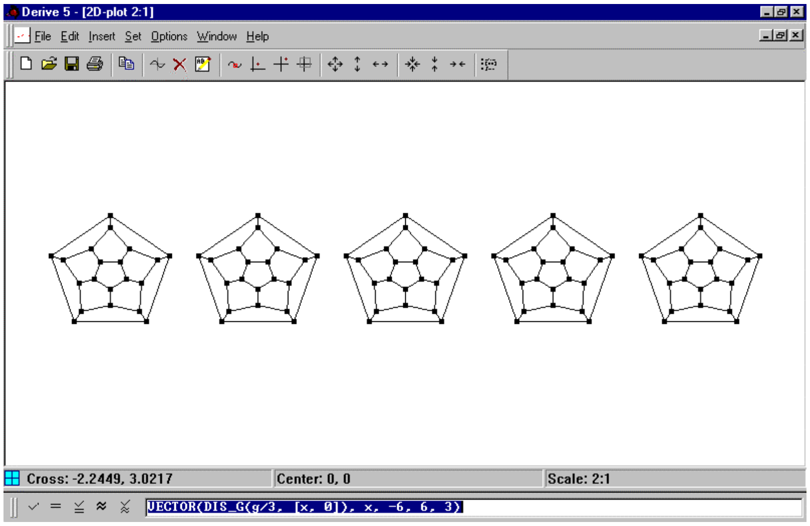


Remark: all basic graphs are drawn with reference to the origin (0,0).

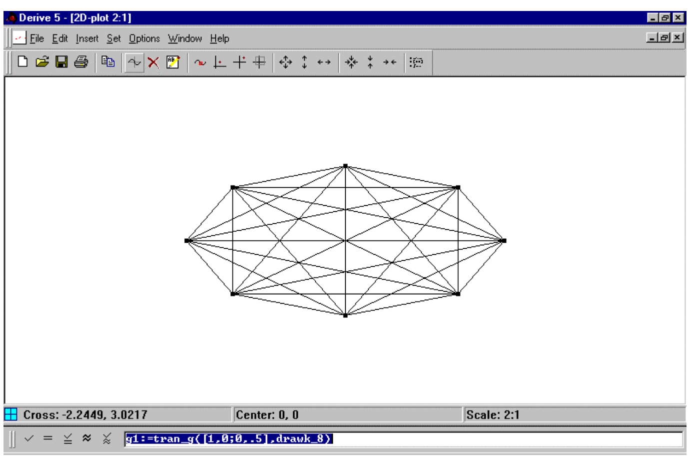
Example 8 Displacing a graph drawing.
Select: Edit > Delete All Plots
Type: g:=drawdodecahedron ↵
Type: dis_g (g, [4,0]) ↵
2D-plot to obtain:



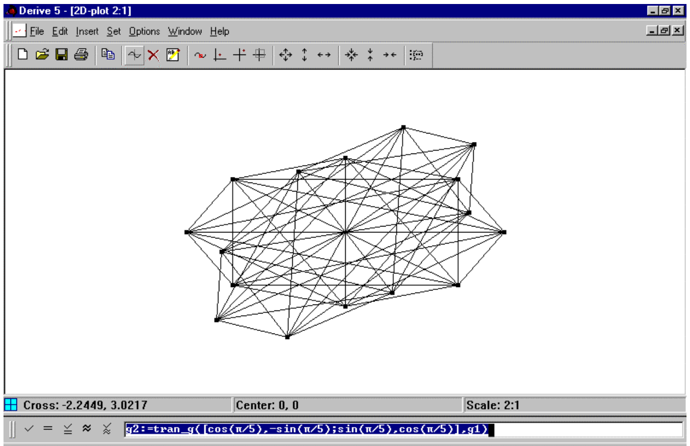
Select: Edit > Delete All Plots
Type: vector(dis_g (g/3, [a, 0]), a, –6, 6, 3) ↵
2D-plot to obtain:



