Lin & Segel's physiological flow model

From the lecture on 2004–09–01

The two differential equations with two initial conditions: One ($v(0) = 0$) given by the problem, and one ($C(0) = \gamma$) to be used in the shooting method.

```
> ode:=epsilon*Diff(v(x),x)=C(x)-1,
      epsilon*v(x)*C(x)-mu^2*Diff(C(x),x)=epsilon*min(x,1):
  initial:=v(0)=0,C(0)=gamma:
  ode[1]; ode[2]; initial;
```

$$\varepsilon \left(\frac{d}{dx} v(x) \right) = C(x) - 1$$

$$\varepsilon\, v(x)\, C(x) - \mu^2 \left(\frac{d}{dx} C(x) \right) = \varepsilon \min(1, x)$$

$$v(0) = 0,\ C(0) = \gamma$$

Here are parameters to be used. Note that some input below needs to be re-evaluated if these are changed!

```
> params:=epsilon=0.5, mu=10.0, lambda=10.0: params;
```

$$\varepsilon = 0.5,\ \mu = 10.0,\ \lambda = 10.0$$

```
> morepar:=kappa=lambda/mu: morepar;
```

$$\kappa = \frac{\lambda}{\mu}$$

Here is the lowest (zeroth) order term in the expansion for v. I could have used Maple to derive it, but that was not my purpose here, so I just typed in what we got by hand.

```
> v0:=x->piecewise(x<1,x-mu*cosh(kappa-1/mu)*sinh(x/mu)/cosh(kappa),
    1-mu*sinh(1/mu)*cosh(kappa-x/mu)/cosh(kappa)):
  'v0(x)'=v0(x);
```

$$v0(x) = \begin{cases} x - \dfrac{\mu \cosh\left(\kappa - \frac{1}{\mu}\right) \sinh\left(\frac{x}{\mu}\right)}{\cosh(\kappa)} & x < 1 \\ 1 - \dfrac{\mu \sinh\left(\frac{1}{\mu}\right) \cosh\left(-\kappa + \frac{x}{\mu}\right)}{\cosh(\kappa)} & otherwise \end{cases}$$

The first order term for C is the derivative of $v0$.

```
> C1:=D(v0);
```

$$C1 := x \to piecewise\left(x < 1,\ 1 - \frac{\cosh\left(\kappa - \frac{1}{\mu}\right) \cosh\left(\frac{x}{\mu}\right)}{\cosh(\kappa)},\ x = 1,\ undefined,\ 1 < x,\ \frac{\sinh\left(\frac{1}{\mu}\right) \sinh\left(\kappa - \frac{x}{\mu}\right)}{\cosh(\kappa)} \right)$$

Here are the two functions with numerical parameters substituted.

Don't forget to re-evaluate if parameter values were changed!

```
> solv0:=unapply(subs(morepar,params,v0(x)),x);
```

$$solv0 := x \to piecewise\left(x < 1, x - \frac{10.0 \cosh(0.9000000000) \sinh(0.1000000000\, x)}{\cosh(1.000000000)}, 1 - \frac{10.0 \sinh(0.1000000000) \cosh(-1.000000000 + 0.1000000000\, x)}{\cosh(1.000000000)} \right)$$

```
> solC1:=unapply(subs(morepar,params,C1(x)),x);
```

$$solC1 := x \to piecewise\left(x < 1, 1 - \frac{\cosh(0.9000000000) \cosh(0.1000000000\, x)}{\cosh(1.000000000)}, x = 1, undefined, 1 < x, - \frac{\sinh(0.1000000000) \sinh(-1.000000000 + 0.1000000000\, x)}{\cosh(1.000000000)} \right)$$

```
> solC1(0); 1+epsilon*%; subs(params,%);
```

$$0.0712822438$$

$$1 + 0.0712822438\, \varepsilon$$

$$1.035641122$$

The latter number above is a reasonable starting point for shooting.

```
> depvars:=v(x),C(x):
```

This creates function(s) which compute numerical solutions to the full system.

