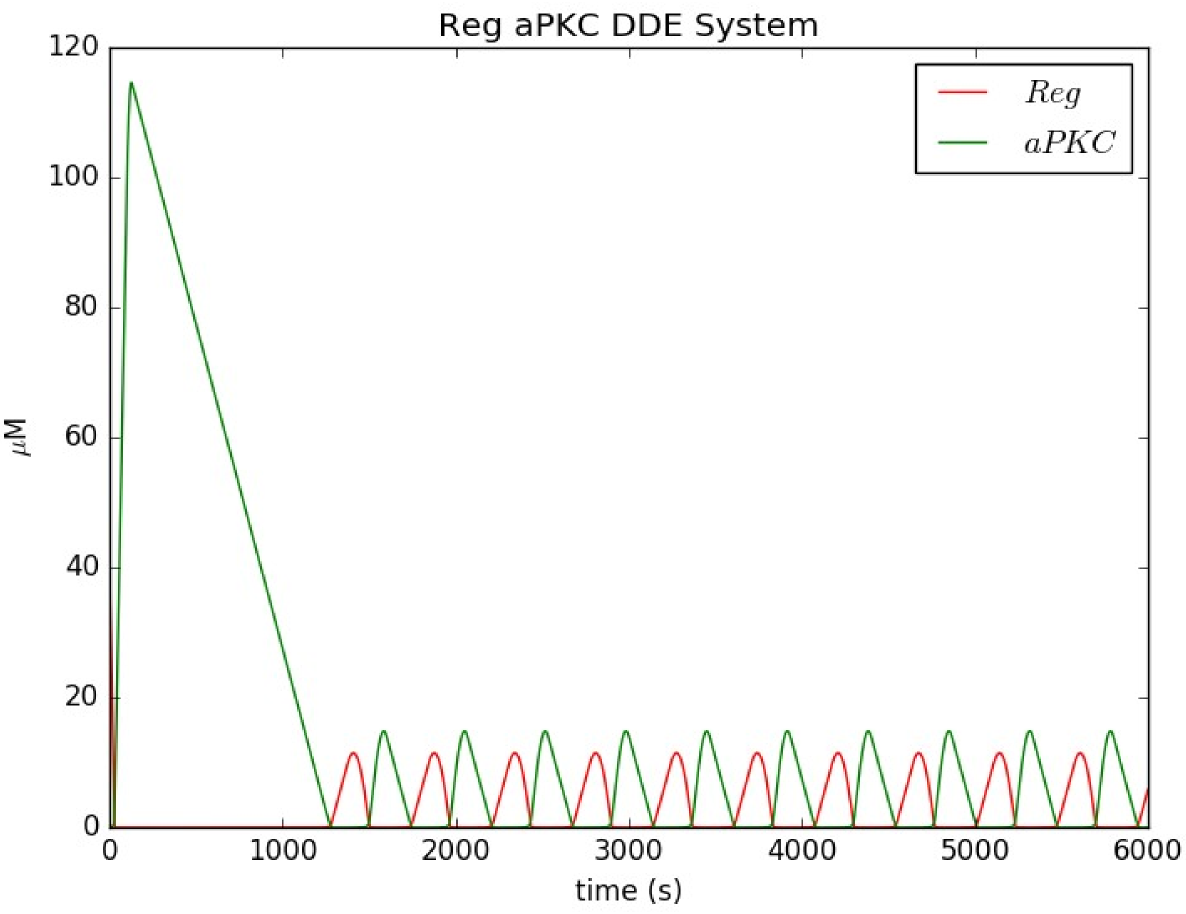
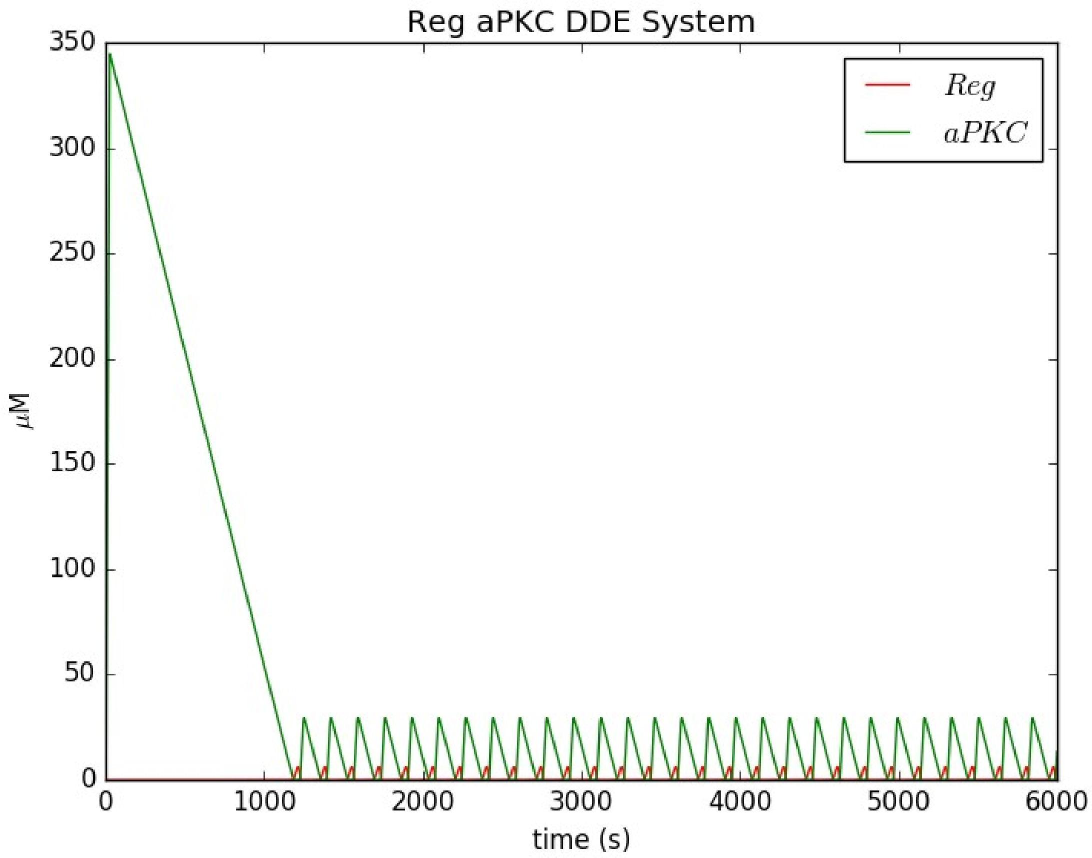
DDEs With Simplified System

k1 = Birth rate of aPKC (also used is delayed recruitment)
k2 = mutual antagonism
k3 = birth rate of Reg
k4 = decomposition of AR

Simplified model. Ignoring AR complex production and migration of aPKC.



The parameters used were:
{
 "k1": 0.03,
 "k2": 0.5,
 "k3": 0.1,
 "tau1": 100.0
}
The initial conditions used we
{
 "A": 10.0,
 "R": 50.0
}



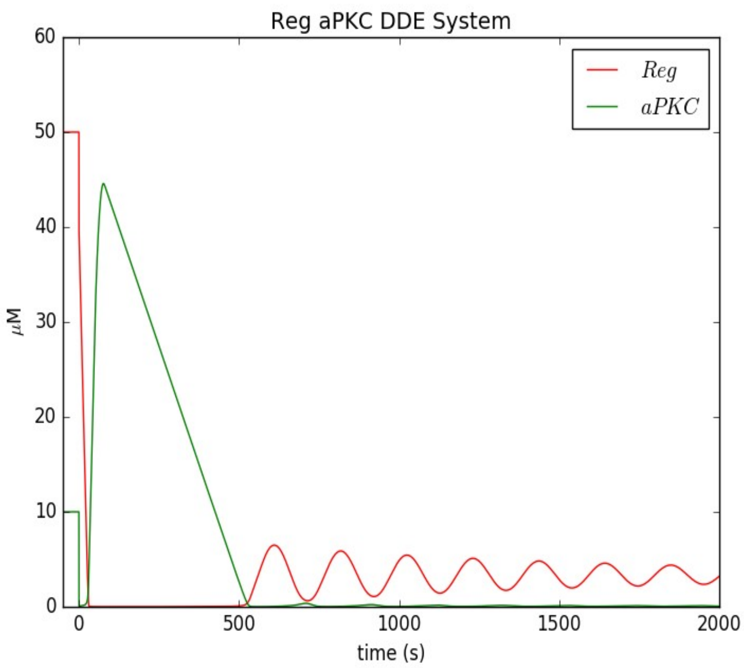
The parameters used were:

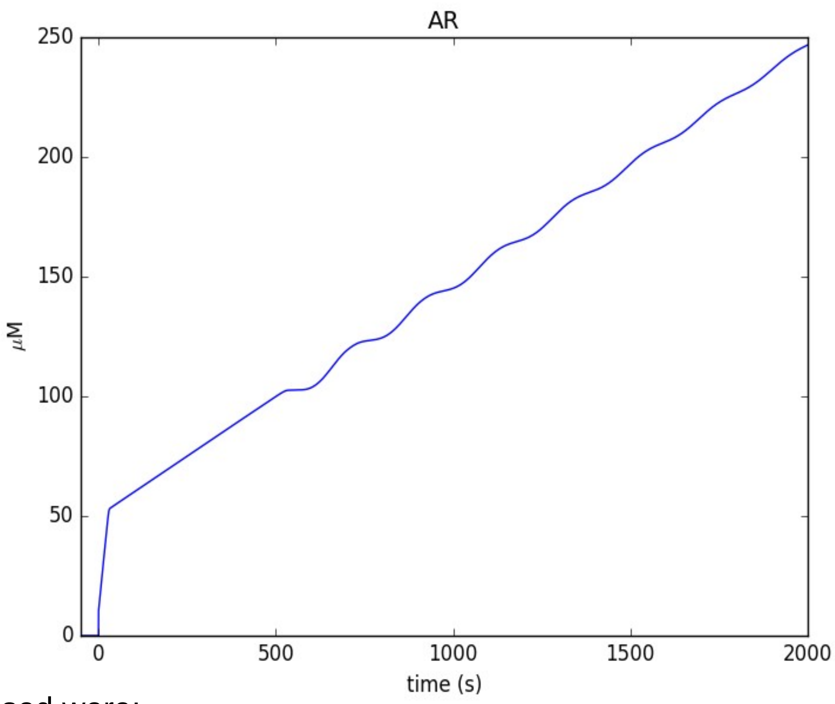
```
{
 "k1": 0.3,
 "k2": 0.5,
 "k3": 0.3,
 "tau1": 25.0
}
```

The initial conditions used were:

```
{
 "A": 10.0,
 "R": 50.0
}
```

Simplified Model: AR complex now included.





