A two-domain model for the Decapentaplegic morphogen gradient in the *Drosophila* wing primordium: Modeling Supplemental

Abed E. Alnaif[1,3], Arthur D. Lander[2,3]
[1]Department of Biomedical Engineering
[2]Department of Developmental and Cell Biology
[3]Center for Complex Biological Systems
[1,2,3]University of California, Irvine

1 General remarks on modeling the Decapentaplegic morphogen gradient

Several mechanisms have been proposed to explain how Decapentaplegic (Dpp) spreads throughout the *Drosophila* wing primordium (wing disc). Nevertheless, it is well established that spreading occurs by a random walk. Ectopic expression of Dpp in a location where Dpp isn't endogenously expressed induces the expression of Dpp target genes (such as *sal* and *omb*) in a circular pattern centered about the cells that are ectopically expressing Dpp [**?**, **?**]. This is evidence that the spreading of Dpp occurs by a random walk, as opposed to directed transport. Thus, even for proposed mechanisms that appear to constitute directed transport of Dpp, such as transport along cellular projections, the process by which Dpp molecules spread must still essentially be of a random walk in nature (e.g., the establishment of the cellular projections could occur by a random walk). All random walk processes are governed by the diffusion equation (described below). Thus, as is conventionally done in modeling morphogen gradients, we proceed with using this equation to model Dpp spreading, without making any assumptions regarding the precise mechanisms by which the spreading occurs, except that it is random and isotropic.

Furthermore, we also follow the convention of considering the problem to be one-dimensional, since the anatomy of the wing disc makes this an adequate approximation.

2 Diffusion equation

The diffusion equation can be derived from applying conservation of mass to Fick's Law, which relates the flux (or net movement), J, of a diffusible molecule (here taken to be the morphogen) to the shape of its concentration profile:

$$J = -D\frac{\partial m(x,t)}{\partial x} \tag{1}$$

where $m(x,t)$ represents the concentration of the morphogen, D is the diffusivity (which describes how quickly the morphogen molecules move back and forth), and $\frac{\partial m(x,t)}{\partial x}$ is the slope of the concentration profile. This equation states that, at any given location, the flux of a molecule performing a random walk is inversely proportional to the slope (or gradient, in the mathematical sense of the word[1]) of the concentration profile at that location, with the proportionality constant being D. The negative sign makes intuitive sense (Fig 1): for a concentration profile having a negative slope (decreasing from left to right), the net movement of molecules would be in the positive direction (to the right), and vice versa.

The diffusion equation in 1D (which can be easily derived by applying conservation of mass to Fick's Law; see "Random Walks in Biology" by Howard Berg and Fig 1) is:

$$\frac{\partial m(x,t)}{\partial t} = D\frac{\partial^2 m(x,t)}{\partial x^2} \tag{2}$$

This states that the change in concentration at any location is proportional to the spatial curvature (at that location) of the concentration profile, with D again being the proportionality constant (Fig 1).

[1] In this document, we refrain from using the word "gradient" according to its mathematical definition of being the slope of some function; instead, we use the word "slope" for this. This is so that we can instead use the word "gradient" for its chemical meaning of describing the function itself. Thus, for example, a phrase such as "morphogen concentration gradient" should be interpreted as being synonymous with "morphogen concentration profile".

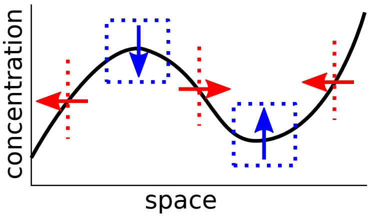


Figure 1: Considering a "toy" concentration profile at a snapshot in time, the red "windows" show that the flux (net movement) of diffusible molecules is in the positive direction (to the right) wherever the slope is negative, and vice versa. From the blue "boxes", we see that where the curvature of the concentration profile is positive (concave up), the fluxes converge and thus diffusion causes the concentration to increase at that location (and vice versa). This assumes that the molecule is only produced and/or consumed at the boundaries, and not anywhere within the boundaries, so that the change in concentration within an imaginary box located anywhere within the boundaries only depends on the fluxes.

3 Source-sink morphogen gradients

It is known from experiments in which Dpp expression is switched off and then back on that the Dpp gradient forms quickly relative to the time scale of development [**?**]. Thus, we follow the convention in morphogen gradient modeling of considering only the time-independent steady-state solutions (when the shape of the gradient stops changing; i.e., when $\frac{\partial m(x,t)}{\partial t} = 0$). Applying this condition to the diffusion equation (Eqn 2):

$$\frac{d^2 m(x,t)}{dx^2} = 0 \tag{3}$$

This states that the concentration profile must have zero curvature at steady state; i.e. it must be a straight line.

