# Using Maple, a General Mathematical Analysis Tool, for NMR Analyses: Introduction

Maple is a powerful collection of routines to aid in the solution of mathematical problems and the analyses of data. Like MATLAB and Mathematica, related mathematical tools used in the sciences, Maple is capable of both symbolic manipulations (solving equations written in terms of variables such as 'x', 'y', and 'z') and numeric calculations. Maple typically defaults to symbolic calculations, but numeric evaluation can easily be specified.

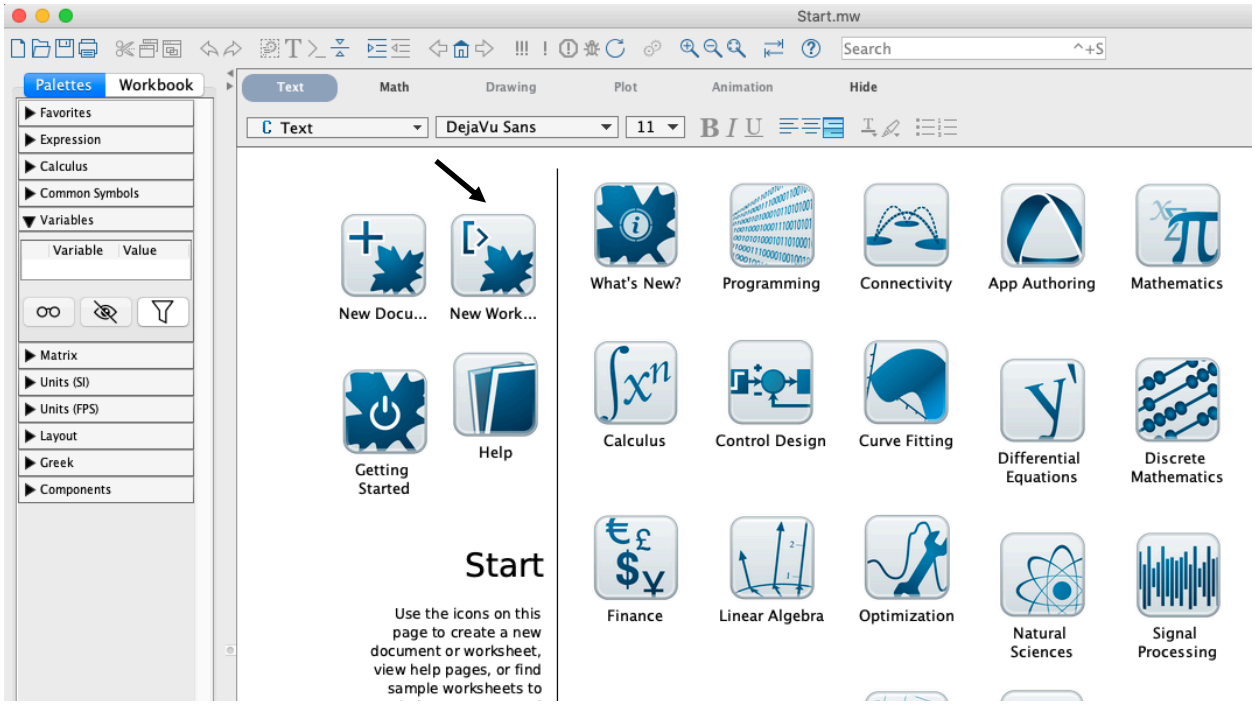
Maple can operate in either the 'Document Mode' or the 'Worksheet Mode'. For these tutorials, we'll use the 'Worksheet Mode', which is the more traditional Maple problem-solving environment and offers some flexibility and capabilities not necessarily available in the 'Document Mode'. In addition, in Maple 2018, the default setting in both 'Document Mode' and 'Worksheet Mode' is '2-D Math' or '2-D Input'. We'll use this setting.

**General Instructions**:
**1**). Install Maple on your computer. In order to proceed, you need to have Maple installed on your computer. On our course website there is a tutorial for doing so.
**2**). Install the Maple 'share' library. You can download the maple 'share' library from our course website in addition to a tutorial for installing it on your computer. Although you don't need the 'share' library to be installed for the tutorial below, you will need it for the next tutorial (product operator calculations). So, you should go ahead and install it now.
**3**). Download the data and worksheets archive from our course website ('maplelab-1-data-worksheets.tar') to the desktop of your computer. Unpack the 'tar' archive ('tar -xf') to give a folder named 'maplelab-1'. Make a directory 'maple' in your home directory (for MacOS, '/Users/username/maple', for Linux, '/home/username/maple'). Copy the 'maplelab-1' folder from your desktop into the 'maple' folder. Confirm the contents as shown below (three Maple worksheets (file extensions '.mw') and one data file ('.txt').

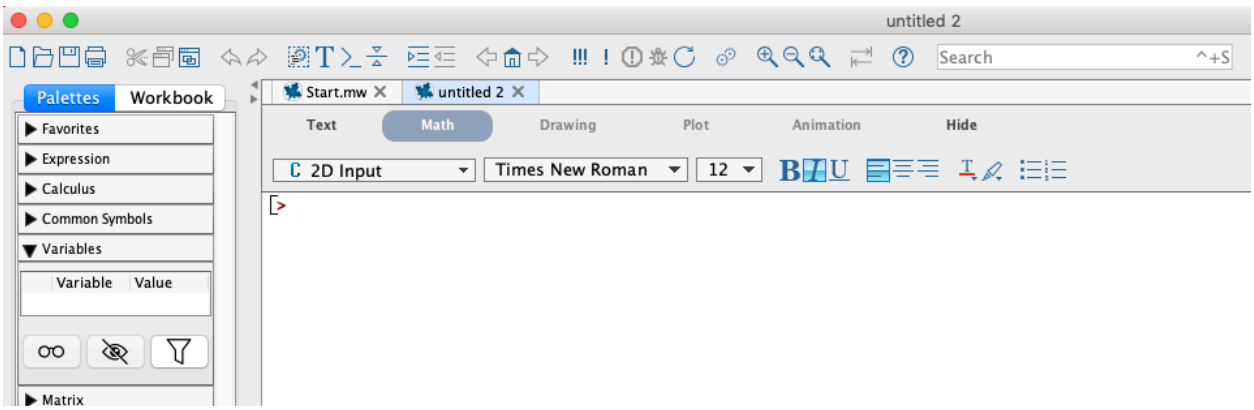
```
MacBook-Pro:~ username$ cd /Users/username/Desktop
MacBook-Pro:Desktop username$ pwd
/Users/username/Desktop
MacBook-Pro:Desktop username $ ls maplelab-1-data-worksheets.tar
maplelab-1-data-worksheets.tar
MacBook-Pro:Desktop username$ tar -xf maplelab-1-data-worksheets.tar
MacBook-Pro:Desktop username$ ls -F maplelab-1
covariance-spectrum.mw*       glucose-anomerization-plot.mw*
glucose-anomerization-data.txt*  intro-maple2018-1.mw*
MacBook-Pro:Desktop username$ mkdir /Users/username/maple
MacBook-Pro:Desktop username$ mv maplelab-1 /Users/username/maple
MacBook-Pro:Desktop username$ cd /Users/username/maple/maplelab-1
MacBook-Pro:maplelab-1 username$ pwd
/Users/username/maple/maplelab-1
MacBook-Pro:maplelab-1 username$ ls -F
covariance-spectrum.mw*       glucose-anomerization-plot.mw*
glucose-anomerization-data.txt*  intro-maple2018-1.mw*
```

