Chapter 19
Data Management

Many of Mathcad's built-in functions and operators are useful for manipulating sets of data. Mathcad therefore provides special mechanisms for importing various types of data into a Mathcad worksheet. And, once you've worked with the data, Mathcad allows you to export the results you've generated.

This chapter also discusses a group of Mathcad functions for reading and writing ASCII text files containing numerical data. These functions are primarily available for compatibility with Mathcad worksheets created in earlier versions of Mathcad.

This chapter contains the following sections:

Introduction to components

Overview of *components*, specialized OLE objects used for managing data in Mathcad.

Importing data

Using the File Read/Write component and the Input Table component to import data from a data file. Using the Input Table component to type data in manually or paste it in from the clipboard.

Exporting data

Using the File Read/Write component and the Output Table component to export data to a data file or to the clipboard.

Exchanging data with other applications

Using the Excel, MATLAB, and Axum components to create links between a Mathcad worksheet and these computational applications. Using the Scriptable Object component to script a custom component.

Functions for reading and writing ASCII data files

Description of ASCII file access functions for reading and writing structured and unstructured data.

Introduction to components

Mathcad provides *components* to exchange data between Mathcad and other applications. A component is a specialized OLE object that you insert into a Mathcad worksheet to create a link between the worksheet and either a data source or another application containing data. Unlike other OLE objects you insert into a worksheet, a component communicates with the mathematical equations in Mathcad, linking your Mathcad computations to the rest of your computing environment. This link allows data to flow dynamically between the Mathcad worksheet and a data source or application.

The following components are available in Mathcad for exchanging data between a Mathcad worksheet and other data sources or applications:

- File Read/Write, for importing and exporting data files in a variety of formats.
- Input Table and Output Table, for quickly getting data in and out of a Mathcad worksheet.
- Axum, for creating Axum graphs in a Mathcad worksheet.
- ***Pro*** Excel, for creating a link between a Mathcad worksheet and an Excel file.
- ***Pro*** MATLAB, for creating a link between a Mathcad worksheet and a MATLAB file.
- ***Pro*** Scriptable Object, for creating a custom component via a scripting language.

Some of these components are used to import data, some to export data, and application components, such as Excel, Axum, and MATLAB, allow you to communicate with other applications installed on your computer.

The data going into a component from Mathcad is called *input*. The data coming out of a component is called *output*. The input and output are passed between a Mathcad worksheet and a component by way of a Mathcad variable. The "input variable" is a variable defined in your Mathcad worksheet. It contains the data which will be passed into a component. Output from a component is used to define a Mathcad variable. This variable is referred to as the "output variable."

How to work with components

The basic steps to inserting a component in order to exchange data with Mathcad are as follows:

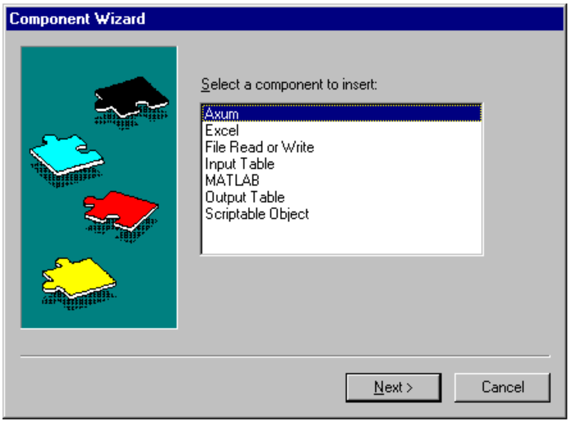
- Insert the component.
- Specify the input variable(s) and output variable(s).
- Configure the component so that it knows what to do with input and what to send as output.

Since some components only take input or only send output, these steps will differ slightly for each component. The ideas presented in the steps, however, should improve your understanding of how components work.

Step 1: Inserting a component

To insert a component into a Mathcad worksheet:

- Click in a blank spot of your Mathcad worksheet where you would like the component.
- Choose **Component** from the **Insert** menu.
- Choose one of the components from the list and click “Next.”



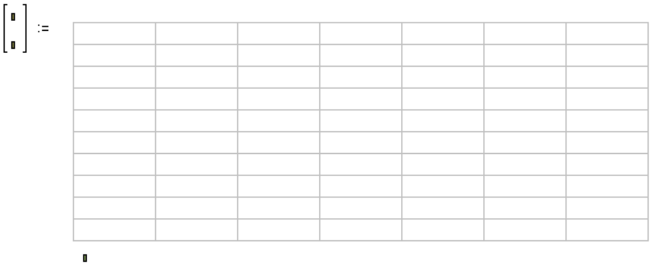
Select a component to insert:

Axum
Excel
File Read or Write
Input Table
MATLAB
Output Table
Scriptable Object

Depending on the component you choose, you may see a Wizard dialog box which lets you specify some properties of the component before it is inserted. Pressing the “Next” buttons continues through the Wizard. You can use the “Back” buttons to go back to a previous page. When you click the “Finish” button, the component will be inserted into your worksheet.

If you don’t see a Wizard when you choose one of the components from the Insert Component dialog box, you’ll immediately see the component, with some default properties, inserted into your worksheet.

Each component has its own look, so you will see something different for each component you choose. All components have one or more placeholders either to the right of the := or at the bottom of the component. For example, the Excel component might look like this when inserted into your worksheet:

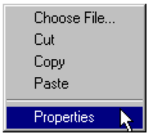


The placeholders you see to the left of the := are for the output variables which will contain the data passed from the component. The placeholder at the bottom of the component is for the name of an input variable the component is getting from Mathcad. The number of placeholders you see to the left or the bottom depend either on the default properties for the component or on the settings you provided in the component's Wizard.

Step 2: Configuring a component

Once you've inserted a component into a worksheet, you usually need to set up the component's properties so that the component knows how to handle any inputs it's getting from Mathcad and what to send as output. To configure the properties for a component:

- Click on the component once to select it.
- Click on it with the right mouse button to see a context menu, like the one shown below for the File Read/Write component. The options on the context menu depend on the component you selected, but usually you'll see the commands **Cut**, **Copy**, and **Properties**. Other options are specific to each component.
- Choose **Properties** from the context menu.



The settings in the Properties dialog box depend on the component you clicked on. The properties dialog for the Output Table component, for example, lets you specify the display format for the numbers in the table. The properties dialog for the Excel component in Mathcad Professional lets you specify the cells in which the input values are stored and the cells from which the output is sent.

Step 3: Exchanging data

Once you've inserted a component into a Mathcad worksheet, filled in the appropriate placeholders, and configured the properties, click away from the component region. At that point, the data exchange takes place: data will pass from the input variable into the component, and the output variable will become defined with the output. This will happen whenever you click on the component and press [**F9**], when the input variables change, or when you choose **Calculate Worksheet** from the **Math** menu.

MathConnex

Pro

The components available in Mathcad are used to connect a Mathcad worksheet to other data sources and applications. If you want to connect these data sources and applications to *each other* as well as to Mathcad, you can use the MathConnex application if you have Mathcad Professional.

In addition to the components available in Mathcad Professional, MathConnex contains a number of other components for manipulating data, such as a Mathcad component for connecting to a Mathcad worksheet. The MathConnex environment lets you connect any of one of the available components to any other component. MathConnex is therefore a tool for controlling data as it flows from one data source or application to another. You can visually design systems of data flow to analyze projects which involve a variety of applications and data sources.

To run MathConnex, click on the MathConnex icon from the Mathcad toolbar, or exit Mathcad and run MathConnex as you would any application. For more information, refer to the *MathConnex Getting Started Guide*.

Importing data

Although you can use Mathcad as a tool for working with variable and function definitions, you can also use Mathcad to perform calculations on data. In order to do so, you first need to import the data into a Mathcad worksheet using either the File Read/Write component or the Input Table component. For a general discussion of components see “Introduction to components” on page 428. You use these components to import data into a Mathcad array. Once the data are in Mathcad as an array, you can use any of Mathcad’s built-in functions and operators.

When you import data into Mathcad you should consider where your data is stored:

- It may be stored in a file created in another application.
- It may have been copied to the clipboard from another application.
- It may be on a piece of paper in front of you.

Mathcad therefore allows you to import data by reading from a data file, by pasting it in from the clipboard, or by typing it directly into a Mathcad worksheet.

Importing data from a data file

Mathcad can import data from a file created in any of a variety of applications, such as:

- Excel (*.xls)
- Professional edition of MATLAB (*.mat)
- Lotus 1-2-3 (*.wk*)
- ASCII editors (*.dat, *.csv, *.prn, *.txt)

When you want to import data from a data file into Mathcad, you should first decide whether you want the data to update automatically in Mathcad whenever it changes later in the data file.

If the data file is likely to change, and you want the changes to be reflected in your Mathcad worksheet whenever it is calculated, you can import and establish a connection between the data file and your Mathcad worksheet. To do so, use the Data File Read/ Write component. If you do not want the data to change in Mathcad if the data file changes later on, you can import without establishing a connection between Mathcad and the data file using the Input Table component.

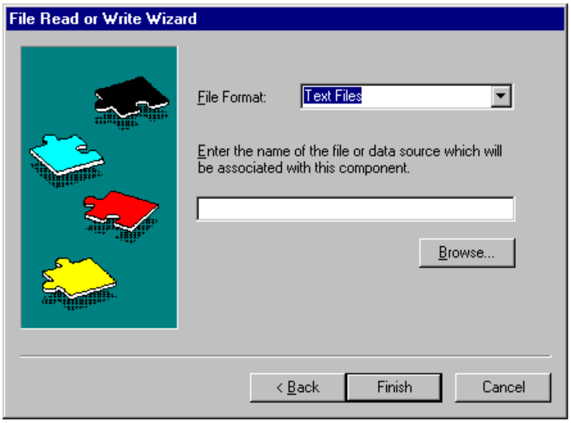
Connecting to a data file

To import data and establish a connection between your Mathcad worksheet and the data file, use the File Read/Write component:

- Click in a blank spot of your worksheet.
- Choose **Component** from the **Insert** menu.
- Choose File Read/Write from the list and click “Next.”

This launches the first part of the File Read or Write Wizard. Choose “Read from a data source” and press “Next” to go to the second page of the Wizard, shown below.

- From the File Format drop-down list in this Wizard, choose the type of data file you want to read.
- Type the path to the data file you want to read or use the “Browse” button to locate it.

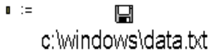


File Read or Write Wizard

File Format: Text Files

Enter the name of the file or data source which will be associated with this component.

