```
> restart;
  with( linalg ):
Warning: new definition for   norm
Warning: new definition for   trace
```

Maple Worksheet for Geometric Programming Homework

Prepared by Douglas B. Meade (meade@math.sc.edu)
26 October 1995

WWW URL: http://www.math.sc.edu/~meade/math524/worksheets/ch2no26.ms

---

Homework Problem: #26, p. 81

NOTE: You should be using Maple as a tool. Don't expect to avoid pencil and paper. In fact, you HW should not appear any different from previous assignments except that messy computations are omitted (with a reference to the fact that they were completed using Maple).

The objective function for the (primal) geometric program is

```
> g := 1000/x/y + 2*x + 2*y + x*y;
```

$$g := 1000 \frac{1}{x\,y} + 2\,x + 2\,y + x\,y$$

The objective function for the dual geometric program is

```
> v := (1000/delta[1])^delta[1] * (2/delta[2])^delta[2] * (2/delta[3])^delta[3] * (1/delt
  a[4])^delta[4];
```

$$v := \left(1000 \frac{1}{\delta_1}\right)^{\delta_1} \left(2 \frac{1}{\delta_2}\right)^{\delta_2} \left(2 \frac{1}{\delta_3}\right)^{\delta_3} \left(\frac{1}{\delta_4}\right)^{\delta_4}$$

```
>
```

The feasible vectors are the solution(s) to the positivity, normality, and orthogonality conditions:

```
> eqP := { delta[1]>0, delta[2]>0, delta[3]>0, delta[4]>0 };
```

$$eqP := \left\{0 < \delta_1, 0 < \delta_2, 0 < \delta_3, 0 < \delta_4\right\}$$

```
> eqN := delta[1] + delta[2] + delta[3] + delta[4] = 1;
```

$$eqN := \delta_1 + \delta_2 + \delta_3 + \delta_4 = 1$$

```
> eqO := -delta[1] + delta[2] +            delta[4] = 0,
         -delta[1] +            delta[3] + delta[4] = 0;
```

$$eqO := -\delta_1 + \delta_2 + \delta_4 = 0, -\delta_1 + \delta_3 + \delta_4 = 0$$

```
>
> F :=solve( {eqN, eqO}, {delta[1],delta[2],delta[3],delta[4]} );
```

$$F := \left\{\delta_2 = \delta_3, \delta_4 = -\frac{3}{2}\delta_3 + \frac{1}{2}, \delta_1 = -\frac{1}{2}\delta_3 + \frac{1}{2}, \delta_3 = \delta_3\right\}$$

NOTE that 1) there are an infinite number of solutions to these equations and 2) these are exactly the same as for the example in the text (see geomprog.ms)..

With the above solutions, the dual objective function becomes

```
> vv := subs( F, v );
```

$$vv := \left(1000\frac{1}{-\frac{1}{2}\delta_3 + \frac{1}{2}}\right)^{\left(-1/2\,\delta_3 + 1/2\right)} \left(\left(2\frac{1}{\delta_3}\right)^{\delta_3}\right)^2 \left(\frac{1}{-\frac{3}{2}\delta_3 + \frac{1}{2}}\right)^{\left(-3/2\,\delta_3 + 1/2\right)}$$

To simplify matters, let s = delta[3].

```
> vv := subs( delta[3]=s, vv );
```

$$vv := \left(1000\frac{1}{-\frac{1}{2}s + \frac{1}{2}}\right)^{\left(-1/2\,s + 1/2\right)} \left(\left(2\frac{1}{s}\right)^{s}\right)^2 \left(\frac{1}{-\frac{3}{2}s + \frac{1}{2}}\right)^{\left(-3/2\,s + 1/2\right)}$$

```
>
```

We still have to enforce the positivity condition.

```
> subs( F, delta[3]=s, eqP );
```

$$\left\{0 < s,\ 0 < -\frac{1}{2}s + \frac{1}{2},\ 0 < -\frac{3}{2}s + \frac{1}{2}\right\}$$

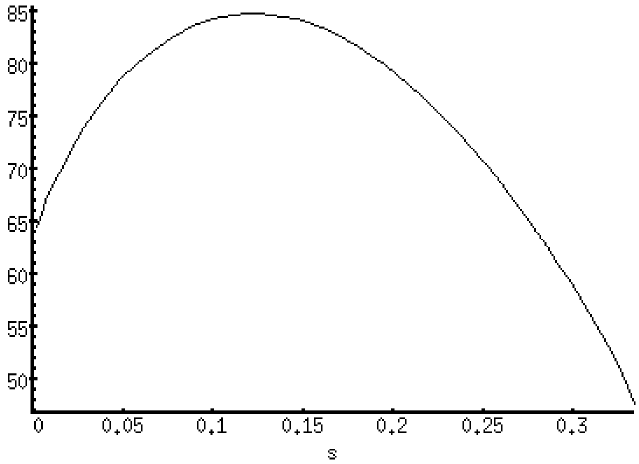
Unfortunately, Maple is not yet able to solve this system of inequalities. It's not difficult to see that the above conditions are satisified when:
s>0
s<1
s<1/3
Thus, we restrict our search to 0 < s < 1/3.

```
>
```

So, how would you find the maximum value of vv on the interval (0,1/3) ?