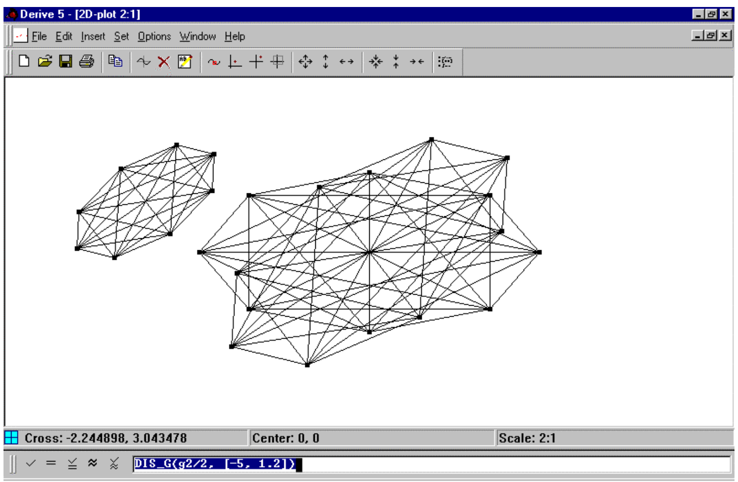
Example 9   Transforming a graph drawing by (pre)multiplying by a matrix.
Select: Edit > Delete All Plots
Type: g1 := tran_g ([1, 0 ; 0, .5], drawk_8) ↵ 2D-plot to obtain:



Type: g2 := tran_g ([cos(π/5), –sin(π/5) ; sin(π/5), cos(π/5)], g1) ↵
and 2D-plot to obtain:



Type: dis_g (g2/2, [–5, 1.2]) ↵ 2D-plot to obtain:



3. Drawing Graphs

NGRAPHS.MTH has a number of user-defined instructions to specify a graph in the form [v,e], where v is a vector of point coordinates and e is a specification for the edges. (See NGRAPHS.DOC for details)

Select: Algebra Window (and maximize)
Close> a:\NGRS.UTIL.DFW
Select: New Blank Document (Left hand button of Standard Menu)
Select: File>Load>Utility> a:\ NGRAPHS.MTH
(In DERIVE 5, note the instruction inserted into the Algebra file.)
Select: 2D-plot Window
Select: Window>Tile Vertically
(Reset the customized settings of Section 1.0 and set the Scale to 1:1)

3.1 Some Examples of Drawing Graphs

Stages for drawing a graph

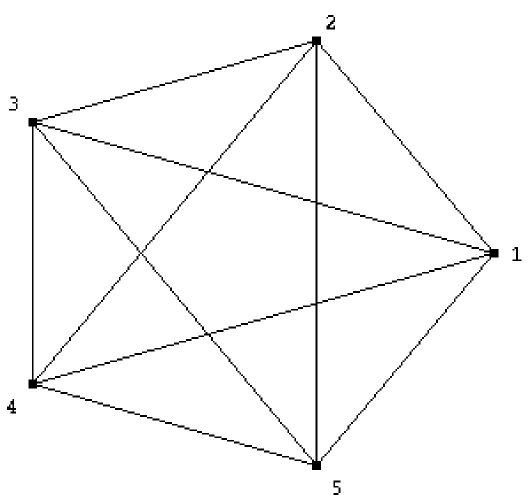
Stage 1: make a rough sketch of the graph; label vertices using positive integers.

Stage 2: analyse good vertex positions and edges vector sequences and convert these into DERIVE vectors and instructions for v and e.

Stage 3: plot DRAW[v,e]in DERIVE's 2D-plot Window

Example 10

Use NGRAPHS to draw:



It is clear that we need a ring of 5 vertices on a circle of radius, 3 say. Also, the first vertex makes an angle of 0 with the x-axis. Therefore, enter:

v := 3 ring (5,0)