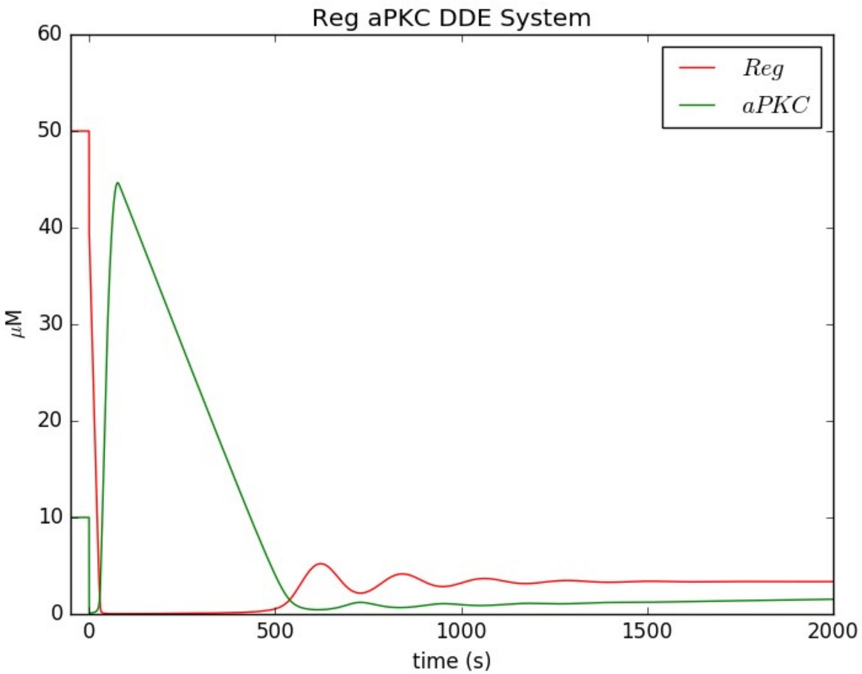
The parameters used were:

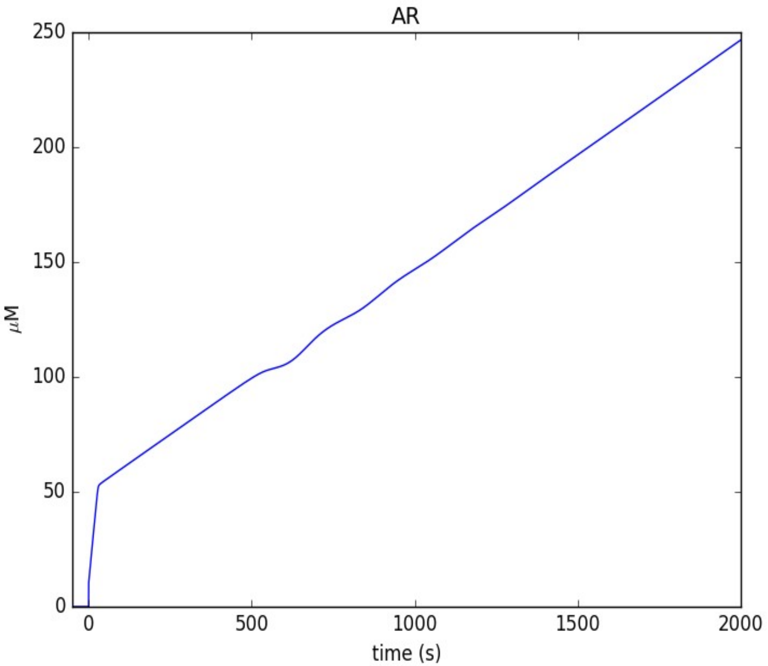
{
 "k1": 0.03,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.0001,
 "tau1": 50.0
}

The initial conditions used were:

{
 "ARi": 0.0,
 "Ai": 10.0,
 "Ri": 50.0
}

Simplified Model: AR complex now included. Increase k4 rate that AR dissociates



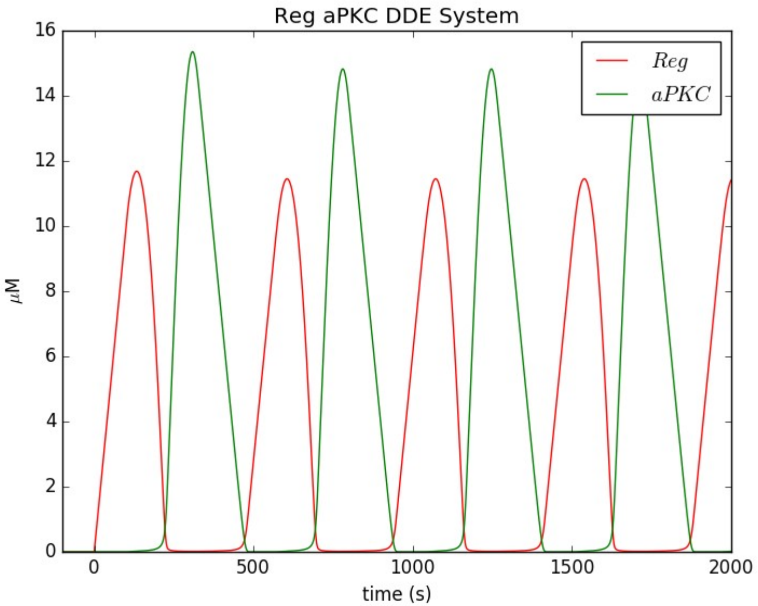


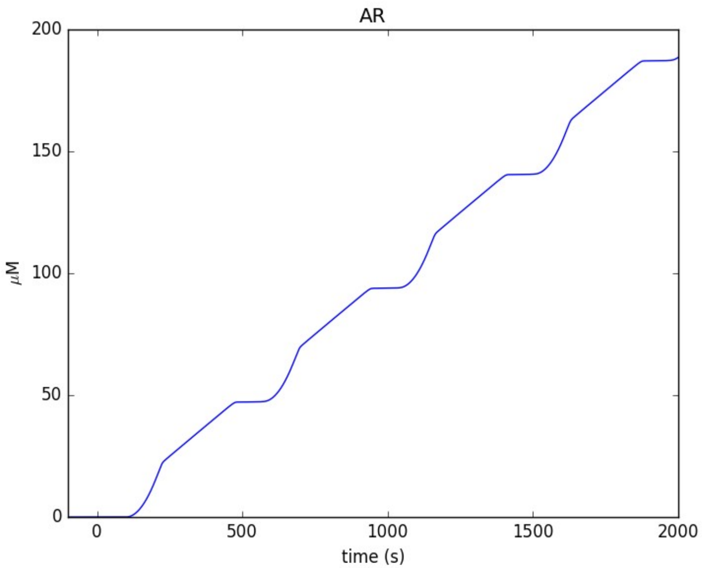
The parameters used were:
{
 "k1": 0.03,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.01,
 "tau1": 50.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Ai": 10.0,
 "Ri": 50.0
}
Increasing k4 causes the oscillations to dampen out.

Conclusions (after meeting with Jimmy)

- K4 adds dampening to the system.
- APKC appears to sustain oscillations.
- Check a k1*tau condition
  - Doesn't appear to be quite the same. There is a sweet spot though.
- Check lower Ri condition.
  - Confirmed. We avoid the initial spike.