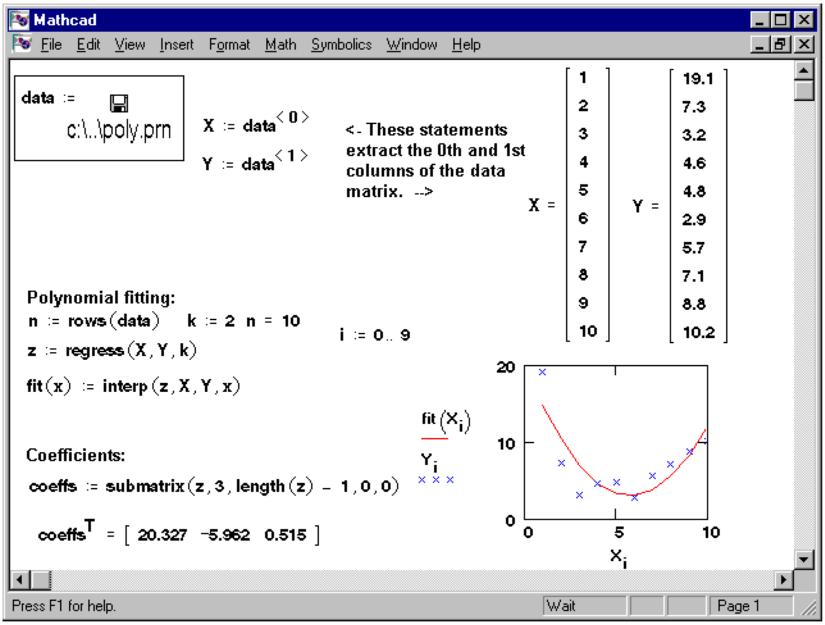
- Press “Finish.” You’ll see the File Read/Write component icon and the path to the data file. For example, if you read from a data file called data.txt, you’ll see:



■ :=

c:\windows\data.txt

In the placeholder that appears, enter the name of the Mathcad variable to which the data from the file will be assigned. When you click outside the component, the data file is read in and the data is assigned to the Mathcad variable you entered into the placeholder. Figure 19-1 shows an example of importing data using the File Read/Write component.



data := c:\..\poly.prn

$X := data^{\langle 0 \rangle}$

$Y := data^{\langle 1 \rangle}$

<- These statements extract the 0th and 1st columns of the data matrix. -->

$X = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{bmatrix}$ $Y = \begin{bmatrix} 19.1 \\ 7.3 \\ 3.2 \\ 4.6 \\ 4.8 \\ 2.9 \\ 5.7 \\ 7.1 \\ 8.8 \\ 10.2 \end{bmatrix}$

Polynomial fitting:

n := rows(data) k := 2 n = 10

i := 0.. 9

z := regress(X, Y, k)

fit(x) := interp(z, X, Y, x)

Coefficients:

coeffs := submatrix(z, 3, length(z) - 1, 0, 0)

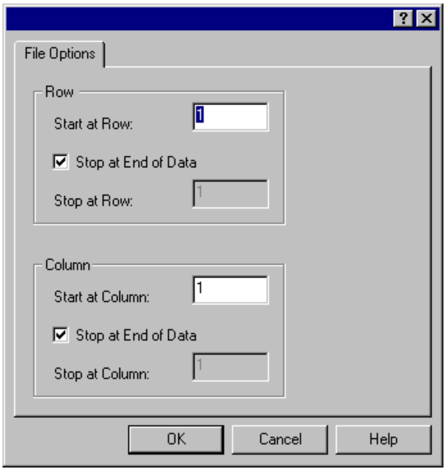
$coeffs^T$ = [20.327 -5.962 0.515]

Figure 19-1: Importing data from a data file. Whenever the data file changes, the data will automatically update in Mathcad.

You can now manipulate the Mathcad variable however you'd like using Mathcad's built-in functions and operators. Each time you calculate the worksheet, Mathcad will re-read the data from the file you have specified.

By default, Mathcad will read in the entire data file and assign the values to a matrix with the variable name you provide. To read in only certain rows or columns of a data file:

- Click once on the component to select it.
- Click with the right mouse button on the component so that you see the context menu for the component (see example on page 430).
- Choose **Properties** from the context menu to bring up the Properties dialog box:



Row

Start at Row: 1

☑ Stop at End of Data

Stop at Row: 1

Column

Start at Column: 1

☑ Stop at End of Data

Stop at Column: 1

- Use this dialog box to specify the row and columns at which to start and stop reading.

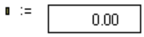
To read in a different data file or a different type of data file:

- Click with the right mouse button on the component and select **Choose File** from the component context menu.
- In the “Files of type” text box, choose the type of file you’d like to import. Use the dialog box to browse to the data file, select the data file, and click “Open.”

Importing once from a data file

You can use the Input Table component to import data once from a data file without establishing a connection between a Mathcad worksheet and the data file. To do so:

- Click in a blank spot of your worksheet.
- Choose **Component** from the **Insert** menu.
- Select Input Table from the list and click “Next.” The Input Table component will be inserted into your worksheet:



- In the placeholder that appears to the left, enter the name of the Mathcad variable to which this data will be assigned.
- Click with the right mouse button on the component so that you see the context menu for the component (see example on page 430).
- Choose **Import**.
- The Read from File dialog box appears. In the “Files of type” text box, choose the type of file you’d like to import. Use the dialog box to browse to the data file and click “Open.”

The data from the data file will appear in your worksheet in a table. See Figure 19-2 for an example.

When you double-click on the table, you can edit the values in it. You'll see scroll bars which let you scroll through the table. You'll also see handles along the sides of the component region. To resize the table, move the cursor to one of these handles so that it changes to a double-headed arrow, press and hold down the mouse button and drag the cursor in the direction you want the table's dimensions to change.

When you click outside the component, the data is assigned as an array to the Mathcad variable you specified. You can now manipulate this Mathcad array however you'd like using Mathcad's built-in functions and operators. Unlike the File Read/Write component, the Import feature of the Input Table component reads the data only when you choose **Import**, not each time you calculate the worksheet.

Importing data from the clipboard

In some cases, you may have data stored in a spreadsheet or another application, but you don't want to import the data as a file into Mathcad. An alternative to importing a file is to copy the data to the clipboard and paste it into Mathcad. To do so:

- Select the data in the other application and copy it to the clipboard. In most applications, you do this by choosing **Copy** from the **Edit** menu.
- Open the Mathcad worksheet in which you want the data to be used.
- In your Mathcad worksheet, type a variable definition such as **`m:=`**
- Click in the placeholder to the right of the := and choose **Paste** from the **Edit** menu. A matrix containing the data you copied will appear.

When you click outside the component, the data you copied will appear in a scrolling table and are assigned as an array to the Mathcad variable you specified. You can manipulate the array however you'd like using Mathcad's built-in functions and operators.

Note that if you paste the copied data into a blank spot of a Mathcad worksheet instead of into a placeholder, you'll also see a matrix containing the data you selected.

When you double-click on the table, you can edit the values in it. You'll see scroll bars which let you scroll through the table. You'll also see handles along the sides of the component region. To resize the table, move the cursor to one of these handles so that it changes to a double-headed arrow, press and hold down the mouse button and drag the cursor in the direction you want the table's dimensions to change.

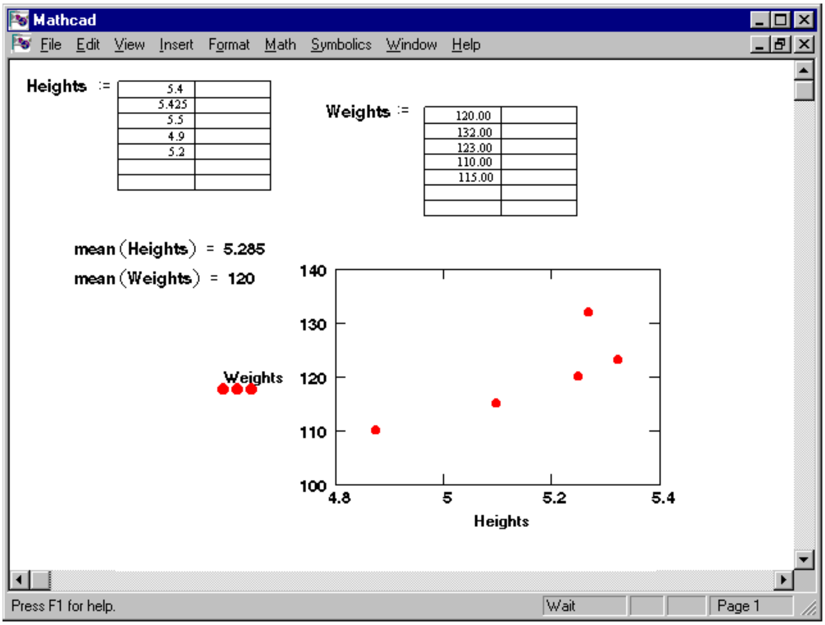
Entering data manually

It is usually most convenient to have your data stored somewhere on your computer so that you can import the data into Mathcad, either as a file or from the clipboard. However, when you don't have a lot of data and it is not already stored in another application, you can enter your data directly into Mathcad. To do so, create an input table using the Input Table component:

- Click in a blank spot in your worksheet and choose **Component** from the **Insert** menu.

- Select **Input Table** from the list and click "Next." The Input Table component will be inserted into your worksheet.
- In the placeholder that appears, enter the name of the Mathcad variable to which this data will be assigned.
- Double-click on the component and enter data into the cells. Each row must have the same number of data values. If you do not enter a number into a cell, Mathcad inserts 0 into the cell.

Figure 19-2 shows two input tables. Notice that when you create an input table, you're actually assigning elements to an array that has the name of the variable you entered into the placeholder. Once you've created an array using an input table, you can manipulate the array using Mathcad's built-in functions and operators. Note that if you have a set of *X* and *Y* data values, you will probably find it easiest to create one input table for the *X*'s and one for the *Y*'s.



Heights :=

5.4	
5.425	
5.5	
4.9	
5.2	

Weights :=

120.00	
132.00	
123.00	
110.00	
115.00	

mean(Heights) = 5.285

mean(Weights) = 120

Press F1 for help.

Figure 19-2: Using the Input Table component to create arrays of data.

When you double-click on the table, you can edit the values in it. You'll see scroll bars which let you scroll through the table. You'll also see handles along the sides of the component region. To resize the table, move the cursor to one of these handles so that it changes to a double-headed arrow, press and hold down the mouse button and drag the cursor in the direction you want the table's dimensions to change.

You can copy data from the Input Table component by selecting data, clicking with the right mouse button on the component, and choosing **Copy Data** from the context menu. You can also paste data into the Input Table component by selecting a cell and choosing **Paste Data** from the context menu.

Another way to create an input table is to use a range variables as subscript for another Mathcad variable name. See the section “Entering a table of numbers” in Chapter 11 for more information.

Exporting data

Once you’ve used Mathcad to perform calculations, you can export the results either to a data file or to the clipboard for pasting into another application. The File Read/ Write component and the Output Table component allow you to do this. For a general discussion of components see “Introduction to components” on page 428. You may use any Mathcad variable you have defined in your worksheet as an input variable to the component. The component either sends the data to a data file or lets you copy it to the clipboard.

Exporting to a data file

Mathcad allows you to export the values stored in a Mathcad variable to a variety of file formats, such as the following:

- Excel (*.xls)
- MATLAB (*.mat) (Professional edition only)
- Lotus 1-2-3 (*.wk*)
- ASCII editors (*.dat, *.csv, *.prn, *.txt)

When you want to export data to a data file, you should first decide whether you want the data to update automatically in the data file when it changes in Mathcad.

