MATH 420 Mathematical Modeling

GoldenOrbWeaver

Spring 2026

Contents

1 One Variable Optimization

Approach to mathematical modeling:

1. Ask the question
2. Select the modeling approach
3. Formulate the model
4. Solve the problem
5. Answer the question

Example. A pig weighing 200 pounds gains 5 pounds per day and costs 45 cents a day to keep. The market price for pigs is 65 cents per pound, but is falling 1 center per day. When should the pig be sold?

Answer. Start at $t = 0$, with t in days. Then we find each component:
Weight of pig:

$$w(t) = 200 + 5t$$

Cost of pig:

$$c(t) = 0.45t$$

Price per pound to sell pig:

$$p(t) = 0.65 - 0.01t$$

Revenue from selling pig:

$$r(t) = w(t)p(t)$$

Profit from selling at time t:

$$p(t) = r(t) - c(t)$$

⊛

However, we aren't sure that our values are accurate. The amount the price falls per day could be different, so we can instead set it to a function $r(t)$ then:

$$p(t) = 0.65 - rt$$

To find the optimal time to sell, solve $p'(t) = 0$ to get t_{max}. We are interested in how sensitive t_{max} is on changes in r, which we call $s(t, r)$:

$$s(t, v) = \lim \frac{\frac{\Delta t}{t}}{\frac{\Delta r}{r}} = \frac{r}{t} * \frac{dt}{dr}$$

We begin to solve:

$$p'(t) = r'(t) - c'(t) = w'(t)p'(t) = w(t)p'(t) - 0.45 = 5(0.65 - rt) + (200 + 5t)(-r) - 0.45 = -10rt - 200r + 2.80 = 0$$

Then, we solve for t_{max}:

$$t_{max} = \frac{2.80 - 200r}{10r} = 0.28r^{-1} - 20$$

Next, we find $\frac{dt_{max}}{dr}$:

$$\frac{dt_{max}}{dr} = -0.28r^{-2}$$

Therefore:

$$S(t, r) = \frac{-r}{t_{max}} * \frac{0.28}{r^2} = \frac{-0.28}{t_{max}r}$$

Substituting t_{max}:

$$S(t,r) = \frac{-0.28}{(0.28r^{-1} - 20)r} = \frac{-0.28}{0.28 - 20r}$$

At our best guess of $r = 0.01$:

$$S(t, 0.01) = \frac{-0.28}{0.28 - 0.2} = \frac{-0.28}{0.08} = \frac{-7}{2}$$

Thus, if r is wrong by $\pm 10\%$, then t_{max} will be wrong by $S(t,r) * 10\% = \frac{7}{2} * 10\% = \pm 35\%$

What if the rate of growth of the pig is also wrong? We redefine $w(t)$:

$$w(t) = 200 + gt$$

Then, we find $S(t,g)$:

$$S(t,g) = \frac{g}{t_{max}} * \frac{dt_{max}}{dg}|_{g=5} \approx 3.0625$$

We can also examine the sensitivity of the outcome:

$$S(P,r) = \frac{r}{P} * \frac{dP}{dr} = \frac{r}{P(t_{max})} * \frac{dP(t)}{dt} * \frac{dt}{dr}|_{t=8} = 0$$

This means that the profit doesn't change that much (to the first order) when the rate of change in the price of the pig varies.

(FIX LAST PART WITH dp/dr, on website. ALSO NEW PLOT)

Example. A manufacturer of color TV sets is planning the introduction of two new products, a 19-inch LCD flat panel set with a manufacturer's suggested retail price (MSRP) of $339 and a 21-inch LCD flat panel set with an MSRP of $399. The cost to the company is $195 per 19-inch set and $225 per 21-inch set, plus an additional $400,000 in fixed costs. In the competitive market in which these sets will be sold, the number of sales per year will affect the average selling price. It is estimated that for each type of set, the average selling price drops by one cent for each additional unit sold. Furthermore, sales of the 19-inch set will affect sales of the 21-inch set and vice-versa. It is estimated that the average selling price for the 19-inch set will be reduced by an additional 0.3 cents for each 21-inch set sold, and the price for the 21-inch set will decrease by 0.4 cents for each 19-inch set sold. How many units of each type of set should be manufactured?