## **Starting Maple**:

For **MacOS**, double-click on the 'Maple 2018.app' (assuming you are using Maple 2018) in the folder/directory 'Maple 2018' in your 'Applications' folder. For convenience, you should probably put the icon on your Dock. For **Linux**, if you have a shortcut on your Desktop, double-click the shortcut icon. Alternatively, you may have a shortcut in your 'Applications' menu (for the 'CentOS installation, a shortcut was automatically placed in the 'Applications' menu under 'Education'. Select it to start the program. If you do not have a shortcut, enter the path to the 'xmaple' application (for the CentOS installation, the path to the 'xmaple' executable is '/opt/local/maple2018/bin/xmaple') to start the program. Notice the 'x' ('xmaple'). This starts the graphical version of the Maple interface, which is what you want. Once Maple starts, select the 'New Worksheet' option to open the program in the 'Worksheet Mode'.

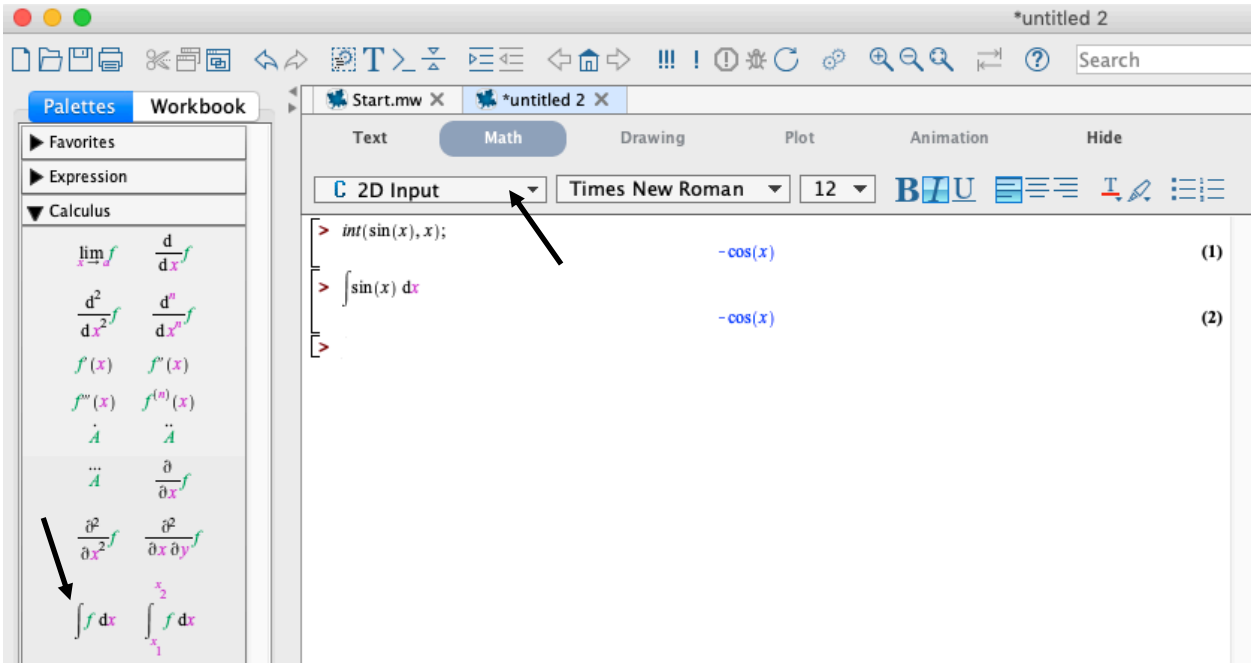



The new worksheet will open. Notice the red command prompt (>). This indicates you are in the 'Worksheet Mode'. Commands are entered at the prompt.



## Using '2-D Input' ("2-D Math"):

The default Maple input style is '2D Input'. There are a number of other options (that can be selected from the drop-down menu), and we'll not attempt to describe them here, but the '2D Input' mode is a very versatile mode for mathematics. Like everything else in Maple, the '2D Input' mode allows for multiple ways to accomplish the same task. For instance, in the examples below, Maple is asked to solve an integral. In the first example, the command to integrate, 'int' is entered at the first red command prompt (>). Enclosed in the parentheses are the two variables, the integrand, 'sin(x)', and the variable of integration, 'x'. A semicolon follows the command, indicating that the output will be shown (a colon suppresses output). In the second example, an expression is selected from the 'Calculus' contextual menu on the left (with a click), and the common, familiar notation for an integral is entered automatically after the command prompt. The notation is edited manually to make 'sin(x)' the integrand. The output of the two commands is the same ('-cos(x)').




In our tutorial, we'll mostly use the first type of notation, which is the older, classical Maple notation. This notation is a bit more versatile, but has the disadvantage that the commands to accomplish various mathematical tasks must be remembered. For our tutorials, all of the commands will be entered in the worksheets, so all you have to do is execute the commands sequentially (enter <return> at the end of each line after the semicolon or colon). This is convenient for tutorials as mistakes in entering commands can make the tutorials laborious.

Also, we'll use the convention that a semicolon or colon must follow each command. As you work through the tutorials, feel free to experiment with commands. If your worksheet ceases to work properly, just reload the original and begin again.

## **Help**:

The Maple 'Help' menu (top bar in MacOS, top of Maple window in Linux) includes the entry 'Maple Help'. Selecting this option opens a comprehensive help utility. For simple help with commands, typing '?command_name' at the red command prompt will give help on that particular command. Usually, a new window will open with an exhaustive analysis of the command and its usage. Try typing '?plot3d' at the command prompt, for instance, so you can get an idea of the usefulness of the help utilities.

## **Basics**:

According to the book, "Maple V by Example" (Marth Abell and James Braselton, Academic Press Professional, 1994), there are five basic rules to Maple V syntax.