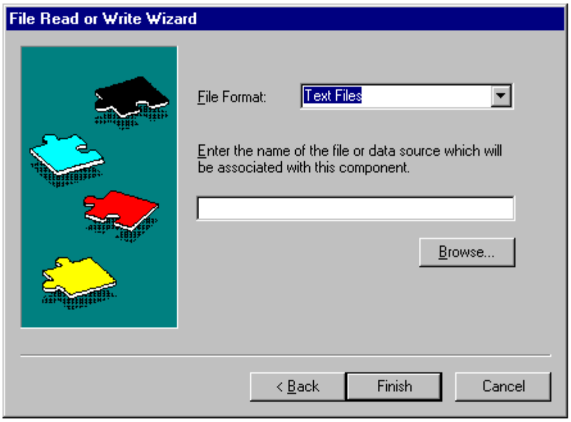
If your results are likely to change in Mathcad, and you want to export the new results to a data file whenever the worksheet is calculated, you can export the data and establish a connection between the data file and your Mathcad worksheet. To do so, use the File Read/Write component. If you do not want the data file to change if the results in Mathcad change later on, you can export without establishing a connection between Mathcad and the data file. To do so, use the Output Table component.

Exporting to a connected data file

To export values stored in a Mathcad array or scalar variable to a data file and establish a connection between your Mathcad worksheet and the data file:

- Click in a blank spot in your worksheet.
- Choose **Component** from the **Insert** menu.
- Select File Read/Write from the list and click “Next.”

This launches the first part of the File Read or Write Wizard in which you can specify whether the component will read from a data file or write to one. Choose “Write to a data source” and press “Next” to go to the second page of the Wizard:



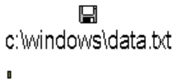
File Read or Write Wizard

File Format: Text Files

Enter the name of the file or data source which will be associated with this component.

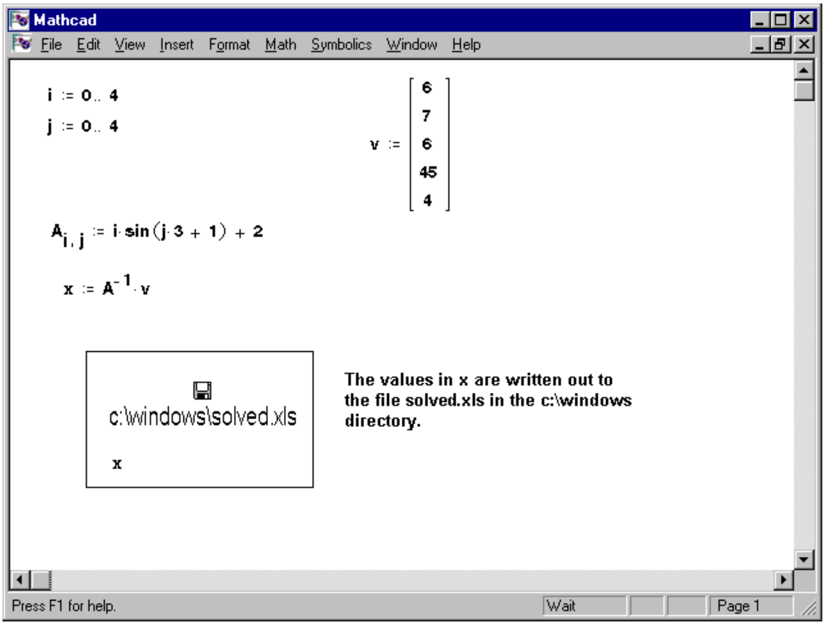
- From the File Format drop-down list in this Wizard, choose the type of data file you want to write.
- Type the path to the data file you want to write or click the “Browse” button to locate it.

Press “Finish.” You’ll see the File Read/Write component icon and the path to the data file:



c:\windows\data.txt

In the placeholder that appears, enter the name of the Mathcad variable containing the data which will be written to the data file. When you click outside the component, all the values in the input variable will be written to the filename you specified. Each time you calculate the worksheet, the data file is rewritten. See Figure 19-3 for an example.



$i := 0 .. 4$

$j := 0 .. 4$

$$v := \begin{bmatrix} 6 \\ 7 \\ 6 \\ 45 \\ 4 \end{bmatrix}$$

$A_{i,j} := i \cdot \sin(j \cdot 3 + 1) + 2$

$x := A^{-1} \cdot v$

c:\windows\solved.xls

x

The values in x are written out to the file solved.xls in the c:\windows directory.

Figure 19-3: Using the File Read/Write component to export data from Mathcad.

To change the name of the data file being created to or to change the type of file being created:

- Click once on the component to select it.
- Click with the right mouse button on the component so that you see the context menu for the component (see example on page 430).
- Select **Choose File.** You will see the Write to File dialog box.
- In the “Files of type” text box, choose the type of file you’d like to create. Use the dialog box to browse to the folder in which the data file will be created and click “Open.”

Exporting once to a data file

To export data without establishing a connection between Mathcad and the data file, use the Output Table component:

- Click in a blank spot in your worksheet.
- Choose **Component** from the **Insert** menu.
- Select Output Table from the list and click “Next.” The Output Table component will be inserted into your worksheet.

- In the placeholder that appears, enter the name of the Mathcad variable containing the data to be exported.
- Click on the component with the right mouse button so that you see the context menu for the component (see example on page 430).
- Choose **Export**. You will see the Write to File dialog box.
- In the "Files of type" text box, select the format of the file you'd like to create. Use the dialog box to browse to the folder in which the data file will be created and enter the name of the data file you wish to create. Then click "Open."

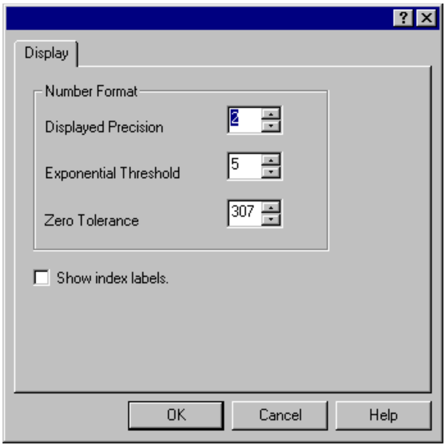
When you click outside the component, the data you see in the Output Table component will be written to the filename you specified. Unlike the File Read/Write component, the export feature in the Output Table component writes the data only when you choose Export, *not* each time you calculate the worksheet.

When you double-click on the table, you'll see scroll bars which let you scroll through the table. You'll also see handles along the sides of the component region. To resize the table, move the cursor to one of these handles so that it changes to a double-headed arrow, press and hold down the mouse button and drag the cursor in the direction you want the table's dimensions to change.

Changing the format of the numbers in the table

The numbers displayed in the Output Table component will display using a default format specific to the component. Unlike most other values displayed in Mathcad, the values in the Output Table component are not affected by the settings in the Number Format dialog box. To change the way the numbers are displayed in the component:

- Click once on the component to select it.
- Click on the component with the right mouse button so that you see the context menu.
- Choose **Properties** from the context menu.
- Go to the Display tab, which looks like this:



Displayed Precision 2

Exponential Threshold 5

Zero Tolerance 307

□ Show index labels.

The settings on the Display tab are as follows:

Displayed Precision: Choose or type an integer *n*. Determines the number of values to the right of the decimal place.

Exponential Threshold: Choose or type an integer *n*. Values smaller than 10^{-n} or greater than 10^{n} are displayed in exponential notation.

Zero Tolerance: Choose or type an integer *n*. Values smaller than 10^{-n} are displayed as zero.

Show index labels: Click this box to display the row and column numbers associated with the values in the table.

Keep in mind that these settings only affect the display of numbers in the Output Table component. They do not affect the values written to a data file if you choose **Export** from the context menu of the Output Table component.

Exchanging data with other applications

As described earlier in this chapter, Mathcad can import and export data files in formats recognized by other applications, including data files saved in spreadsheet programs. This is useful when you simply want to connect to a data file as opposed to the application in which it was created.

For some work, however, you may need access to functionality in the other application as well as the data. It is convenient, then, to set up a link between Mathcad and the other application. A link allows you to exchange data and have immediate access to the other application from within Mathcad.

By linking an application to your Mathcad worksheet, you can:

- Send values from Mathcad to the application.
- Use the application to manipulate the data dynamically without actually leaving the Mathcad environment.
- Send values from the application back to Mathcad.

Although performing tasks of all three kinds takes full advantage of the link, you can also use the link to perform just one or two of them.

To create a connection between Mathcad and another application, you'll use an application component. (For a general discussion of components, see "Introduction to components" on page 428.) When you insert an application component into your Mathcad worksheet, you'll actually see a small window on that application's environment embedded in your Mathcad worksheet. When you double-click on the component, Mathcad's menus and toolbars change to those of the other application. This gives you access to the application without leaving the Mathcad environment.

Mathcad provides components for creating links to Microsoft Excel, MathSoft's Axum (a data analysis and graphics application), and MATLAB from The MathWorks, Inc. The Scriptable Object component allows you to create a custom component to any other application using a scripting language.

The Excel component

Pro

You can use the File Read/Write component to import and export data to and from an Excel file as described in the sections "Connecting to a data file" on page 432 and "Exporting to a connected data file" on page 437. However, there may be times when you want to exchange data between an Excel file and a Mathcad worksheet, and you need brief access to the functionality in Excel, not just the data stored there. For example, you may have some values in Mathcad that you want to send to Excel, use Excel for some manipulation, and send values back to Mathcad to continue working.

The Excel component lets you:

- Pass input variable from Mathcad into an Excel file.
- Double-click on the Excel component to use functionality from Excel.
- Send values from an Excel file which define output variables in Mathcad.

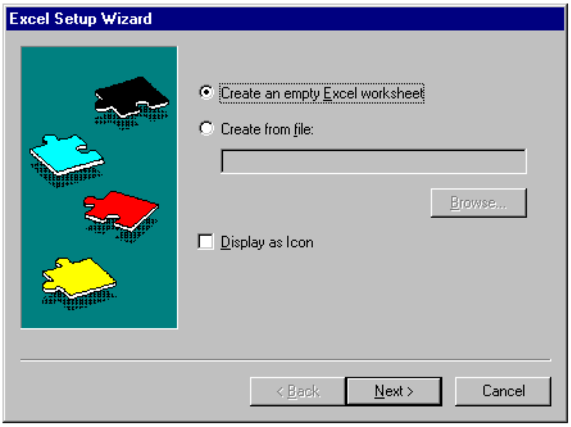
Inserting an Excel component

Before you insert an Excel component into a Mathcad worksheet, Excel needs to be installed on your system, but it does not need to be running.

To insert an Excel component into a Mathcad worksheet:

- Click in a blank spot in your worksheet. If you want to send values to the component from a Mathcad variable defined in your worksheet, make sure you click below or to the right of the variable definition.
- Choose **Component** from the **Insert** menu.