Decrease R_i to see if it removes the initial spike = True

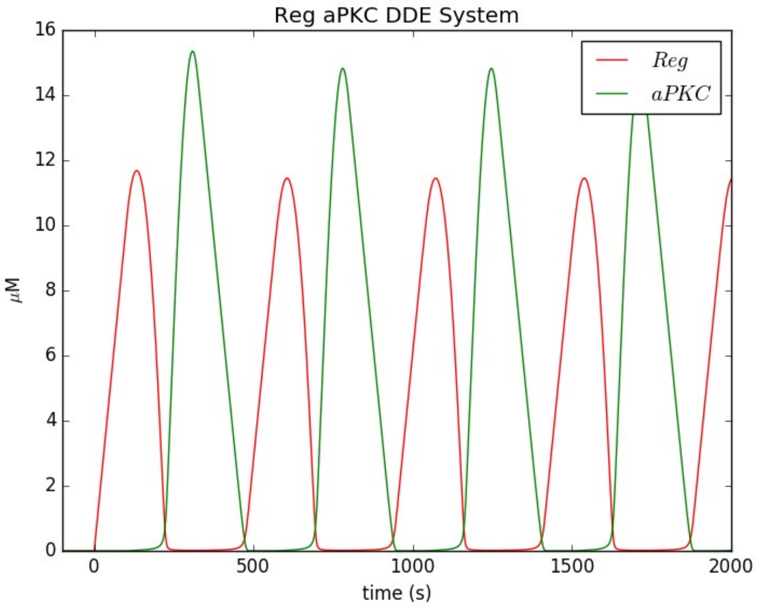


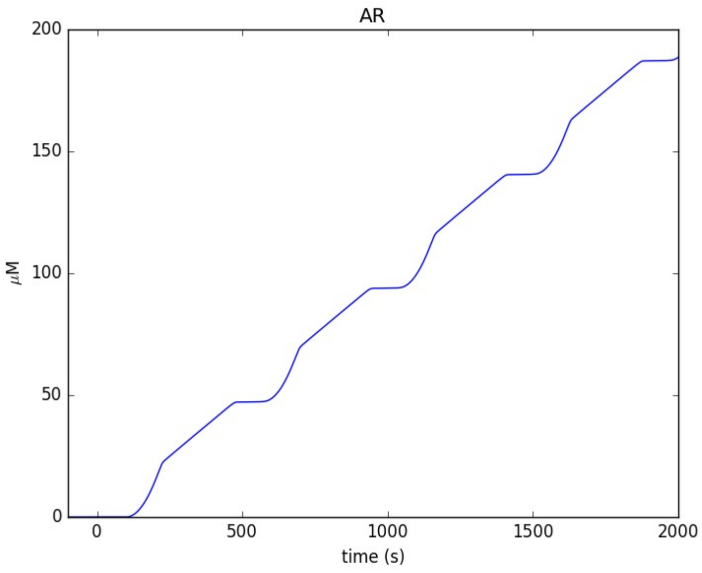


The parameters used were:
{
 "k1": 0.03,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.0,
 "tau1": 100.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}

Hence, as expected started with a lower Ri,
does not cause the initial spike of aPKC

Is there a k1*tau*e > 1 connection?





The parameters used were:

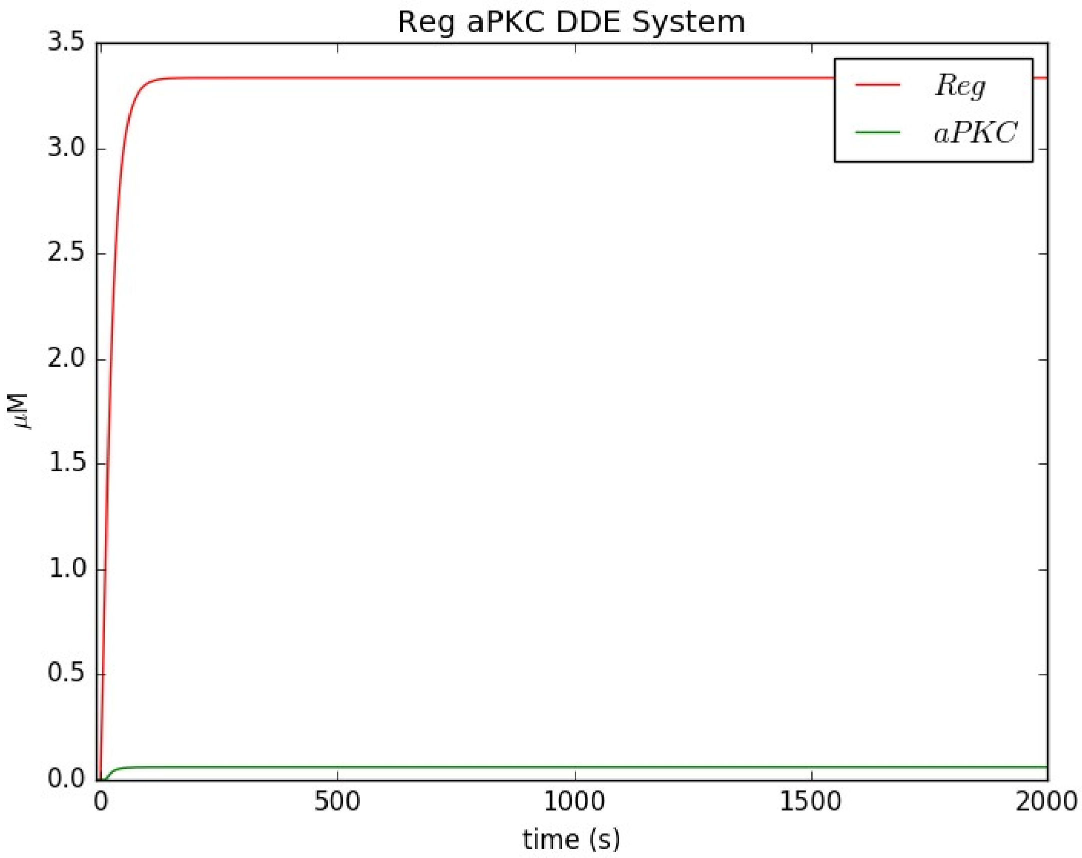
```
{
 "k1": 0.03,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.0,
 "tau1": 100.0
}
```

The initial conditions used were:

```
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
```

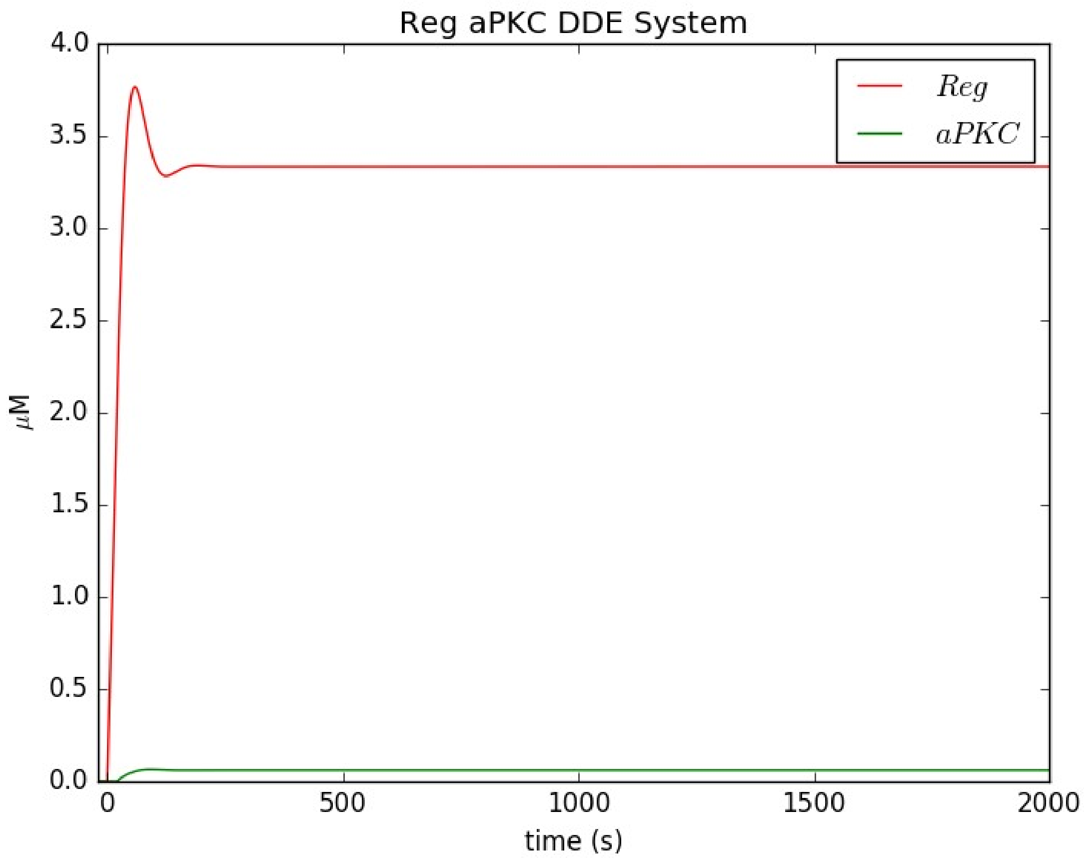
k1*tau*e = 8.15484548538

Is there a k1*tau*e > 1 connection?



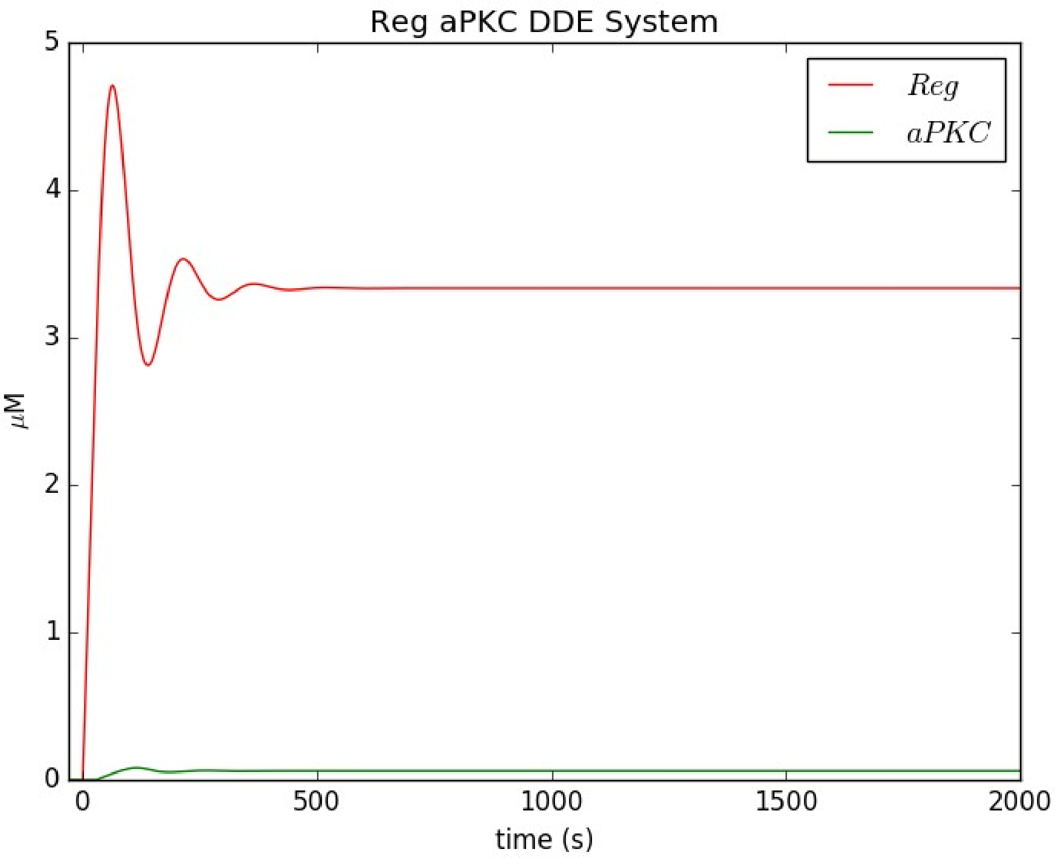
The parameters used were:
```
{
 "k1": 0.03,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.0,
 "tau1": 10.0
}
```
The initial conditions used were:
```
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
```
k1*tau*e = 0.815484548538

Is there a k1*tau*e > 1 connection?



