General model definition and calculation of the invasion fitness

Here is a very simple model of an environmentally transmitted pathogen. I consider a population with two strains of pathogen (infected hosts are labeled I and I_m, depending on whether they are infected with the “resident” or “mutant,” respectively). Both susceptible and infected hosts give birth at the per-capita rate $r(N)$, where N is the level of nutrients. Susceptible hosts die at a background rate m and become infected due to contact with free-living pathogens at the rate βZ. Infected hosts die due to background causes (at the rate m) and die due to infection at the rate v, where virulence v is a function of both the parasite’s exploitation of the host ϵ and on nutrients, $v(\epsilon, N)$. Essentially we are assuming that virulence depends on pathogen abundance within the host, which should depend on both exploitation and nutrients, but this form is general enough to reflect any nutrient-dependent tolerance within the host. Thus, more nutrients could mean more parasites, or it could mean less per-parasite virulence. Hosts recover at the rate $\gamma(N)$, and reduced nutrient uptake could improve recovery (adaptive anorexia) or impair it. Infected hosts shed free-living pathogens into the environment at the rate $\lambda(v(\epsilon, N))$. This functional form suggests that shedding depends on virulence (i.e., on the number of parasites within the host). Free-living pathogens die at the background rate δ and are removed from the environment by contact with susceptible hosts.

We assume that there are two strains of parasite that differ in their exploitation rates (ϵ for the resident and ϵ_m for the mutant). This leads to the following model:

```
dS = r[N] (S + Q) (1 - (S + Q)/k) - m S - β S Z + γ[N] Q - β S Zm;
dQ = β S Z - (m + v[ϵ, N] + γ[N]) Q;
dZ = λ[v[ϵ, N]] Q - β S Z - δ Z;
dQm = β S Zm - (m + v[ϵm, N] + γ[N]) Qm;
dZm = λ[ϵm, N] Qm - β S Zm - δ Zm;
```

To find the invasion fitness, we must first define the Jacobian for this system:

```
J = Simplify[{{D[dS, S], D[dS, Q], D[dS, Z], D[dS, Qm], D[dS, Qm]},
    {D[dQ, S], D[dQ, Q], D[dQ, Z], D[dQ, Qm], D[dQ, Zm]},
    {D[dZ, S], D[dZ, Q], D[dZ, Z], D[dZ, Qm], D[dZ, Zm]},
    {D[dQm, S], D[dQm, Q], D[dQm, Z], D[dQm, Qm], D[dQm, Zm]},
    {D[dZm, S], D[dZm, Q], D[dZm, Z], D[dZm, Qm], D[dZm, Zm]}}];
```

The invasion fitness is calculated from the Jacobian evaluated at the mutant-free equilibrium:

```
MatrixForm[J /. {Qm -> 0, Zm → 0}]
```

$$\begin{pmatrix} -m - Z\beta + r[N] - \frac{2(Q+S)\,r[N]}{k} & r[N] - \frac{2(Q+S)\,r[N]}{k} + \gamma[N] & -S\beta & 0 & 0 \\ Z\beta & -m - v[\epsilon, N] - \gamma[N] & S\beta & 0 & 0 \\ -Z\beta & \lambda[v[\epsilon, N]] & -S\beta - \delta & 0 & 0 \\ 0 & 0 & 0 & -m - v[\epsilon m, N] - \gamma[N] & S\beta \\ 0 & 0 & 0 & \lambda[\epsilon m, N] & -S\beta - \delta \end{pmatrix}$$

Because this matrix is block upper-triangular, its eigenvalues are given by the eigenvalues of the 3x3 matrix in the upper-left (which determines the stability of the system excluding the

mutant parasite, so we assume these eigenvalues are negative) and the eigenvalues of the bottom-right submatrix. The mutant can invade if any of the eigenvalues of this matrix are positive. Rather than calculate these eigenvalues directly, we make use of the next-generation theorem to determine the system stability.

```
F = {{0, S β}, {λ[ϵm, N], 0}};
V = {{m + v[ϵm, N] + γ[N], 0}, {0, S β + δ}};
(J[[4 ;; 5, 4 ;; 5]] /. {Qm → 0, Zm → 0}) == F - V
(* check to ensure that F-V is identical to the bottom-right submatrix *)
(* Check that the assumptions of the NGT are satisfied *)
Eigenvalues[-V] (* must be negative *)
Inverse[V] (* must be nonnegative *)
```

```
True
```

$$\{-S\,\beta-\delta,\ -m-v[\epsilon m,\ N]-\gamma[N]\}$$

$$\left\{\left\{\frac{1}{m+v[\epsilon m,\ N]+\gamma[N]},\ 0\right\},\ \left\{0,\ \frac{1}{S\,\beta+\delta}\right\}\right\}$$

The mutant can invade if the spectral radius of the matrix F.V^{-1} is greater than one.

```
Eigenvalues[F.Inverse[V]]
```

$$\left\{-\frac{\sqrt{S}\ \sqrt{\beta}\ \sqrt{\lambda[\epsilon m,\ N]}}{\sqrt{S\,\beta+\delta}\ \sqrt{m+v[\epsilon m,\ N]+\gamma[N]}},\ \frac{\sqrt{S}\ \sqrt{\beta}\ \sqrt{\lambda[\epsilon m,\ N]}}{\sqrt{S\,\beta+\delta}\ \sqrt{m+v[\epsilon m,\ N]+\gamma[N]}}\right\}$$

Thus the invasion fitness is

```
Rmut = Eigenvalues[F.Inverse[V]][[2]]^2
```

$$\frac{S\,\beta\,\lambda[\epsilon m,\ N]}{(S\,\beta+\delta)\ (m+v[\epsilon m,\ N]+\gamma[N])}$$

Evolutionary analysis of the general model

$$\text{Rmut} = \frac{S\,\beta\,\lambda[\epsilon m,\ N]}{(S\,\beta+\delta)\ (m+v[\epsilon m,\ N]+\gamma[N])}$$

$$\frac{S\,\beta\,\lambda[\epsilon m,\ N]}{(S\,\beta+\delta)\ (m+v[\epsilon m,\ N]+\gamma[N])}$$

The invasion fitness can be written as $\frac{\beta S^*}{\beta S^*+\delta}\,\lambda(\epsilon_m, N)\,\frac{1}{m+v(\epsilon_m,N)+\gamma(N)}$, which has the biological interpretation of the probability that a free-living infectious stage encounters a host times the shedding rate during an infection times the expected duration of infection.

```
InfectProb Shedding[ϵm, N] InfectDuration[ϵm, N]
InfectProb InfectDuration[ϵm, N] Shedding[ϵm, N]
```

Thus the fitness gradient can be written in the following way. Since the change in infection duration with exploitation is necessarily negative, it is clear that this entire expression will be equal to zero only when

$\frac{1}{\text{InfectDuration}}\,\text{InfectDuration}^{(1,0)}[\epsilon m, N] = \frac{1}{\text{Shedding}}\,\text{Shedding}^{(1,0)}[\epsilon m, N]$. **In other words, any zero of the fitness gradient (which defines a potential endpoint of evolution) will satisfy that, at this exploitation rate ϵ_m, the relative change in infection duration with a change in exploitation is exactly the same as the relative change in shedding with a change in exploitation. In other words, at this point, any relative gain in shedding with an increase in exploitation will be exactly matched by the relative decrease in infection duration.**

```
D[InfectProb Shedding[ϵm, N] InfectDuration[ϵm, N], ϵm]
```

$$\text{InfectProb}\,\text{Shedding}[\epsilon m, N]\,\text{InfectDuration}^{(1,0)}[\epsilon m, N] + \text{InfectProb}\,\text{InfectDuration}[\epsilon m, N]\,\text{Shedding}^{(1,0)}[\epsilon m, N]$$

Working with the general model, the fitness gradient is:

```
D[Rmut, ϵm] // Simplify
```

$$\left(S\,\beta\,\left(-\lambda[\epsilon m, N]\,v^{(1,0)}[\epsilon m, N] + (m + v[\epsilon m, N] + \gamma[N])\,\lambda^{(1,0)}[\epsilon m, N]\right)\right) / \left(\left(S\,\beta + \delta\right)\,(m + v[\epsilon m, N] + \gamma[N])^2\right)$$