- Select Excel from the list and click “Next.” This launches the first part of the Excel Setup Wizard:

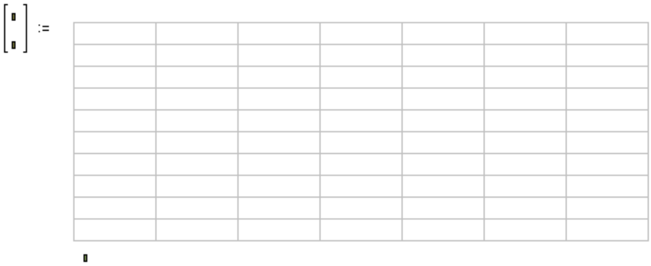


- If you want to connect a file you’ve already created, choose “Create from file,” and type the path name in the text box or use the Browse button to locate the file; then click “Open.” If you don’t want to connect to a previously created file, choose “Create an empty Excel Worksheet.”
- Click the Display as Icon box if you want to see an icon in your Mathcad worksheet rather than a portion of the Excel file.

When you click on the “Next” button, the Wizard will bring you through other dialog boxes where you will specify:

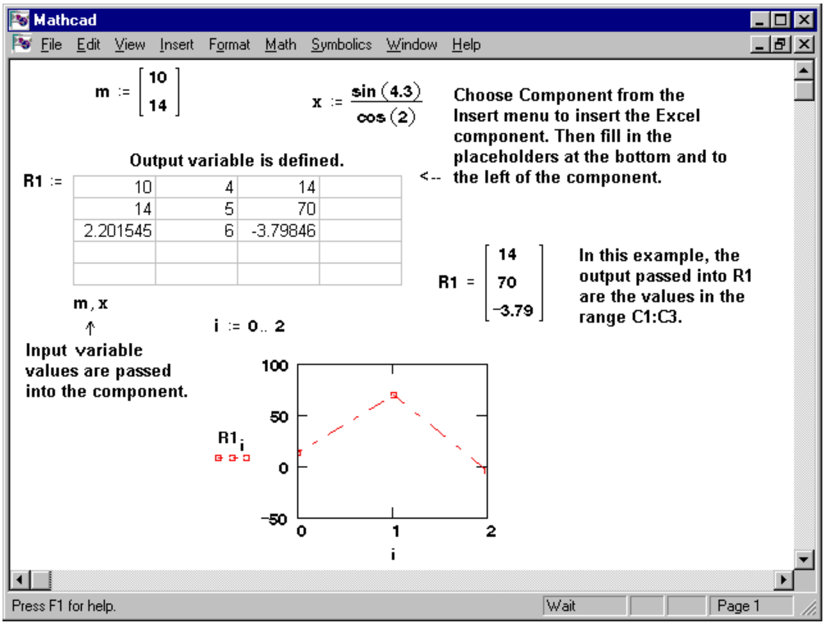
- **The number of input and output variables.** You can supply between 0 and 4 input variables and between 0 and 4 output variables.
- **Input ranges:** the cells in which the values of each Mathcad input variable will be stored. Enter the starting cell, which is the cell that will hold the element in the upper left corner of an array of values. For example, for an input variable containing a 3×3 matrix of values, you can specify A1 as the starting cell, and the values will be placed in the cells A1 through C3.
- **Output ranges:** the cells whose values will define the Mathcad output variables. For example, enter C2:L11 to extract the values in cells C2 through L11 and create a 10×10 matrix.

When you finish using the Wizard, you’ll see the Excel component in your worksheet with placeholders for the input and output variables. For example, an Excel component that will be sent one input variable and whose output will define two output variables would look like this:



Enter the input variables in the bottom placeholders. Enter the output variables into the placeholders to the left of the :=.

Figure 19-4 shows an example of an Excel component inserted into a Mathcad worksheet.



$m := \begin{bmatrix} 10 \\ 14 \end{bmatrix}$

$x := \frac{\sin(4.3)}{\cos(2)}$

Choose Component from the Insert menu to insert the Excel component. Then fill in the placeholders at the bottom and to the left of the component.

Output variable is defined.

R1 := <--

10	4	14	
14	5	70	
2.201545	6	-3.79846	

m , x

↑

Input variable values are passed into the component.

$i := 0..2$

$R1 = \begin{bmatrix} 14 \\ 70 \\ -3.79 \end{bmatrix}$

In this example, the output passed into R1 are the values in the range C1:C3.

Press F1 for help. Wait Page 1

Figure 19-4: Inserting an Excel component into Mathcad to exchange data between an Excel file and a Mathcad worksheet.

When you click outside the component, input variables are sent to Excel from Mathcad and a range of cells are sent from Excel back to Mathcad and assigned to output variables. Once the output variables are defined, you can manipulate them however you'd like using Mathcad's built-in functions and operators.

By default, the Excel component displays only some of the rows and columns. To see more or fewer rows and columns, double-click on the component so that you see handles along the sides of the component region. Move the cursor to one of these handles so that it changes to a double-headed arrow. Press and hold down the mouse button and drag the cursor in the direction you want the component's dimensions to change.

Making changes to the inputs and outputs

Once you set up an Excel component, you may find that you want to add or remove input or output variables, or you may want to change the cell ranges for inputs and outputs which you initially specified in the Setup Wizard. To do so:

- Click once on the component to select it.
- Click on the component with the right mouse button to bring up the context menu.
- To add another output, choose “Add Output Variable.” Another placeholder will appear to the left of the :=.
- To add another input, choose “Add Input Variable.” Another placeholder will appear at the bottom of the component.
- To eliminate an output, choose “Remove Output Variable.” An output placeholder or variable to the left of the := will disappear. If there was more than one output, the last output placeholder or variable will disappear.
- To eliminate an input, choose “Remove Input Variable.” An input variable or placeholder at the bottom of the component will disappear. If there was more than one input, the placeholder furthest to the right will disappear.

When you add input or output variables, you’ll need to specify which cells in the component will store the new input and which will provide the new output. To do so:

- Click once on the component to select it.
- Click on the component with the right mouse button until you see the context menu.
- Choose **Properties** from the context menu.
- Choose either the Inputs tab or the Outputs tab and specify a range of cells for each input and each output.

You should also follow these steps if you want to change the cell ranges for inputs and outputs you initially specified in the Setup Wizard.

Accessing Excel

After inserting an Excel component into a Mathcad worksheet, you can use the component to perform calculations in Excel, provided you have installed Excel on your system. To do so:

- Double-click on the Excel component in the Mathcad worksheet. The Excel component opens and the menus and toolbars change to Excel menus and toolbars.
- Edit the Excel component however you’d like.
- Click back in the Mathcad worksheet to have the component recalculate and to resume working in Mathcad.

The MATLAB component

Pro

You can use the File Read/Write component to import and export data to and from a MATLAB file as described in the sections “Connecting to a data file” on page 432 and

"Exporting to a connected data file" on page 437. However, there may be times when you want to exchange data between a MATLAB worksheet and a Mathcad worksheet, and you need brief access to MATLAB's functionality, provided you have installed MATLAB (Professional edition) on your system. For example, you may have some values in Mathcad that you want to send to MATLAB, use MATLAB to manipulate them, and send values back to MATLAB.

A MATLAB component lets you:

- Pass values stored in variables in Mathcad into MATLAB variables.
- Double-click on the MATLAB component to use MATLAB to process the data.
- Send data from MATLAB variables into Mathcad variables.

Before you insert an MATLAB component into a Mathcad worksheet, MATLAB needs to be installed on your system, but it does not need to be running.

Inserting a MATLAB component

To insert a MATLAB component into a Mathcad worksheet:

- Click in a blank spot in your worksheet. If you want to send values to the MATLAB component from a Mathcad variable, make sure you click below or to the right of the variable definition.
- Choose **Component** from the **Insert** menu.
- Select MATLAB from the list and click "Next." The MATLAB component will be inserted into your worksheet.
- In the placeholder that appears at the bottom, enter the name of the Mathcad input variables to pass into the MATLAB component. In the placeholder that appears to the left of the component, enter the names of the Mathcad output variables to be defined. You can add more placeholders or remove some in the same way that you add and remove inputs and outputs for the Excel component as described in the section "Making changes to the inputs and outputs" on page 445.

By default, the input variables will be sent into MATLAB variables named **`in0`**, **`in1`**, **`in2`**, and so on refer. The MATLAB variables **`out0`**, **`out1`**, **`out2`**, and so on will define the output variables to be created in Mathcad. To change these names:

- Choose **Properties** from the component's context menu.
- Use the Outputs and Inputs tabs to type in new names for the inputs and outputs.

Accessing MATLAB

Once you know what MATLAB variable names take the input and provide the output, you should set up your MATLAB component just as you would create a file in MATLAB. Be sure to define the variables which will be passed as output. To use the MATLAB component to perform calculations in MATLAB:

- Double-click on the MATLAB component in the Mathcad worksheet.
- The MATLAB component opens a text window for entering MATLAB commands.

- Edit the MATLAB worksheet however you'd like.

When you click outside the component, input variables are sent to MATLAB and arrays are assigned to output variables in Mathcad. Once the output variables are created in Mathcad, you can manipulate them however you'd like using Mathcad's built-in functions and operators.

The Axum component

Axum is a technical graphing and data analysis application available from MathSoft. Axum gives you access to over 75 2D and 3D graph types. You also get complete control over the look of your graphs because you can point and click to make just about any change you want.

If you have Axum 5 (patch level 5.03 or higher) installed on your system, the Axum component brings some of this graphing power to your Mathcad worksheet and gives you the power to create publication-quality graphs from your Mathcad data without leaving the Mathcad environment. (For a general discussion of components, see "Introduction to components" on page 428.)

Using the Axum component you can:

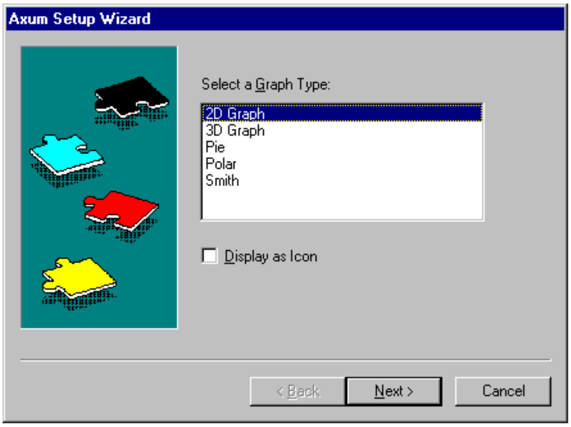
- Create over 50 different types of 2D and 3D graphs using Mathcad data. For example, the Axum component lets you create the following: a 2D Area Chart, a 2D Grouped Bar Chart, and a 3D Regression Plot.
- Double-click on the Axum component to use Axum to format every detail of your graph.

Inserting an Axum Graph

Before you insert an Axum component into a Mathcad worksheet, Axum needs to be installed on your system, but it does not need to be running.