Answer. Let x_1 be the number of 19" TVs sold and x_2 be the number of 21" TVs sold. Then, our price for the 19" TVs is:

$$p_1(x_1, x_2) = 339 - 0.01x_1 - 0.003x_2$$

And our price for the 21" TVs is:

$$p_2(x_1, x_2) = 399 - 0.004x_1 - 0.01x_2$$

The total cost to produce TVs is:

$$C(x_1, x_2) = 400000 + 195x_1 + 225x_2$$

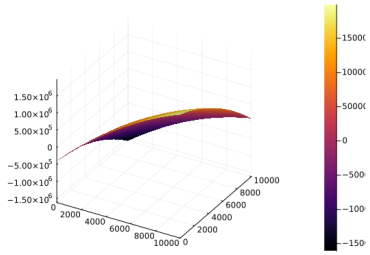
Our revenue is:

$$R(x_1, x_2) = x_1 p_1(x_1, x_2) + x_2 p_2(x_1, x_2)$$

Our profit is:

$$P(x_1, x_2) = R(x_1, x_2) - C(x_1, x_2)$$

```
p1(x1,x2)=339-0.01*x1-0.003*x2
p2 (generic function with 1 method)

p2(x1,x2)=399-0.004*x1-0.01*x2
p2 (generic function with 1 method)

R(x1,x2)=p1(x1,x2)*x1+p2(x1,x2)*x2
R (generic function with 1 method)

C(x1,x2)=400000+195x2+225x2
C (generic function with 1 method)

P(x1,x2)=R(x1,x2)-C(x1,x2)
P (generic function with 1 method)

import Pkg; Pkg.add("plots")
Show hidden output
Next steps: Explain error

using Plots

surface(0:100:10000,0:100:10000,P)
```



FIX THIS WRONG EQUATION

Note. When using computer algebra systems, use fractions instead of decimals because computers using floating point decimal systems, which can cause errors.

We are trying to maximize $P(x_1, x_2)$ on the set:

$$S = \{(x_1, x_2) : x_1 \geq 0, x_2 \leq 0\} \cap \mathbb{Z} \times \mathbb{Z}$$

$$\frac{\partial p}{\partial x_1} = \frac{\partial R}{\partial x_1} - \frac{\partial C}{\partial x_1}$$

(FINISH WRITING THIS SOLUTION, ADD JULIA PART)

⊛

What if the elasticity of 19" televisions was a instead of 0.01?
We first find x_1 and x_2 as functions of a, and then find:

$$S(x_1, a) = \frac{a}{x_1} * \frac{dx_1}{da}|_{a=0.01, x_1 optm.} = -\frac{400}{351}$$

and

$$S(x_2, a) = \frac{a}{x_2} * \frac{dx_2}{da}|_{a=0.01, x_2 optm.} = \frac{9695}{36153}$$

Example. We reconsider the color TV problem introduced in the previous section. There we assumed that the company has the potential to produce any number of TV sets per year. Now we will introduce constraints based on the available production capacity. Consideration of these two new products based on the available production capacity. Consideration of these two new came about because the company plans to discontinue manufacturing of some older models, thus providing excess capacity at its assembly plant. This excess capacity could be sued to increase production of other existing production lines, but the company feels that the new products will be more profitable
FINISH WRITING

FINISH WRITING NOTES FOR SOLUTION, SIMILAR TO MATH 487
Look into math modeling competitions (SIAM, consortium) download and learn julia. FIX FORMATTING'

Example. Maximize $x + 2x + 3z$ over the set $x^2 + y^2 + z^2 = 3$

Answer. We have $f(x, y, z) = x + 2y + 3z$ and $g(x, y, z) = x^2 + y^2 + z^2$

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \text{ and}$$

⊛

FINISH WRITING ANSWER

We now perform sensitivity analysis on our answer to this question, with a being the elasticity of demand:

$$S(x_1, a) = \frac{dx_1}{da} * \frac{a}{x_1}$$

$$S(X_2, a) = \frac{dx_2}{da} * \frac{a}{x_2}$$

We first set up an equation with the Lagrange multiplier (REPHRASE):

$$\nabla P = \lambda \nabla g \text{ max condition}$$

$$g(x_1, x_2) = x_1 + x_2$$

$$\text{Constraint is } g(x_1, x_2) = 10000$$

We then solve:

$$\nabla g = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\nabla P = \begin{bmatrix} \frac{\partial P}{\partial x_1} \\ \frac{\partial P}{\partial x_2} \end{bmatrix}$$

(ADD PART WITH JULIA)
We get:

$$x_1 = \frac{\frac{-831}{500} + \frac{13}{1000}\lambda}{\frac{49}{1000000} - \frac{1}{25}a}$$

$$x_1 = \frac{144 - \frac{348000}{7}a - \lambda + \frac{2000}{7}a\lambda}{\frac{7}{1000} - \frac{40}{7}a}$$