Any zero of the fitness gradient will satisfy $\left(-\lambda[\epsilon m, N]\,v^{(1,0)}[\epsilon m, N] + (m + v[\epsilon m, N] + \gamma[N])\,\lambda^{(1,0)}[\epsilon m, N]\right) = 0$. **Does such a point represent a fitness maximum? To answer that question, we need to determine the sign of the second derivative of the invasion fitness with respect to the evolving trait, when the trait is at the value ϵ^* that causes the fitness gradient to vanish. Mathematically, the singular ϵ^* will satisfy:**

$$\text{Solve}\left[\left(\text{D}[\text{Rmut}, \epsilon m]\ /.\ \epsilon m \to \epsilon^*\right) == 0,\ \lambda^{(1,0)}[\epsilon^*, N]\right]$$

$$\left\{\left\{\lambda^{(1,0)}[\epsilon^*, N] \to \frac{\lambda[\epsilon^*, N]\,v^{(1,0)}[\epsilon^*, N]}{m + v[\epsilon^*, N] + \gamma[N]}\right\}\right\}$$

We can plug that condition into the second derivative expression to determine what is required for the singular ϵ^* to be a fitness maximum.

$$\text{Simplify}\left[\left(\text{D}[\text{Rmut}, \{\epsilon m, 2\}]\ /.\ \epsilon m \to \epsilon^*\right)\ /.\ \left\{\lambda^{(1,0)}[\epsilon^*, N] \to \frac{\lambda[\epsilon^*, N]\,v^{(1,0)}[\epsilon^*, N]}{m + v[\epsilon^*, N] + \gamma[N]}\right\}\right]$$

$$\left(S\,\beta\,\left(-\lambda[\epsilon^*, N]\,v^{(2,0)}[\epsilon^*, N] + (m + v[\epsilon^*, N] + \gamma[N])\,\lambda^{(2,0)}[\epsilon^*, N]\right)\right) / \left(\left(S\,\beta + \delta\right)\,(m + v[\epsilon^*, N] + \gamma[N])^2\right)$$

The singular ϵ^* will be a fitness maximum (and thus a true evolutionarily stable strategy) if $\frac{\partial^2 v}{\partial \epsilon^2} \geq 0$ and $\frac{\partial^2 \lambda}{\partial \epsilon^2} < 0$, in other words if shedding is a saturating function of exploitation but virulence is either a linear or increasing function of exploitation. This is a classic result in evolutionary epidemiology theory: regardless of which trait is evolving, its benefits must increase at a slower rate than the costs for there to be an intermediate singular strategy that is evolutionarily stable.

To understand how changing nutrients will affect the ES exploitation value, let's recall that when $\epsilon_m = \epsilon^*$, the following is true (the singular strategy condition):

$$\left(\left(-\lambda[\epsilon m, N]\,v^{(1,0)}[\epsilon m, N] + (m + v[\epsilon m, N] + \gamma[N])\,\lambda^{(1,0)}[\epsilon m, N]\right)\ /.\ \epsilon m \to \epsilon^*\right) == 0$$

$$-\lambda[\epsilon^*, N]\,v^{(1,0)}[\epsilon^*, N] + (m + v[\epsilon^*, N] + \gamma[N])\,\lambda^{(1,0)}[\epsilon^*, N] == 0$$

Implicitly, the value of ϵ^* that satisfies that expression is itself an expression of *N.* Let's make that implicit dependence explicit:

```
-λ[ϵ*, N] v^(1,0)[ϵ*, N] + (m + v[ϵ*, N] + γ[N]) λ^(1,0)[ϵ*, N] == 0 /. ϵ* → ϵ*[N]
-λ[ϵ*[N], N] v^(1,0)[ϵ*[N], N] + (m + v[ϵ*[N], N] + γ[N]) λ^(1,0)[ϵ*[N], N] == 0
```

We can take the derivative of this expression with respect to *N* to determine how the ES exploitation rate will change as a function of *N*...

```
D[-λ[ϵ*[N], N] v^(1,0)[ϵ*[N], N] + (m + v[ϵ*[N], N] + γ[N]) λ^(1,0)[ϵ*[N], N] == 0, N]
(γ′[N] + v^(0,1)[ϵ*[N], N] + (ϵ*)′[N] v^(1,0)[ϵ*[N], N]) λ^(1,0)[ϵ*[N], N] -
  v^(1,0)[ϵ*[N], N] (λ^(0,1)[ϵ*[N], N] + (ϵ*)′[N] λ^(1,0)[ϵ*[N], N]) -
  λ[ϵ*[N], N] (v^(1,1)[ϵ*[N], N] + (ϵ*)′[N] v^(2,0)[ϵ*[N], N]) +
  (m + v[ϵ*[N], N] + γ[N]) (λ^(1,1)[ϵ*[N], N] + (ϵ*)′[N] λ^(2,0)[ϵ*[N], N]) == 0
```

...and then solving for *∂ϵ*/∂N*

```
Solve[D[-λ[ϵ*[N], N] v^(1,0)[ϵ*[N], N] + (m + v[ϵ*[N], N] + γ[N]) λ^(1,0)[ϵ*[N], N] == 0, N],
  (ϵ*)′[N]] // FullSimplify
{{(ϵ*)′[N] → (-λ^(0,1)[ϵ*[N], N] v^(1,0)[ϵ*[N], N] + (γ′[N] + v^(0,1)[ϵ*[N], N]) λ^(1,0)[ϵ*[N], N] -
        λ[ϵ*[N], N] v^(1,1)[ϵ*[N], N] + (m + v[ϵ*[N], N] + γ[N]) λ^(1,1)[ϵ*[N], N]) /
     (λ[ϵ*[N], N] v^(2,0)[ϵ*[N], N] - (m + v[ϵ*[N], N] + γ[N]) λ^(2,0)[ϵ*[N], N])}}
```

The denominator is the inverse of the evolutionary stability expression, and so is positive. The numerator thus determines whether increasing nutrients increases or decreases exploitation. From this expression and our sure knowledge of the signs of some of the derivatives (*λ*’>0, *λ*’’ < 0, $\frac{\partial v}{\partial \epsilon}$ > 0), we can make some statements about the conditions under which ES exploitation will increase or decrease.

```
-λ^(0,1)[ϵ*[N], N] v^(1,0)[ϵ*[N], N] + (γ′[N] + v^(0,1)[ϵ*[N], N]) λ^(1,0)[ϵ*[N], N] -
 λ[ϵ*[N], N] v^(1,1)[ϵ*[N], N] + (m + v[ϵ*[N], N] + γ[N]) λ^(1,1)[ϵ*[N], N]
(* At the ESS the following must be true: λ^(1,0)[ϵ*,N] / v^(1,0)[ϵ*,N] = λ[ϵ*,N] / (m+v[ϵ*,N]+γ[N]) *)
(-λ^(0,1)[ϵ*[N], N] v^(1,0)[ϵ*[N], N] + (γ′[N] + v^(0,1)[ϵ*[N], N]) λ^(1,0)[ϵ*[N], N] -

     λ[ϵ*[N], N] v^(1,1)[ϵ*[N], N] + (m + v[ϵ*[N], N] + γ[N]) λ^(1,1)[ϵ*[N], N] /.
    λ[ϵ*[N], N] -> λ^(1,0)[ϵ*[N], N] (m + v[ϵ*[N], N] + γ[N]) / v^(1,0)[ϵ*[N], N]) ==

  (-λ^(0,1)[ϵ*[N], N] v^(1,0)[ϵ*[N], N] + (γ′[N] + v^(0,1)[ϵ*[N], N]) λ^(1,0)[ϵ*[N], N] -

     (m + v[ϵ*[N], N] + γ[N])
      (λ^(1,0)[ϵ*[N], N] v^(1,1)[ϵ*[N], N] / v^(1,0)[ϵ*[N], N] - λ^(1,1)[ϵ*[N], N])) // Simplify
-λ^(0,1)[ϵ*[N], N] v^(1,0)[ϵ*[N], N] + (γ′[N] + v^(0,1)[ϵ*[N], N]) λ^(1,0)[ϵ*[N], N] -
 λ[ϵ*[N], N] v^(1,1)[ϵ*[N], N] + (m + v[ϵ*[N], N] + γ[N]) λ^(1,1)[ϵ*[N], N]

True
```

$$\left(-\lambda^{(0,1)}[\epsilon^*[N],N]\;v^{(1,0)}[\epsilon^*[N],N]+\left(\gamma'[N]+v^{(0,1)}[\epsilon^*[N],N]\right)\lambda^{(1,0)}[\epsilon^*[N],N]-(m+v[\epsilon^*[N],N]+\gamma[N])\left(\frac{\lambda^{(1,0)}[\epsilon^*[N],N]\;v^{(1,1)}[\epsilon^*[N],N]}{v^{(1,0)}[\epsilon^*[N],N]}-\lambda^{(1,1)}[\epsilon^*[N],N]\right)\right)==\left(\gamma'[N]+v^{(0,1)}[\epsilon^*[N],N]\right)v^{(1,0)}[\epsilon^*[N],N]\left(\left(\frac{\lambda^{(1,0)}[\epsilon^*[N],N]}{v^{(1,0)}[\epsilon^*[N],N]}-\frac{\lambda^{(0,1)}[\epsilon^*[N],N]}{\left(\gamma'[N]+v^{(0,1)}[\epsilon^*[N],N]\right)}\right)-\frac{(m+v[\epsilon^*[N],N]+\gamma[N])\;v^{(1,1)}[\epsilon^*[N],N]}{\left(\gamma'[N]+v^{(0,1)}[\epsilon^*[N],N]\right)v^{(1,0)}[\epsilon^*[N],N]}\left(\frac{\lambda^{(1,0)}[\epsilon^*[N],N]}{v^{(1,0)}[\epsilon^*[N],N]}-\frac{\lambda^{(1,1)}[\epsilon^*[N],N]}{v^{(1,1)}[\epsilon^*[N],N]}\right)\right)\;\text{// Simplify}$$

