Math 420/620: Quiz 2 Solutions

1. Forest scientists estimate the growth rates of competing hardwood and softwood trees can be modeled by

$$\frac{dH}{dt} = r_1 H\Big(1 - \frac{a_1 S + H}{K_1}\Big)$$
$$\frac{dS}{dt} = r_2 S\Big(1 - \frac{S + a_2 H}{K_2}\Big)$$

Here the variables H and S represent the biomass of hardwood and softwood in tons per acre; the parameters r_i represent the intrinsic growth rate of each type of tree; K_i represents the maximum sustainable biomass of each type of tree in the absence of competition; and a_i represents the effects of competition.

The parameters are estimated to be

i	1	2
r_i	0.10	0.25
K_i	10000	6000
a_i	0.5	0.5

The total rate of increase in biomass is $f(H,S) = dH/dt + dS/dt$.

(i) Create a contour plot showing level curves of $f(H,S)$ over the domain $H \in [0,4000]$ and $S \in [0,4000]$.

I used the Plots library in Julia to create the contour plot as follows:

```
$ julia
               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.10.10 (2025-06-27)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia> r1=0.10; r2=0.25;

julia> K1=10000.0; K2=6000.0;

julia> a1=0.5; a2=0.5;

julia> dHdt(H,S)=r1*H*(1-(a1*S+H)/K1)
dHdt (generic function with 1 method)

julia> dSdt(H,S)=r2*S*(1-(S+a2*H)/K2)
dSdt (generic function with 1 method)
```

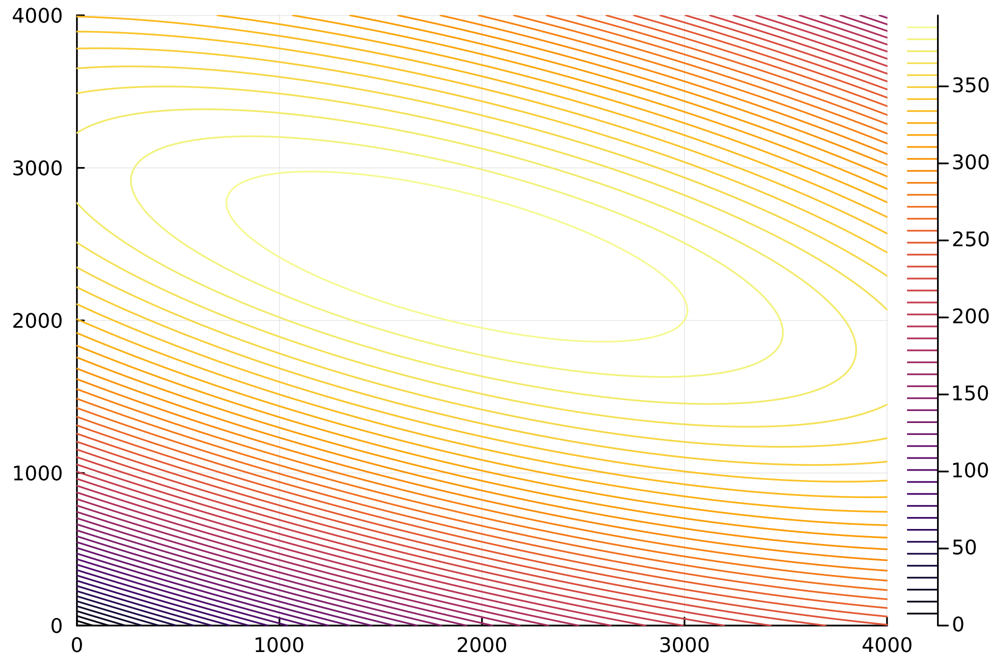
```
julia> f(H,S)=dHdt(H,S)+dSdt(H,S)
f (generic function with 1 method)

julia> using Plots

julia> P1=contour(0:10:4000,0:10:4000,f,levels=50)

julia> savefig(P1,"q2p1.pdf");
```