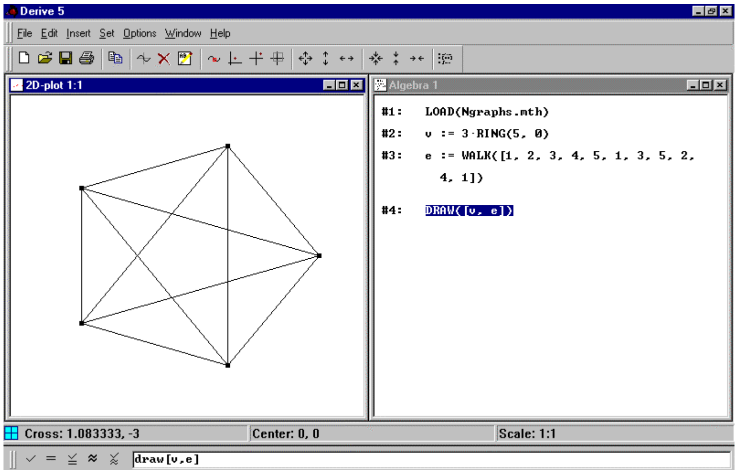
In NGRAPHS.MTH (Version 3) we can specify the edge set as a walk, path, trail (when drawing, NGRAPHS.MTH can use any of these) or a cycle or a circuit.
For example, enter:

e := walk[1,2,3,4,5,1,3,5,2,4,1]

(or e:=cycle[1,2,3,4,5,1,3,5,2,4])

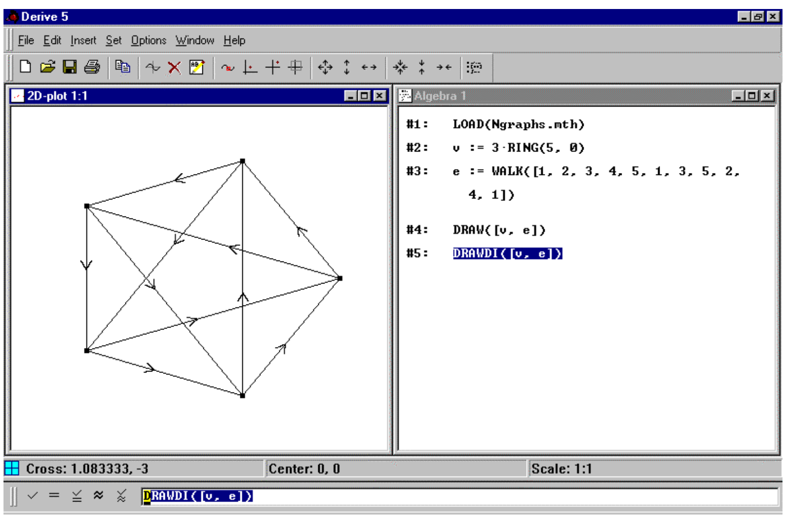
Contents

To draw the graph enter and 2D-plot: draw[v,e]



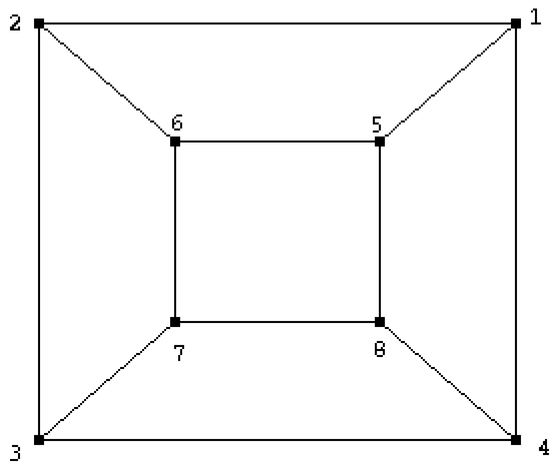
To label vertices, use the 2D-plot annotation facilities of DERIVE.

To draw Example 10 as a directed graph, enter and 2D-plot: drawdi[v,e]
(Apply parameters to rest of plot list (On))



Contents

Example 11 (Drawing a cube graph from scratch)