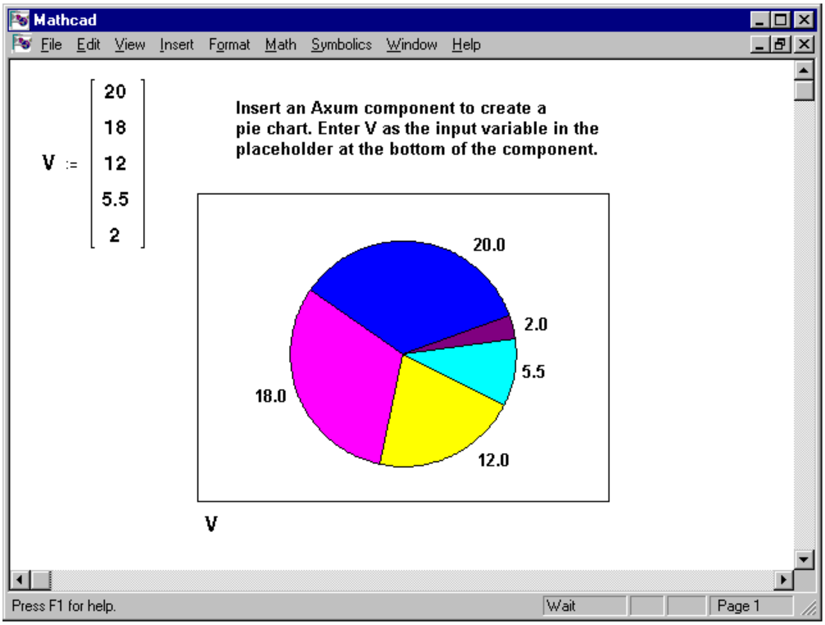
To insert an Axum component into a Mathcad worksheet:

- Create the arrays which will provide input to the Axum component and which Axum will display in a graph. For information on the number and type of arrays required for each type of graph, choose **Mathcad-Axum Link** from Axum's **Help** menu and go to the section titled "Mathcad data specifications for creating Axum graphs."
- Click in a blank spot in your worksheet. Make sure you click below or to the right of the arrays.
- Choose **Component** from the **Insert** menu.
- Select Axum from the list and click "Next." This launches the first part of the Axum Setup Wizard.



- Choose they type of graph you would like to insert and click “Next” to go to the next page of the Wizard.
- Choose a plot type. The available choices depend on the type of graph you selected.
- Click “Finish.” The Axum component will be inserted into your worksheet.
- Enter the appropriate array variables in the placeholders that appear.

When you click outside the component, the array data will be displayed in the Axum plot type you selected in the Wizard. Each time you calculate the worksheet, the data is redisplayed in Axum. See Figure 19-5 for an example.



$$V := \begin{bmatrix} 20 \\ 18 \\ 12 \\ 5.5 \\ 2 \end{bmatrix}$$

Insert an Axum component to create a pie chart. Enter V as the input variable in the placeholder at the bottom of the component.

Figure 19-5: The Axum component lets you create graph types that aren’t directly available in Mathcad.

Editing an Axum Graph

After inserting an Axum component into a Mathcad worksheet, you can format the graph using Axum's formatting options. To do so:

- Double-click on the Axum graph in the Mathcad worksheet.
- The menus and toolbars change to Axum's menus and toolbars. Edit the Axum graph however you'd like.
- Click back in the Mathcad worksheet to recalculate the component and to resume working in Mathcad.

The Scriptable Object component

Pro Although Mathcad has components for exchanging data between a Mathcad worksheet and Excel, MATLAB, and Axum, you can actually exchange data between a Mathcad worksheet and any other application that supports OLE Automation, even if Mathcad does not have a specific component to do so. You can use the Scriptable Object component to create a custom component that:

- Sends values from Mathcad to the application.
- Uses the application to manipulate the data without actually leaving Mathcad.
- Sends values from the other application back to Mathcad.

You can create a custom scriptable object from any object you can insert into a Mathcad worksheet. For example, you may want to send some values from Mathcad to Lotus 1-2-3, manipulate the values there, and send results back to the Mathcad worksheet.

To create a Scriptable Object component, you must:

- Be proficient in a supported scripting language, such as Microsoft VBScript or JScript, that is installed on your system.
- Have some knowledge of the way the other application has implemented OLE.
- Have the other application available on your system.

More on scripting languages

Before you insert a Scriptable Object component into a Mathcad worksheet, you need to have a scripting language installed on your system. As this *User's Guide* goes to press, the following two scripting languages are supported: Microsoft VBScript (Visual Basic Scripting Edition) version 2 and Microsoft JScript (an implementation of JavaScript) version 2. Both of these scripting languages can currently be downloaded at no charge from Microsoft, Inc.'s Web sites:

`http://www.microsoft.com/vbscript`
`http://www.microsoft.com/jscript`

VBScript is a strict *subset* of the Visual Basic for Applications language used in Microsoft Excel, Project, Access, and the Visual Basic 4.0 development system. VBScript is designed to be a lightweight interpreted language, so it does not use strict types (only Variants). Also, because VBScript is intended to be a safe subset of the language, it does not include file input/output or direct access to the underlying operating system.

JScript is a fast, portable, lightweight interpreter for use in applications that use ActiveX Controls, OLE automation servers, and Java applets. JScript is directly comparable to VBScript (not Java). Like VBScript, JScript is a pure interpreter that processes source code rather than producing stand-alone applets.

The syntax and techniques used in the scripting language you choose are beyond the scope of this *Getting Started Guide*. For more information about Microsoft VBScript and JScript, consult Microsoft, Inc.'s Web sites.

Inserting a Scriptable Object

To insert a Scriptable Object component into a Mathcad worksheet:

- Click in a blank spot in your worksheet. If you want to send values to the object from a Mathcad variable, make sure you click below or to the right of the variable definition.
- Choose **Component** from the **Insert** menu.
- Select Scriptable Object from the list and click "Next."

This launches the Scripting Wizard. The Wizard first prompts you to specify the OLE server application from which you want to create a Scriptable Object. The Object to Script scrolling list shows the available applications on your system. Choose an application that supports the OLE2 automation interface (consult the documentation for the application for details).

You must specify:

- Whether the component will be a new file or whether you will insert a file that already exists.
- Whether you will see the actual file in your Mathcad worksheet or if you'll see an icon.

When you click on the "Next" button, the Wizard will bring you through other dialog boxes where you will specify:

- Which scripting language you will use.
- The type of object you want to script.
- The name of the object.
- The number of inputs and outputs the object will accept and provide.

When you finish using the Wizard, you'll see a Scriptable Object component in your worksheet with placeholders for the input and output variables. Enter the input variables in the bottom placeholders. Enter the output variables into the placeholders to the left of the :=.

Figure 19-6 shows a simple example of a scripted object—in this case, a picture of a face. The object accepts two input values and sends out two output values. The input values determine whether the face smiles or winks.

Object model

The Scriptable Object component has the following predefined objects, properties, and methods that enable you to configure it to work as a Mathcad component.

Collections

- **`Inputs`** and **`Outputs`** are predefined *collections* of DataValue objects (see below) containing the Scriptable Object's inputs and the outputs, respectively.
- The **`Count`** property can be used to query the total number of elements in the collection. For example, **`Outputs.Count`** returns the number of output ports.
- The **`Item`** method is used to specify an individual element in the collection. To refer to a particular input or output, use the notation **`Inputs.Item(`***n***`)`** or **`Outputs.Item(`***n***`)`**, where *n* is the index of the input or output. The index *n* always starts from 0. Since **`Item`** is the default method, languages such as VBScript and JScript let you drop the method name to imply the default method. For example, **`Inputs(0)`** is equivalent to **`Inputs.Item(0)`** and references the first input.

DataValue objects

- The **`Value`** property accesses a DataValue's real part. For example, in VBScript or JScript **`Inputs(0).Value`** returns the real part of the first input.
- The **`IValue`** property accesses a DataValue's imaginary part. For example, in VBScript or JScript **`Outputs(1).IValue`** returns the imaginary part of the second output. If there is no imaginary part, the **`IValue`** portion returns "NIL."
- The **`IsComplex`** property returns "TRUE" if a DataValue has a valid imaginary part; this property returns "FALSE" otherwise. For example, the expression **`(inputs(0).IsComplex)`** returns "FALSE" if the first input has only a real part.
- The **`Rows`** and **`Cols`** properties yield the number of rows and columns.

Global methods

- The **`alert`** function takes a single string parameter that is presented to the user as a standard modal Windows message box with an "OK" button.
- The **`errmsg`** function takes a single string parameter that displays as an error message from within the script and causes the script to stop execution. The second parameter is also a string, but it is optional. It is used to display the source of the error.

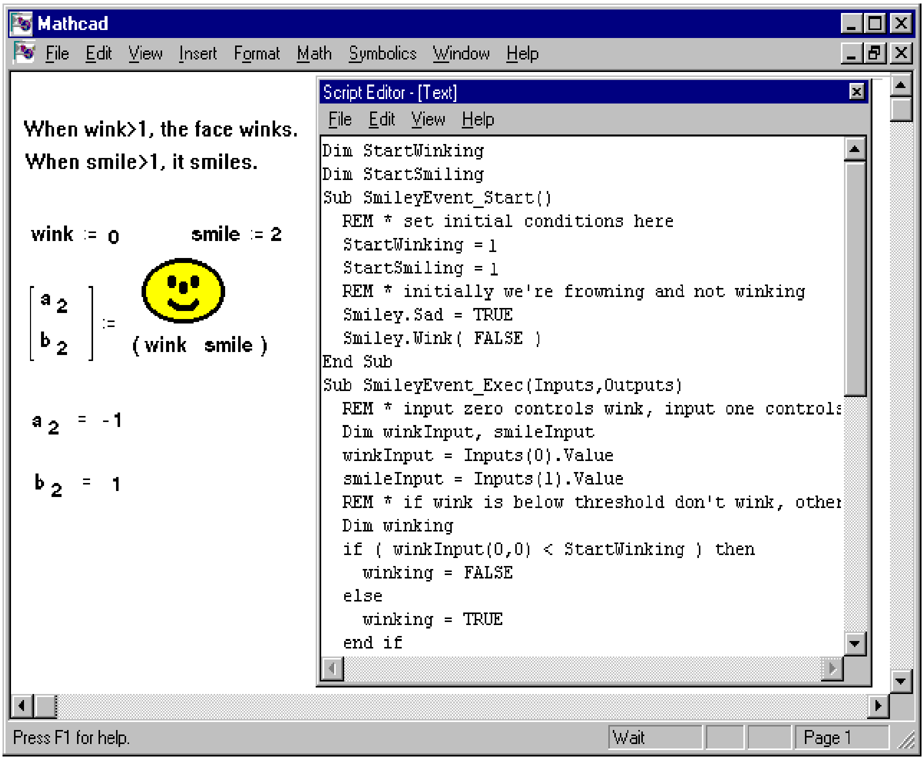
Note: In JScript, the names of functions, methods, objects, and properties are case sensitive, while in VBScript they are not.