The resulting contour plot was



(ii) Determine the values of H and S that maximize f and verify the maximum is consistent with your contour plot. I used the Symbolics library in Julia to solve $\nabla f = 0$ as follows:

```
julia> using Symbolics

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

julia> @variables H,S
2-element Vector{Num}:
 H
 S

julia> dfdH=expand(D(f(H,S),H))
```

```
0.1 - 2.0e-5H - 2.5833333333333332e-5S

julia> dfdS=expand(D(f(H,S),S))
0.25 - 2.5833333333333332e-5H - 8.333333333333333e-5S

julia> Asym=-[D(dfdH,H) D(dfdH,S);D(dfdS,H) D(dfdS,S)]
2×2 Matrix{Num}:
     2.0e-5  2.58333e-5
 2.58333e-5  8.33333e-5

julia> bsym=[substitute(dfdH,[H=>0,S=>0]),
             substitute(dfdS,[H=>0,S=>0])]
2-element Vector{Num}:
  0.1
 0.25

julia> A=Symbolics.value.(Asym) # for efficiency
2×2 Matrix{Float64}:
 2.0e-5      2.58333e-5
 2.58333e-5  8.33333e-5

julia> b=Symbolics.value.(bsym)
2-element Vector{Float64}:
 0.1
 0.25

julia> X=A\b
2-element Vector{Float64}:
 1876.3029881862403
 2418.3460736622656
```

It follows that $H \approx 1876.3$ and $S \approx 2418.3$ are the optimal biomass levels.

(iii) Compute the sensitivities $S(H, r_1)$ and $S(H, r_2)$ of the optimal biomass levels H to the intrinsic growth rates r_1 and r_2.

By definition

$$S(H, r_1) = \frac{r_1}{H}\frac{dH}{dr_1}\Big|_{r_1=0.1}.$$

Compute this using Julia as

```
julia> @variables r1
1-element Vector{Num}:
 r1
```

```
julia> f(H,S) # check it looks right
0.25(1 + (-0.5H - S) / 6000.0)*S + (1 + (-H - 0.5S) / 10000.0)*H*r1

julia> dfdH=expand(D(f(H,S),H))
-2.0833333333333333e-5S + r1 - 0.0002H*r1 - 5.0e-5S*r1

julia> dfdS=expand(D(f(H,S),S))
0.25 - 2.0833333333333333e-5H - 8.333333333333333e-5S - 5.0e-5H*r1

julia> Asym=-[D(dfdH,H) D(dfdH,S);D(dfdS,H) D(dfdS,S)]
2×2 Matrix{Num}:
              0.0002r1  2.08333e-5 + 5.0e-5r1
 2.08333e-5 + 5.0e-5r1             8.33333e-5

julia> bsym=[substitute(dfdH,[H=>0,S=>0]),
             substitute(dfdS,[H=>0,S=>0])]
2-element Vector{Num}:
   r1
 0.25

julia> Xsym=Asym\bsym
2-element Vector{Num}:
 (r1 + (-(0.25 + (-2.0833333333333333e-5 - 5.0e-5r1) / 0.0002)*(2.08333
33333333333e-5 + 5.0e-5r1)) / (8.333333333333333e-5 + (-((2.08333333333
33333e-5 + 5.0e-5r1)^2)) / (0.0002r1))) / (0.0002r1)
(0.25 + (-2.0833333333333333e-5 - 5.0e-5r1) / 0.0002) / (8.333333333333
333e-5 + (-((2.0833333333333333e-5 + 5.0e-5r1)^2)) / (0.0002r1))

julia> dHdr1=D(Xsym[1],r1)
(-0.0002(r1 + (-(0.25 + (-2.0833333333333333e-5 - 5.0e-5r1) / 0.0002)*(
2.0833333333333333e-5 + 5.0e-5r1)) / (8.333333333333333e-5 + (-((2.0833
333333333333e-5 + 5.0e-5r1)^2)) / (0.0002r1)))) / (4.0e-8(r1^2)) + (1 +
 ((0.25 + (-2.0833333333333333e-5 - 5.0e-5r1) / 0.0002)*(2.083333333333
3333e-5 + 5.0e-5r1)*((0.0002((2.0833333333333333e-5 + 5.0e-5r1)^2)) / (
4.0e-8(r1^2)) + (-0.0001(2.0833333333333333e-5 + 5.0e-5r1)) / (0.0002r1
))) / ((8.333333333333333e-5 + (-((2.0833333333333333e-5 + 5.0e-5r1)^2)
) / (0.0002r1))^2) + (-5.0e-5(0.25 + (-2.0833333333333333e-5 - 5.0e-5r1
) / 0.0002) + 0.25(2.0833333333333333e-5 + 5.0e-5r1)) / (8.333333333333
333e-5 + (-((2.0833333333333333e-5 + 5.0e-5r1)^2)) / (0.0002r1))) / (0.
0002r1)

julia> SHr1=substitute(r1/X[1]*dHdr1,r1=>0.10)
2.3684657555401123
```