**1**) The arguments of functions are given in parentheses (…..).
**2**) A semicolon (;) or colon (:) must be included at the end of each command (note, the colon suppresses output, and the semicolon now is optional, although it is a good idea to include it).
**3**) Multiplication is represented by an asterisk (*). However, in Maple 2018 (and perhaps some previous versions) even though you type an asterisk, it unfortunately shows up as a dot (•).
**4**) Powers are denoted by ^ (you indicate an exponent by typing the caret sign, shift-6). However, in Maple 2018 (and perhaps some previous versions), the display that you see is an exponent (old display: 3^5, new display: $3^5$).
**5**) If you get no response or an obviously incorrect response, most likely you entered the command incorrectly.

Because of #5, and to make the tutorials go smoothly, you will only need to execute commands, not enter them, to run the tutorials (see below).

## **Begin this tutorial**:

Start Maple. From the 'File' menu, choose 'Open' and **open the file 'intro-maple2018-1.mw'**. (for MacOS, /Users/username/maple/maplelab-1/intro-maple2018-1.mw', for Linux, '/home/username/maple/ maplelab-1/intro-maple2018-1.mw'). This file has a list of Maple commands, but no output. So, you don't have to type in all of the commands. This is so that you can see how Maple works without worrying about incorrectly typing in the commands.

**All you have to do is put the cursor after the semicolon on each line and hit <return>. This will execute the command and give you the output shown below (in blue). Start at the top and execute each command in sequence from top to bottom.**

Here we'll demonstrate some Maple math basics. The command 'restart' clears the Maple memory. The next command specifies multiplication of 2 by 2 (2 x 2). In Maple, you type an asterisk (*) to indicate multiplication, but the program inserts a dot (•) instead. You see the output is '4', as expected. In this case Maple evaluated the expression. Likewise, the second operation adds 3 and 7 to give 10. When Maple is given 8/3, it does not evaluate the expression (we'll force evaluation later), which is the way Maple behaves sometimes. The next operation demonstrates that several commands can be placed on a single line. You only have to hit <return> after the last semicolon (not after each one) for all commands to be executed. Next, Maple is forced to evaluate 8/3 using 'evalf' ('evaluate function'). The next command multiples 6 by 6 (6 x 6), but the output is suppressed (colon used instead of semicolon). The output is still there, you just can't see it. The following command multiplies 2 by the output of the previous command (the % sign indicates the output of the previous command). So, 2 is multipled by 26 to give 72. Use 'restart' to clear the memory before proceeding.

```
> restart;
> 2·2;
```

$$4 \quad (1)$$

```
> 3 + 7;
```

$$10 \quad (2)$$

```
> 8/3;
```

$$\frac{8}{3} \quad (3)$$

```
> 2·2; 3 + 7; 8/3;
```

$$4$$
$$10$$
$$\frac{8}{3} \quad (4)$$

```
> evalf(8/3);
```

$$2.666666667 \quad (5)$$

```
> 6·6 :
> 2·(%);
```

$$72 \quad (6)$$

```
> restart;
>
```

Here we'll demonstrate how to define functions and plot them. After a restart, a variable '*f* ' is defined (:= is the general way to define a variable), which is a function of *x* ('*f*(*x*)'). The function '*f*(*x*)' is equal to $x^3$+20 $x^2$+3 (to type the arrow, type dash (-) followed by the right caret (shift-period)). A second variable, '*g*', is defined to be a function of x and y ('*g*(*x*,*y*)'). The function '*f*(*x*)' is plotted for values of x from -30 to 10, and '*g*(*x*,*y*)' is plotted (3D plot) for values of '*x*' from -π to π (Pi is π), and for values of '*y*' from - π to π.

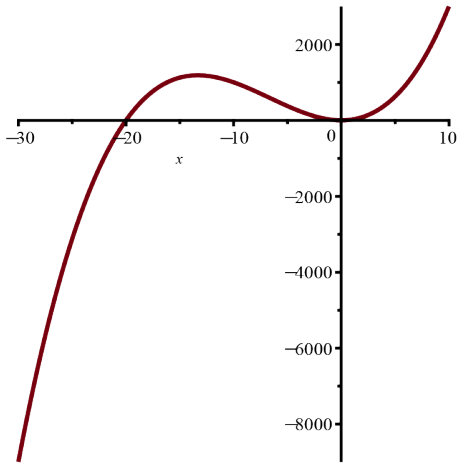
```
> restart;
> f := x->x^3 + 20*x^2 + 3;
```

$$f := x \mapsto x^3 + 20\,x^2 + 3 \qquad (1)$$

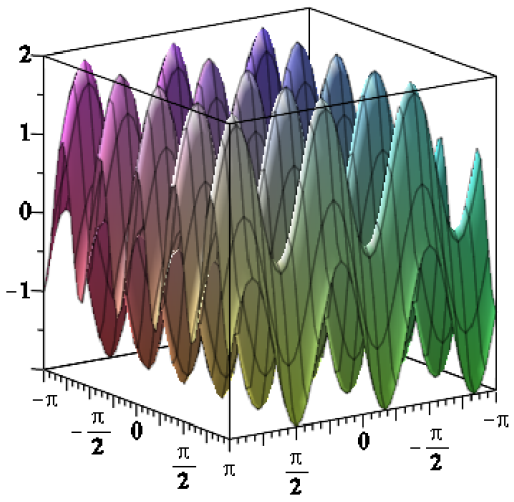
```
> g := (x,y)->sin(3*x) + cos(5*y);
```

$$g := (x, y) \mapsto \sin(3\,x) + \cos(5\,y) \qquad (2)$$

```
> plot(f(x), x=-30..10);
```




```
> plot3d(g(x,y), x=-Pi..Pi, y=-Pi..Pi);
```




```
> restart;
>
```

Here are some examples of evaluating, solving, and simplifiying functions. First, a variable '*x*' is defined, but the output suppressed. It is then evaluated with 'evalf'. Next, a variable '*f*' is defined as a function of both '*x*' and '*y*' ('*f*(*x*,*y*)'). It is then factored with the 'factor' command, then expanded again with 'expand' to give the original function (here '%' is used to indicate the output of the previous command). Next, a variable '*h*' is defined as a function of '*x*' ('*h*(*x*)') and is simplified with the 'simplify' command (Maple will try to simplify functions if it can, which is easy for standard trigonometric identities). In the last example, a quadratic equation '*f*(*x*)' is defined and solved for values of '*f*(*x*)' =0, 5, and -2 (note: 'I' indicates an imaginary number).

```
> restart;
> x := Pi^2 - Pi :
> evalf(x);
```