True

$$D[D[1/(m+v[\epsilon,N]+\gamma[N]),\epsilon],N]\;\text{// Simplify}$$

$$\left(2\left(\gamma'[N]+v^{(0,1)}[\epsilon,N]\right)v^{(1,0)}[\epsilon,N]-(m+v[\epsilon,N]+\gamma[N])\;v^{(1,1)}[\epsilon,N]\right)\Big/(m+v[\epsilon,N]+\gamma[N])^3$$

$$2\left(\gamma'[N]+v^{(0,1)}[\epsilon,N]\right)v^{(1,0)}[\epsilon,N]-(m+v[\epsilon,N]+\gamma[N])\;v^{(1,1)}[\epsilon,N]>0;$$

$$2\;>\;\frac{(m+v[\epsilon,N]+\gamma[N])\;v^{(1,1)}[\epsilon,N]}{\left(\gamma'[N]+v^{(0,1)}[\epsilon,N]\right)v^{(1,0)}[\epsilon,N]}$$

We can get some further insights into this complicated expression by plugging in our functions.

True

$$\left(\left(\frac{\lambda^{(1,0)}[\epsilon^*[N],N]}{v^{(1,0)}[\epsilon^*[N],N]}-\frac{\lambda^{(0,1)}[\epsilon^*[N],N]}{\left(\gamma'[N]+v^{(0,1)}[\epsilon^*[N],N]\right)}\right)-\frac{(m+v[\epsilon^*[N],N]+\gamma[N])\;v^{(1,1)}[\epsilon^*[N],N]}{\left(\gamma'[N]+v^{(0,1)}[\epsilon^*[N],N]\right)v^{(1,0)}[\epsilon^*[N],N]}\left(\frac{\lambda^{(1,0)}[\epsilon^*[N],N]}{v^{(1,0)}[\epsilon^*[N],N]}-\frac{\lambda^{(1,1)}[\epsilon^*[N],N]}{v^{(1,1)}[\epsilon^*[N],N]}\right)\right)\;/.\;\left\{\lambda^{(1,0)}[\epsilon^*[N],N]\to D\left[\frac{\lambda 0\;\epsilon^*\;N}{1+\epsilon^*\;N},\epsilon^*\right],\;\lambda^{(0,1)}[\epsilon^*[N],N]\;\text{->}\;D\left[\frac{\lambda 0\;\epsilon^*\;N}{1+\epsilon^*\;N},N\right],\;\lambda^{(1,1)}[\epsilon^*[N],N]\to D\left[D\left[\frac{\lambda 0\;\epsilon^*\;N}{1+\epsilon^*\;N},N\right],\epsilon^*\right]\right\}\;/.\;\left\{v^{(0,1)}[\epsilon^*[N],N]\to v0\;\epsilon^*,\;v^{(1,0)}[\epsilon^*[N],N]\to v0\;N,\;v^{(1,1)}[\epsilon^*[N],N]\to v0\right\}\;\text{// Simplify}$$

$$\frac{\lambda 0\left(\gamma'[N]+\epsilon^*\left(-2\,m-2\,v[\epsilon^*[N],N]-2\,\gamma[N]+N\,\gamma'[N]\right)\right)}{v0\;(1+N\,\epsilon^*)^3\left(v0\;\epsilon^*+\gamma'[N]\right)}$$

Epidemiological analysis of the general model

One thing that is interesting to look at, epidemiologically, is how nutrients will affect infection prevalence. The epidemiological system is:

$$dS=r[N]\;(S+Q)\left(1-\frac{S+Q}{k[N]}\right)-m\,S-\beta\,S\,Z+\gamma[N]\;Q;$$

$$dQ=\beta\,S\,Z-(m+v[\epsilon,N]+\gamma[N])\;Q;$$

$$dZ=\lambda[\epsilon,N]\;Q-\beta\,S\,Z-\delta\,Z;$$

At equilibrium, the infection prevalence will be

```
Eq = Solve[{dQ == 0, dZ == 0}, {S, Q}]
```

$$\text{Simplify}\left[\frac{\text{Eq[[1, 2, 2]]}}{\text{Eq[[1, 1, 2]]} + \text{Eq[[1, 2, 2]]}}\right]$$

$$\left\{\left\{S \to -\frac{\delta\,(m + v[\epsilon, N] + \gamma[N])}{\beta\,(m + v[\epsilon, N] + \gamma[N] - \lambda[\epsilon, N])},\ Q \to \frac{Z\,\delta}{-m - v[\epsilon, N] - \gamma[N] + \lambda[\epsilon, N]}\right\}\right\}$$

$$\frac{Z\,\beta}{m + Z\,\beta + v[\epsilon, N] + \gamma[N]}$$

Of course, the equilibrium abundance of free-living parasites, *Z*, will depend on nutrients as well. Thus the change in infection prevalence with nutrients will be given by

$$\text{D}\left[\frac{\text{Z[N]}\,\beta}{\text{m} + \text{Z[N]}\,\beta + \text{v}[\epsilon, \text{N}] + \gamma[\text{N}]}, \text{N}\right] \text{ // Simplify}$$

$$\frac{\beta\,\left((m + v[\epsilon, N] + \gamma[N])\,Z'[N] - Z[N]\,\left(\gamma'[N] + v^{(0,1)}[\epsilon, N]\right)\right)}{\left(m + v[\epsilon, N] + \beta\,Z[N] + \gamma[N]\right)^2}$$

Evolutionary analysis of specific cases

It is clear from the above work that changing how virulence varies with exploitation and nutrients, or changing how nutrients varies with nutrients, can both affect how the ES exploitation changes with nutrients. For simplicity, we will hold the relationship between virulence and exploitation and nutrients constant and vary only how nutrients affect recovery.

In particular, we will assume that $v(\epsilon, N) = \epsilon N$: under such a model, we are essentially assuming that parasite abundance increases with both exploitation and nutrients in a linear way. We will also assume that shedding rate depends on virulence according to $\lambda(v(\epsilon, N)) = \frac{\lambda_0\,\epsilon N}{h + \epsilon N}$.

We can plug this into the invasion fitness expression:

```
Rmut /. {v[ϵm, N] → ϵm N} /. λ[ϵm, N] → λ0 ϵm N/(h + ϵm N)
```

$$\frac{S\,\beta\,\epsilon m\,\lambda 0\,N}{(S\,\beta + \delta)\,(h + \epsilon m\,N)\,(m + \epsilon m\,N + \gamma[N])}$$

```
D[D[λ0 ϵm N/(h + ϵm N), ϵm], N] // Simplify
```

$$\frac{h\,\lambda 0\,(h - \epsilon m\,N)}{(h + \epsilon m\,N)^3}$$

And we can take the derivative with respect to the evolving trait, set equal to ϵ, and solve to find the ES exploitation:

```
Solve[
 (D[Rmut /. {v[ϵm, N] → ϵm N} /. λ[ϵm, N] → λ0 ϵm N/(h + ϵm N), ϵm] /. {ϵm → ϵ}) == 0, ϵ]
```

$$\left\{\left\{\epsilon \to -\frac{\sqrt{h}\,\sqrt{m + \gamma[N]}}{N}\right\}, \left\{\epsilon \to \frac{\sqrt{h}\,\sqrt{m + \gamma[N]}}{N}\right\}\right\}$$

Thus, the ES exploitation is:

```
ESϵ =
 Solve[(D[Rmut /. {v[ϵm, N] → ϵm N} /. λ[ϵm, N] → λ0 ϵm N / (h + ϵm N), ϵm] /. {ϵm → ϵ}) == 0,
   ϵ][[2, 1, 2]]
```

$$\frac{\sqrt{h}\,\sqrt{m+\gamma[N]}}{N}$$

How nutrients will affect the ES ϵ value is given by the following expression:

```
D[ESϵ, N] // Simplify
```

$$\frac{\sqrt{h}\,\left(-2\,m-2\,\gamma[N]+N\,\gamma'[N]\right)}{2\,N^{2}\,\sqrt{m+\gamma[N]}}$$

Thus, if increasing nutrients do not strongly increase recovery ($\gamma'(N) \leq 0$ or even $\gamma'(N) = c$ (recovery is linear function of nutrients) then increasing nutrients should decrease the ES exploitation rate. We can confirm these semi-analytical results with numerical experiments.