A good way to begin is to get a picture of the function.

```
> plot( vv, s=0..1/3 );
```




```
>
```

In this case we can learn much about the solution, which will be denoted ss (for s-star). However, we can also look for the exact solution -- using Calculus.

> dv := diff(vv, s);

$$dv := \%1^{(-1/2\,s+1/2)}\left(-\frac{1}{2}\ln(\%1)+\frac{1}{2}\right)\left(\left(2\,\frac{1}{s}\right)^{s}\right)^{2}\left(\frac{1}{-\frac{3}{2}\,s+\frac{1}{2}}\right)^{(-3/2\,s+1/2)}$$
$$+\,2\,\%1^{(-1/2\,s+1/2)}\left(\left(2\,\frac{1}{s}\right)^{s}\right)^{2}\left(\frac{1}{-\frac{3}{2}\,s+\frac{1}{2}}\right)^{(-3/2\,s+1/2)}\left(\ln\left(2\,\frac{1}{s}\right)-1\right)$$
$$+\,\%1^{(-1/2\,s+1/2)}\left(\left(2\,\frac{1}{s}\right)^{s}\right)^{2}\left(\frac{1}{-\frac{3}{2}\,s+\frac{1}{2}}\right)^{(-3/2\,s+1/2)}\left(-\frac{3}{2}\ln\left(\frac{1}{-\frac{3}{2}\,s+\frac{1}{2}}\right)+\frac{3}{2}\right)$$

$$\%1 := 1000\,\frac{1}{-\frac{1}{2}\,s+\frac{1}{2}}$$

The critical points of v are the zeroes of the derivative of v:

> cp := solve(dv=0, s);

$$cp := \mathrm{RootOf}(973\,\_Z^4+54\,\_Z^3-36\,\_Z^2+10\,\_Z-1)$$

Amazingly, the zeroes of v' are simply the roots of the above fourth-order polynomial (in _Z). As such, the exact solution can be obtained.

> cp := allvalues(cp):

What can we see from this? Note that there are four roots (as expected). The last two roots appear to be complex-valued, so the maximizer should come from one of the first two roots. Which one? And, are you sure this falls in the needed range?

To find out, let's ask Maple for floating-point approximations to these roots.

> evalf(cp);

$$-.3126322090,\ .06746340497-.1494958747\,I,\ .06746340497+.1494958747\,I,\ .1222069407$$

Thus, the fourth root is our optimal choice for s (and this is consistent with our graph).

```
> ss := cp[4];
```

$$ss := .1222069407$$

The maximum value of the dual objective function is

```
> vvs := subs( s=ss, vv );
```

$$vvs := 84.82073538$$

and this value is attained with delta as follows (remember that delta*[3] = ss):*

```
> ds := subs( delta[3]=ss, F );
```

$$ds := \{\delta_2 = .1222069407, \delta_4 = .3166895889, \delta_1 = .4388965297, .1222069407 = .1222069407\}$$

```
>
```

The last step is to take the solution to the dual geometric program and compute the solution of the (primal) geometric program (which will be denoted xs and ys). According to Theorem 2.5.2, we know the solution to the primal and dual geometric programs are related by:

delta[1] * v(delta*) = u[1](t*) = 1000 / xs / ys*
delta[2] * v(delta*) = u[2](t*) = 2 * xs*
delta[3] * v(delta*) = u[3](t*) = 2 * ys*
delta[4] * v(delta*) = u[4](t*) = xs * ys*

While this could be a real mess, look at the middle two equations -- they tell us exactly how to compute the solution to the original problem.

```
> xs := subs( ds, ss*vvs )/2:
  evalf( xs );
```

5.182841290

```
> ys := subs( ds, ss*vvs )/2:
  evalf( ys );
```

5.182841290

The fact that x = y* should not be surprising. (Why?)*

As a final check, let’s look at the minimum value of g.

```
> subs( x=xs, y=ys, g ):
  evalf( " );
```

84.82073536

Does this agree with what you expect? Explain.

END OF WORKSHEET

```
>
```