$$6.728011750 \tag{1}$$

```
> restart;
> f := (x, y) -> x^2 + 2*x*y + y^2;
```

$$f := (x, y) \mapsto x^2 + 2\,y\,x + y^2 \tag{2}$$

```
> factor(f(x, y));
```

$$(x + y)^2 \tag{3}$$

```
> expand(%);
```

$$x^2 + 2\,y\,x + y^2 \tag{4}$$

```
> h := x -> sin(x)^2 + cos(x)^2;
```

$$h := x \mapsto \sin(x)^2 + \cos(x)^2 \tag{5}$$

```
> simplify(h(x));
```

$$1 \tag{6}$$

```
> restart;
> f := x -> x^2 - 2*x + 1;
```

$$f := x \mapsto x^2 - 2\,x + 1 \tag{7}$$

```
> solve(f(x) = 0);
```

$$1, 1 \tag{8}$$

```
> solve(f(x) = 5);
```

$$\sqrt{5} + 1, -\sqrt{5} + 1 \tag{9}$$

```
> solve(f(x) = -2);
```

$$1 + I\sqrt{2}, 1 - I\sqrt{2} \tag{10}$$

```
> restart;
>
```

Above, Maple solved a quadratic equation for particular values of the function. Maple can also solve equations symbolically. The first example below is the standard quadratic equation. Using 'solve', Maple solves to give the two possible solutions in symbolic form. The other examples below are using Maple for calculus. The first example is an integral ('int' for integration), where $x^2$ is the integrand and '*x*' is the variable of integration (*dx*). The second is another integral where ($ax^2$+cos(*x*)-1/*x*) is the integrand and '*x*' is again the variable of integration. These are equivalent to:

$$\int x^2 \, dx \qquad \int \left(ax^2 + \cos(x) - \frac{1}{x}\right) dx$$

The last example is for differentiation. Here Maple is differentiating ('diff') the output from the previous command with respect to '*x*':

$$\frac{d\left(\frac{ax^3}{3} - \ln(x) + \sin(x)\right)}{dx}$$

So, as expected, the integrand from the previous integral is returned.

```
> restart;
> f := x→a·x^2 + b·x + c;
```

$$f := x \mapsto a\,x^2 + b\,x + c \tag{1}$$

```
> solve(f(x) = 0, x);
```

$$\frac{-b + \sqrt{-4\,a\,c + b^2}}{2\,a}, \quad -\frac{b + \sqrt{-4\,a\,c + b^2}}{2\,a} \tag{2}$$

```
> restart;
> int(x^2, x);
```

$$\frac{x^3}{3} \tag{3}$$

```
> int(a·x^2 + cos(x) - 1/x, x);
```

$$\frac{a\,x^3}{3} - \ln(x) + \sin(x) \tag{4}$$

```
> diff(%, x);
```

$$a\,x^2 + \cos(x) - \frac{1}{x} \tag{5}$$

```
> restart;
```

Maple is especially useful for matrix operations. First, Maple is very flexible in how matrices are created/defined. There are many different ways to accomplish this in Maple. Below we'll show just a few examples of how this can be done, but there are many more.

In the first example below, matrix 'A' is defined with 2 rows and 3 columns as shown, whereas matrix 'B' is defined with 3 rows and 2 columns. The first operation evaluates the transpose of matrix B, so the rows and columns are exchanged, as shown. Note, instead of 'eval' or 'evalf' (for functions), here the command is 'evalm', the 'm' denoting 'matrix'. The next example adds matrix A and the transpose of matrix B. For matrices, the ampersand sign (&) must be used for matrix additions, multiplications, etcetera. So, '&+' indicates matrix addition, '&*' (which sometimes will be displayed as '&•') indicates matrix multiplication, etcetera.

The last operation attempts to add matrices 'A' and 'B'. Because matrix 'A' does not have the same numbers of rows and columns as matrix 'B', they cannot be added by this simple operation. So, the pink error message indicates this (incompatible matrix dimensions).

```
> restart;
> A := matrix(2, 3, [1, 2, 3, 4, 5, 6]); B := matrix(3, 2, [a, b, c, d, e, f]);
```

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$B := \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \quad (1)$$

```
> evalm(transpose(B));
```

$$\begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix} \quad (2)$$

```
> evalm(A&+transpose(B));
```

$$\begin{bmatrix} 1+a & 2+c & 3+e \\ 4+b & 5+d & 6+f \end{bmatrix} \quad (3)$$

```
> evalm(A&+B);
Error, (in linalg:-matadd) matrix dimensions incompatible
> restart;
```

Most of the operations for matrices are in a Maple library called 'LinearAlgebra'. In order to be able to perform most matrix operations, this library has to be invoked, i.e. loaded, for Maple to access the available routines. This is done using the 'with' command, as shown below. The output is suppressed by the colon (:). You can substitute a semicolon if you wish to see the long list of matrix operations supported by the 'LinearAlgebra' library. Below, matrix 'A' is defined first, then its inverse is calculated and named matrix 'B'. These two matrices are then multiplied, but not yet evaluated, so, in this case, Maple just returns the statement that these two matrices are multiplied. In the last operation ('evalm'), the matrices are actually multipled, and, as expected when a matrix is multiplied by its inverse, the unity matrix is returned.

```
> restart;
> with(LinearAlgebra) :
> A := Matrix(3, 3, [[1, 2, 9], [1, 1, 1], [0, 1, 1]]);
```

$$A := \begin{bmatrix} 1 & 2 & 9 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{1}$$

```
> B := MatrixInverse(A);
```

$$B := \begin{bmatrix} 0 & 1 & -1 \\ -\frac{1}{7} & \frac{1}{7} & \frac{8}{7} \\ \frac{1}{7} & -\frac{1}{7} & -\frac{1}{7} \end{bmatrix} \tag{2}$$

```
> A&*B;
```

$$\begin{bmatrix} 1 & 2 & 9 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \&* \begin{bmatrix} 0 & 1 & -1 \\ -\frac{1}{7} & \frac{1}{7} & \frac{8}{7} \\ \frac{1}{7} & -\frac{1}{7} & -\frac{1}{7} \end{bmatrix} \tag{3}$$

```
> evalm(A&*B);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

```
> restart;
>
```

Here are shown some vector operations. Again, it is best to invoke the 'LinearAlgebra' library because many of the matrix and vector operations require functions/routines in that library. Here, again, we've suppressed the output of the 'with(LinearAlgebra)' command with a colon, but feel free to replace it with a semicolon to see the output if you like. As with matrices, there are innumerable ways to define vectors, below we've just shown one way. A column vector 'A' and a row vector 'B' are defined. The first operation gives the dot product of 'B' and 'A' (&*). This is evaluated with 'evalm' to give a scalar (a+2b+3c). The next operation gives the dot product of matrix 'C' and 'A' to give the corresponding scalar (14). The final operation gives the so-called outer product of the two vectors, which is a matrix.