Case 1: recovery is a linear function of nutrients $\gamma(N) = \gamma_0 N$

If recovery is a linear function of nutrients, the ES ϵ is:

```
ESϵ /. {γ[N] → γ0 N}
```

$$\frac{\sqrt{h}\,\sqrt{m+\gamma0\,N}}{N}$$

As nutrients change, this ES ϵ changes as well, and as expected, the ES ϵ decreases with an increase in nutrients.

```
D[ESϵ /. {γ[N] → γ0 N}, N] // Simplify
```

$$-\frac{\sqrt{h}\,\left(2\,m+\gamma0\,N\right)}{2\,N^{2}\,\sqrt{m+\gamma0\,N}}$$

Case 2: recovery is an accelerating function of nutrients $\gamma(N) = \gamma_0 N^3$

If recovery is a accelerating function of nutrients, the ES ϵ is:

```
ESϵ /. {γ[N] → γ0 N^3}
```

$$\frac{\sqrt{h}\,\sqrt{m+\gamma0\,N^{3}}}{N}$$

As nutrients change, this ES ϵ changes as well; here the ES ϵ can either increase or decrease with nutrients, depending on the values of m, γ_0, and N.

$$\text{D}\left[\text{ES}\epsilon\ \text{/. }\left\{\gamma[\text{N}]\to\gamma0\ \text{N}^3\right\},\ \text{N}\right]\ \text{// Simplify}$$

$$\frac{\sqrt{h}\ \left(-2\ m+\gamma0\ N^3\right)}{2\ N^2\ \sqrt{m+\gamma0\ N^3}}$$

Case 3: recovery is a decreasing function of nutrients $\gamma(\mathbf{N}) = \frac{\gamma_0\,\mathbf{N}}{\eta+\mathbf{N}^3}$

If recovery is a decreasing function of nutrients, the ES ϵ is:

$$\text{ES}\epsilon\ \text{/. }\left\{\gamma[\text{N}]\to\frac{\gamma_0\ \text{N}}{\eta+\text{N}^3}\right\}$$

$$\frac{\sqrt{h}\ \sqrt{m+\frac{N\,\gamma_0}{\eta+N^3}}}{N}$$

As nutrients change, this ES ϵ changes as well, and as expected, the ES ϵ decreases with an increase in nutrients.

$$\text{D}\left[\text{ES}\epsilon\ \text{/. }\left\{\gamma[\text{N}]\to\frac{\gamma_0\ \text{N}}{\eta+\text{N}^3}\right\},\ \text{N}\right]\ \text{// Simplify}$$

$$-\frac{\sqrt{h}\ \left(2\ m\ \left(\eta+N^3\right)^2+N\ \left(\eta+4\ N^3\right)\ \gamma_0\right)}{2\ N^2\ \left(\eta+N^3\right)^2\ \sqrt{m+\frac{N\,\gamma_0}{\eta+N^3}}}$$

Epidemiological analysis of specific cases

From the above analysis, we know that the change in infection prevalence with nutrients will be given by the sign of:

$$\frac{\beta\ \left(\left(m+v[\epsilon,\ N]+\gamma[N]\right)\ Z'[N]-Z[N]\ \left(\gamma'[N]+v^{(0,1)}[\epsilon,\ N]\right)\right)}{\left(m+v[\epsilon,\ N]+\beta\ Z[N]+\gamma[N]\right)^2}$$

$$\frac{\beta\ \left(\left(m+v[\epsilon,\ N]+\gamma[N]\right)\ Z'[N]-Z[N]\ \left(\gamma'[N]+v^{(0,1)}[\epsilon,\ N]\right)\right)}{\left(m+v[\epsilon,\ N]+\beta\ Z[N]+\gamma[N]\right)^2}$$

For the functional forms specific in the preceding section, we know that $v^{(0,1)}[\epsilon,\ N] = \epsilon$ and that, at the ES exploitation,

$$\left(\left(m+v[\epsilon,\ N]+\gamma[N]\right)\ Z'[N]-Z[N]\ \left(\gamma'[N]+v^{(0,1)}[\epsilon,\ N]\right)\right)\ \text{/.}$$
$$\left\{v[\epsilon,\ N]\to\epsilon\ N\ ,\ v^{(0,1)}[\epsilon,\ N]\to\epsilon\right\}\ \text{/. }\epsilon\to\text{ES}\epsilon$$

$$\left(m+\gamma[N]+\sqrt{h}\ \sqrt{m+\gamma[N]}\right)\ Z'[N]-Z[N]\ \left(\frac{\sqrt{h}\ \sqrt{m+\gamma[N]}}{N}+\gamma'[N]\right)$$

Numerical experiments

Evolutionary consequences of altered nutrients

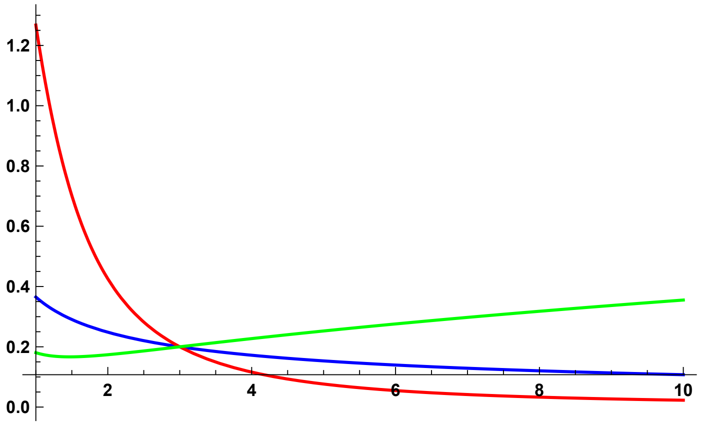
Here we show how changing nutrients affects the ES exploitation as parameters change. We have chosen the parameter such that, at *N* = 5, the evolutionarily stable production is the same across the 3 recovery functions. We can then explore what happens with anorexia (*N* < 5) or overfeeding (*N* > 5).

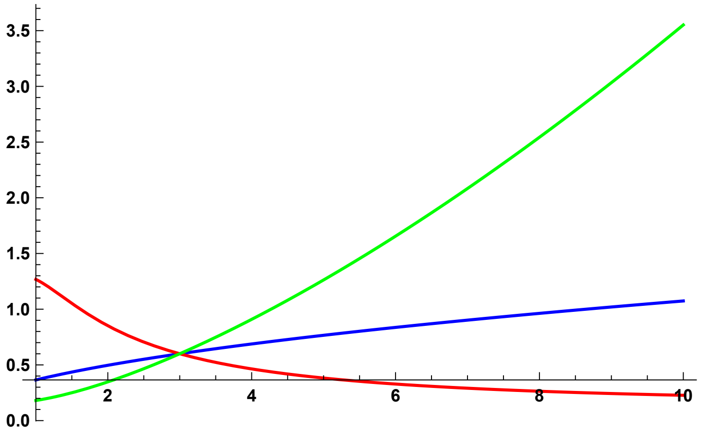
$$\text{ES}\epsilon = \frac{\sqrt{h}\ \sqrt{m+\gamma[N]}}{N}$$

