Linear Programming Notes for Maple

1. Basic Linear Algebra in Maple

To use these commands, type `with(linalg);` first.

(1) To define a matrix in Maple: (This example is a 2×3 matrix)

```
A:=matrix(2,3,[1,2,3,4,5,6]);
A:=matrix([ [1,2,3],[4,5,6] ]);
```

Reminder: In Matlab, it was: `A=[1,2,3;4,5,6];`

(2) To define a vector in Maple, you can either define an $n \times 1$ or $1 \times n$ matrix, or use the vector command:

```
b:=vector([a,b,c]);
b:=matrix(3,1,[a,b,c]);
```

When you use "vector", Maple will assume it is a row or a column depending on how matrix-vector multiplication is defined. Defining a vector using `matrix` forces it to be a row/column vector.

(3) Scalar Multiplication: In this example, compute γA

```
A:=matrix(2,3,[1,2,3,4,5,6]);
gA:=evalm(gamma*A);
```

(4) Add two matrices (must be the same sizes):

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

(5) Matrix Transpose, Determinant, Inverse, Rank:

```
A:=matrix(2,2,[1,2,b,4]);
transpose(A);
det(A);
inverse(A);
rank(A);
```

(6) Matrix-Vector, Matrix-Matrix multiplication: Use `&*` rather than `*`

```
b:=vector([3,c]);
evalm(A&*b);
evalm(b&*A);
```

Compare that with the following (One of these gives an error):

```
b:=matrix(2,1,[3,c]);
evalm(A&*b);
evalm(b&*A);
```

And Matrix-Matrix multiplication (One of these gives an error):

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

(7) To identify a particular element of a matrix, $A[i, j]$ (Note that these are square brackets; in Matlab these would be parentheses).

2. Solving Systems of Equations

There are always three possibilities when solving $Ax = b$ for x: A unique solution, No solution, or an infinite number of solutions (free variables).

(1) Maple can give you the explicit equations you're working with:

```
A:=matrix(3,3,[3,1,1,1,2,1,1,1,7]);
b:=vector([10,20,30]);
geneqns(A,[x,y,z],b);
```

(2) To solve the system directly: `linsolve(A,b);`

(3) In Linear Algebra, we use Gaussian Elimination with Pivoting to put an augmented matrix into RREF (Row Reduced Echelon Form). Maple can do that for you if you want to see it (sometimes its handy when there is no solution or an infinite number of solutions):

```
C:=augment(A,b);
rref(C);
```

(4) Here's an example of a system with an infinite number of solutions:

```
A:=matrix(2,2,[3,1,6,2]);
b:=vector([4,8]);
linsolve(A,b);
```

Maple uses $\_t_1$, $\_t_2$, $\_t_3, \ldots$ to represent the variable for the first column, second column, etc. Compare this to the RREF of the augmented matrix:

```
Ab:=augment(A,b);
rref(Ab);
```

(5) Here's an example of a system with no solution (Again, compare it with the RREF):

```
A:=matrix(2,2,[1,1,1,1]);
b:=vector([1,2]);
linsolve(A,b);
```

3. A Specialty plot: INEQUAL

EXAMPLE: Plot the region in the plane defined by the following set of inequalities:

$$\begin{aligned} x_1 &\leq 90 \\ x_2 &\leq 60 \\ 5x_1 + 6x_2 &\leq 600 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

First we'll define the set of constraints, then we'll plot them. To use `inequal`, we first need to call `with(plots)`

```
with(plots):
CS:=[ x[1]<=90, x[2]<=60, 5*x[1]+6*x[2]<=600, x[1]>=0, x[2]>=0 ];
inequal(CS,x[1]=1..120,x[2]=0..120, optionsfeasible=(color=white),
optionsexcluded=(color=yellow) );
```

NOTE: This will only plot linear inequalities and only in the plane.

4. Linear Programming in Maple

Before using these commands, type `with(simplex):`

(1) We can solve for the intersection points of the boundary in Maple. For example, to solve for the intersection point between constraints 1 and 5 we would write:

```
Eqns1:=convert( { CS[1],CS[2] }, equality );
solve(Eqns1);
```

(2) EXAMPLE: Solve using Maple: Maximize $30x_1 + 40x_2$ so that all the previous constraints in `CS` are satisfied. We'll start from scratch so that all relevant commands are right here:

```
with(simplex):
CS:=[ x[1]<=90, x[2]<=60, 5*x[1]+6*x[2]<=600];
z:=30*x[1]+40*x[2];
Sol:=maximize(z, { seq( CS[i],i=1..3)},NONNEGATIVE);
assign(Sol); z;
```

The last line will give you what the maximum value is. Using `NONNEGATIVE` will save some typing- this is in place of the last set of inequalities.

(3) EXAMPLE: We can get Maple to tell us what it is computing by using `infolevel[simplex]`. In this example, we have an unbounded problem so there is no solution. Without using `infolevel`, Maple returns an empty solution so we would be unsure as to why it is empty.

```
CS:=[x[1]<=20,x[2]>=10];
Z:=x[1]+x[2];
Sol:=maximize(Z, {seq(CS[i],i=1..2)});
infolevel[simplex]:=1;
Sol:=maximize(Z, {seq(CS[i],i=1..2)});
```

5. Exercises:

Use Maple to solve the following problems:

(1) Solve the system:

$$\begin{aligned} 3a - b + 5c &= 2 \\ 2a - b - c &= 1 \\ b - 3c &= 5 \end{aligned}$$

(2) Solve the equation $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$, if

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 5 & 6 & 0 & 0 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 90 \\ 60 \\ 600 \end{bmatrix}$$

(3) Maximize $x_1 + x_2$ subject to:

$$\begin{aligned} 2x_1 + x_2 &\le 29 \\ x_1 + 2x_2 &\le 25 \\ x_1 &\ge 2 \\ x_2 &\ge 5 \end{aligned}$$

Also plot the feasible region.

(4) Maximize $x_1 + x_2 + x_3 + x_4 + x_5$ subject to:

$$\begin{aligned} x_1 + x_2 &\leq 100 \\ x_3 + x_4 &\leq 70 \\ x_2 + x_3 + 2x_4 + 5x_5 &\leq 250 \\ x_i &\geq 0 \qquad 1 \leq i \leq 5 \end{aligned}$$

(5) Minimize $2x_1 + 4x_2$ subject to:

$$\begin{aligned} x_1 + 5x_2 &\leq 100 \\ 4x_1 + 2x_2 &\geq 20 \\ x_1 + x_2 &= 10 \\ x_i &\geq 0 \qquad 1 \leq i \leq 2 \end{aligned}$$

And plot the feasible region.