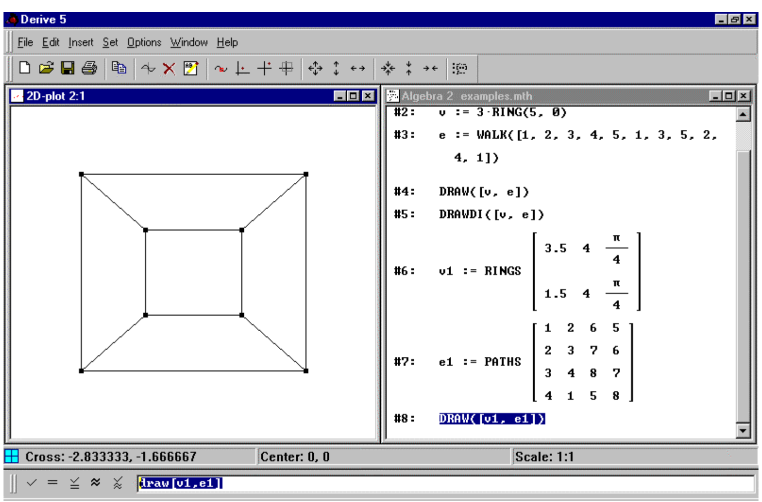
We require two rings of four vertices, on circles of different radii, with the first vertex on each ring offset by an angle $\pi/4$ from the direction of the x-axis. Therefore enter:

v1 := rings [3.5, 4, $\pi/4$; 1.5, 4, $\pi/4$]

For the edge set we could use a single walk, but this would mean going over some edges more than once. It is more efficient to draw the cube graph using 4 paths of length 3 as follows:

e1 := paths [1, 2, 6, 5 ; 2 , 3, 7, 6 ; 3 , 4, 8, 7 ; 4, 1, 5 , 8]

To draw the cube graph enter and 2D-plot: draw[v1,e1]



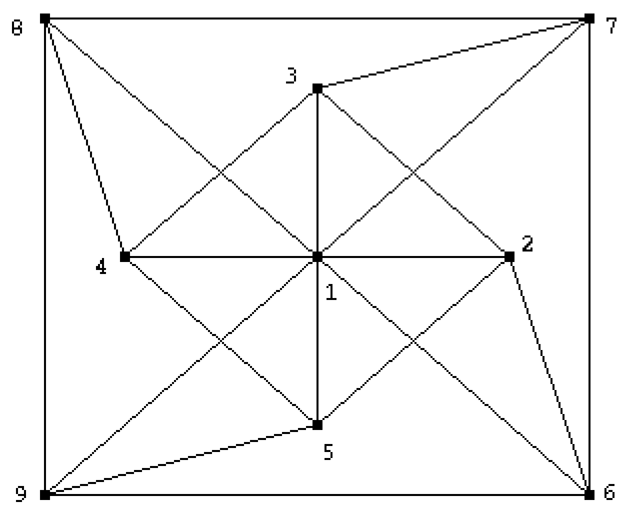
Notes: (i) in the Algebra Window, note the format of RINGS and PATHS;
(ii) enter and 2D-plot: drawdi[v1,e1] to draw the cube graph with diedges.

Fourth International Derive TI-89/92 Conference

Exercise 1   Draw a graph representing the Atomium Building, Brussels.

The Atomium (in Brussels) is a remarkable building. It is a cube structure of 8 large spherical rooms, linked by tubular staircase corridors. In addition, it has a central spherical room linked by corridors/lifts to the other 8 rooms at the corners of the cube.
If you ever visit Brussels, it is a "must" to go and see. For more information about the Atomium consult: http://www.atomium.be/

Draw the following graph of the Atomium:



Remarks

(i) To position the vertices, think of three rings: the first ring is of radius 0, with one vertex offset at 0; the second ring is of radius 2, with 4 vertices (offset at 0); the third ring is of radius 4, with 4 vertices (offset at $-\pi/4$).

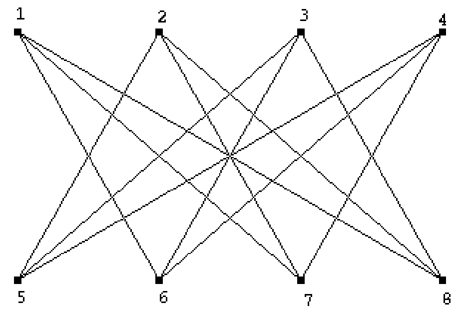
(ii) For the edges, since all of its vertices are even, it follows that this graph has an Eulerian cycle (circuit).

(iii) Once you have specified v and e, to draw this graph you have to enter and 2D-plot draw[v,e].