$$\frac{\sqrt{h}\ \sqrt{m+\gamma[N]}}{N}$$

```
gEqual1 = Solve[(ESϵ /. {γ[N] → γ0 N} /. {h → 1, m → 0.02, N → 3}) == 0.2, γ0][[1]]
gEqual2 = Solve[(ESϵ /. {γ[N] → γ0 N^3} /. {h → 1, m → 0.02, N → 3}) == 0.2, γ0][[1]]
gEqual3 =
 Solve[(ESϵ /. {γ[N] → (γ0 N)/(η + N^3)} /. {h → 1, m → 0.02, η → 1} /. N → 3) == 0.2, γ0][[1]]
Show[Plot[ESϵ /. {γ[N] → γ0 N} /. {h → 1, m → 0.02} /. gEqual1,
  {N, 1, 10}, PlotStyle → {Blue}, PlotRange → All],
 Plot[ESϵ /. {γ[N] → (γ0 N)/(η + N^3)} /. {h → 1, m → 0.02, η → 1} /. gEqual3,
  {N, 1, 10}, PlotStyle → {Red}, PlotRange → All],
 Plot[ESϵ /. {γ[N] → γ0 N^3} /. {h → 1, m → 0.02} /. gEqual2,
  {N, 1, 10}, PlotStyle → {Green}, PlotRange → All]]
Show[Plot[ESϵ N /. {γ[N] → γ0 N} /. {h → 1, m → 0.02} /. gEqual1,
  {N, 1, 10}, PlotStyle → {Blue}, PlotRange → All],
 Plot[ESϵ N /. {γ[N] → (γ0 N)/(η + N^3)} /. {h → 1, m → 0.02, η → 1} /. gEqual3,
  {N, 1, 10}, PlotStyle → {Red}, PlotRange → All],
 Plot[ESϵ N /. {γ[N] → γ0 N^3} /. {h → 1, m → 0.02} /. gEqual2,
  {N, 1, 10}, PlotStyle → {Green}, PlotRange → All]]
```

```
{γ0 → 0.113333}

{γ0 → 0.0125926}

{γ0 → 3.17333}
```





```
ES1 = Table[ESϵ /. {γ[N] → γ0 N} /. {h → 1, m → 0.02} /. gEqual1, {N, 1, 10, 0.01}];
ES2 =
  Table[ESϵ /. {γ[N] → γ0 N/(η + N^3)} /. {h → 1, m → 0.02, η → 1} /. gEqual3, {N, 1, 10, 0.01}];
ES3 = Table[ESϵ /. {γ[N] → γ0 N^3} /. {h → 1, m → 0.02} /. gEqual2, {N, 1, 10, 0.01}];

Export["~/Box Sync/Anorexia_Parasite_Evolution/Plotting/ES_prod_eq5.csv", ES1]
Export["~/Box Sync/Anorexia_Parasite_Evolution/Plotting/ES_prod_eq7.csv", ES2]
Export["~/Box Sync/Anorexia_Parasite_Evolution/Plotting/ES_prod_eq6.csv", ES3]
```

```
~/Box Sync/Anorexia_Parasite_Evolution/Plotting/ES_prod_eq5.csv

~/Box Sync/Anorexia_Parasite_Evolution/Plotting/ES_prod_eq7.csv

~/Box Sync/Anorexia_Parasite_Evolution/Plotting/ES_prod_eq6.csv
```

Epidemiological consequences

There are a couple of ways to explore this. First, we can ask what the infection prevalence is when the parasite is at the evolutionarily stable strategy.

To get around the problem of needing to figure out, each time, which equilibrium to choose, simply simulate the system and compute the infection prevalence numerically. Note that, right now, you have to manually change the recovery function - I'm sure there is some slick way to do this by passing the function directly.

```
epiDynES1[{Nvalue_?NumberQ}] :=
  Module[{params, δ, r, k, m, λ0, γ0, h, v, β, η, N, dS, S, Q, dQ, Z, dZ, λ, ϵ, γ, Soln},
   params = {δ → 0.8, r → 0.5, k → 1.5, m → 0.02, λ0 → 100,
     γ0 → 0.113333, h → 1, v → 0.05, β → 2, η → 1, N → Nvalue};
   dS = r (S + Q) (1 - (S + Q)/k) - m S - β S Z + γ[N] Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → √h √(m + γ[N])/N /. γ[N] → γ0 N /. params;
   dQ = β S Z - (m + v[ϵ, N] + γ[N]) Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → √h √(m + γ[N])/N /. γ[N] → γ0 N /. params;
   dZ = λ[v[ϵ, N]] Q - β S Z - δ Z /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → √h √(m + γ[N])/N /. γ[N] → γ0 N /. params;
   Soln = NDSolve[{S'[t] == (dS /. {S → S[t], Q → Q[t], Z → Z[t]}),
      Q'[t] == (dQ /. {S → S[t], Q → Q[t], Z → Z[t]}),
      Z'[t] == (dZ /. {S → S[t], Q → Q[t], Z → Z[t]}),
      S[0] == (k/2 /. params),
      Q[0] == 0.1, Z[0] == 0.1}, {S, Q, Z}, {t, 0, 1000}];
   Return[{(S[1000] /. Soln)[[1]], ((S[1000] + Q[1000]) /. Soln)[[1]],
     ((Q[1000]/(S[1000] + Q[1000])) /. Soln)⟦1⟧, (Z[1000] /. Soln)[[1]]}]
  ];
epiDynES2[{Nvalue_?NumberQ}] :=
  Module[{params, δ, r, k, m, λ0, γ0, h, v, β, η, N, dS, S, Q, dQ, Z, dZ, λ, ϵ, γ, Soln},
   params = {δ → 0.8, r → 0.5, k → 1.5, m → 0.02, λ0 → 100,
     γ0 → 0.012592592592592593`, h → 1, v → 0.05, β → 2, η → 1, N → Nvalue};
   dS = r (S + Q) (1 - (S + Q)/k) - m S - β S Z + γ[N] Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → √h √(m + γ[N])/N /. γ[N] → γ0 N^3 /. params;
   dQ = β S Z - (m + v[ϵ, N] + γ[N]) Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → √h √(m + γ[N])/N /. γ[N] → γ0 N^3 /. params;
   dZ = λ[v[ϵ, N]] Q - β S Z - δ Z /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → √h √(m + γ[N])/N /. γ[N] → γ0 N^3 /. params;
   Soln = NDSolve[{S'[t] == (dS /. {S → S[t], Q → Q[t], Z → Z[t]}),
      Q'[t] == (dQ /. {S → S[t], Q → Q[t], Z → Z[t]}),
      Z'[t] == (dZ /. {S → S[t], Q → Q[t], Z → Z[t]}),
      S[0] == (k/2 /. params),
      Q[0] == 0.1, Z[0] == 0.1}, {S, Q, Z}, {t, 0, 1000}];
```

```
    Return[{(S[1000] /. Soln)[[1]], ((S[1000] + Q[1000]) /. Soln)[[1]],
      ((Q[1000] / (S[1000] + Q[1000])) /. Soln)⟦1⟧, (Z[1000] /. Soln)[[1]]}]
   ];
epiDynES3[{Nvalue_?NumberQ}] :=
  Module[{params, δ, r, k, m, λ0, γ0, h, v, β, η, N, dS, S, Q, dQ, Z, dZ, λ, ϵ, γ, Soln},
   params = {δ → 0.8, r → 0.5, k → 1.5, m → 0.02, λ0 → 100,
     γ0 → 3.173333333333334`, h → 1, v → 0.05,  β → 2, η → 1, N → Nvalue};
   dS = r (S + Q) (1 - (S + Q)/k) - m S - β S Z + γ[N] Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → (√h √(m + γ[N]))/N /. γ[N] → γ0 N/(η + N^3) /. params;
   dQ = β S Z - (m + v[ϵ, N] + γ[N]) Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → (√h √(m + γ[N]))/N /. γ[N] → γ0 N/(η + N^3) /. params;
   dZ = λ[v[ϵ, N]] Q - β S Z - δ Z /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → (√h √(m + γ[N]))/N /. γ[N] → γ0 N/(η + N^3) /. params;
   Soln = NDSolve[{S'[t] == (dS /. {S → S[t], Q → Q[t], Z → Z[t]}),
      Q'[t] == (dQ /. {S → S[t], Q → Q[t], Z → Z[t]}),
      Z'[t] == (dZ /. {S → S[t], Q → Q[t], Z → Z[t]}),
      S[0] == (k/2 /. params),
      Q[0] == 0.1, Z[0] == 0.1}, {S, Q, Z}, {t, 0, 1000}];
   Return[{(S[1000] /. Soln)[[1]], ((S[1000] + Q[1000]) /. Soln)[[1]],
     ((Q[1000] / (S[1000] + Q[1000])) /. Soln)⟦1⟧, (Z[1000] /. Soln)[[1]]}]
  ];

inprevEvo1 = Table[epiDynES1[{nval}][[3]], {nval, 0.1, 10, 0.1}];
SdenEvo1 = Table[epiDynES1[{nval}][[1]], {nval, 0.1, 10, 0.1}];
totPopEvo1 = Table[epiDynES1[{nval}][[2]], {nval, 0.1, 10, 0.1}];
totZEvo1 = Table[epiDynES1[{nval}][[4]], {nval, 0.1, 10, 0.1}];
SetDirectory["~/Box Sync/Anorexia_Parasite_Evolution/Plotting"];
Export["inprevEvo1.csv", inprevEvo1];
Export["SdenEvo1.csv", SdenEvo1];
Export["totPopEvo1.csv", totPopEvo1];
Export["totZEvo1.csv", totZEvo1];
```

```
inprevEvo2 = Table[epiDynES2[{nval}][[3]], {nval, 0.1, 10, 0.1}];
SdenEvo2 = Table[epiDynES2[{nval}][[1]], {nval, 0.1, 10, 0.1}];
totPopEvo2 = Table[epiDynES2[{nval}][[2]], {nval, 0.1, 10, 0.1}];
totZEvo2 = Table[epiDynES2[{nval}][[4]], {nval, 0.1, 10, 0.1}];
Export["inprevEvo2.csv", inprevEvo2];
Export["SdenEvo2.csv", SdenEvo2];
Export["totPopEvo2.csv", totPopEvo2];
Export["totZEvo2.csv", totZEvo2];
```