The boundary conditions determine the properties of this line (y-intercept and slope). For the left boundary (at the morphogen source), we specify a constant flux of morphogen molecules, which is akin to saying that the morphogen is released from its production region at a constant rate. According to Fick's Law (Eqn. 1), this condition is specified by constraining the slope of the profile at the left boundary:

$$\left.\frac{\partial m(x,t)}{\partial x}\right|_{x=0} = -j \tag{4}$$

where $j \equiv J/D$ is a constant related to the flux.

For a morphogen that is constantly being produced, its gradient will only reach steady-state if the morphogen is consumed by the tissue or leaks out of the tissue; otherwise the concentration of the morphogen will just keep increasing and never reach steady state. According to the source-sink model, the morphogen experiences zero consumption everywhere except at a location far from its source, where there is a "sink" that consumes the morphogen so quickly that its concentration is essentially zero there. A simple way to model the sink is to constrain the concentration of the morphogen to be zero at $x = L$ (with L denoting the location of the sink):

$$m(x = L, t) = 0 \tag{5}$$

For these boundary conditions, the steady-state solution of the system (refer to the attached Mathematica code for calculations) is a line which goes from jL at $x = 0$ to 0 at $x = L$ (Fig 2A):

$$m(x) = j(L - x) \tag{6}$$

Note that the amplitude of the morphogen gradient is proportional to both the morphogen's production rate (j) and the location of the sink (L) (Fig 2D). Both of these make sense intuitively. The amplitude is greater for larger values of j since more morphogen molecules are being released into the tissue. And, for larger values of L, the concentration of the morphogen at the source increases because the sink is located farther away.

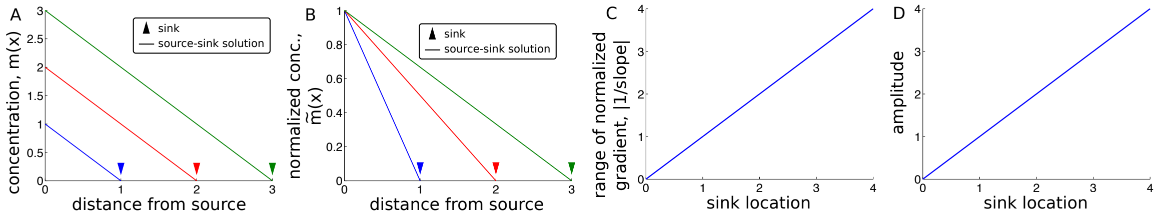


Figure 2: **(A)** Steady-state solutions of the source-sink model (with a constant-flux boundary condition at $x = 0$) for different locations of the sink. **(B)** Steady-state solutions from (A) normalized to the amplitudes of the gradients. **(C,D)** The range and amplitude of the gradient increase proportionally with the sink's distance from the source. The range is taken to be the magnitude of the inverse of the normalized gradient's slope $\left(\left|\frac{1}{d\widetilde{m}(x)/dx}\right|\right)$, and the amplitude is taken to be the concentration near the morphogen source ($m(x = 0)$). Parameters for all panels: $j = 1$.

It is interesting that the steady-state gradient (Equations 3 and 6) only depends on j and L, and not on the diffusivity D. The diffusivity only affects the time to reach steady-state.

Also of importance is the shape of the gradient after it has been normalized to its amplitude. This is important because the normalized gradient more accurately reflects the range of the intercellular communication mediated by the morphogen. For example, considering a particular location within the tissue, an increase in the absolute concentration of a morphogen could be because of an increase in the range or because of an increase in the production rate of the morphogen (or many other reasons). But, an increase in the normalized concentration could only be because of an increase in the morphogen's range of action. Considering the normalized gradients is also important for interpreting the experimental data, since it is hard to accurately measure absolute concentrations using fluorescence microscopy.

Let $\widetilde{m}(x)$ represent the gradient after it has been normalized to its amplitude:

$$\begin{aligned}\widetilde{m}(x) &\equiv \frac{m(x)}{m(x=0)} \\ &= \frac{m(x)}{jL} \\ &= 1 - \frac{x}{L}\end{aligned} \tag{7}$$

The slope of the normalized gradient is $-{}^{1}\!/\!{}_{L}$; thus, as L is increased, the gradient becomes shallower and the range is increased (note that this isn't the case on the absolute scale (Eqn. 6), where the slope is $-j$ and independent of L). In fact, the normalized gradient exhibits perfect scale-invariance with respect to L (Fig 2B,C)[1]. This gives source-sink gradients the unique ability to automatically scale with tissue size: if L is taken to be the size of the tissue, then, as the tissue size grows by some amount, the slope of the gradient will change by exactly the same amount, thus automatically preserving the proportions of patterns in differently sized tissues. This scale-invariance of the normalized solution can be easily seen in Eqn. 7 in that, whereever x appears, it appears as ${}^{x}\!/\!{}_{L}$. The ability of source-sink morphogen gradients to automatically scale with tissue size is why they were originally postulated to underly pattern formation [**?**].