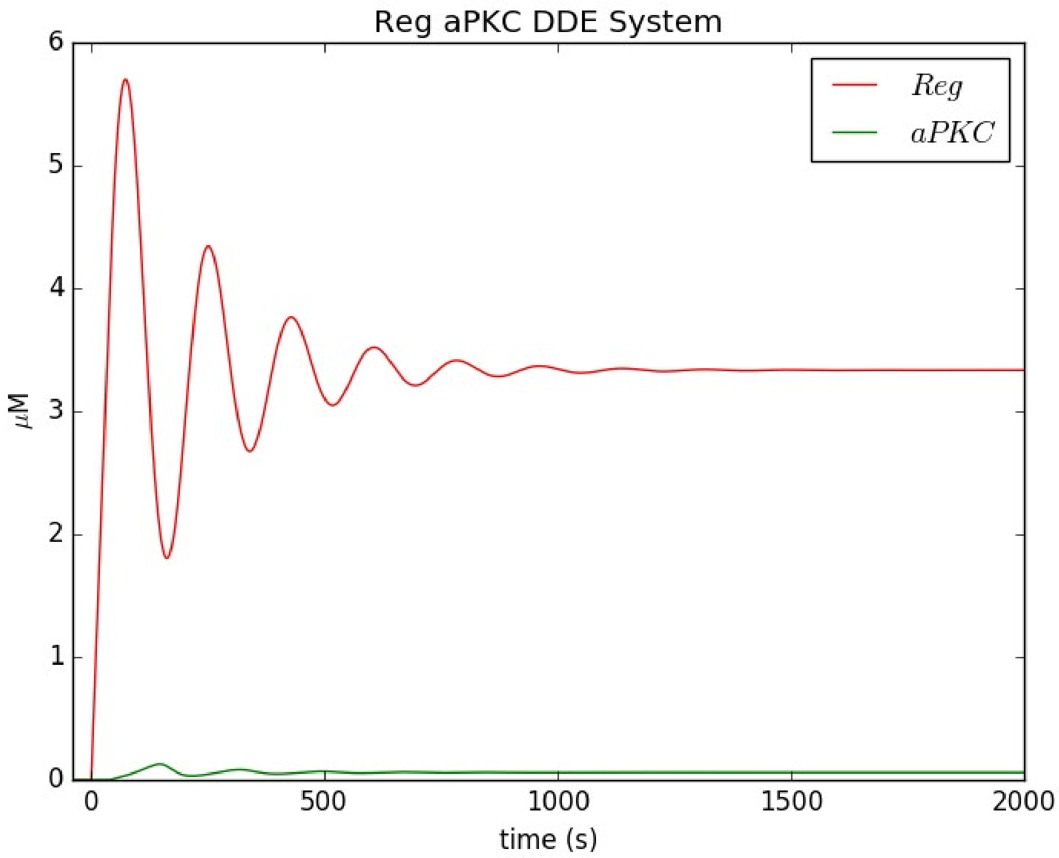
The parameters used were:
{
 "k1": 0.03,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.0,
 "tau1": 20.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 1.63096909708

Is there a k1*tau*e > 1 connection?



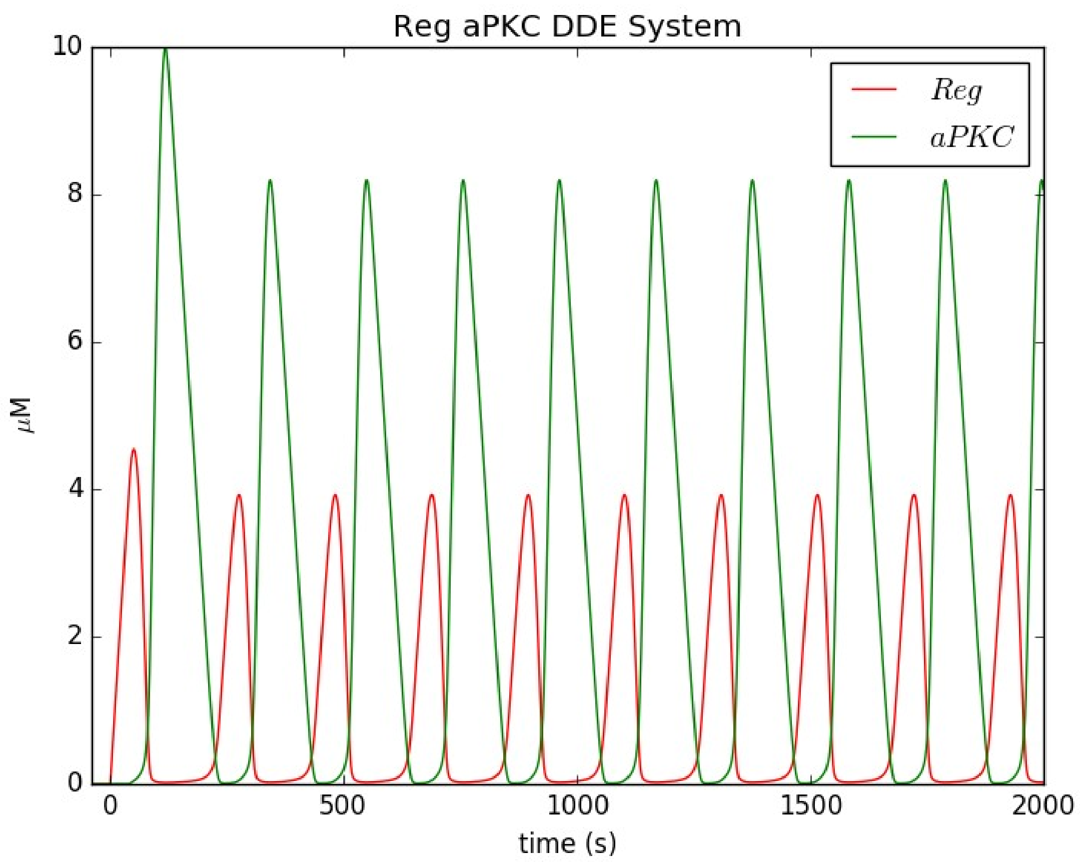
The parameters used were:
{
"k1": 0.03,
"k2": 0.5,
"k3": 0.1,
"k4": 0.0,
"tau1": 30.0
}
The initial conditions used were:
{
"ARi": 0.0,
"Ai": 0.0,
"Ri": 0.0
}
k1*tau*e = 2.44645364561

Is there a k1*tau*e > 1 connection?



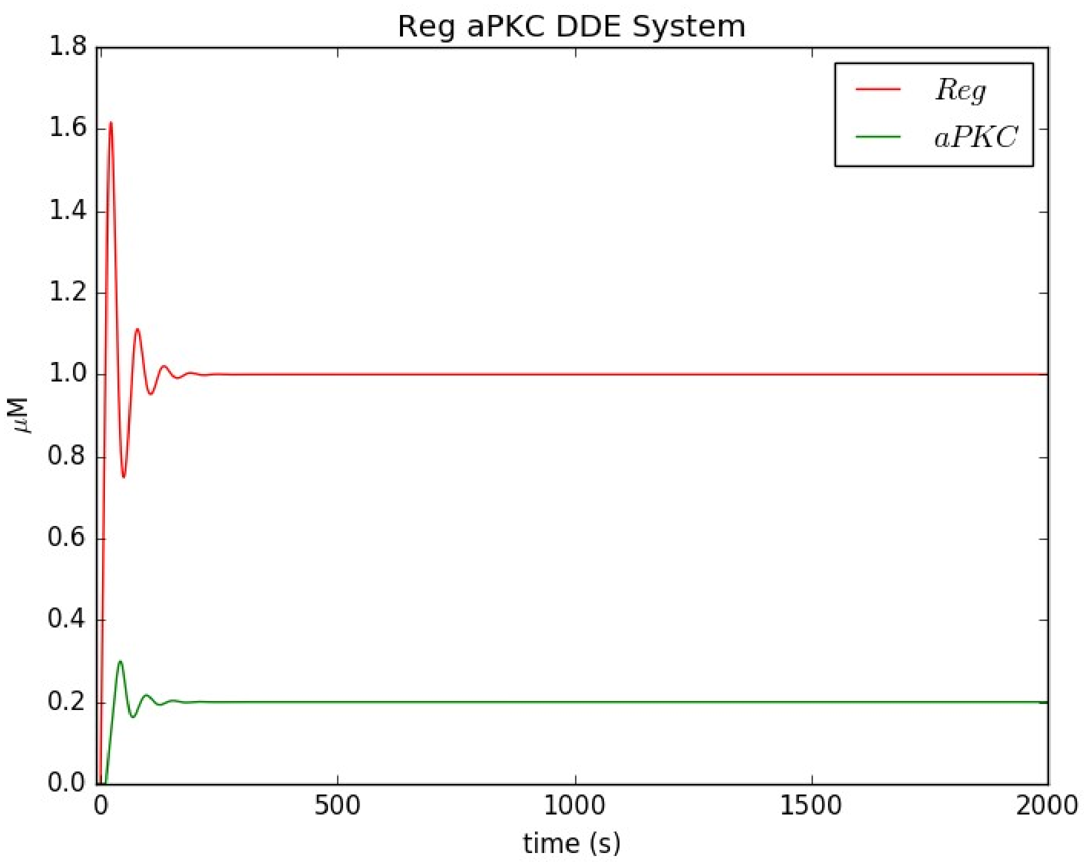
The parameters used were:
{
 "k1": 0.03,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.0,
 "tau1": 40.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 3.26193819415

Is there a k1*tau*e > 1 connection?