```
> restart;
> with(LinearAlgebra) :
> A := <1, 2, 3>;
```

$$A := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \tag{1}$$

```
> B := <a|b|c>;
```

$$B := \begin{bmatrix} a & b & c \end{bmatrix} \tag{2}$$

```
> (B&*A);
```

$$\begin{bmatrix} a & b & c \end{bmatrix} \&* \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \tag{3}$$

```
> evalm(B&*A);
```

$$a + 2\,b + 3\,c \tag{4}$$

```
> C := <1|2|3>;
```

$$C := \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \tag{5}$$

```
> evalm(C&*A);
```

$$14 \tag{6}$$

```
> OuterProductMatrix(A, C);
```

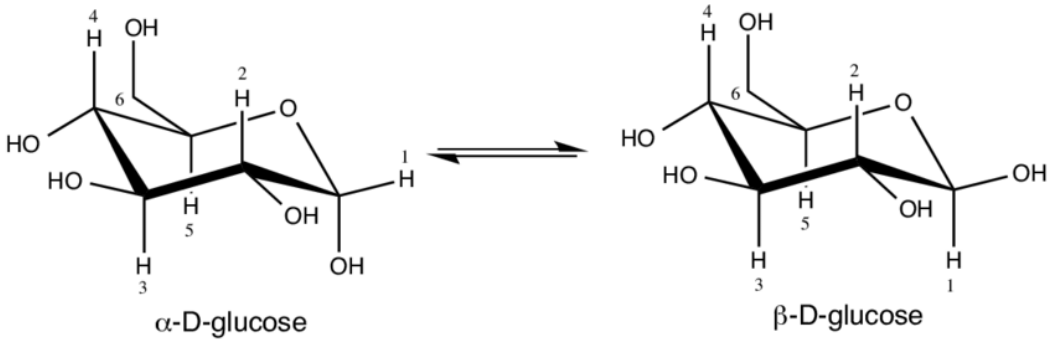
$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix} \tag{7}$$

```
> restart;
>
```

## **Reading in (NMR) data and plotting**:

Maple is capable of reading data from external files (so you don't have to type it all in), and, likewise, it is capable of plotting and analyzing data. Periodically, software packages meant for handling specific types of data (NMR data, for instance), may be unable to perform specific operations on the data, or may be unable to otherwise manipulate or analyze the data in ways that are not standard in some sense. For instance, in NMR spectroscopy, periodically new experiments or procedures are introduced by researchers that produce data that standard NMR processing and analysis software packages are simply unable to handle. Software packages like Maple, Mathematica, and Matlab, therefore, are attractive for manipulating and otherwise mathematically analyzing and displaying/plotting such data.

To illustrate reading data from external files with Maple and plotting the data, we'll consider the results of an NMR experiment to monitor the anomerization of D-glucose. D-glucose exists in two pyranose ring structures (chair conformations) referred to as the $\alpha$ and $\beta$ anomers, which are shown below. Typically, when glucose is crystallized, it does so as the $\alpha$ anomer. When placed in aqueous solution, there is a net conversion of $\alpha$ to $\beta$ anomer as a function of time until equilibrium is reached.




In pyranose ring form, the anomeric hydrogen (the hydrogen labeled '1' in the figure) is coupled only to the single neighboring hydrogen (labeled '2' in the figure). In the $\alpha$ anomer, the dihedral angle between H1 and H2 is about 60 degrees, so the coupling constant is relatively small (Karplus relationship). In the $\beta$ anomer the dihedral angle is about 180 degrees, so the coupling constant is much larger. The signal from the anomeric hydrogen (H1) of the $\alpha$ anomer is a doublet with a small splitting that is downfield from the signal of the $\beta$ anomer, which also is a doublet but with a larger splitting.

The data we will be reading into Maple and plotting will be from an NMR experiment where crystalline $\alpha$-D-glucose is dissolved in $D_2O$ and 1D NMR spectra are collected as a function of time to monitor the equilibration of $\alpha$-D-glucose and $\beta$-D-glucose. The raw data have been Fourier transformed, so the data we will be reading are amplitudes in the frequency domain (the signals). Only the small region of the spectra where the anomeric hydrogen signals appear is shown. There are 40 spectra, each with 80 points defining the amplitudes.

**Start Maple**. From the 'File' menu, choose 'Open' and **open the file 'glucose-anomerization-plot.mw'** (for MacOS, /Users/username/maple/maplelab-1/glucose-anomerization-plot.mw', for Linux, '/home/username/maple/ maplelab-1/glucose-anomerization-plot.mw'). This file has a list of Maple commands, but no output. So, you don't have to type in all of the commands, you just have to do is hit <return> after the semicolon at the end of each line (you don't have to type any commands). HOWEVER, you will have to tell Maple where the data (that you will be reading in) is. This is done with the 'currentdir' command (see below). You should have a directory/folder 'maple' in your home directory, and a subdirectory called 'maplelab-1'. The data file should be in that directory ('glucose-anomerization-data.txt'). So, in the worksheet, **edit the 'currentdir' as shown below** (shown is for a typical Unix/Linux system, for MacOS it would be something like '/Users/username/maple/maplelab-1').

After 'restart', the current directory (the folder that contains the data file) is defined with the 'currentdir' command. The next command, 'readdata' reads in the data from the file 'glucose-anomerization-data.txt' into a matrix named 'A'. The argument '80' tells Maple that there are 80 data points. Maple will automatically figure out that there are 40 lines and it will read in all 40 of them (so, a 40 x 80 matrix). The output is suppressed with the colon (:), but, if you like, you can use a semicolon instead to see all of the data.

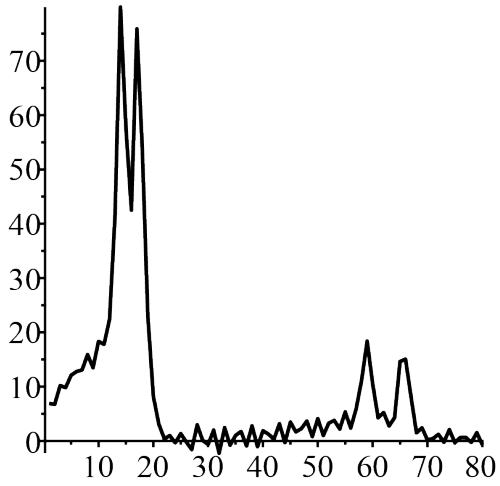
```
> restart;
> currentdir("/home/username/maple/maplelab-1");
                    "/home/username/maple/maplelab-1"                    (1)
> A := readdata("glucose-anomerization-data.txt", 80) :
```

The following command (*with*(*plots*):) loads some routines/procedures that Maple uses to plot data. Again, the output is suppressed with a colon, but, if you like, feel free to substitute a semicolon to see the output. The following command simply displays the first row ('1'), which consists of all of the values (amplitudes) for each of the 80 points ('1..80') in that row. This corresponds to the first spectrum that was acquired in the experiment.