Scripting the object

To start scripting an object:

- Click once on the component to select it.
- Click on the component with the right mouse button.
- Choose **Edit Script** from the context menu.

You'll see a new window called the Script Editor containing three subroutine stubs in which you insert your own scripting code. Figure 19-6 shows a portion of the completed VBScript script for an object called **`SmileyEvent`**.



When wink>1, the face winks.
When smile>1, it smiles.

$wink := 0 \qquad smile := 2$

$$\begin{bmatrix} a_2 \\ b_2 \end{bmatrix} := (wink \;\; smile)$$

$a_2 = -1$

$b_2 = 1$

```
Dim StartWinking
Dim StartSmiling
Sub SmileyEvent_Start()
  REM * set initial conditions here
  StartWinking = 1
  StartSmiling = 1
  REM * initially we're frowning and not winking
  Smiley.Sad = TRUE
  Smiley.Wink( FALSE )
End Sub
Sub SmileyEvent_Exec(Inputs,Outputs)
  REM * input zero controls wink, input one controls
  Dim winkInput, smileInput
  winkInput = Inputs(0).Value
  smileInput = Inputs(1).Value
  REM * if wink is below threshold don't wink, other
  Dim winking
  if ( winkInput(0,0) < StartWinking ) then
    winking = FALSE
  else
    winking = TRUE
  end if
```

Figure 19-6: A scripted object. The Script Editor window shows a portion of the VBScript code used to configure the object.

The script you write will usually contain at a minimum the following three subroutines:

- A *starting* routine, called once when execution of the component begins. This is a good place to initialize variables, open files for reading and writing, etc.
- An *execution* routine that by default takes as arguments the collections **`Inputs`** and **`Outputs`**.
- A *stopping* routine, called once when execution of the component stops.

The commands in each section are executed in sequence whenever the Mathcad worksheet is calculated. What you will include in these subroutines is determined largely by the properties of the OLE object you are scripting; consult the documentation for the server or control.

Choose **Close & Return** from the Script Editor’s **File** menu when you have completed your script and want to resume working in the Mathcad worksheet.

Functions for reading and writing ASCII data files

If the data file you want to import or export to is a plain ASCII file, you can use components for importing and exporting data as described in the first three sections of this chapter. Alternatively, you can use Mathcad’s built-in functions for reading plain ASCII data files. These built-in functions are provided mainly for compatibility with worksheets created in earlier versions of Mathcad, but you can use them in a Mathcad 7 worksheet if you like.

Files in plain ASCII format consist only of numbers separated by commas, spaces, or carriage returns. The following are some examples of files that the built-in functions can read, assuming the data are in ASCII format:

- A file containing experimental data captured with data-acquisition hardware and software.
- A file created by printing from a spreadsheet program to the disk.
- A column of numbers typed into a word processor and saved in plain ASCII format.
- Output from a BASIC program.
- Data downloaded from a mainframe database.

The numbers in the data files can be integers like **3** or **–1**, floating-point numbers like **2.54**, or E-format numbers like **4.51E-4** (for $4.51 \cdot 10^{-4}$).

For example, this list of numbers would be a valid line in an ASCII data file:

```
200, 50 25.1256, 16E–2, –16.125E15
```

Mathcad also has functions for saving data in ASCII files. Data files saved by Mathcad with these functions contain numbers separated by spaces and carriage returns.

ASCII data files can be either *structured* or *unstructured*. Structured data files contain the data organized in rows and columns. Unstructured data files contain numbers, but not necessarily in rows and columns. Mathcad has file access functions for reading and writing both structured and unstructured ASCII data files.

Note: Mathcad documents themselves are not valid data files.

Arguments to file access functions

The argument you supply to any Mathcad file access function described in this section is a *string expression*—or a variable to which a string is assigned—that corresponds either to:

- the name of a data file in the working directory of the Mathcad worksheet you're currently working on; or
- a full or relative path to a data file located elsewhere on a local or network file system.

As described in Chapter 8, "Variables and Constants," a string in Mathcad is a special kind of math expression that always appears between double quotes, and you start a string expression in an *empty math placeholder* by typing the double-quote (**"**) key.

If the data file you want to use is located in the working directory of the current Mathcad worksheet, the string expression you supply to a file access function can simply contain the name of the file you are reading from or writing to. For example, the string expression **`"data2.prn"`** (quotes included) refers to the file **`data2.prn`** located in the working directory of the current Mathcad worksheet. Any worksheet containing such string expressions in file access functions can be safely moved to another file system with a different directory structure, provided that you move any required data files along with your worksheet.

Pro If you have Mathcad Professional, you may use the *concat* function (described in Chapter 13, "Built-in Functions,") to concatenate the Mathcad built-in variable CWD, which is a string corresponding to the current directory of the worksheet, with any string argument to a file access function. In this way you will ensure that any file access functions in your worksheet will work properly should you move your worksheet and data files to another file system.

If the file to read from or write to is not in the working directory, the argument to a file access function must be a string expression containing a full or relative path to the data file. For example, you could use the string expression **`"C:\data\trial1.dat"`** to identify the file **`trial1.dat`** located in the **`C:\data`** folder. If you include path information in your string expressions, your file access functions may not work if you move your worksheet to another file system that does not have a similar directory structure.

See the section "Strings" in Chapter 8 for more information about creating and editing strings. For examples of how to use string expressions as arguments to data file access functions, see Figure 19-7 though Figure 19-13.

ASCII data file access functions

Mathcad includes six functions for accessing numerical data stored in ASCII data files: *READ*, *WRITE*, *APPEND*, *READPRN*, *WRITEPRN*, and *APPENDPRN*. They each take as an argument a string expression as described above.

These functions have the following properties:

- You must type the function name all in uppercase. Alternatively, you may insert the function into your document by choosing **Function** from the **Insert** menu and double-clicking on the function name.
- If Mathcad cannot find a data file, it marks the file-access function with an error message indicating that it could not find the file you specified. If Mathcad tries to read a file and the format is incorrect, it marks the function with an error message.
- The *WRITE*, *APPEND*, *WRITEPRN*, and *APPENDPRN* functions must appear alone on the left side of a definition.
- Each new equation reopens the data file. When you read data, for example, each new equation starts reading at the beginning of the file.
- If two equations in the same document use *WRITE* or *WRITEPRN* with the same file, the data from the second equation will overwrite the data from the first. Use *APPEND* or *APPENDPRN* when you don't want to overwrite data. These functions add to an existing file instead of overwriting it.

The table below describes these six functions. In this table:

- **A** represents an array, either vector or matrix.
- v_i represent the individual elements of vector **v**.
- *file* is a string expression containing the name of a file in the working directory or a path to a file outside the working directory. For more information, see section “Arguments to file access functions” on page 454.
- *i* is a range variable.

The functions *READ*, *WRITE* and *APPEND* can be used with range variables. The remaining functions never use range variables.

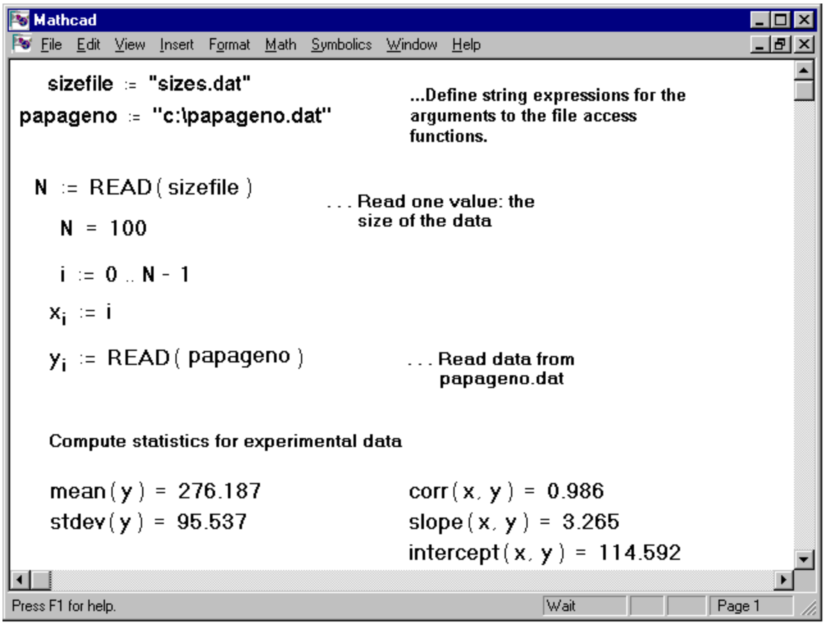
Function	Meaning
READ(*file*)	Read a value from a data file. Returns a scalar. Usually used as follows: v_i := READ(*file*)
WRITE(*file*)	Write a value to a data file. If file already exists, replace it with new file. Must be used in a definition of the following form: WRITE(*file*) := v_i
APPEND(*file*)	Append a value to an existing file. Must be used in a definition of the following form: APPEND(*file*) := v_i
READPRN(*file*)	Read a structured data file. Returns a matrix. Each line in the data file becomes a row in the matrix. The number of elements in each row must be the same. Usually used as follows: **A** := READPRN(*file*)
WRITEPRN(*file*)	Write a matrix into a data file. Each row becomes a line in the file. Must be used in a definition of the form: WRITEPRN(*file*) := **A**

APPENDPRN(*file*)	Append a matrix to an existing file. Each row in the matrix becomes a new line in the data file. Must be used in a definition of the following form: APPENDPRN(*file*) := **A** Existing data must have as many columns as **A**.

Reading and writing data in unstructured files

This section discusses how to use *READ*, *WRITE*, and *APPEND* to read and write data in unstructured files: files containing numbers, but not necessarily in rows and columns.

Figure 19-7 shows two ways to use *READ* to read data from a file.



sizefile := "sizes.dat"

papageno := "c:\papageno.dat"

...Define string expressions for the arguments to the file access functions.

$N := READ(sizefile)$

$N = 100$

. . . Read one value: the size of the data

$i := 0 .. N - 1$

$x_i := i$

$y_i := READ(papageno)$

. . . Read data from papageno.dat

Compute statistics for experimental data

$mean(y) = 276.187$

$stdev(y) = 95.537$

$corr(x, y) = 0.986$

$slope(x, y) = 3.265$

$intercept(x, y) = 114.592$

Figure 19-7: The READ function.