Shooting consists of changing the value of **gamma** in the line below until ...

```
> dsolve(subs(params,gamma=1.035522,[ode,initial]),numeric,[depvars],outpu
  t=listprocedure);
  solv:=subs(%,v(x)): solC:=subs(%%,C(x)):
```

$$[x = \mathbf{proc}(x) \ \ldots \ \mathbf{end\ proc}, v(x) = \mathbf{proc}(x) \ \ldots \ \mathbf{end\ proc}, C(x) = \mathbf{proc}(x) \ \ldots \ \mathbf{end\ proc}]$$

```
> solv(subs(params,lambda)); solC(subs(params,lambda));
```

$$0.349912219011387136$$

$$1.00000065346696188$$

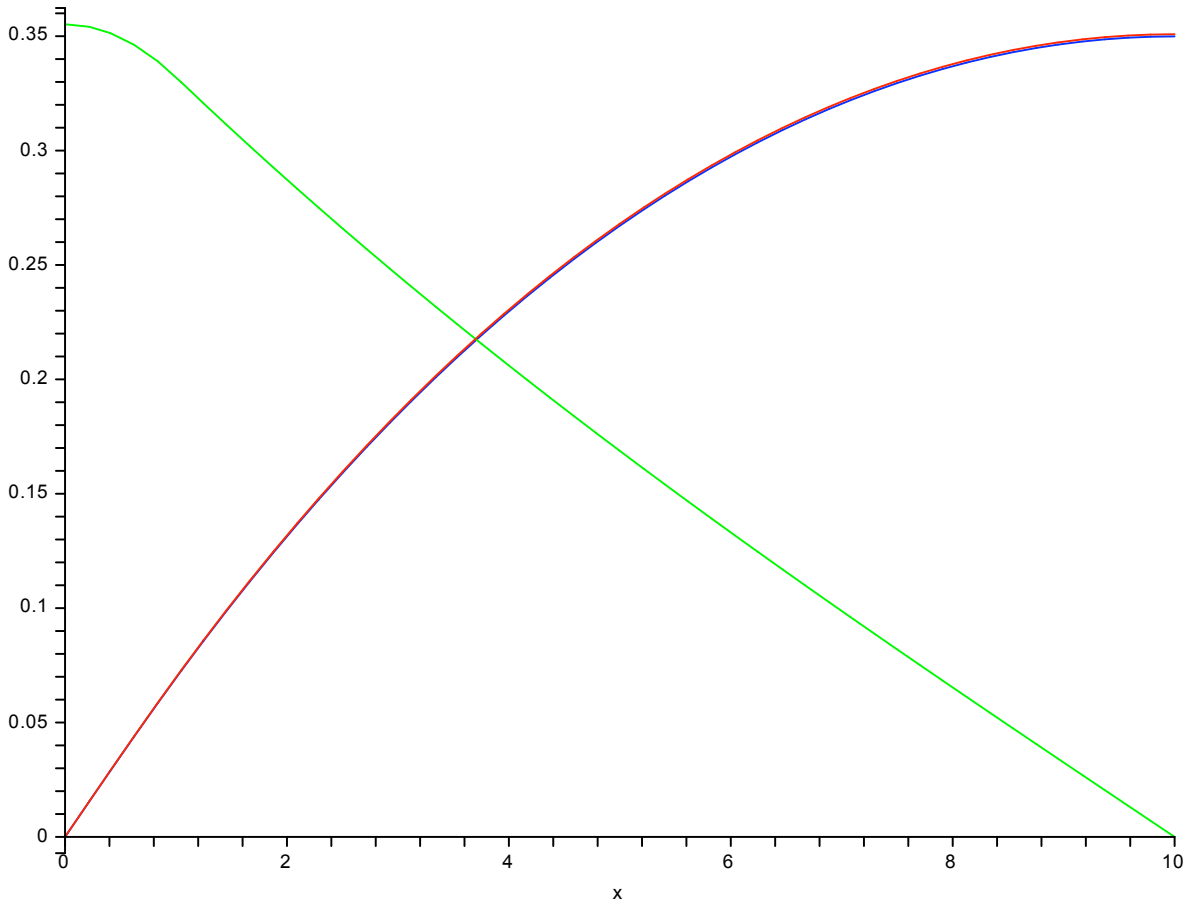
... until the last number above comes close to 1. (Of course, we could have used a numerical method to search efficiently, but a few iterations by hand yield a pretty good value.)

Now plot the data. Use separate plot commands and put the plots together with plots[display] in order to have control of the colors.
The "exact" solution, obtained by the shooting method and solving the equations numerically, is blue.

The perturbation solution, to lowest order in ε, is red.

A scaled version of $C - 1$ is shown in green.

```
> plots[display](
    plot(solv(x),x=0..subs(params,lambda),color=blue),
    plot(solv0(x),x=0..subs(params,lambda),color=red),
    plot(10*(solC(x)-1),x=0..subs(params,lambda),color=green));
```


0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
2
4
6
8
10
x

>