```
> with(plots) :
> A[1, 1..80];
[6.88674, 6.71932, 10.1826, 9.81554, 12.0841, 12.7721, 13.0638, 15.8922, 13.4764, 18.3412,    (2)
  17.7667, 22.3273, 41.2496, 79.8992, 57.6429, 42.5073, 75.8843, 53.2645, 22.8668,
  8.21004, 3.08898, 0.406383, 0.965, -0.35866, 1.33142, -0.187674, -1.59647, 2.94416,
  0.076275, -0.663215, 1.97814, -2.26676, 2.41783, -0.786294, 0.999904, 1.72891,
  -0.919743, 2.75346, -1.05469, 1.86125, 1.23796, 0.328664, 3.14554, -0.323768,
  3.452, 1.65393, 2.25359, 3.64313, 0.798591, 4.09284, 0.998589, 3.26487, 3.76633,
  2.18372, 5.32162, 2.36547, 5.91841, 11.2497, 18.3552, 10.5576, 4.25486, 5.23857,
  2.75403, 4.26366, 14.6109, 15.0597, 8.13963, 1.48505, 2.4092, 0.165846, 0.418552,
  1.21156, -0.242511, 2.06176, -0.316915, 0.629331, 0.640378, -0.281724, 1.50842,
  -0.683636]
```

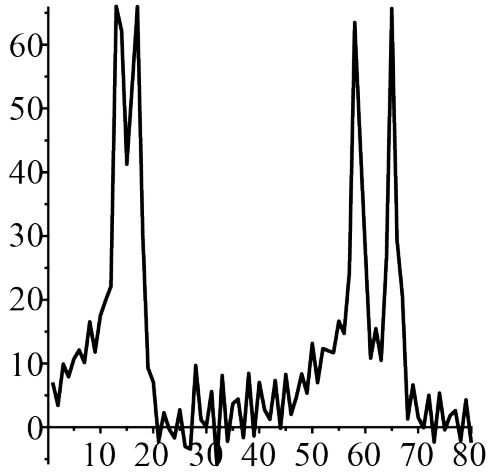
Finally, the final command, 'listplot', plots all 80 points in the first row of 'A' (the first spectrum). These are the points whose amplitudes are listed with the previous command. This is the first spectrum. The doublet at the left (downfield) corresponds to the $\alpha$ anomer of D-glucose, and the upfield doublet (right) is the $\beta$ anomer. The signal of the $\alpha$ anomer is much larger than the signal for the $\beta$ anomer, as would be expected. Note the larger coupling constant for the $\beta$ anomer, as predicted.

```
> listplot(A[1, 1..80]);
```




The 'listplot' command above plotted the first spectrum. You can plot any of the 40 spectra with similar listplot commands. For instance, the 'listplot' command below plots the 32nd row of 'A' (the 32nd spectrum). As you can see, the anomerization process results in a net conversion of the $\alpha$ anomer to the $\beta$ anomer, such that at this time point the relative concentrations of the two anomers are approximately equal, as judged by the relative sizes of the two signals.

```
> listplot(A[32, 1..80]);
```

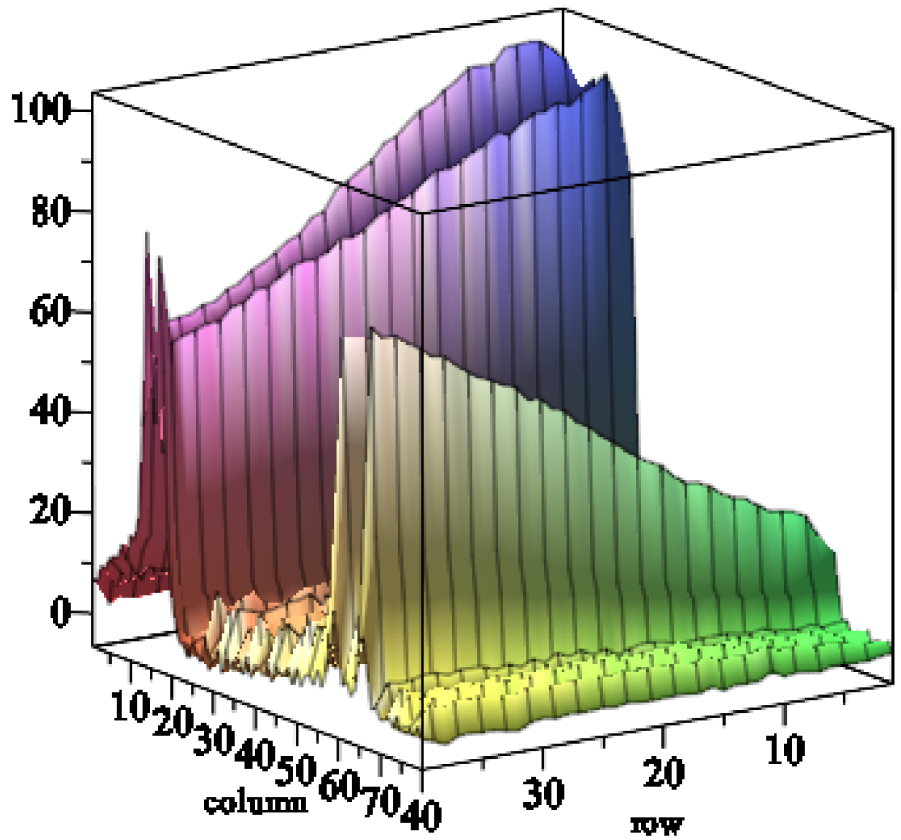


Finally, Maple can plot all of the spectra simultaneously. The 'Matrix' command below displays some information about our data matrix (40 rows by 80 columns, i.e. 80 data points per row or per spectrum), and some other information. If you use the same command only with a lower case m on matrix (i.e. 'B:=matrix(A);'), then the entire matrix of data will be the output (give it a try if you like). The last command, 'matrixplot', plots all of the spectra as a pseudo-3D plot. In Maple, you can put your cursor on this plot and hold down the left mouse button and you can reorient this plot in any way you like.

```
> B := Matrix(A);
```

$$B := \begin{bmatrix} 40 \text{ x } 80 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \quad (3)$$