Sensitivity analysis:

$$S(x_1, a) = \frac{a}{x_1}\frac{dx_1}{da} = \frac{a}{x_1}(\frac{\partial x_1}{\partial a} + \frac{\partial x_1}{\partial \lambda}\frac{d\lambda}{da}) \tag{1}$$

$$= \frac{a}{x_1}\Big[(\frac{\frac{-831}{500} + \frac{13}{1000}\lambda}{(\frac{49}{1000000} - \frac{1}{25}a)^2})(\frac{1}{25}) + \frac{13}{1000} * \frac{1}{\frac{49}{1000000} - \frac{1}{25}a}\frac{d\lambda}{da}\Big] \tag{2}$$

We evaluate this at $a = 0.01$ and the corresponding optimal production level to get the elasticity at that point.

Setting up:

```
julia> using Symbolics

julia> @variables a, x1, x2, lambda
4-element Vector{Num}:
      a
     x1
     x2
 lambda

julia> b=[lambda, lambda]
2-element Vector{Num}:
 lambda
 lambda

julia> b=[BigInt(144)-lambda, BigInt(174)-lambda]
2-element Vector{Num}:
 144 - lambda
 174 - lambda

julia> A=[BigInt(2)*a BigInt(7)//1000; BigInt(7)//1000 BigInt(2)//100]
2×2 Matrix{Num}:
      2a  7//1000
 7//1000    1//50

julia> X=A\b
2-element Vector{Num}:
 (174 - lambda + ((-1//50)*(144 - lambda - (2000//7)*a*(174 - lambda))) /
 ((7//1000) - (40//7)*a)) / (7//1000)
                                        (144 - lambda - (2000//7)*a*(174
 - lambda)) / ((7//1000) - (40//7)*a)

julia> a0=BigInt(1)//100
1//100

julia> g(x1,x2)=x1+x2
g (generic function with 1 method)
```

Solving for maximum:

```
julia> myLambda=Symbolics.solve_for(g(X[1],X[2])~BigInt(10000), lambda)
(-10000 + (144 - (348000//7)*a) / ((7//1000) - (40//7)*a) + (174 + ((-1//
50)*(144 - (348000//7)*a)) / ((7//1000) - (40//7)*a)) / (7//1000)) / (-((
-1 + (2000//7)*a) / ((7//1000) - (40//7)*a)) - ((-1 + ((-1//50)*(-1 + (20
00//7)*a)) / ((7//1000) - (40//7)*a)) / (7//1000)))

julia> lambda0=substitute(myLambda, a=>a0)
24//1

julia> substitute(X[1], [a=>a0, lambda=>lambda0])
50000//13

julia> x10=substitute(X[1], [a=>a0, lambda=>lambda0])
50000//13

julia> x20=substitute(X[2], [a=>a0, lambda=>lambda0])
80000//13
```

Differentiating:

```
julia> p1(x1,x2)=339-a*x1-3//1000*x2
p1 (generic function with 1 method)

julia> p2(x1,x2)=399-4//1000*x1-1//100*x2
p2 (generic function with 1 method)

julia> C(x1,x2)=400_000+195*x1+225*x2
C (generic function with 1 method)

julia> R(x1,x2)=x1*p1(x1,x2)+x2*p2(x1,x2)
R (generic function with 1 method)

julia> P(x1,x2)=R(x1,x2)-C(x1,x2)
P (generic function with 1 method)

julia> substitute(P(x10,x20), a=>a0)
6920000//13

julia> D(f,x)=expand_derivatives(Differential(x)(f))
D (generic function with 1 method)

julia> pardxlda=expand(simplify(D(X[1],a)))
(-(831//612500) + (13//1225000)*lambda) / ((49//1000000000000) - (1//
12500000)*a + (1//30625)*(a^2))

julia> pardx1dlambda=expand(simplify(D(X[1],lambda)))
(13//1000) / ((49//1000000) - (1//25)*a)

julia> pardx1da=expand(simplify(D(X[1],a)))
(-(831//612500) + (13//1225000)*lambda) / ((49//1000000000000) - (1//
12500000)*a + (1//30625)*(a^2))
```

```
julia> simplify(expand(myLambda))
((143//6125) - (52//49)*a) / ((3//24500) + (2//49)*a)

julia> dlambda=D(simplify(expand(myLambda)),a)
((-2//49)*((143//6125) - (52//49)*a)) / (((3//24500) + (2//49)*a)^2)
+ (-52//49) / ((3//24500) + (2//49)*a)
```

And finally evaluating $S(x_1, a)$:

```
julia> Sx1a=substitute(a/x1*tmp1, [a=>a0, lambda=>lambda0, x1=>x10])
-10//13
```

We can repeat the same procedure to find $S(x_2, a)$, but that it left as an exercise to the reader as I'd prefer for my notes to not be too long.