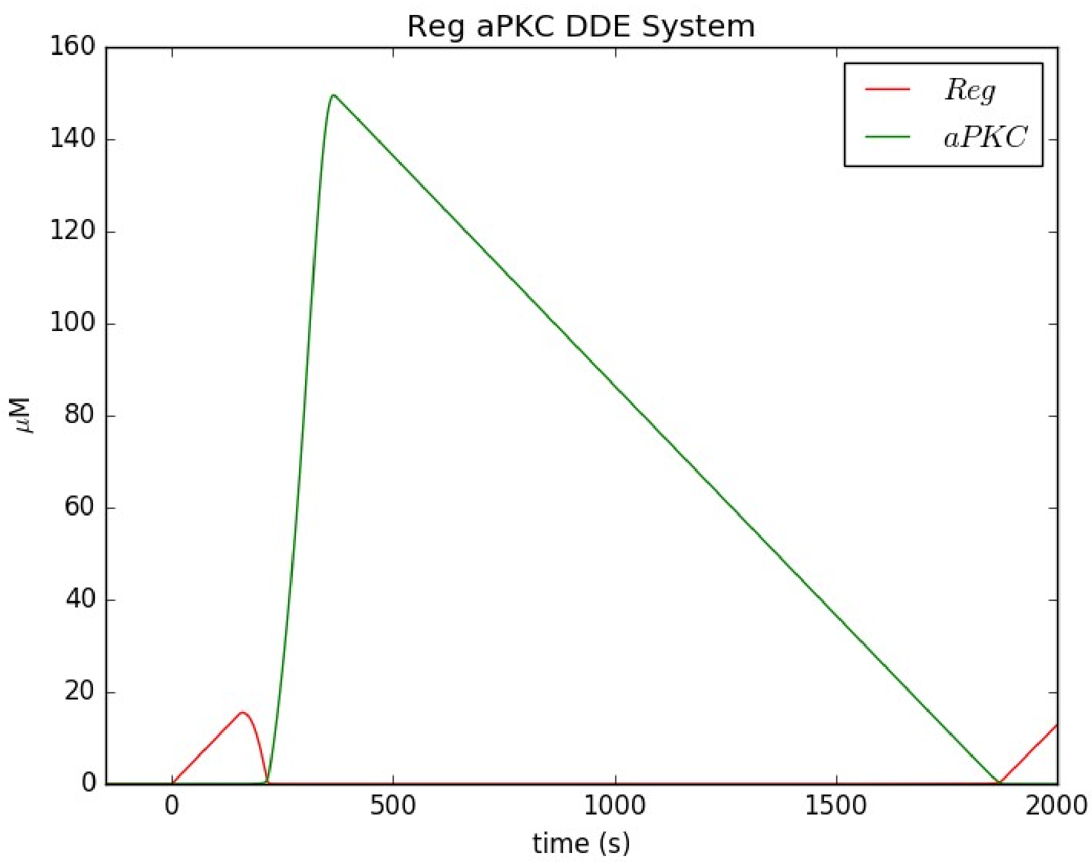
The parameters used were:
{
 "k1": 0.1,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0,
 "tau1": 40.0
}
The initial conditions used were
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 10.8731273138

Is there a k1*tau*e > 1 connection?



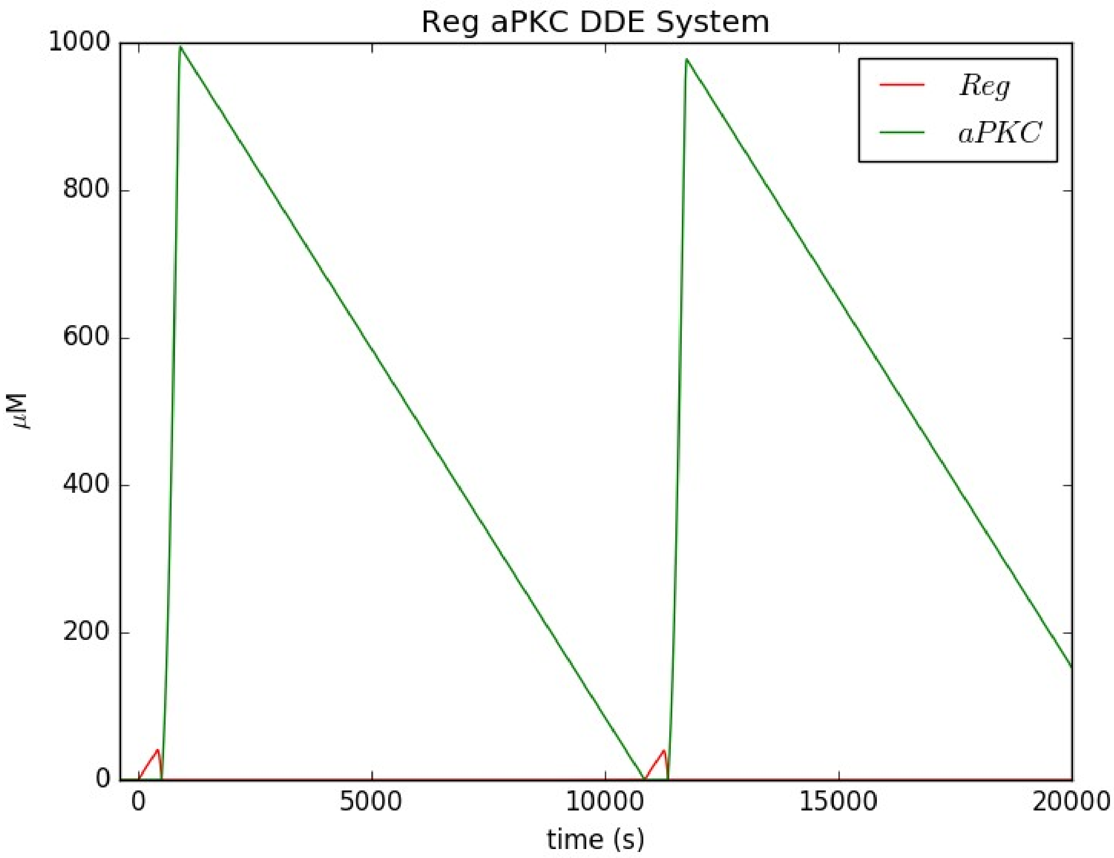
The parameters used were:
{
 "k1": 0.1,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0,
 "tau1": 10.0
}
The initial conditions used were
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 2.71828182846

Is there a k1*tau*e > 1 connection?



The parameters used were:
{
 "k1": 0.1,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0,
 "tau1": 150.0
}
The initial conditions used were
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 40.7742274269

Is there a k1*tau*e > 1 connection?



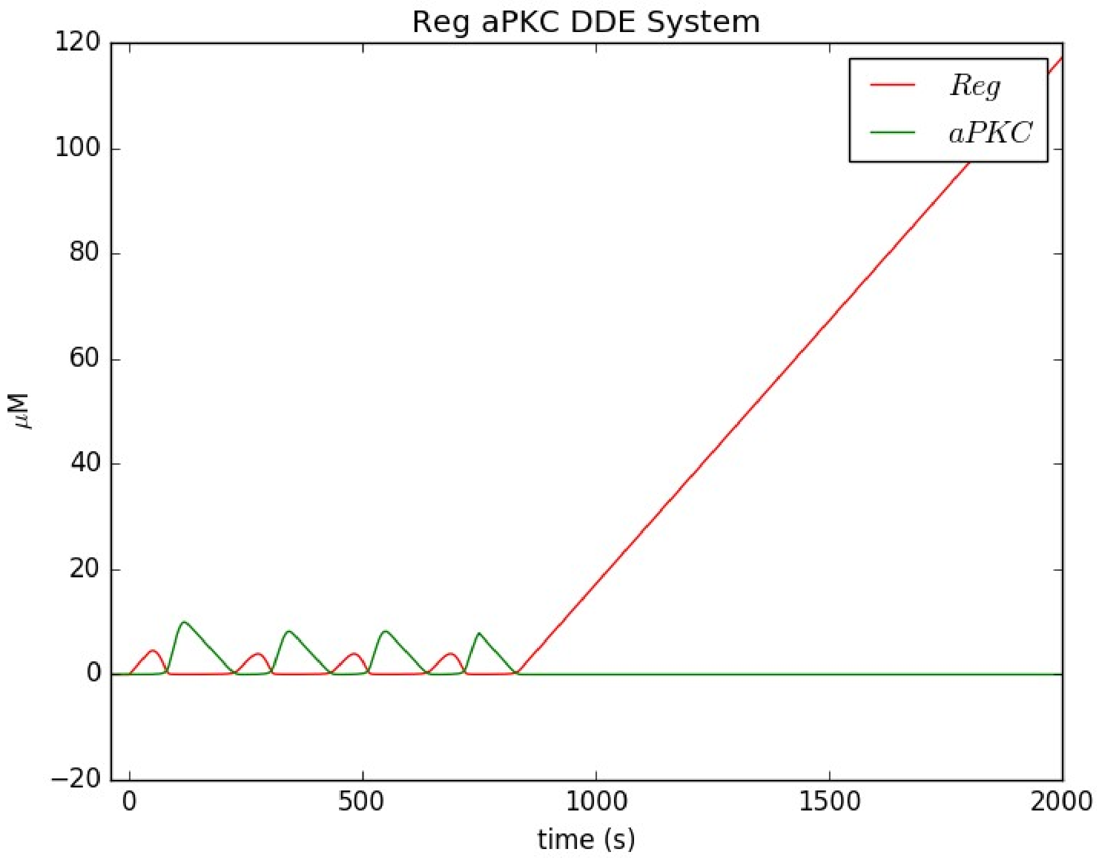
The parameters used were:
{
 "k1": 0.1,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0,
 "tau1": 400.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 108.731273138

*Notice the time scale.