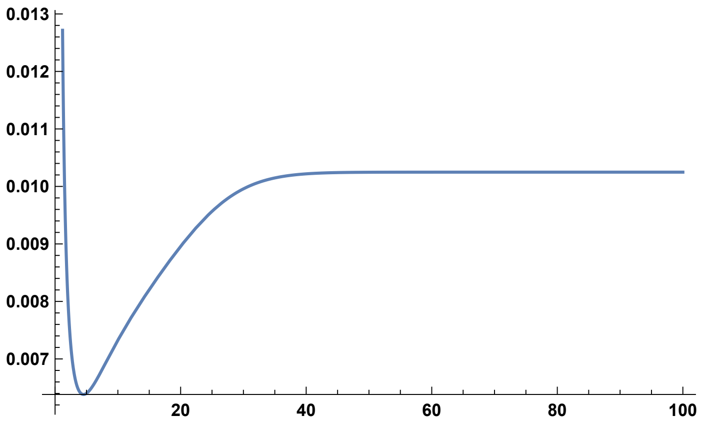
```
inprevEvo3 = Table[epiDynES3[{nval}][[3]], {nval, 0.1, 10, 0.1}];
SdenEvo3 = Table[epiDynES3[{nval}][[1]], {nval, 0.1, 10, 0.1}];
totPopEvo3 = Table[epiDynES3[{nval}][[2]], {nval, 0.1, 10, 0.1}];
totZEvo3 = Table[epiDynES3[{nval}][[4]], {nval, 0.1, 10, 0.1}];
Export["inprevEvo3.csv", inprevEvo3];
Export["SdenEvo3.csv", SdenEvo3];
Export["totPopEvo3.csv", totPopEvo3];
Export["totZEvo3.csv", totZEvo3];
```

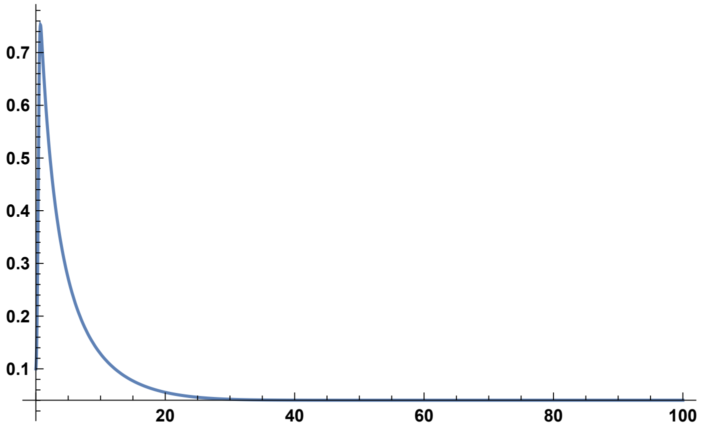
```
inprevEvo3 = Table[epiDynES3[{nval}][[3]], {nval, 0.1, 10, 0.1}];
```

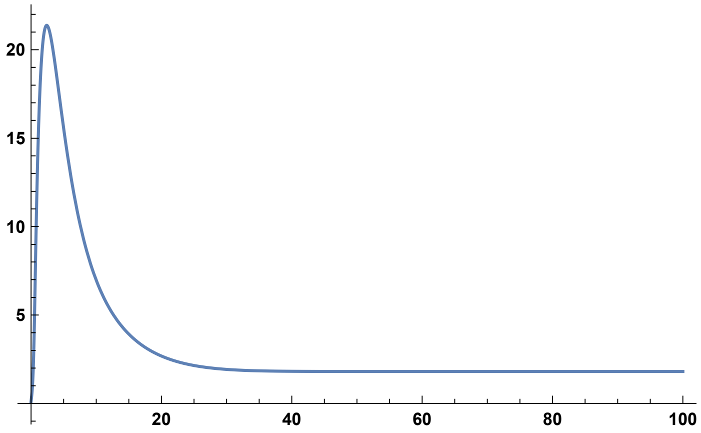
```
params = {δ → 0.8, r → 0.5, k → 1.5,  m → 0.02, λ0 → 100,
   γ0 → 3.173333333333334`, h → 1, v → 0.05,  β → 2, η → 1, N → 0.1};
dS = r (S + Q) (1 - (S + Q)/k) - m S - β S Z + γ[N] Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
     ϵ → (Sqrt[h] Sqrt[m + γ[N]])/N /. γ[N] → (γ0 N)/(η + N^3) /. params;
dQ = β S Z - (m + v[ϵ, N] + γ[N]) Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
     ϵ → (Sqrt[h] Sqrt[m + γ[N]])/N /. γ[N] → (γ0 N)/(η + N^3) /. params;
dZ = λ[v[ϵ, N]] Q - β S Z - δ Z /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
     ϵ → (Sqrt[h] Sqrt[m + γ[N]])/N /. γ[N] → (γ0 N)/(η + N^3) /. params;
Soln1 = NDSolve[{S'[t] == (dS /. {S → S[t], Q → Q[t], Z → Z[t]}),
   Q'[t] == (dQ /. {S → S[t], Q → Q[t], Z → Z[t]}),
   Z'[t] == (dZ /. {S → S[t], Q → Q[t], Z → Z[t]}),
   S[0] == (k/2 /. params),
   Q[0] == 0.1, Z[0] == 0.1}, {S, Q, Z}, {t, 0, 1000}]
Q[1000]/(Q[1000] + S[1000]) /. Soln1
```

```
{{S → InterpolatingFunction[ ⊞  Domain: {{0., 1.00 × 10^3}}
                                 Output: scalar ],

  Q → InterpolatingFunction[ ⊞  Domain: {{0., 1.00 × 10^3}}
                                 Output: scalar ],

  Z → InterpolatingFunction[ ⊞  Domain: {{0., 1.00 × 10^3}}
                                 Output: scalar ]}}

{0.797736}
```

```
Plot[S[t] /. Soln1, {t, 0, 100}]
Plot[Q[t] /. Soln1, {t, 0, 100}, PlotRange → All]
Plot[Z[t] /. Soln1, {t, 0, 100}, PlotRange → All]
```