3.1 Concluding remarks on source-sink gradients

For understanding the subsequent sections, the key points to remember about source-sink gradients are that, once they've reached steady-state:

- The gradient is linear, falling to zero at the sink.

[1] The absolute gradient would also exhbit scale-invariance with respect to L if the left boundary condition were to constrain the morphogen concentration, instead of the flux (for example, if the morphogen's production region were able to sense the morphogen concentration and adjust the production rate in order to maintain a fixed concentration). However, in the case of the Dpp gradient, it appears that the production rate of Dpp stays approximately constant over time, though it is known that Dpp signaling within the production region feeds back to inhibit the production of Dpp.

- The amplitude of the gradient increases as the sink is moved farther away from the source.
- The shape of the gradient (after it has been normalized) only depends on the location of the sink, and the range of the gradient increases as the sink is moved farther away from the source.

4 Uniform consumption model

Uniform consumption models lie at the opposite extreme from source-sink models. Whereas source-sink models assume that the morphogen experiences zero consumption everywhere except at the sink (where it is quickly consumed), uniform consumption models assume that the morphogen is consumed throughout the tissue, and moreover that the consumption rate constant is, well, constant throughout space.

This can be modeled by reaction-diffusion equations, in which reaction terms are added to the diffusion equation (Eqn. 2). The simplest way to model consumption of Dpp is according to linear first-order mass-action kinetics, ${}^{dm(t)}/_{dt} = -\alpha m(t)$, where α is the consumption rate constant. Adding this term to the diffusion equation:

$$\frac{\partial m(x,t)}{\partial t} = D\frac{\partial^2 m(x,t)}{\partial x^2} - \alpha m(x,t) \tag{8}$$

where α is assumed to be spatially uniform (i.e., independent of x).

As before, the left boundary condition (at $x = 0$) is assumed to have a constant flux. In order to avoid "boundary effects" (which will be described in greater detail later), let's start by putting the right boundary at infinity ($L = \infty$).

For the same reason stated in Section 3, we will again consider only the steady-state (setting $\frac{\partial m(x,t)}{\partial t} = 0$). With the specified boundary conditions, the steady-state solution of the system is a decaying exponential (Fig 3A; see Mathematica code for calculation):

$$m(x) = m_0 e^{-\frac{x}{\lambda_{exp}}} \tag{9}$$

where $m_0 \equiv m(x = 0) = j\lambda_{exp}$ is the amplitude of the gradient and $\lambda_{exp} \equiv \sqrt{{}^{D}/_{\alpha}}$ is the decay length of the morphogen gradient – the distance over which the morphogen concentration decreases by a factor of ${}^{1}/_{e} = 37\%$ (an exponential function will always change by the same factor over a certain distance, regardless of where on the abscissa we are). Considering the definition of λ_{exp}, an interesting observation is that it's not so much the values of D and α that matter for determining the steady-state shape of the gradient as much as the ratio of the two values, ${}^{D}//_{\alpha}$.

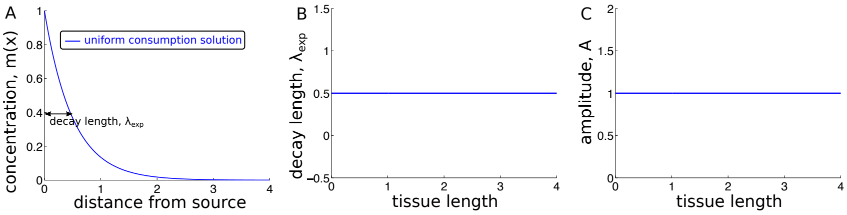


Figure 3: (A) The steady-state solution of the uniform consumption model is a decaying exponential, characterized be a decay length λ_{exp}. **(B,C)** The decay length and amplitude of a uniform consumption gradient do not depend on tissue size. Parameters for all panels: $j = 2$, $\lambda_{exp} = 0.5$.

4.1 Concluding remarks on the uniform consumption model

The concentration gradient of a morphogen experiencing spatially uniform consumption is predicted to be a decaying exponential, characterized by a decay length λ_{exp}. λ_{exp} thus describes the range of the morphogen gradient – the larger the value of λ_{exp} the greater the range.

An important difference between the uniform consumption and source-sink models is what controls the range of the morphogen gradient. The range of a source-sink gradient is entirely determined by the location of the sink, and not by any properties of the morphogen itself. On the other hand, the range of uniform consumption gradients is determined entirely by the properties of the morphogen molecules –

how quickly they diffuse and how quickly they are consumed. Thus, for the uniform consumption model, the range does not scale with tissue size (Fig 3B). Also, whereas the amplitude of source-sink gradients may depend on the location of the sink, those of uniform consumption gradients do not depend on any geometrical properties of the tissue (Fig 3C).