Conclusion - Is there a k1*tau*e > 1 connection?

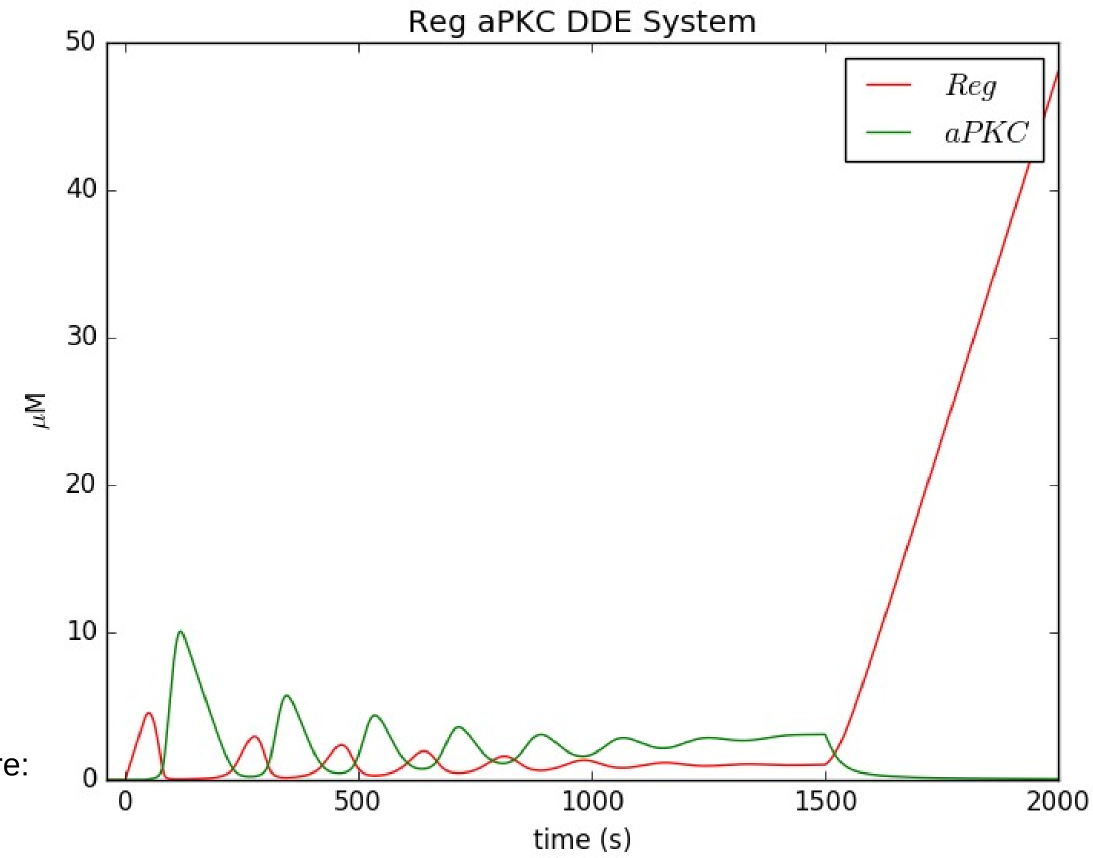
- It doesn't seem to have the same condition attached. A literature search for coupled systems should be carried out to find a similar mechanism.
- However, we do see that there is a sweet spot for k1*tau to reside in for oscillations on this time scale.
- Update after next section

Investigate Ac sustaining oscillations. Implement 1-Heavi(t-750).



The parameters used were:
{
 "k1": 0.1,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0,
 "tau1": 40.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 10.8731273138

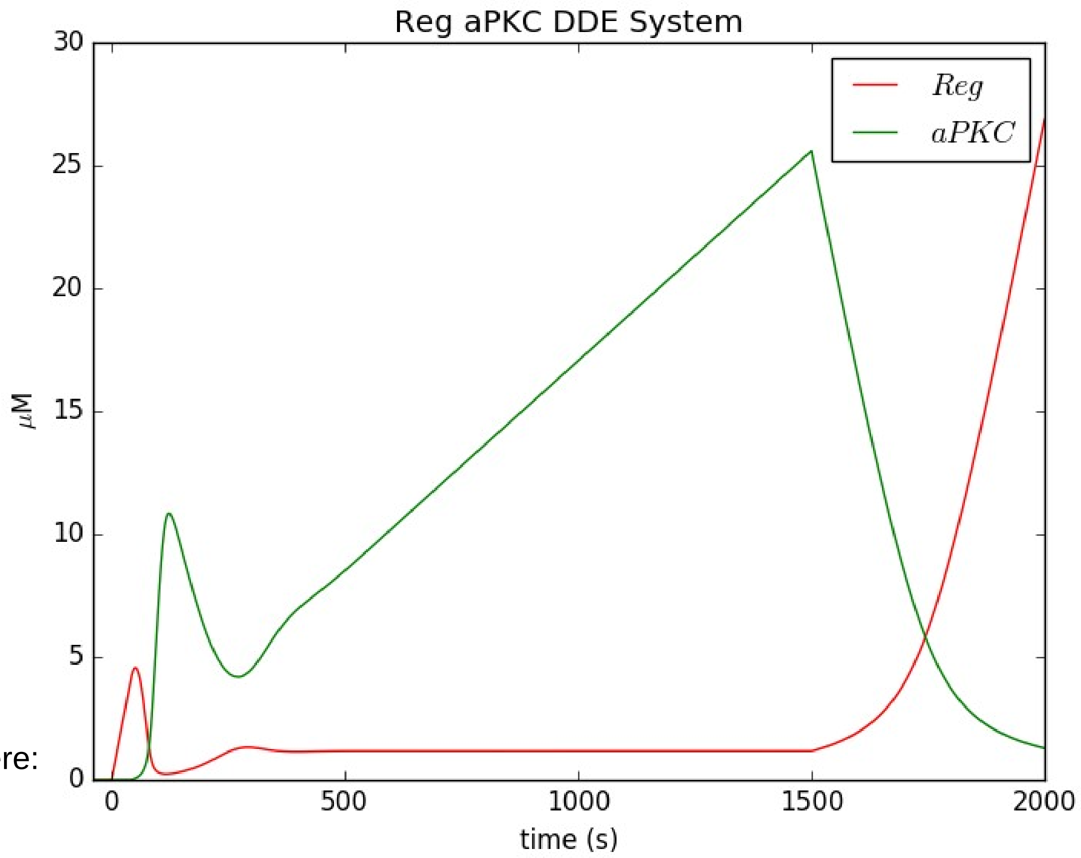
Composite Effect of k4 and “Ac”



The parameters used were:
{
 "k1": 0.1,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.01,
 "tau1": 40.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 10.8731273138

K1 turns off at t=1500.
K4 needs to be low or there are not any oscillations.

Composite Effect of k4 and “Ac”



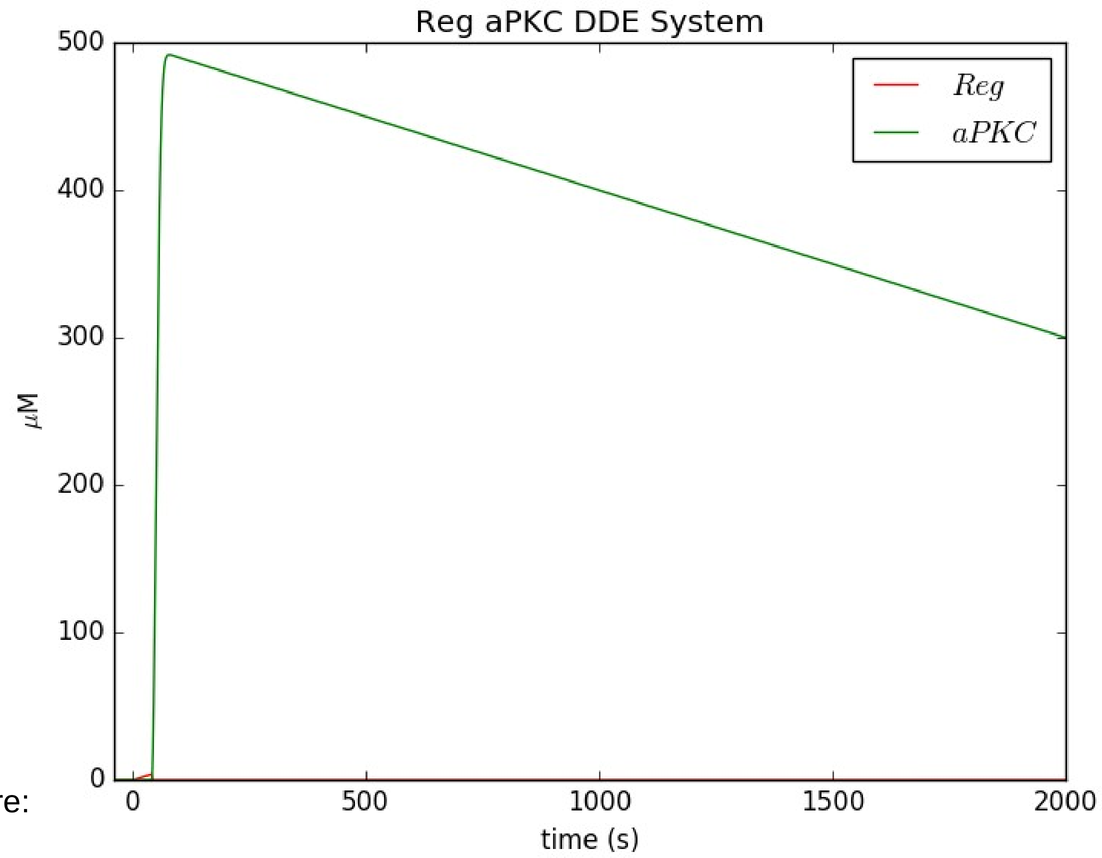
The parameters used were:
{
 "k1": 0.1,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.1,
 "tau1": 40.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 10.8731273138