Thus $S(H, r_1) \approx 2.3685$.

Similarly,

```
julia> r1=0.10 # reset r1 back to the estimated value
0.1

julia> @variables r2
1-element Vector{Num}:
 r2

julia> dfdH=expand(D(f(H,S),H)) # recompute
0.1 - 2.0e-5H - 5.0e-6S - 8.333333333333333e-5S*r2

julia> dfdS=expand(D(f(H,S),S))
-5.0e-6H + r2 - 8.333333333333333e-5H*r2 - 0.0003333333333333333S*r2

julia> Asym=-[D(dfdH,H) D(dfdH,S);D(dfdS,H) D(dfdS,S)]
2×2 Matrix{Num}:
                2.0e-5  5.0e-6 + 8.33333e-5r2
 5.0e-6 + 8.33333e-5r2          0.000333333r2

julia> bsym=[substitute(dfdH,[H=>0,S=>0]),
             substitute(dfdS,[H=>0,S=>0])]
2-element Vector{Num}:
 0.1
  r2

julia> Xsym=Asym\bsym
2-element Vector{Num}:
 (0.1 + (-(5.0e-6 + 8.333333333333333e-5r2)*(-5000.0(5.0e-6 + 8.3333333
33333333e-5r2) + r2)) / (0.0003333333333333333r2 - 49999.99999999999((5
.0e-6 + 8.333333333333333e-5r2)^2))) / 2.0e-5
(-5000.0(5.0e-6 + 8.333333333333333e-5r2) + r2) / (0.000333333333333333
3r2 - 49999.99999999999((5.0e-6 + 8.333333333333333e-5r2)^2))

julia> dHdr2=D(Xsym[1],r2)
(((5.0e-6 + 8.333333333333333e-5r2)*(0.0003333333333333333 - 8.33333333
3333332(5.0e-6 + 8.333333333333333e-5r2))*(-5000.0(5.0e-6 + 8.333333333
333333e-5r2) + r2)) / ((0.0003333333333333333r2 - 49999.99999999999((5.
0e-6 + 8.333333333333333e-5r2)^2))^2) + (-0.5833333333333334(5.0e-6 + 8
.333333333333333e-5r2) - 8.333333333333333e-5(-5000.0(5.0e-6 + 8.333333
333333333e-5r2) + r2)) / (0.0003333333333333333r2 - 49999.99999999999((
5.0e-6 + 8.333333333333333e-5r2)^2))) / 2.0e-5
```

```
julia> SHr2=substitute(r2/X[1]*dHdr2,r2=>0.25)
-2.3684657555401114
```

shows that $S(H, r_2) \approx -2.3685$.

I found it interesting that $S(H, r_1) = -S(H, r_2)$. This may have something to do with the structure of the original system and the fact that r_i multiplies the entire left size in each of the equations. Or is it just luck?