(iv) This graph also a Hamiltonian cycle. Can you draw an example of one of these in a different plot colour?

Contents

<u>Example 12</u> (Drawing a cube graph as a bipartite graph)



(Satisfy yourself that this is a cube graph!?)

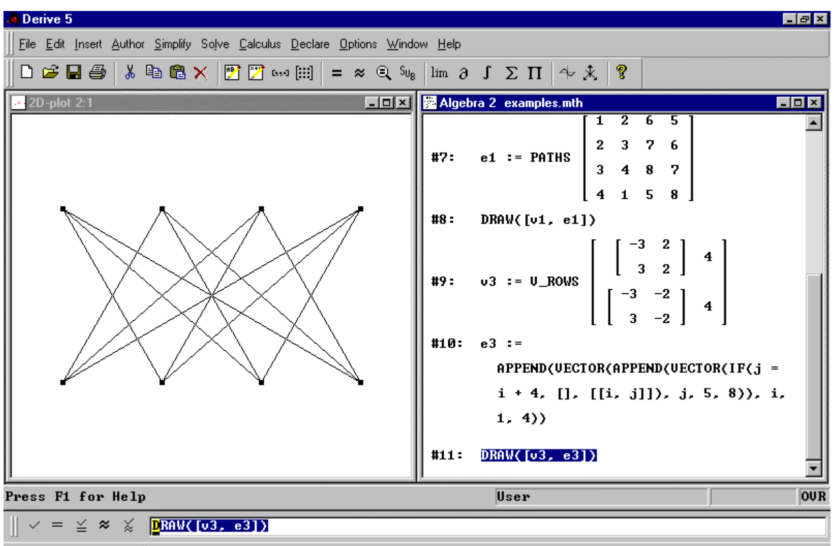
For the vertex positions enter:
v3 := v_rows [[–3, 2 ; 3, 2], 4 ; [–3, –2 ; 3, –2], 4]

(Note: this specifies two rows of (four) vertices with the beginning and end points of each row indicated.)

We shall use a bit of sophisticated DERIVE algebra for the edges. Enter:
e3 := appendvector(appendvector(if(j=i+4, [], [[i, j]]), j, 5, 8), i, 1, 4)
The double square brackets are used so that [i, j] stays, but [ ] is eliminated, by the append operation.

(You could also enter e3 simply as: e3:=[1,6;1,7;1,8;2,5;2,7;2,8;3,5;3,6;3,8;4,5;4,6;4,7].)

To draw, enter and 2D-plot: draw[v3, e3]



Note the way v3 is displayed in the Algebra Window.

4. Interlude

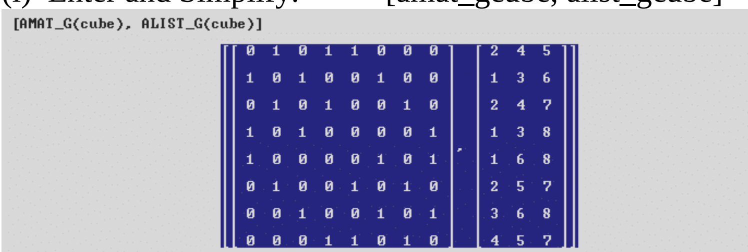
The vertex/edge information for a network graph can be stored in a variety of forms, and NGRAPHS.MTH has instructions for converting between them.

4.1 Adjacency Matrices and Adjacency Lists

Example 13   Adjacency matrices and lists of standard graphs.

Select: maximized Algebra Window

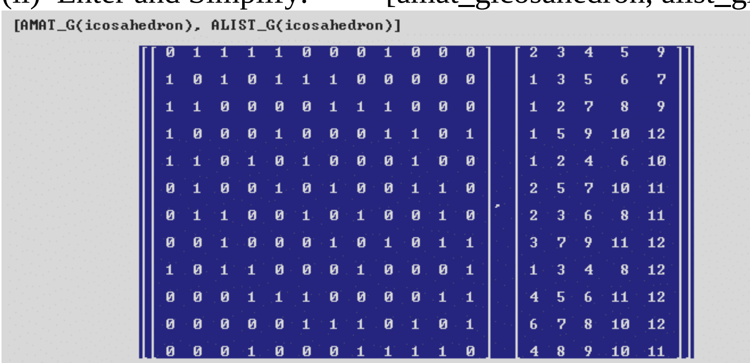
(i) Enter and Simplify: [amat_gcube, alist_gcube]