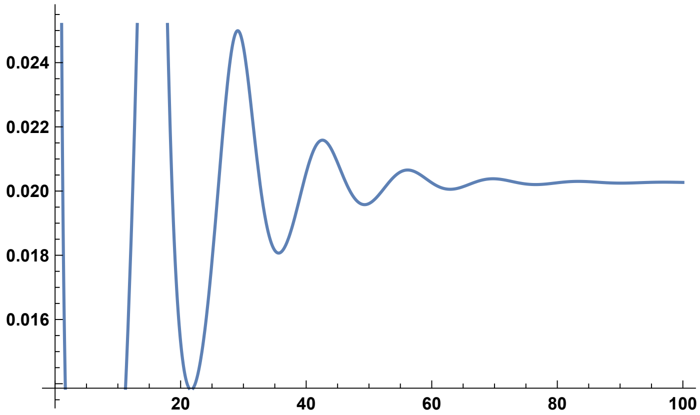


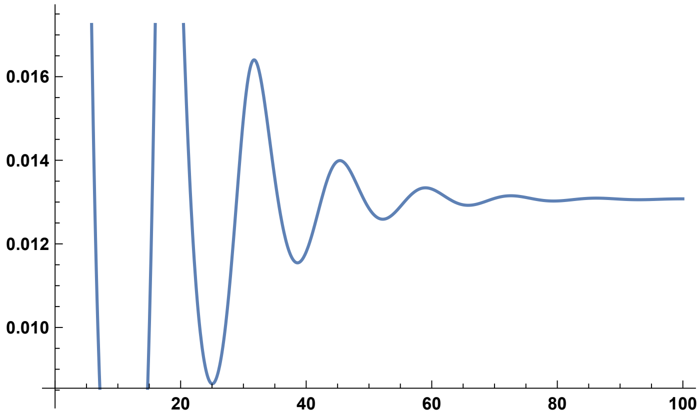


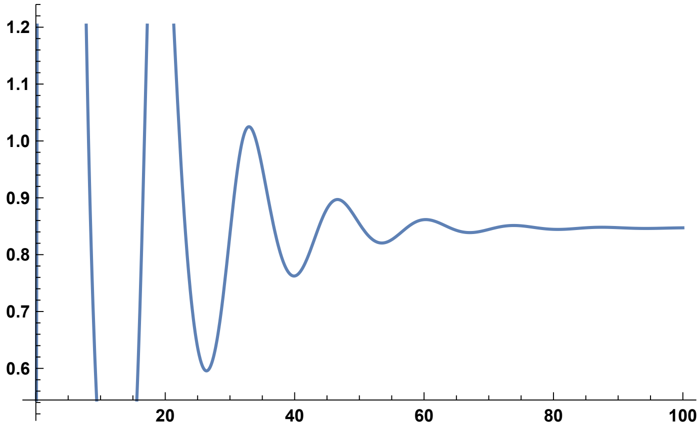


```
params = {δ → 0.8, r → 0.5, k → 1.5,  m → 0.02, λ0 → 100,
    γ0 →  3.173333333333334`, h → 1, v → 0.05,  β → 2, η → 1, N → 0.5};
dS = r (S + Q) (1 - (S + Q)/k) - m S - β S Z + γ[N] Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
      ϵ → (√h √(m + γ[N]))/N /. γ[N] → (γ0 N)/(η + N^3) /. params;
dQ = β S Z - (m + v[ϵ, N] + γ[N]) Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
      ϵ → (√h √(m + γ[N]))/N /. γ[N] → (γ0 N)/(η + N^3) /. params;
dZ = λ[v[ϵ, N]] Q - β S Z - δ Z /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
      ϵ → (√h √(m + γ[N]))/N /. γ[N] → (γ0 N)/(η + N^3) /. params;
Soln2 = NDSolve[{S'[t] == (dS /. {S → S[t], Q → Q[t], Z → Z[t]}),
    Q'[t] == (dQ /. {S → S[t], Q → Q[t], Z → Z[t]}),
    Z'[t] == (dZ /. {S → S[t], Q → Q[t], Z → Z[t]}),
    S[0] == (k/2 /. params),
    Q[0] == 0.1, Z[0] == 0.1}, {S, Q, Z}, {t, 0, 1000}]
Q[1000]/(Q[1000] + S[1000]) /. Soln2
```

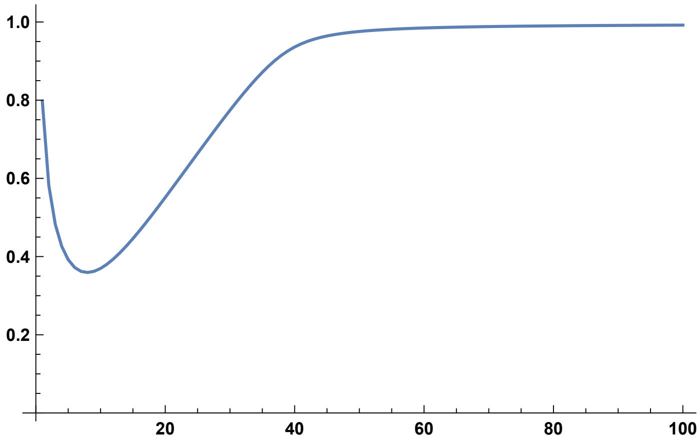
```
Plot[S[t] /. Soln2, {t, 0, 100}] (* More susceptible hosts *)
Plot[Q[t] /. Soln2, {t, 0, 100}]
Plot[Z[t] /. Soln2, {t, 0, 100}](* Very few parasites in the environment *)
```







```
ListLinePlot[inprevEvo3]
```