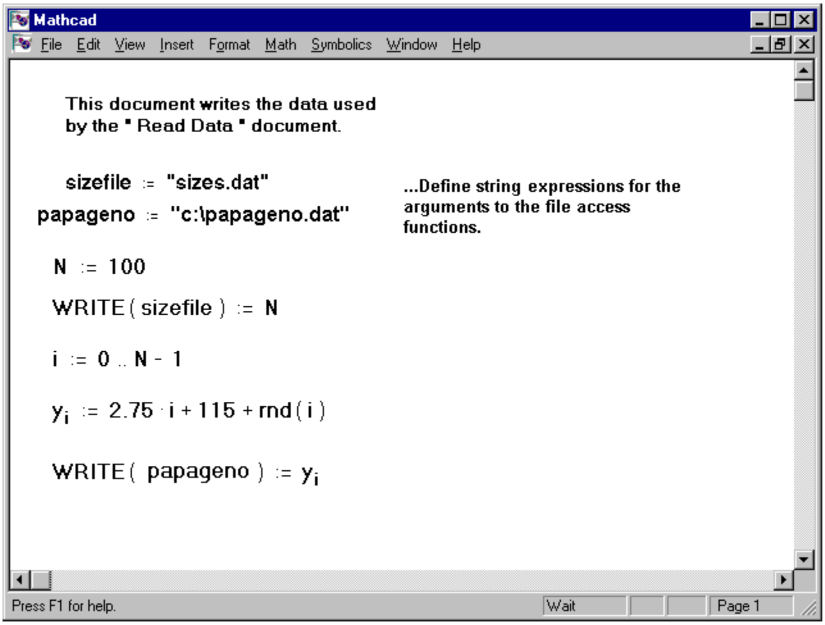
The first *READ* equation defines *N* as the first value in the data file **`sizes.dat`**. The second *READ* equation fills the array **y** with the first 100 numbers in the data file **`papageno.dat`** located in the root directory.

When Mathcad reads data with the *READ* function:

- Each new equation reopens the file and starts reading from the beginning. You cannot read two successive sets of data from the same file by using two separate *READ* equations.

- In an equation with *READ* and a range variable, Mathcad reads one value for each value of the range variable. If the data runs out before the range variable, Mathcad just stops looking for more data. If the range variable ends before the end of the data, Mathcad ignores the extra data in the file.

Figure 19-8 shows how to use the *WRITE* function to write data to a data file.



This document writes the data used by the " Read Data " document.

sizefile := "sizes.dat"

papageno := "c:\papageno.dat"

...Define string expressions for the arguments to the file access functions.

$N := 100$

$WRITE(sizefile) := N$

$i := 0 .. N - 1$

$y_i := 2.75 \cdot i + 115 + rnd(i)$

$WRITE(papageno) := y_i$

Figure 19-8: Writing data with the WRITE function.

The first *WRITE* equation writes a single number to the data file **`sizes.dat`**. The second *WRITE* equation writes *N* numbers to the data file **`papageno.dat`**, one number for each value of the range variable *i*.

When Mathcad writes data to a file, it separates successive numbers with spaces. Mathcad also inserts line breaks to keep the lines shorter than 80 characters. When you use *WRITE*, all values are saved to the file with maximum precision, regardless of the global format of the document.

Mathcad ignores units when it writes data to a data file.

Like the *READ* function, the *WRITE* function reopens the file and starts at the beginning for each new equation. If you want to write data to a file from several different equations, use the *APPEND* function instead of the *WRITE* function in the second and subsequent equations.

Warning: If you use the *WRITE* function on the same file in two separate equations, the data from the second equation will overwrite the data from the first equation.

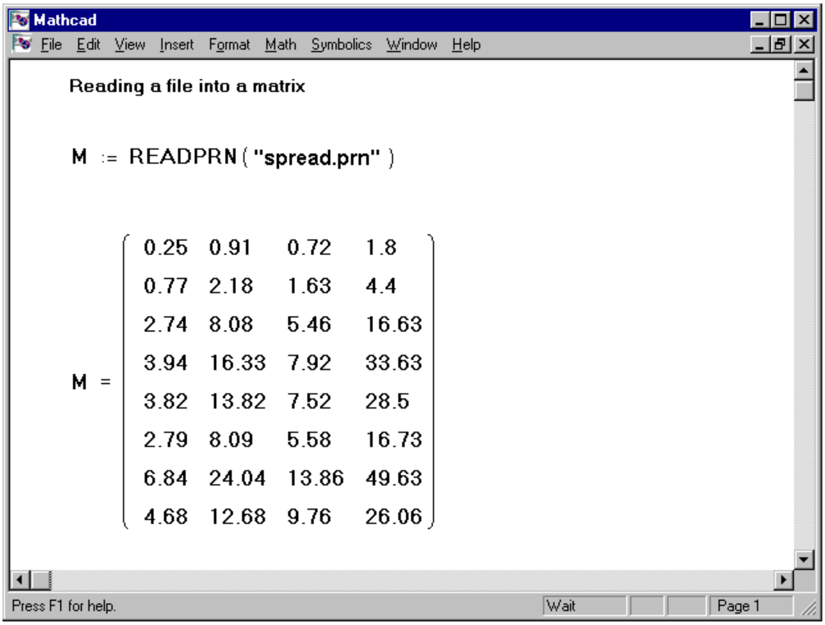
Reading and writing data in structured files

This section discusses how to use *READPRN*, *WRITEPRN*, and *APPENDPRN* to read and write data in structured files. A structured data file is a data file with a fixed number of values per line. For example, if you print a rectangular area from a spreadsheet into a file, the resulting rows and columns of numbers will be a structured file.

Suppose you have an ASCII text file containing the data shown below. These numbers could come from a spreadsheet or from any other source.

```
0.25     0.91    0.72     1.8
0.77     2.18    1.63     4.4
2.74     8.08    5.46    16.63
3.94    16.33    7.92    33.63
3.82    13.82    7.52    28.5
2.79     8.09    5.58    16.73
6.84    24.04   13.86    49.63
4.68    12.68    9.76    26.06
```

Figure 19-9 shows a Mathcad worksheet in which these numbers are read into a matrix.



Reading a file into a matrix

$M := \mathrm{READPRN}(\text{"spread.prn"})$

$$M = \begin{pmatrix} 0.25 & 0.91 & 0.72 & 1.8 \\ 0.77 & 2.18 & 1.63 & 4.4 \\ 2.74 & 8.08 & 5.46 & 16.63 \\ 3.94 & 16.33 & 7.92 & 33.63 \\ 3.82 & 13.82 & 7.52 & 28.5 \\ 2.79 & 8.09 & 5.58 & 16.73 \\ 6.84 & 24.04 & 13.86 & 49.63 \\ 4.68 & 12.68 & 9.76 & 26.06 \end{pmatrix}$$

Figure 19-9: Reading spreadsheet data into a matrix.

The *READPRN* function reads the entire data file, determines the number of rows and columns, and creates a matrix out of the data.

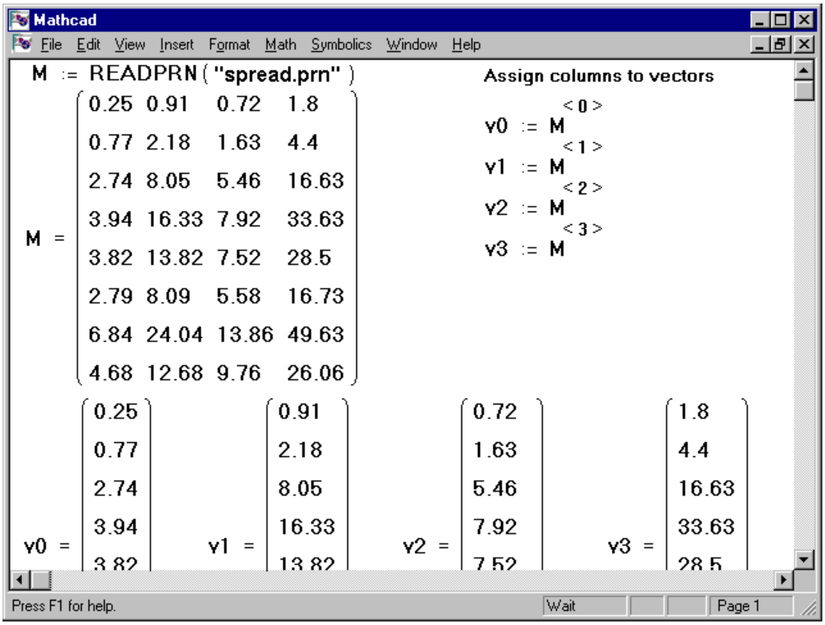
When Mathcad reads data with the *READPRN* function:

- Each instance of the *READPRN* function reads an entire data file.
- All lines in the data file must have the same number of values. (Lines containing no values are ignored.) If the lines in the file have differing numbers of values, Mathcad marks the *READPRN* equation with an error message.
- The *READPRN* function ignores text in the data file.

- The result of reading the data file is an *m*-by-*n* matrix, where *m* is the number of lines containing data in the file and *n* is the number of values per line. To define a matrix out of the numbers in a data file, type an equation like **M** := `READPRN("file.prn")`. *Do not use subscripts on* **M**. *READPRN* returns a matrix, so no subscripts are necessary.

Warning: Each line in the data file must contain the same number of values. If you leave gaps where Mathcad expects numbers, the *READPRN* function will not be able to read the file. Mathcad determines where one number ends and the next begins by looking for spaces or commas.

Sometimes each column of values in a data file represents a different variable. Figure 19-10 shows how to use superscripts to create a vector from each column in the data file.



$M := \mathrm{READPRN}(\text{"spread.prn"})$

Assign columns to vectors

$v0 := M^{\langle 0 \rangle}$

$v1 := M^{\langle 1 \rangle}$

$v2 := M^{\langle 2 \rangle}$

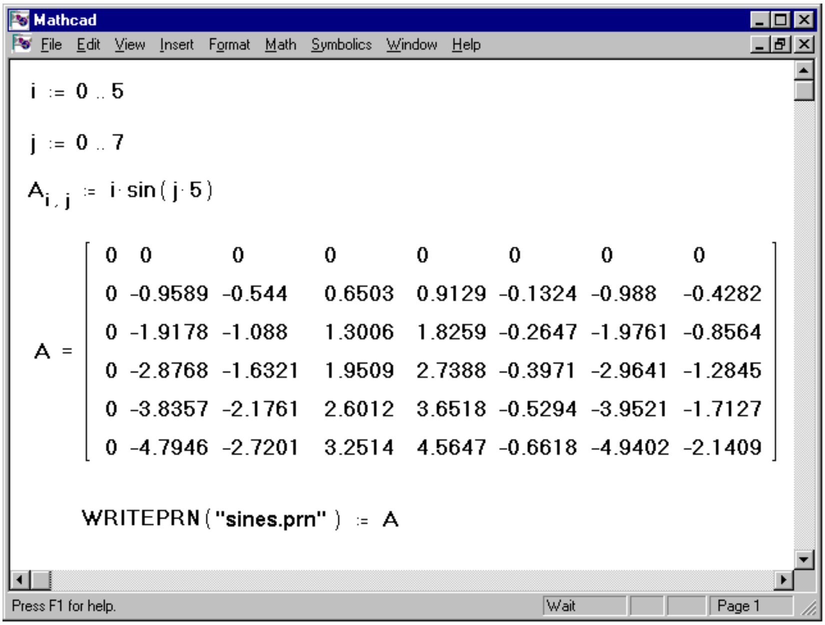
$v3 := M^{\langle 3 \rangle}$

$$M = \begin{pmatrix} 0.25 & 0.91 & 0.72 & 1.8 \\ 0.77 & 2.18 & 1.63 & 4.4 \\ 2.74 & 8.05 & 5.46 & 16.63 \\ 3.94 & 16.33 & 7.92 & 33.63 \\ 3.82 & 13.82 & 7.52 & 28.5 \\ 2.79 & 8.09 & 5.58 & 16.73 \\ 6.84 & 24.04 & 13.86 & 49.63 \\ 4.68 & 12.68 & 9.76 & 26.06 \end{pmatrix}$$

$$v0 = \begin{pmatrix} 0.25 \\ 0.77 \\ 2.74 \\ 3.94 \\ 3.82 \end{pmatrix} \quad v1 = \begin{pmatrix} 0.91 \\ 2.18 \\ 8.05 \\ 16.33 \\ 13.82 \end{pmatrix} \quad v2 = \begin{pmatrix} 0.72 \\ 1.63 \\ 5.46 \\ 7.92 \\ 7.52 \end{pmatrix} \quad v3 = \begin{pmatrix} 1.8 \\ 4.4 \\ 16.63 \\ 33.63 \\ 28.5 \end{pmatrix}$$

Figure 19-10: Assigning a variable to each column of data from a data file.