Larger tau1 value leads to non-oscillatory dynamics.

Conclusion – k4 and “Ac”

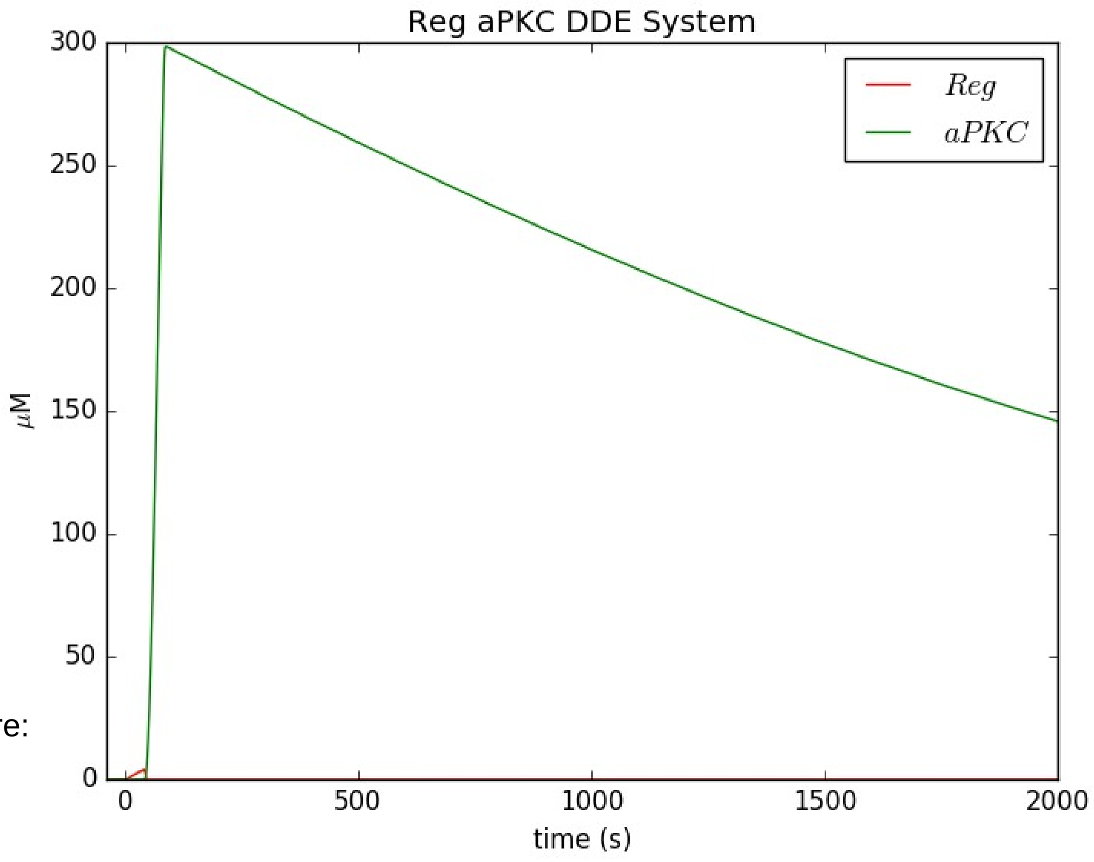
- It is possible to use a finite Ac supply to sustain oscillations for a set amount of time based on preliminary study. Therefore, adding in Ac looks like a promising direction to follow.
- K4 in a sense dampens oscillations, but it also has a propensity to ruin ideal oscillations.
- Based on the way the model is shaping up the biological cost we would play to keep k4 low is that AR has a low “off-rate” which incurs relatively little cost.
- K4 is much more important in determining the length of oscillations. Increasing Ac is not enough to sustain oscillations for t~6000s.
  - Figures omitted as of now.

Implementing Migration



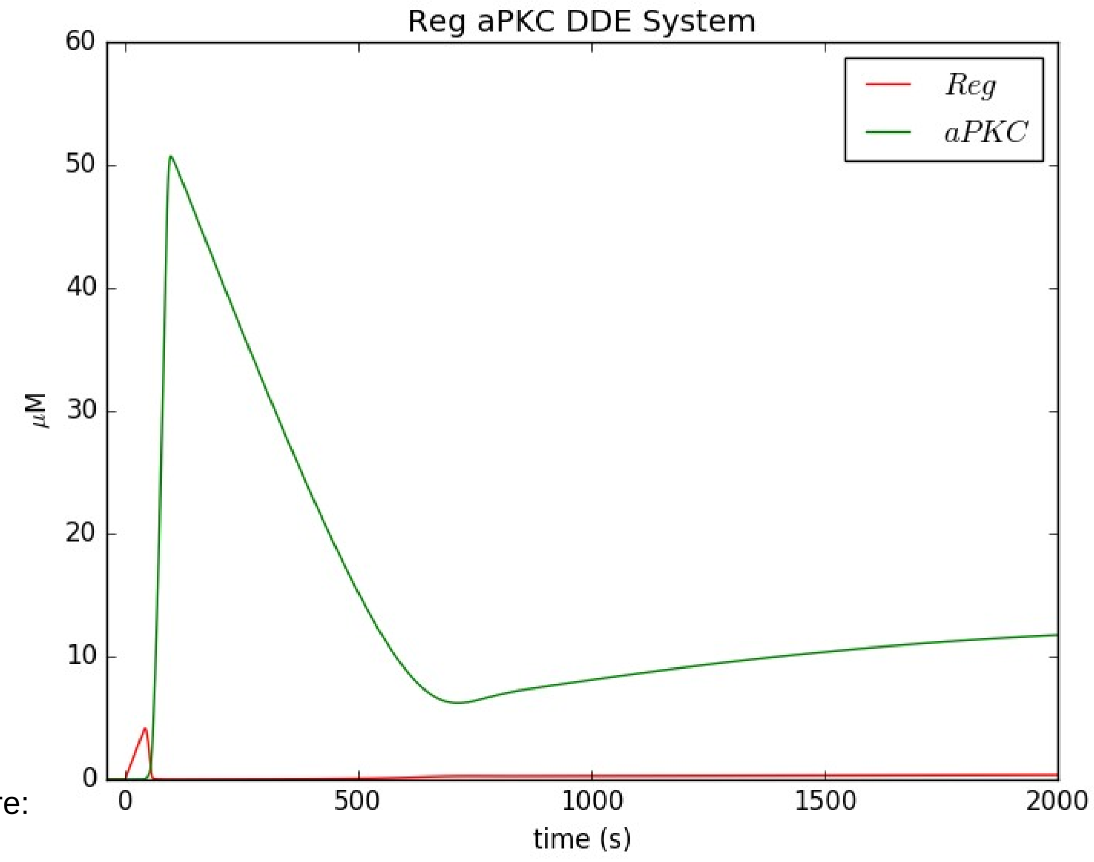
The parameters used were:
{
 "k1": 0.1,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.01,
 "tau1": 40.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Aci": 500,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 10.8731273138