[AMAT_G(cube), ALIST_G(cube)]

$$\left[\begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 2 & 4 & 5 \\ 1 & 3 & 6 \\ 2 & 4 & 7 \\ 1 & 3 & 8 \\ 1 & 6 & 8 \\ 2 & 5 & 7 \\ 3 & 6 & 8 \\ 4 & 5 & 7 \end{bmatrix}\right]$$

(ii) Enter and Simplify: [amat_gicosahedron, alist_gicosahedron]



[AMAT_G(icosahedron), ALIST_G(icosahedron)]

$$\left[\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 2 & 3 & 4 & 5 & 9 \\ 1 & 3 & 5 & 6 & 7 \\ 1 & 2 & 7 & 8 & 9 \\ 1 & 5 & 9 & 10 & 12 \\ 1 & 2 & 4 & 6 & 10 \\ 2 & 5 & 7 & 10 & 11 \\ 2 & 3 & 6 & 8 & 11 \\ 3 & 7 & 9 & 11 & 12 \\ 1 & 3 & 4 & 8 & 12 \\ 4 & 5 & 6 & 11 & 12 \\ 6 & 7 & 8 & 10 & 12 \\ 4 & 8 & 9 & 10 & 11 \end{bmatrix}\right]$$

Adjacency matrices and lists can be extracted for all of the standard graph specifications saved in NGRAPHS.MTH. The results for complete bipartite graphs are interesting:

(iii) for example, Enter and Simplify: [amat_gkb_(3,5), alist_gkb_(3,5)]

[AMAT_G(KB_(3, 5)), ALIST_G(KB_(3, 5))]

$$\left[\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, [[4, 5, 6, 7, 8], [4, 5, 6, 7, 8], [4, 5, 6, 7, 8], [1, 2, 3], [1, 2, 3], [1, 2, 3], [1, 2, 3], [1, 2, 3]]\right]$$

Converting between edges and adjacency matrices and lists.

(iv) Enter and simplify: [e3, amate3, alistmatamate3]

[e3, AMAT(e3), ALISTMAT(AMAT(e3))]

$$\left[\begin{bmatrix} 1 & 6 \\ 1 & 7 \\ 1 & 8 \\ 2 & 5 \\ 2 & 7 \\ 2 & 8 \\ 3 & 5 \\ 3 & 6 \\ 3 & 8 \\ 4 & 5 \\ 4 & 6 \\ 4 & 7 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 6 & 7 & 8 \\ 5 & 7 & 8 \\ 5 & 6 & 8 \\ 5 & 6 & 7 \\ 2 & 3 & 4 \\ 1 & 3 & 4 \\ 1 & 2 & 4 \\ 1 & 2 & 3 \end{bmatrix}\right]$$

Compare the edge set with its adjacency matrix and list.

(v) Enter: a:= [0, 2 , 1, 1 ; 2, 2, 0, 1 ; 1, 0, 0, 1 ; 1, 1, 1, 0]

$$a := \begin{bmatrix} 0 & 2 & 1 & 1 \\ 2 & 2 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Notice that matrix a is symmetric, with even entries on the main diagonal. Therefore, it can be processed as the adjacency matrix of a graph or a digraph.
Enter and simplify: [edgesa, diedgesa, alistmata]

[EDGES(a), DIEDGES(a), ALISTMAT(a)]

$$\left[\begin{bmatrix}1&2\\1&2\\1&3\\1&4\\2&2\\2&4\\3&4\end{bmatrix}, \begin{bmatrix}1&2\\1&2\\2&1\\2&1\\1&3\\3&1\\1&4\\4&1\\2&2\\2&2\\2&4\\4&2\\3&4\\4&3\end{bmatrix}, [[2, 2, 3, 4], [1, 1, 2, 2, 4], [1, 4], [1, 2, 3]]\right]$$

Note that, this graph has multiple edges and loops. It can be drawn as either a graph or a digraph.