```
> matrixplot(A);
```




```
> restart;
>
```

## **Calculating and plotting correlation spectra (covariance matrices)**:

A useful procedure that has come on the NMR scene relatively recently is the use of 2D correlation spectra based on covariance matrices. This is well-exemplified by the so-called "statistical TOCSY" method (see references below). These rely on a statistical procedure for establishing whether signals in a spectrum are correlated in their evolution under some parameter variation, for instance, how they vary with time. As an example, we can consider a simple chemical reaction where a single reactant is converted to a single product. If NMR spectra of the reactant and product (the reaction mixture) are acquired as a function of time, there will be two types of correlated signals that will be observed. Two Signals from the same compound (either the reactant or product) will obviously vary in the same way from spectrum to spectrum, and thus are said to be positively correlated. Two signals, one from the reactant and one from the product, will vary in opposite ways from spectrum to spectrum (one will decrease as the other increases), and are said to be negatively correlated. In a more complex mixture with multiple reactants and products, two signals from different compounds that vary randomly with respect to one another will not be correlated (zero correlation).

A plot of a covariance matrix looks very much like some typical 2D NMR spectra. For instance, in a normal 2D TOCSY spectrum, signals are correlated in two dimensions (i.e. crosspeaks are observed) if they are in the same spin system and are scalar coupled. In a statistical TOCSY spectrum, cross-peaks connect all signals from nuclei in a single compound because these are positively correlated (has nothing to do with scalar coupling), and other cross peaks will connect signals from nuclei that are negatively correlated (perhaps with opposite phase of positively correlated signals). Interestingly, these statistical methods also have the beneficial property of transferring the resolution seen in the direct dimension to the indirect dimension. The references below describe some of these methods.

Cloarec et al. (2005) "Statistical total correlation spectroscopy: An exploratory approach for latent biomarker identification from metabolic H-1 NMR data sets". *Analytical Chemistry* **77:** 1282-1289.

Zhang, F.L., and Bruschweiler, R. (2007) "Robust deconvolution of complex mixtures by covariance TOCSY spectroscopy". *Angewandte Chemie-International Edition* **46:** 2639-2642.

In the following exercise, using Maple we'll create a 2D correlation spectrum from the 1D spectra collected for the anomerization of D-glucose. So, as a function of time, we should observe some signals that are positively correlated (signals from nuclei in the $\alpha$ anomer should be positively correlated with one another, and signals from nuclei in the $\beta$ anomer should be positively correlated with one another). Likewise, we should observe signals that are negatively correlated (as the anomerization proceeds, the $\alpha$ anomer is converted into the $\beta$ anomer, so signals from these should be negatively correlated). We should be able to observe these positive and negative correlations as signals of opposite sign in the 2D spectrum.

The correlation-based NMR methods require construction of a correlation matrix, or variance/covariance matrix, from the spectra. A variance/covariance matrix is a type or particular case of a cross-product matrix. A cross-product matrix is constructed by multiplication of a matrix by its transpose (for matrix 'A', '$A^TA$'), and thus is a square, symmetric matrix. To construct a variance/covariance matrix, the data first have to be "centered" (subtract the mean of each column from each element of the column) before transposing and taking the cross product. Technically, after the cross product, the data should be scaled (divided) by the number of rows of the original data matrix, but this isn't necessary in our case.

**Start Maple**. From the 'File' menu, choose 'Open' and **open the file 'covariance-spectrum.mw'** that is in your 'maplelab-1' directory. After a restart to clear the memory, both the 'linear algebra' and 'plots' libraries must be loaded using the 'with' command. Then, read in the same dataset that you did for the previous exercise ('glucose-anomerization-data.txt') in the same way as you did before. The directory where the data are stored must be specified with the 'currentdir' command. We'll read our data into a matrix we'll call 'A' using the 'readdata' command as we did previously. Our data is a 40 x 80 matrix (40 spectra, each one row of the matrix, with 80 data points/amplitudes per row, so 80 columns). The 'for j from 1 to 80' and 'for I from 1 to 40' loops sum the data in each of the 40 columns and subtract the average from each element in the column ("centering"). The last command defines our centered matrix, and the output gives some information about it.

```
> restart;
> with(LinearAlgebra) :
> with(plots) :
> currentdir "/home/username/maple/maplelab-1";
                          "/home/username/maple/maplelab-1"        (1)
> A := readdata("glucose-anomerization-data.txt", 80) :
> for j from 1 to 80 do
      s := 0 :
      for i from 1 to 40 do
          s := (A[i,j] + s)
      end do:
      for i from 1 to 40 do
          A[i,j] := A[i,j]-s/40
      end do:
  end do:
> A := Matrix(A);
```

$$A := \begin{bmatrix} 40\ x\ 80\ Matrix \\ Data\ Type:\ anything \\ Storage:\ rectangular \\ Order:\ Fortran\_order \end{bmatrix} \qquad (2)$$

The first of the next two commands calculates the transpose of the matrix, and names it 'B', and the second multiplies the transpose 'B' by 'A' ('$A^TA$') to give the square (80x80) and symmetric matrix C. The final command plots this matrix, which is our 2D 'spectrum'. The positive diagonal peaks/signals are our original spectrum. The positive off-diagonal peaks near the diagonal are the positively correlated peaks (those for nuclei in the same molecule), and the negative cross-peaks indicate negatively correlated nuclei, those in different molecules (representing the conversion of the $\alpha$ anomer to the $\beta$ anomer). In Maple you can rotate the 3D plot any way you like (put the cursor on the plot, hold down the left mouse button, rotate).

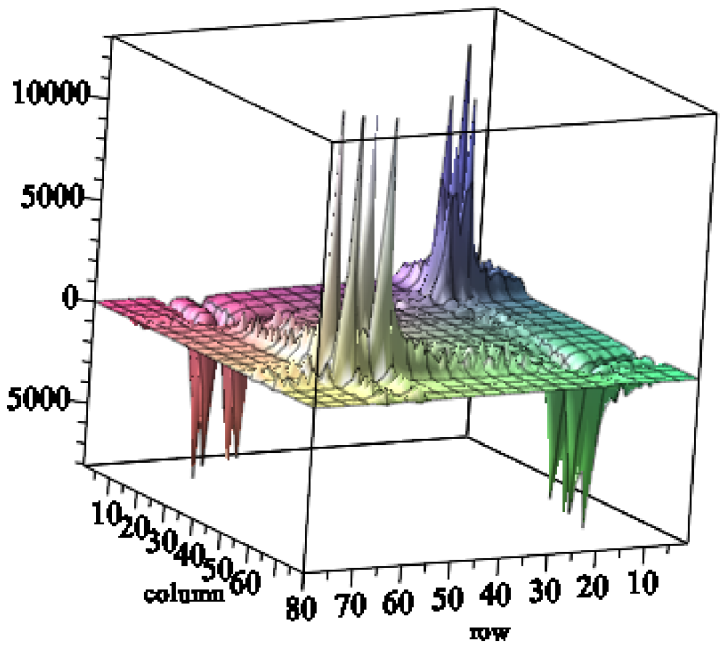
```
> B := Matrix(Transpose(A));
```

$$B := \begin{bmatrix} 80\ x\ 40\ Matrix \\ Data\ Type:\ anything \\ Storage:\ rectangular \\ Order:\ Fortran\_order \end{bmatrix} \quad (3)$$