Migration



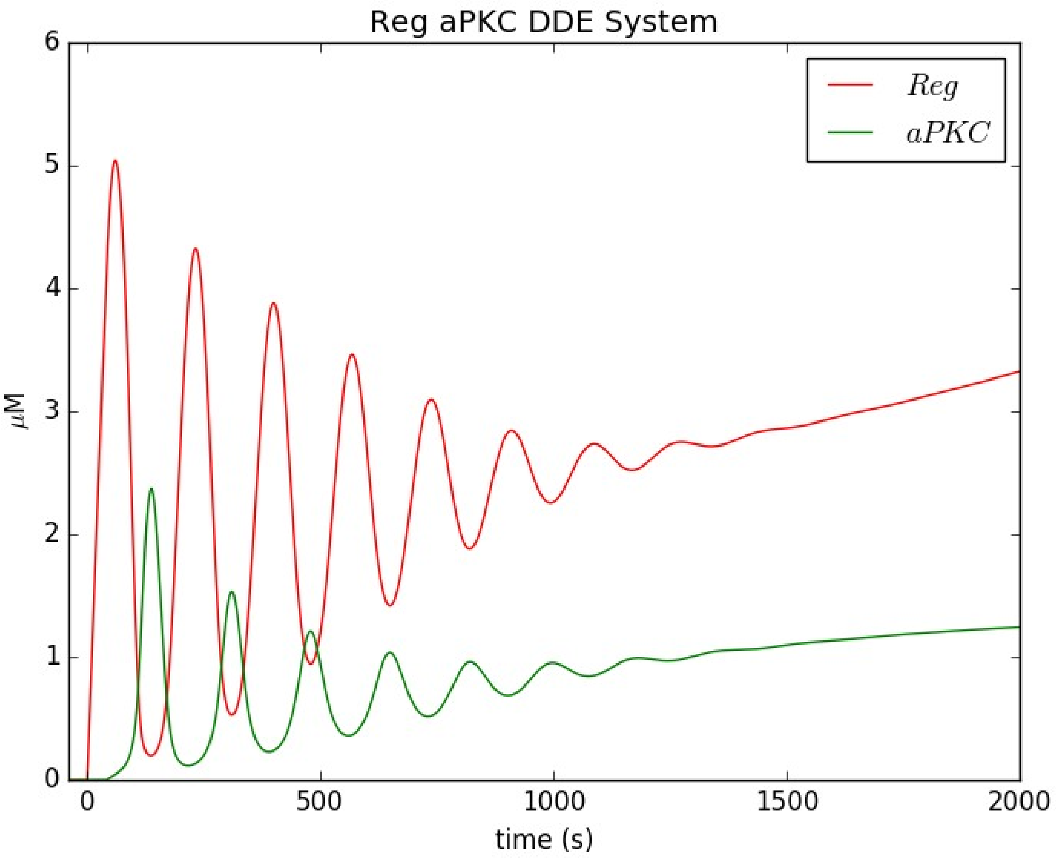
The parameters used were:
{
 "k1": 0.01,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.01,
 "tau1": 40.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Aci": 500,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 1.08731273138

Migration



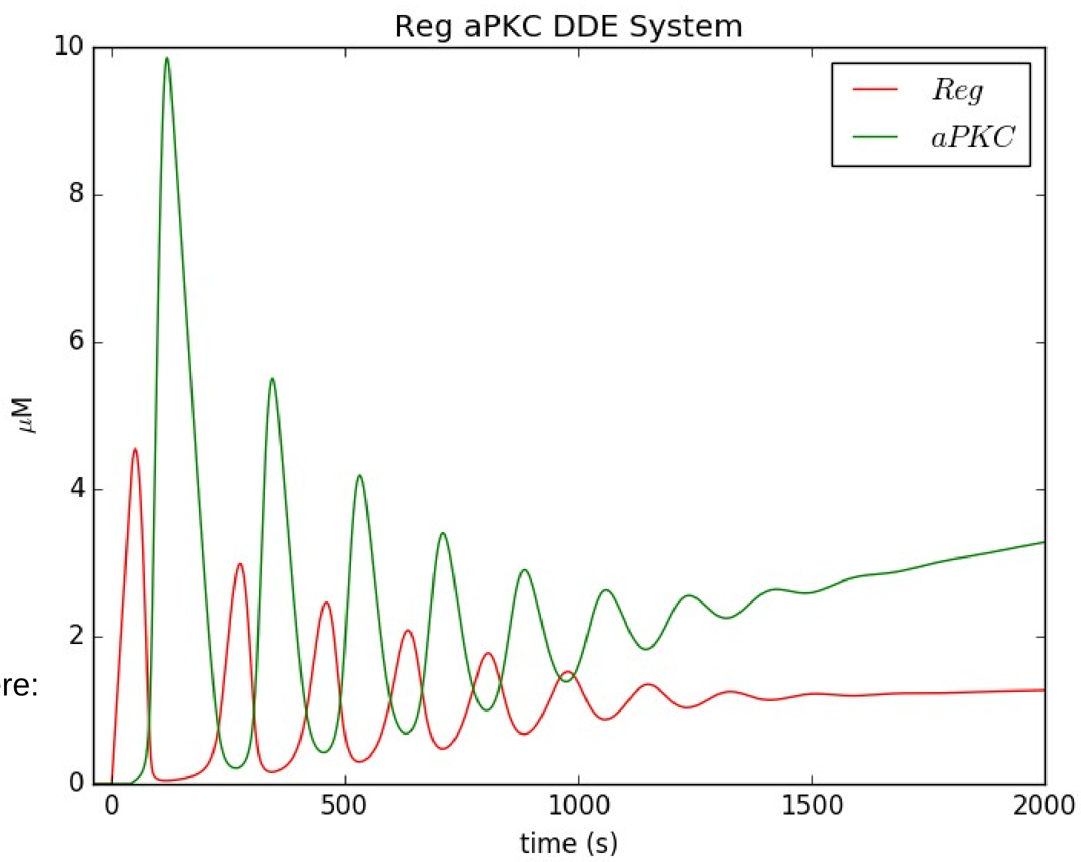
The parameters used were:
{
 "k1": 0.001,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.01,
 "tau1": 40.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Aci": 500,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 0.108731273138

Migration



The parameters used were:
{
 "k1": 0.0001,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.01,
 "tau1": 40.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Aci": 500,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 0.0108731273138

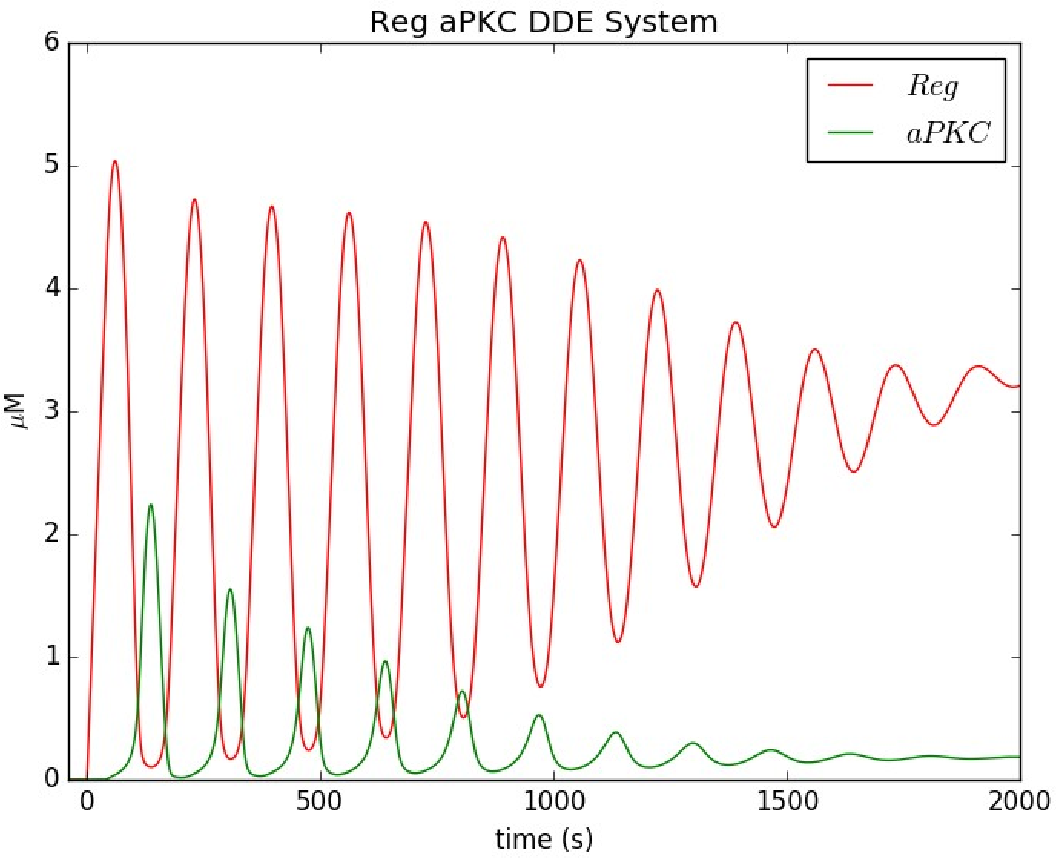
Sustaining Oscillations w/ Migration



The parameters used were:
{
 "k1": 0.0001,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.01,
 "tau1": 40.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Aci": 1000,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 0.0108731273138

We see that increasing Ac can sustain oscillations but the gain is minimal. As we continue to increase this term we would also have to pay the price of smaller k1 value.

Sustaining Oscillations w/ Migration



The parameters used were:
{
 "k1": 0.0001,
 "k2": 0.5,
 "k3": 0.1,
 "k4": 0.001,
 "tau1": 40.0
}
The initial conditions used were:
{
 "ARi": 0.0,
 "Aci": 500,
 "Ai": 0.0,
 "Ri": 0.0
}
k1*tau*e = 0.0108731273138

We see k4 plays a much larger role in sustaining the oscillations.

Conclusions - Implementing Migration

- When implementing migration, k1*R*A is too large and too much (all) aPKC migrates too quickly.
- Hypothesis concerning the oscillatory condition.
  - Basically, now that the birth term is being multiplied by Ac (O(10^2)) I need to decrease k1 by 10^-2. Likewise, multiplication by R which starts out as O(10) thus another 10^-1 and we recover the oscillations.
  - Instead of k1*tau>e, it is possible k1*tau*R*Ac ~ 10-20. This is a rough estimate and running back through all the simulations and obtaining this value could be valuable.
- Increasing initial value of Ac will be able to sustain oscillations longer but the gains are minimal. Increasing Ac to a value to sustain oscillations for the time scale that we need (6000s, length of dorsal closure process), we would have to pay the price of a much smaller k1 value.
- K4 plays a much larger role in sustaining oscillations.

Questions

- Now that we see why k1 and k4 must best be low in order to sustain oscillations, is there a biological story as to why this could be implemented mathematically?
- Should we forgo these modelling assumptions and modify them to match the desired behavior?