Next, we find $S(P, a)$ (similar workflow in Julia of finding derivatives and substituting):

```
julia> substitute(SPa, [a=>a0, lambda=>lambda0, x1=>x10])
-625//2249
```

figure out where to add section 2, reformat and fix

2 Computational Methods for Optimization

Example. Reconsider the pig problem of Example 1.1, but now take into account the fact that the growth rate of the pig is not constant. Assume that the pig is young, so that the growth rate is increasing. When should we sell the pig for maximum profit?

Answer. We start with our equations from 1.1:

$$w(t) = 200 + 5t$$
$$C(t) = 0.45t$$
$$p(t) = 0.65 - 0.01t$$
$$R(t) = w(t)p(t)$$
$$P(t) = R(t) - C(t)$$

Since the rate of growth now increases with time, we rewrite the equation for weight as:

$$w(t) = w_0 e^{ct}$$

We want this to be consistent with 1.1, so we solve for w_0:

$$w(t) = 200 = w_0 e^{ct} = w_0$$

Then, we solve for c using the fact that the pig initially grows at 5 pounds per day:

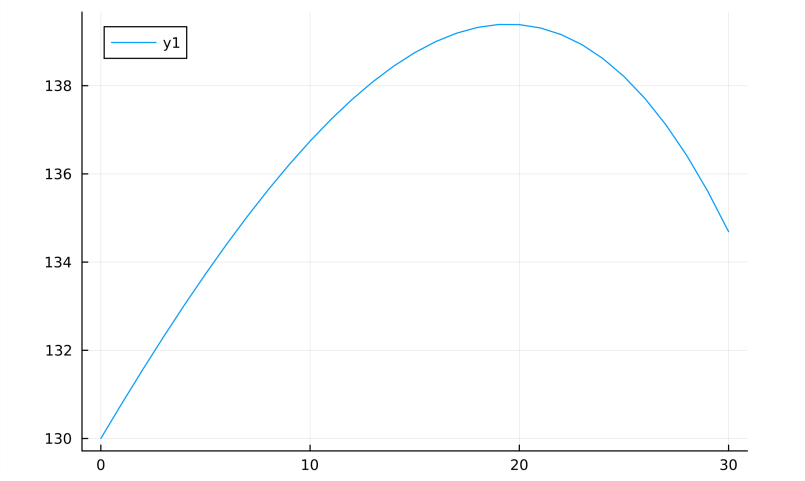
$$w'(0) = 5 = w_0 c e^{ct}|_{t=0} = 200c \tag{3}$$

$$\Rightarrow c = \frac{1}{40} \tag{4}$$

We can rewrite the rate of growth in differential form:

$$\frac{dw}{dt} = constant * w(t)$$

We then set this up in Julia and plot:



We start our search for the maximum by finding $\frac{dP}{dt}$:

```
julia> D(f,x)=expand_derivatives(Differential(x)(f))
D (generic function with 1 method)

julia> @variables t
1-element Vector{Num}:
 t

julia> dP=D(P(t),t)
-0.45 - 2.0exp(0.025t) + 5.0(0.65 - 0.01t)*exp(0.025t)

julia> expand(dP)
-0.45 + 1.25exp(0.025t) - 0.05t*exp(0.025t)
```

Since this isn't linear, we can't easily solve a system of equations to find the maximum. Instead, we use successive numerical approximations, called **Newton's method**:

ADD DRAWING AND EXPLANATION OF TANGENT LINE THING

```
julia> g(x)=x-fdP(x)/fd2p(x)
g (generic function with 1 method)

julia> g(t)
t + (0.45 - 1.25exp(0.025t) + 0.05t*exp(0.025t)) / (-0.01875exp(0.025t) - 0.00125t*exp(0.025t))

julia> g(20)
19.475684642955773
```

We repeat this several times, plugging the resulting back into $g(t)$, until it converges:

```
julia> g(g(20))
19.46816097315497

julia> g(g(g(20)))
19.46815944416128

julia> g(g(g(g(20))))
19.46815944416122

julia> g(g(g(g(g(20)))))
19.46815944416122
```

(ask how he saves old julia things) add section breaks, check textbook!!!!!!
install julia lab and jupyter
Maybe make notes for each lecture in separate documents and then combine? Figure out this weekend.

Definition 2.1. Sensitiv