The other way to explore this is to ask what would happen to infection prevalence if ϵ remained fixed at the ES for a particular nutrient level, and then nutrient level changes. For this, I need a version that checks to make sure that everything remains stable.

```
statesNotAtES1[{Nvalue_?NumberQ, ESe_?NumberQ}] :=
  Module[{params, δ, r, k, m, λ0, ɣ0, h, v, β, η, N, dS, S, Q, dQ,
     Z, dZ, λ, ϵ, ɣ, Soln, ret, dt, St, FirstPeak, SecondPeak, T},
   params = {δ → 0.8, r → 0.5, k → 1.5,  m → 0.02, λ0 → 100,
     ɣ0 →  0.113333, h → 1, v → 0.05,  β  → 2, η → 1, N → Nvalue};
   dS = r (S + Q) (1 - (S + Q)/k) - m S - β S Z + ɣ[N] Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → ESe /. ɣ[N] → ɣ0 N /. params;
   dQ = β S Z - (m + v[ϵ, N] + ɣ[N]) Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /. ϵ → ESe /.
      ɣ[N] → ɣ0 N /. params;
   dZ = λ[v[ϵ, N]] Q - β S Z - δ Z /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /. ϵ → ESe /.
      ɣ[N] → ɣ0 N /. params;
   Soln = NDSolve[{S'[t] == (dS /. {S → S[t], Q → Q[t], Z → Z[t]}),
      Q'[t] == (dQ /. {S → S[t], Q → Q[t], Z → Z[t]}),
      Z'[t] == (dZ /. {S → S[t], Q → Q[t], Z → Z[t]}),
      S[0] == (k/2 /. params),
      Q[0] == 0.1, Z[0] == 0.1}, {S, Q, Z}, {t, 0, 1000}];
   If[Abs[((S[1000] - S[990]) /. Soln)[[1]]] < 10^-6,
    (* if the system has gone to an equilibrium return the
     abundances of all three variables and infection prevalence *)
    ret = {"stable", S[1000] /. Soln[[1]], (S[1000] + Q[1000]) /. Soln[[1]],
      (Q[1000]/(S[1000] + Q[1000])) /. Soln[[1]], Z[1000] /. Soln[[1]]},
    (* if the system is cycling *)
    dt = 0.01;
    St = Table[(S[t] /. Soln)⟦1⟧, {t, 800, 900, dt}];
    (* find the first peak of the cycle *)
```

```
      FirstPeak = 800 + Position[St, Max[St]]⟦1, 1⟧ dt;
      St = Table[(S[t] /. Soln)⟦1⟧, {t, FirstPeak, 1000, dt}];
      (* Cycle period *)
      T = Position[
          Table[(St⟦t⟧ - St⟦t + 1⟧)/(St⟦t + 1⟧ - St⟦t + 2⟧), {t, 1, Length[St] - 2}], _? (# < 0 &)]⟦2, 1⟧ dt;
      (* Next peak *)
      SecondPeak = FirstPeak + T;
      (* Average abundance of susceptibles,
      infecteds, infection prevalence, and free-living parasites *)
      ret = {"unstable",
        1/T Total[Table[S[t] /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]],
        1/T Total[Table[(S[t] + Q[t]) /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]],
        1/T Total[Table[Q[t]/(S[t] + Q[t]) /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]],
        1/T Total[Table[Z[t] /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]]}
    ];
    Return[ret]
  ];
statesNotAtES2[{Nvalue_?NumberQ, ESe_?NumberQ}] :=
  Module[{params, δ, r, k, m, λ0, γ0, h, v, β, η, N, dS, S, Q, dQ,
    Z, dZ, λ, ϵ, γ, Soln, ret, dt, St, FirstPeak, SecondPeak, T},
    params = {δ → 0.8, r → 0.5, k → 1.5, m → 0.02, λ0 → 100,
      γ0 → 0.012592592592592593`, h → 1, v → 0.05, β → 2, η → 1, N → Nvalue};
    dS = r (S + Q) (1 - (S + Q)/k) - m S - β S Z + γ[N] Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
        ϵ → ESe /. γ[N] → γ0 N^3 /. params;
    dQ = β S Z - (m + v[ϵ, N] + γ[N]) Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /. ϵ → ESe /.
       γ[N] → γ0 N^3 /. params;
    dZ = λ[v[ϵ, N]] Q - β S Z - δ Z /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /. ϵ → ESe /.
       γ[N] → γ0 N^3 /. params;
    Soln = NDSolve[{S'[t] == (dS /. {S → S[t], Q → Q[t], Z → Z[t]}),
       Q'[t] == (dQ /. {S → S[t], Q → Q[t], Z → Z[t]}),
       Z'[t] == (dZ /. {S → S[t], Q → Q[t], Z → Z[t]}),
       S[0] == (k/2 /. params),
       Q[0] == 0.1, Z[0] == 0.1}, {S, Q, Z}, {t, 0, 1000}];
    If[Abs[((S[1000] - S[990]) /. Soln)[[1]]] < 10^-6,
      (* if the system has gone to an equilibrium return the
       abundances of all three variables and infection prevalence *)
      ret = {"stable", S[1000] /. Soln[[1]], (S[1000] + Q[1000]) /. Soln[[1]],
        (Q[1000]/(S[1000] + Q[1000])) /. Soln[[1]], Z[1000] /. Soln[[1]]},
      (* if the system is cycling *)
      dt = 0.01;
      St = Table[(S[t] /. Soln)⟦1⟧, {t, 800, 900, dt}];
      (* find the first peak of the cycle *)
      FirstPeak = 800 + Position[St, Max[St]]⟦1, 1⟧ dt;
      St = Table[(S[t] /. Soln)⟦1⟧, {t, FirstPeak, 1000, dt}];
      (* Cycle period *)
      T = Position[
          Table[(St⟦t⟧ - St⟦t + 1⟧)/(St⟦t + 1⟧ - St⟦t + 2⟧), {t, 1, Length[St] - 2}], _? (# < 0 &)]⟦2, 1⟧ dt;
      (* Next peak *)
      SecondPeak = FirstPeak + T;
      (* Average abundance of susceptibles,
      infecteds, infection prevalence, and free-living parasites *)
      ret = {"unstable",
        1/T Total[Table[S[t] /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]],
        1/T Total[Table[(S[t] + Q[t]) /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]],
        1/T Total[Table[Q[t]/(S[t] + Q[t]) /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]],
        1/T Total[Table[Z[t] /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]]}
     ];
     Return[ret]
    ];


statesNotAtES3[{Nvalue_?NumberQ, ESe_?NumberQ}] :=
  Module[{params, δ, r, k, m, λ0, γ0, h, v, β, η, N, dS, S, Q, dQ,
    Z, dZ, λ, ϵ, γ, Soln, ret, dt, St, FirstPeak, SecondPeak, T},
   params = {δ → 0.8, r → 0.5, k → 1.5, m → 0.02, λ0 → 100,
     γ0 → 3.173333333333334`, h → 1, v → 0.05, β → 2, η → 1, N → Nvalue};
   dS = r (S + Q) (1 - (S + Q)/k) - m S - β S Z + γ[N] Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /.
       ϵ → ESe /. γ[N] → γ0 N/(η + N^3) /. params;
   dQ = β S Z - (m + v[ϵ, N] + γ[N]) Q /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /. ϵ → ESe /.
     γ[N] → γ0 N/(η + N^3) /. params;
   dZ = λ[v[ϵ, N]] Q - β S Z - δ Z /. v[ϵ, N] → ϵ N /. λ[ϵ N] → λ0 ϵ N/(h + ϵ N) /. ϵ → ESe /.
     γ[N] → γ0 N/(η + N^3) /. params;
   Soln = NDSolve[{S'[t] == (dS /. {S → S[t], Q → Q[t], Z → Z[t]}),
      Q'[t] == (dQ /. {S → S[t], Q → Q[t], Z → Z[t]}),
      Z'[t] == (dZ /. {S → S[t], Q → Q[t], Z → Z[t]}),
      S[0] == (k / 2 /. params),
      Q[0] == 0.1, Z[0] == 0.1}, {S, Q, Z}, {t, 0, 1000}];
   If[Abs[((S[1000] - S[990]) /. Soln)[[1]]] < 10^-6,
    (* if the system has gone to an equilibrium return the
     abundances of all three variables and infection prevalence *)
    ret = {"stable", S[1000] /. Soln[[1]], (S[1000] + Q[1000]) /. Soln[[1]],
      (Q[1000] / (S[1000] + Q[1000])) /. Soln[[1]], Z[1000] /. Soln[[1]]},
    (* if the system is cycling *)
    dt = 0.01;
    St = Table[(S[t] /. Soln)⟦1⟧, {t, 800, 900, dt}];
    (* find the first peak of the cycle *)
    FirstPeak = 800 + Position[St, Max[St]]⟦1, 1⟧ dt;
    St = Table[(S[t] /. Soln)⟦1⟧, {t, FirstPeak, 1000, dt}];
    (* Cycle period *)
    T = Position[
        Table[(St⟦t⟧ - St⟦t + 1⟧) / (St⟦t + 1⟧ - St⟦t + 2⟧), {t, 1, Length[St] - 2}], _? (# < 0 &)]⟦2, 1⟧ dt;
    (* Next peak *)
    SecondPeak = FirstPeak + T;
    (* Average abundance of susceptibles,
    infecteds, infection prevalence, and free-living parasites *)
    ret = {"unstable",
      1/T Total[Table[S[t] /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]],
      1/T Total[Table[(S[t] + Q[t]) /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]],
      1/T Total[Table[Q[t] / (S[t] + Q[t]) /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]],
      1/T Total[Table[Z[t] /. Soln, {t, FirstPeak, SecondPeak, dt}] dt][[1]]}
   ];
   Return[ret]
  ];

(* The ϵ value at the point where the three functions agree is
Table[statesNotAtES1[{nval, 0.2}][[1]], {nval, 0.1, 10, 0.1}]
```

```
{stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable}
```

```
Table[statesNotAtES2[{nval, 0.2}][[1]], {nval, 0.1, 10, 0.1}]
```

```
{stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable}
```

```
Table[statesNotAtES3[{nval, 0.2}][[1]], {nval, 0.1, 10, 0.1}]
```

```
{stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, stable, stable,
 stable, stable, stable, stable, stable, stable, stable, unstable, unstable,
 unstable, unstable, unstable, unstable, unstable, unstable, unstable, unstable,
 unstable, unstable, unstable, unstable, unstable, unstable, unstable, unstable,
 unstable, unstable, unstable, unstable, unstable, unstable, unstable, unstable,
 unstable, unstable, unstable, unstable, unstable, unstable, unstable, unstable}
```

```
inprevNoEvo1 = Table[statesNotAtES1[{nval, 0.2}][[4]], {nval, 0.1, 10, 0.1}];
SdenNoEvo1 = Table[statesNotAtES1[{nval, 0.2}][[2]], {nval, 0.1, 10, 0.1}];
totPopNoEvo1 = Table[statesNotAtES1[{nval, 0.2}][[3]], {nval, 0.1, 10, 0.1}];
totZNoEvo1 = Table[statesNotAtES1[{nval, 0.2}][[5]], {nval, 0.1, 10, 0.1}];
SetDirectory["~/Box Sync/Anorexia_Parasite_Evolution/Plotting"];
Export["inprevNoEvo1.csv", inprevNoEvo1];
Export["SdenNoEvo1.csv", SdenNoEvo1];
Export["totPopNoEvo1.csv", totPopNoEvo1];
Export["totZNoEvo1.csv", totZNoEvo1];

inprevNoEvo2 = Table[statesNotAtES2[{nval, 0.2}][[4]], {nval, 0.1, 10, 0.1}];
SdenNoEvo2 = Table[statesNotAtES2[{nval, 0.2}][[2]], {nval, 0.1, 10, 0.1}];
totPopNoEvo2 = Table[statesNotAtES2[{nval, 0.2}][[3]], {nval, 0.1, 10, 0.1}];
totZNoEvo2 = Table[statesNotAtES2[{nval, 0.2}][[5]], {nval, 0.1, 10, 0.1}];
Export["inprevNoEvo2.csv", inprevNoEvo2];
Export["SdenNoEvo2.csv", SdenNoEvo2];
Export["totPopNoEvo2.csv", totPopNoEvo2];
Export["totZNoEvo2.csv", totZNoEvo2];

inprevNoEvo3 = Table[statesNotAtES3[{nval, 0.2}][[4]], {nval, 0.1, 10, 0.1}];
SdenNoEvo3 = Table[statesNotAtES3[{nval, 0.2}][[2]], {nval, 0.1, 10, 0.1}];
totPopNoEvo3 = Table[statesNotAtES3[{nval, 0.2}][[3]], {nval, 0.1, 10, 0.1}];
totZNoEvo3 = Table[statesNotAtES3[{nval, 0.2}][[5]], {nval, 0.1, 10, 0.1}];
Export["inprevNoEvo3.csv", inprevNoEvo3];
Export["SdenNoEvo3.csv", SdenNoEvo3];
Export["totPopNoEvo3.csv", totPopNoEvo3];
Export["totZNoEvo3.csv", totZNoEvo3];
```