```
> C = (B • A);
```

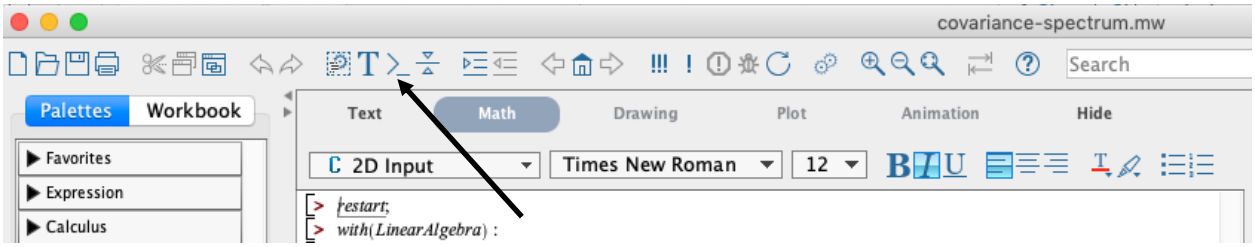
$$C = \begin{bmatrix} 80\ x\ 80\ Matrix \\ Data\ Type:\ float_8 \\ Storage:\ rectangular \\ Order:\ Fortran\_order \end{bmatrix} \quad (4)$$

```
> matrixplot(B • A);
```

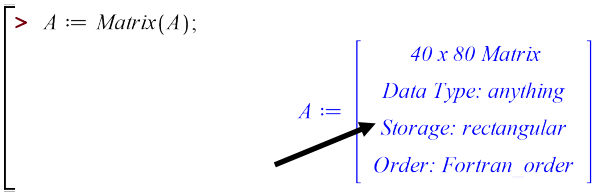


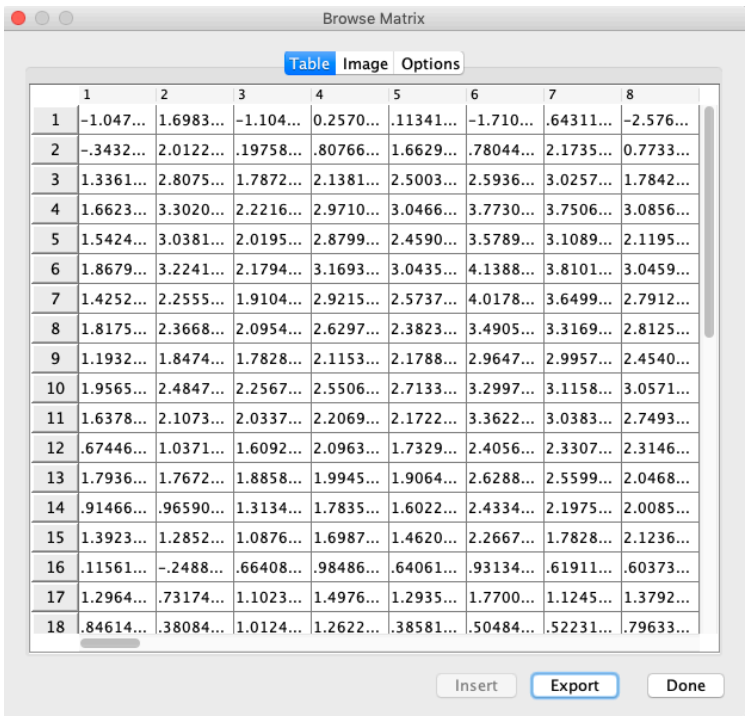

```
> restart ;
>
```

One note. When the matrices are defined or calculated, the Maple output is just some information about the matrix (how many rows and columns, and other stuff). Typically, Maple in the worksheet mode will only display the actual values of all of the matrix elements if the matrices are 10 rows by 10 columns or smaller. However, there are some methods available for showing the contents of matrices. One way is to force Maple to show the matrices, and you can do this if you like. At the top of the worksheet, just after the 'restart' command, you can insert a line with the command 'interface(rtablesize=infinity);'. To insert a line, put the cursor at the end of the 'restart' line, and then click the '>_' button on the top menu. This will insert a new line below the 'restart' line with a cursor, ready to accept a command. Once you insert the 'interface(rtablesize=infinity);' command, all of the elements of the matrices will be shown (when there is a semi-colon at the end of the line, and a colon will still suppress the output).




Another way is to invoke the 'Browse Matrix' tool. You do this by putting the cursor on the representation of the matrix (here, right, on the representation for our centered matrix 'A') and double clicking. This will bring up the 'Browse Matrix' window (right), which has some nice options including the ability to export the matrix.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | -1.047... | 1.6983... | -1.104... | 0.2570... | .11341... | -1.710... | .64311... | -2.576... |
| 2 | -.3432... | 2.0122... | .19758... | .80766... | 1.6629... | .78044... | 2.1735... | 0.7733... |
| 3 | 1.3361... | 2.8075... | 1.7872... | 2.1381... | 2.5003... | 2.5936... | 3.0257... | 1.7842... |
| 4 | 1.6623... | 3.3020... | 2.2216... | 2.9710... | 3.0466... | 3.7730... | 3.7506... | 3.0856... |
| 5 | 1.5424... | 3.0381... | 2.0195... | 2.8799... | 2.4590... | 3.5789... | 3.1089... | 2.1195... |
| 6 | 1.8679... | 3.2241... | 2.1794... | 3.1693... | 3.0435... | 4.1388... | 3.8101... | 3.0459... |
| 7 | 1.4252... | 2.2555... | 1.9104... | 2.9215... | 2.5737... | 4.0178... | 3.6499... | 2.7912... |
| 8 | 1.8175... | 2.3668... | 2.0954... | 2.6297... | 2.3823... | 3.4905... | 3.3169... | 2.8125... |
| 9 | 1.1932... | 1.8474... | 1.7828... | 2.1153... | 2.1788... | 2.9647... | 2.9957... | 2.4540... |
| 10 | 1.9565... | 2.4847... | 2.2567... | 2.5506... | 2.7133... | 3.2997... | 3.1158... | 3.0571... |
| 11 | 1.6378... | 2.1073... | 2.0337... | 2.2069... | 2.1722... | 3.3622... | 3.0383... | 2.7493... |
| 12 | .67446... | 1.0371... | 1.6092... | 2.0963... | 1.7329... | 2.4056... | 2.3307... | 2.3146... |
| 13 | 1.7936... | 1.7672... | 1.8858... | 1.9945... | 1.9064... | 2.6288... | 2.5599... | 2.0468... |
| 14 | .91466... | .96590... | 1.3134... | 1.7835... | 1.6022... | 2.4334... | 2.1975... | 2.0085... |
| 15 | 1.3923... | 1.2852... | 1.0876... | 1.6987... | 1.4620... | 2.2667... | 1.7828... | 2.1236... |
| 16 | .11561... | -.2488... | .66408... | .98486... | .64061... | .93134... | .61911... | .60373... |
| 17 | 1.2964... | .73174... | 1.1023... | 1.4976... | 1.2935... | 1.7700... | 1.1245... | 1.3792... |
| 18 | .84614... | .38084... | 1.0124... | 1.2622... | .38581... | .50484... | .52231... | .79633... |