<u>Example 14</u> Testing for paths and trails.

In Version 3 of NGRAPHS.MTH I have included terms: walk, path, trail, cycle,circuit. When drawing, NGRAPHS makes no distinction between walk, path and trail, and no distinction between cycle and circuit. It is up to the operator to use these terms in context.
However, in Version 3 of NGRAPHS.MTH, there is an instruction that will test whether all of the elements of a vector are distinct.
DISTINCT(v) = 0 $\Leftrightarrow$ all elements of v are distinct.

Using this instruction it is straightforward to test whether a walk is a path (through distinct vertices) or a trail (through distinct edges).
For example,

Simplyfying:	distinct [1,2,3,4,5,6]	(= 0, indicates a path)
Simplifying:	distinctwalk[1,1,2,3,1,4,5]	(= 0, indicates a trail)

Try out some tests for yourself.

5. Drawing Graphs and Digraphs with Multiple Edges and Loops

These can be drawn from the adjacency matrix or list for a graph or digraph. NGRAPHS.MTH processes the adjacency matrix for drawing a graph differently from drawing a digraph.

In 2D-plot Window select: Window>Tile Vertically.
Delete old graph.

5.1 Examples

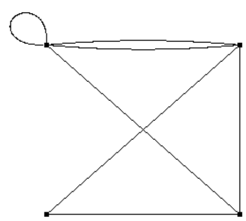
Example 15 (Drawing a graph and digraph from a simple adjacency matrix)

Enter: a:= [0, 2 , 1, 1 ; 2, 2, 0, 1 ; 1, 0, 0, 1 ; 1, 1, 1, 0]
to obtain:

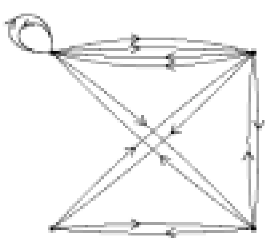
$$a := \begin{bmatrix} 0 & 2 & 1 & 1 \\ 2 & 2 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Enter: v4 := 3ring(4,π/4)

Enter and 2D-plot: draw[v4, a]
(Apply parameters to rest of plot list (On))



Delete old graph and enter and 2D-plot: drawdi[v4, a]
(Apply parameters to rest of plot list (On))



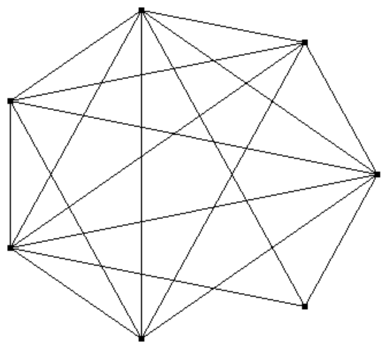
Example 16

DRAW can only draw simple graphs from edge sets specified in terms of walks, cycles, etc. To draw graphs with multiple edges from these, convert into adjacency matrices.

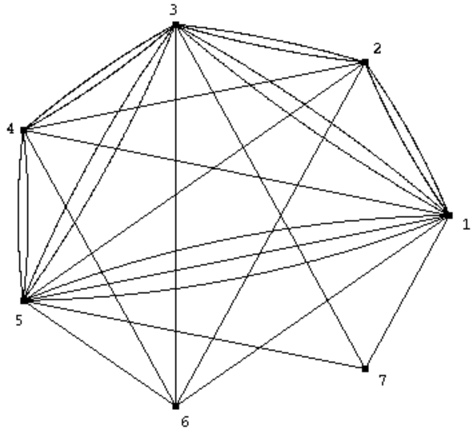
Enter: vv:= 3.5ring(7,0)

Enter: ee:=append[cyclecount4,completecount6,complete[1,3,5,7]]

Delete old graph. Enter and plot: draw[vv, ee]



Delete old graph. Enter and plot: draw[vv, amatee]
(Apply parameters to rest of plot list (On))



Can you spot:
the cycle [1,2,3,4];
the complete subgraph on [1,2,3,4,5,6];
the complete subgraph on [1,3,5,7] ?