Figure 19-11 shows how to use the *WRITEPRN* function to write data to a structured data file.



$i := 0 .. 5$

$j := 0 .. 7$

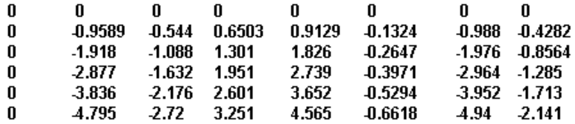
$A_{i,j} := i \cdot \sin(j \cdot 5)$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.9589 & -0.544 & 0.6503 & 0.9129 & -0.1324 & -0.988 & -0.4282 \\ 0 & -1.9178 & -1.088 & 1.3006 & 1.8259 & -0.2647 & -1.9761 & -0.8564 \\ 0 & -2.8768 & -1.6321 & 1.9509 & 2.7388 & -0.3971 & -2.9641 & -1.2845 \\ 0 & -3.8357 & -2.1761 & 2.6012 & 3.6518 & -0.5294 & -3.9521 & -1.7127 \\ 0 & -4.7946 & -2.7201 & 3.2514 & 4.5647 & -0.6618 & -4.9402 & -2.1409 \end{bmatrix}$$

$\text{WRITEPRN}(\text{"sines.prn"}) := A$

Figure 19-11: Writing data to a structured data file.

When you calculate the document in Figure 19-11, Mathcad creates a data file containing the following numbers:



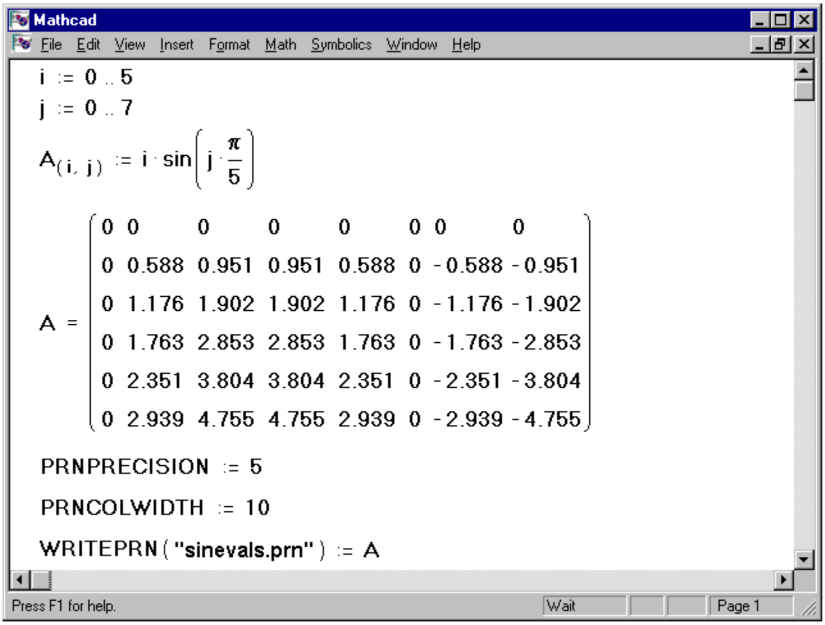
```
0      0       0      0      0      0        0      0
0      -0.9589 -0.544 0.6503 0.9129 -0.1324  -0.988 -0.4282
0      -1.918  -1.088 1.301  1.826  -0.2647  -1.976 -0.8564
0      -2.877  -1.632 1.951  2.739  -0.3971  -2.964 -1.285
0      -3.836  -2.176 2.601  3.652  -0.5294  -3.952 -1.713
0      -4.795  -2.72  3.251  4.565  -0.6618  -4.94  -2.141
```

Unlike the *WRITE* function, the *WRITEPRN* function writes out the data in columns. The precision of the data written out and column widths are controlled by PRNPRECISION and *PRNCOLWIDTH* which you can set by choosing **Options** from the **Math** menu and clicking on the Built-in Variables tab. In the example above, note that since the *PRNPRECISION* is set to four, the numbers are written to four decimal places. Since *PRNCOLWIDTH* is set to eight, each column has space for eight characters. Since *PRNPRECISION* and *PRNCOLWIDTH* can be varied independently, you must take care to choose them in such a way that the column width can accommodate all the digits, together with a space to separate the columns.

When you use *WRITEPRN*:

- Equations using *WRITEPRN* must be in a specified form as follows: On the left should be WRITEPRN(*file*), where *file* is a string expression containing a filename or a pathname. This is followed by a definition symbol (:=) and a matrix expression. *Do not use range variables or subscripts with* WRITEPRN.

- Each new equation writes a new file. If two equations write to the same file, the data written by the second equation will overwrite the data written by the first. Use *APPENDPRN* if you want to append values to a file instead of overwriting the file.
- The built-in variables *PRNCOLWIDTH* and *PRNPRECISION* determine the format of the data file that Mathcad creates. The current value of *PRNCOLWIDTH* specifies the width of the columns (in characters). The current value of *PRNPRECISION* specifies the number of significant digits used. By default, *PRNCOLWIDTH*=8 and *PRNPRECISION*=4. To change these values, choose **Options** from the **Math** menu and edit the numbers on the Built-In Variables tab, or enter definitions in your Mathcad document above the *WRITEPRN* function, as shown in Figure 19-12.
- If the array you are writing is either a *nested* array (an array whose elements are themselves arrays) or a complex array (an array whose elements are complex) then *WRITEPRN* will *not* create a simple ASCII file. Instead, *WRITEPRN* creates a file using a special format unlikely to be readable by other applications. This file can, however, be read by Mathcad's *READPRN* function.



$i := 0 \,..\, 5$

$j := 0 \,..\, 7$

$A_{(i,\,j)} := i \cdot \sin\left(j \cdot \frac{\pi}{5}\right)$

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.588 & 0.951 & 0.951 & 0.588 & 0 & -0.588 & -0.951 \\ 0 & 1.176 & 1.902 & 1.902 & 1.176 & 0 & -1.176 & -1.902 \\ 0 & 1.763 & 2.853 & 2.853 & 1.763 & 0 & -1.763 & -2.853 \\ 0 & 2.351 & 3.804 & 3.804 & 2.351 & 0 & -2.351 & -3.804 \\ 0 & 2.939 & 4.755 & 4.755 & 2.939 & 0 & -2.939 & -4.755 \end{pmatrix}$$

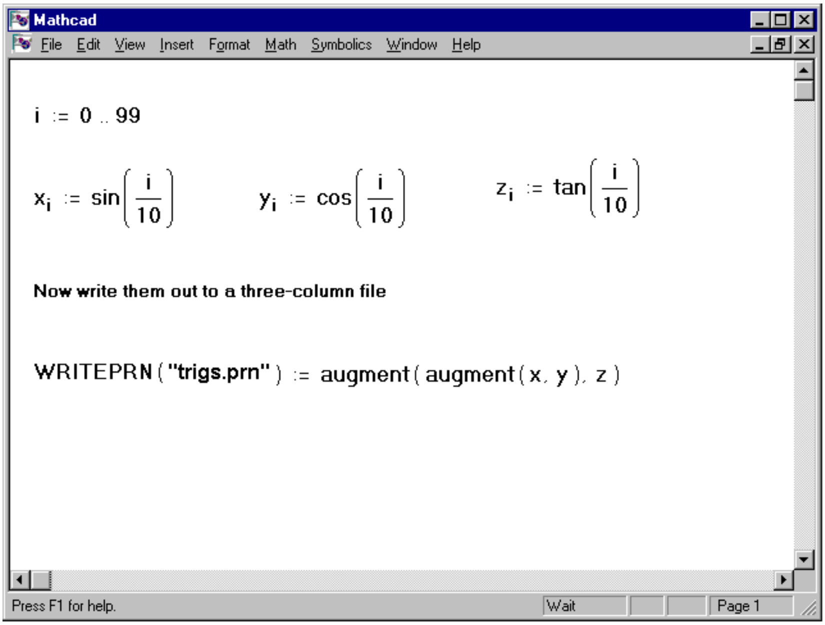
$PRNPRECISION := 5$

$PRNCOLWIDTH := 10$

$WRITEPRN(\text{"sinevals.prn"}) := A$

Figure 19-12: A document that creates a data file with columns 10 characters wide containing numbers with 5 significant digits.

By using the *augment* function, you can concatenate several variables and write them all to a data file. Figure 19-13 demonstrates how to do this.



$i := 0 .. 99$

$x_i := \sin\left(\frac{i}{10}\right) \qquad y_i := \cos\left(\frac{i}{10}\right) \qquad z_i := \tan\left(\frac{i}{10}\right)$

Now write them out to a three-column file

WRITEPRN("trigs.prn") := augment(augment(x, y), z)

Figure 19-13: Writing several concatenated vectors.

Advantages of using READPRN and WRITEPRN

READPRN is generally preferable to *READ*. When the data values are regularly listed out in columns, *READPRN* brings the data into Mathcad in a readily accessible form.

If some lines in a data file have more data values than others, data values may be missing. Use a text editor to replace the missing values with zeros before you use *READPRN*.

READ is required for files in which numbers for a single variable are scattered across several lines in the file. This includes files created by the *WRITE* function, in which there are as many numbers on each line as will fit.

Remember: use a range variable subscript to read with *READ*; do *not* use a subscript to read with *READPRN*.

WRITEPRN generally produces more readable files than *WRITE* because the data values are neatly lined up in rows and columns. However, *WRITE* produces smaller files than *WRITEPRN* because it doesn't have to add spaces to line up the numbers.

Use *WRITE* instead of *WRITEPRN* when you want to crowd as many values as possible into a small data file. *WRITE* creates a data file with only one space between each value and the next.

Pro *WRITEPRN* and *READPRN* allow you to write out and read in nested arrays created in Mathcad Professional. For more information on nested arrays, see the section “Nested arrays” in Chapter 10.