Maple Lab #1 for Math 406

Note: For homework assignments: write (type!) your name at the top!

1. Basics

> restart; # clears everything

Note that # starts an in-line comment (ignored by Maple)

1.1 Help features: (a) By using ? and the command (a new window appears):

> ?sqrt

(b) by typing the command (or by placing the cursor on a word) and pressing F2:

sqrt

1.2 Evaluating numbers (in decimal form):

Note that Maple distinguishes between numbers and their decimal approximations:

> s2 := sqrt(2);

\[ s2 := \sqrt{2} \]  

(1)

s2 is the exact square root of 2, whereas the following is an approximation to 50 decimal places:

> s2approx := evalf(s2, 50);

\[ s2approx := 1.414213562373095048801688724209698075696718753769 \]  

(2)

1.3 Comparing numbers/expressions:

> evalb(s2^2 = 2);

\[ true \]  

(3)

> evalb(s2approx^2 = 2);

\[ false \]  

(4)

Thus, squaring s2 is exactly equal to 2, whereas squaring s2approx is not.

2. Number Theory

2.1 Quotient and remainder:

> q := iquo(123, 39); r := irem(123, 39);

\[ q := 3 \]

\[ r := 6 \]  

(5)

Check that q*39 + r = 123:

> q*39 + r;

\[ 123 \]  

(6)

A better way to check this is as follows:

> evalb(q*39 + r = 123);

\[ true \]  

(7)

2.2 (a) The gcd of two numbers m, n:

> igcd(13843, 14351);

\[ 127 \]  

(8)

(b) The extended gcd: find x, y such that mx + ny = gcd(m,n)

> igcdex(13843, 14351, 'x', 'y'); x, y;

\[ 127 \]

\[ 28, -27 \]  

(9)

> 13843*x + 14351*y;

\[ 127 \]  

(10)
2.3 Free variables of their values. (Note that if no value has been assigned to a name (or if MAPLE does not recognize a command), then MAPLE returns the name (or repeats the command).)

```maple
> unassign('x', 'y'); x, y;
```

2.4 Modular arithmetic:

The modular inverse is computed as follows:

```maple
> modp(1/12, 17);
```

Thus, $10 \times 12 = 1$ in $(\mathbb{Z}/17\mathbb{Z})^x$. Check this:

```maple
> modp(10-12, 17);
```

3. Lists

3.1 Lists and their elements:

(a) The following defines a list called "L" (one cannot use "list" as a name):

```maple
> L := [5, 4, 3, 1, 2, 7];
```

(b) The 5th element of the list "L" is given as follows:

```maple
> L[5];
```

3.2 List of lists:

(a) The following is a list whose elements are other lists (and/or numbers):

```maple
> lst := [L, [1, 2], 5];
```

(b) The 2nd element of the list "lst" is thus the list $[1, 2]$:

```maple
> lst[2];
```

(c) The double index $lst[1][2]$ refers to the 2nd element in the list $lst[1]$ (= lis):

```maple
> lst[1, 2]; lst[1][2];
```

3.3 The `sequence` command `seq(,..)` is useful for constructing lists that follow a pattern:

```maple
> seq(k^2, k = 1..10);
```

This is the list of the 1st ten squares. Note that without the square brackets, we only get a `sequence`:

```maple
> seq(k^2, k = 1..10);
```

4. Matrices and Linear Algebra:

4.1 Defining matrices:

```maple
> A := Matrix(2, 3, [1, 2, 3, 4, 5, 6]); # row matrix
```

```maple
> At := <<1, 2, 3|4, 5, 6>>; # column matrix
```

```maple
A :=
```

\[
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{pmatrix}
\]
Augmenting matrices:

\[
\begin{bmatrix}
1 & 4 \\
2 & 5 \\
3 & 6
\end{bmatrix}
\]

\(At := \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}\) (21)

\[AAt := \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}\]

\(AtA := \begin{bmatrix} 17 & 22 & 27 \\ 22 & 29 & 36 \\ 27 & 36 & 45 \end{bmatrix}\) (22)

\[AAt^{-1} := \begin{bmatrix} \frac{77}{54} & -\frac{16}{27} \\ \frac{16}{27} & \frac{7}{27} \end{bmatrix}\] (23)

4.2 Matrices and modular arithmetic (Computing modulo n):

\[AA := \langle A | A \rangle; A4 := \langle A | A | A \rangle;\]

\[AA := \begin{bmatrix} 1 & 2 & 3 & 1 & 2 & 3 \\ 4 & 5 & 6 & 4 & 5 & 6 \end{bmatrix}\]

\[A4 := \begin{bmatrix} \text{2 x 12 Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}\] (24)

Note that Maple does not automatically display matrices of size > 10.

To display larger matrices, reset the maximum size of the displayed matrices:

\[\text{interface(}\text{rtablesize = 15); A4;}\]

\[
\begin{bmatrix}
1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \\
4 & 5 & 6 & 4 & 5 & 6 & 4 & 5 & 6 & 4 & 5 & 6
\end{bmatrix}
\]

(25)

4.2 Matrices and modular arithmetic (Computing modulo n):

\[A4 \mod 3;\]

\[
\begin{bmatrix}
1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0
\end{bmatrix}
\]

(26)

\[AAt^{-1} \mod 5;\]

\[
\begin{bmatrix}
3 & 2 \\
2 & 1
\end{bmatrix}
\]

(27)
4.2 Working with matrices: this requires the `LinearAlgebra` package
(see course web page for details)

```maple
> with(LinearAlgebra):
> SubMatrix(A4, 1..2, 1..6); # extracting a submatrix

\[
\begin{bmatrix}
1 & 2 & 3 & 1 & 2 & 3 \\
4 & 5 & 6 & 4 & 5 & 6
\end{bmatrix}
\]
```

```maple
> ColumnOperation(A4, [1, 3]); # interchange columns 1 and 3 of A4

\[
\begin{bmatrix}
3 & 2 & 1 & 2 & 3 & 1 & 2 & 3 \\
6 & 5 & 4 & 4 & 5 & 6 & 4 & 5 & 6
\end{bmatrix}
\]
```

```maple
> A4 mod 3;

\[
\begin{bmatrix}
1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \\
1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0
\end{bmatrix}
\]
```

5. Defining our own functions and procedures

5.1 Functions (simple expressions):

```maple
> f := n \rightarrow n^2 - 2 \cdot n + 1;
```

\[f := n \rightarrow n^2 - 2n + 1\]

This defines a function called "f". To evaluate it at n = 12, write \(f(12)\):

```maple
> f(12);
```

\[121\]

The following is the sequence (respectively, list) of the first 10 values of \(f(n)\):

```maple
> seq(f(k), k = 1..10); [seq(f(k), k = 1..10)];
```

\[0, 1, 4, 9, 16, 25, 36, 49, 64, 81\]

5.2 Compound expressions (and/or algorithms): use "proc ... end;".

Example: Write a program "sumnos" which sums (or adds) the first n integers and returns both \(n\) and the sum in a list, i.e., \(\text{sumnos}(n) = [n, 1 + 2 + \ldots + n]\).

```maple
> sumnos := proc(n) local i, s;
        s := 0; for i to n do s := s + i; od; return([n, s]); end;
```

```maple
> sumnos := proc(n) local i, s := 0; for i to n do s := s + i end do; return([n, s]) end proc
```

Note that this program uses the looping construction: for ... do; ... (commands) ... od;

To execute this program, say for \(n=10\), use the program name and the value 10:

```maple
> sumnos(10);
```

\[[10, 55]\]

Thus, \(1 + 2 + \ldots + 10 = 55\).