Also enter and plot each of the following:
drawdi[vv, ee]
drawdi[vv, amatee]
drawb([vv, ee],–2)

<u>Example 17</u> A Markov Chain.

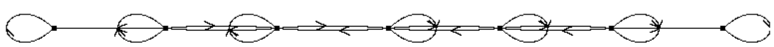
Enter: ml := [1 ; 1, 2, 3 ; 2, 3, 4 ; 3, 4, 5 ; 4, 5, 6 ; 5, 6, 7 ; 7] (“ml” not “m-one”)

Enter and Simplify: ma := amatlist(ml)

Enter: mv:= v_row([–7, 0 ; 7, 0], 7)

In 2D-plot Window select: Window>Tile Horizontally, and zoom in once vertically.

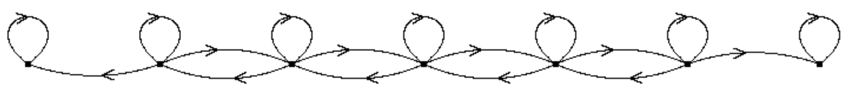
Enter and plot: draw[mv, ma]
to obtain:



This is not a very good representation of the Markov chain. We can do better by setting a bow factor on the edges and selecting an angle of $\pi/2$ (to the x-axis) for the axis of each loop as follows:

Delete the old plot and enter and 2D-plot:

drawdibl([mv, ma],2, π/2) (“bl” not “b-one”)
to obtain:



The same plot can also be obtained directly from the list by:

drawlistdibl([mv, ml],2, π/2) (“bl” not “b-one”)

Final Remarks

(i) Once you have loaded NGRAPHS.MTH as a Utility File, if you save the new file as a DFW file (preferably using a different filename), then all of the utilities of NGRAPHS.MTH will be saved and ready to use when you load this new file again. It will no longer be necessary to load NGRAPHS.MTH again to use the DFW file.

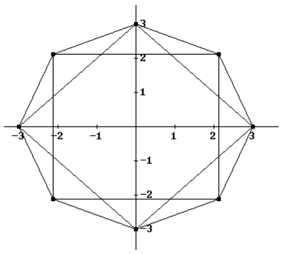
(ii) NGRAPHS.MTH is under continual development. I am interested in any suggestions for improvements and additions. Most of the development takes place in providing utilities for new types of graphs.

(iii) Many of the customized settings of DERIVE 5 can be saved in the .INI File for future use.

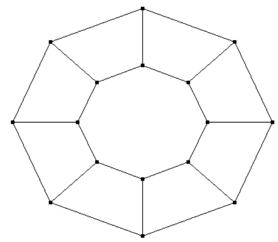
6 Some Exercises

Exercise 2 Draw the following graphs:

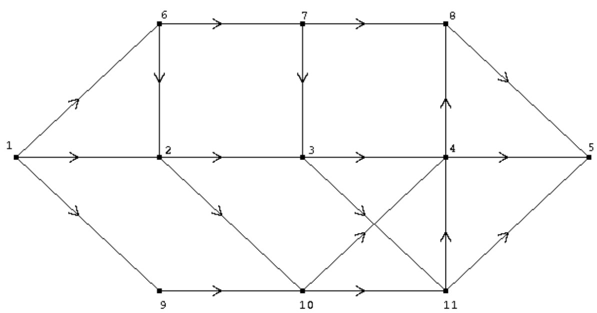
(i)



(ii)

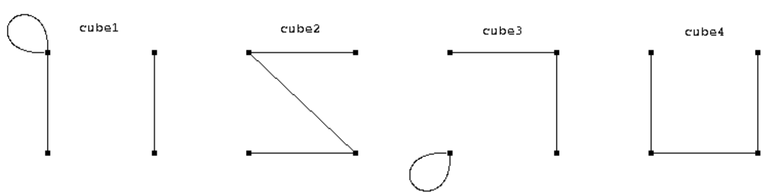


Exercise 3 Draw the graph of the following activity network:



Exercise 4

(i) Draw each of the graphs for the cubes of "instant insanity".



(ii) Combine them to form the following single graph