5 Uniform consumption with sink

Next, consider what happens to the solution of the uniform consumption model when the right boundary is no longer assumed to be at infinity. As in the case of the source-sink model (Section 3), assume that the right boundary is a sink, so that the concentration of the morphogen is forced to be zero there: $m(x = L) = 0$. The solutions for $L = 1$ and $L = 2$ are shown in Fig4A[1]. The sink at $x = L$ creates a "boundary effect" in which the morphogen gradient is pulled downwards. Note that this boundary effect propagates into the tissue a distance of approximately λ_{exp} – the *intrinsic* decay length of the morphogen gradient (the decay length the gradient would assume if it weren't for the sink). Thus, the solution closely follows the exponential until approximately λ_{exp} from the sink.

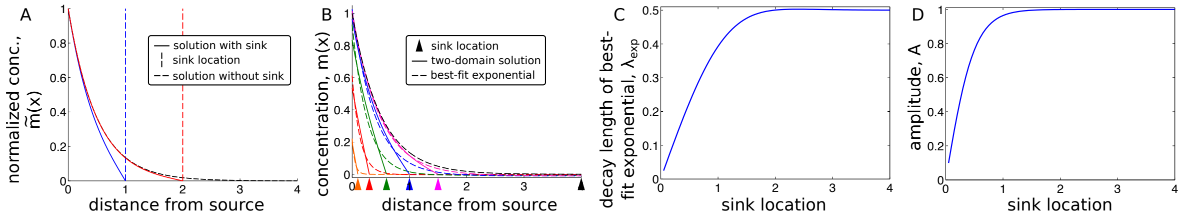


Figure 4: (A) A comparison of the uniform consumption gradient with and without a sink. The gradients have been normalized to their amplitude, in order to allow comparisons of their shapes. The presence of the sink creates a "boundary effect" which pulls down the gradient, thus causing it to become more linear. Note that this boundary effect propagates a distance of approximately $\lambda_{exp} = 0.5$ into the proximal domain. The effect of the sink on the shape of the gradient is stronger when the sink is located closer to the source. **(B)** The solution of the mixed source-sink/uniform consumption model for different locations of the sink. The dashed-curves represent the best-fit decaying exponentials, $y = Ae^{-\frac{x}{\lambda_{exp}}} + b$ (where b is a offset term added to account for the fact that we also include this term when fitting experimental data, due to the presence of background in the signal). Before fitting exponentials, we made sure to specify that $m(x > L) = 0$. **(C,D)** Both the range and amplitude of the gradient increase monotonically and assymptotically as the sink is moved farther away from the source. The range is taken to be the decay length (λ_{exp}) of the best-fit exponential, and the amplitude is taken to be $m(x = 0)$. Parameters for all panels: $j = 2$, $\lambda_{exp} = 0.5$.

When the sink is close to the source (i.e., when L is small), the shape of the morphogen gradient is approximately linear, with a slope that is largely determined by the location of the sink; thus, the gradient behaves much like a source-sink gradient. In contrast, when the sink is far from the source, the value of the exponential is already so small near the sink, and so the sink's effect on the overall gradient is negligible; thus, the gradient behaves like a uniform consumption gradient. And, for intermediate locations of the sink, the gradient exhibits a mix of source-sink and uniform consumption characteristics.

This transition from behaving like a source-sink gradient to behaving like a uniform consumption gradient manifests itself as the location of the sink is moved farther and farther from the source of the morphogen, while all other parameters are kept constant (Fig 4B). Both the range and amplitude of the gradient increase (similar to source-sink gradients), until the sink is far enough from the source that its effect on the gradient becomes negligible (Fig 4C,D).

[1] The steady-state solution of this model is (see attached Mathematica code):

$$m(x) = m_0 \operatorname{csch}\left(\frac{L}{\lambda_{exp}}\right) \sinh\left(\frac{L - x}{\lambda_{exp}}\right) \tag{10}$$

where $m_0 \equiv j\lambda_{exp} \tanh\left(\frac{L}{\lambda_{exp}}\right)$ is the amplitude of the gradient.

5.1 Concluding remarks on the effect of sinks on an otherwise uniform consumption gradient

When a sink is introduced into a uniform consumption gradient, it creates a boundary effect which propagates into the tissue a distance approximately equal to the intrinsic decay length (λ_{exp}) of the morphogen gradient. This causes the gradient to deviate from a purely exponential gradient, towards the linear gradient that is the steady-state solution of the source-sink model. Thus, the morphogen gradient exhibits a mix of source-sink and uniform consumption characteristics, with the balance of this mix depending on the distance of the sink from the source (relative to the intrinsic decay length).

This mixed model is a good starting point for understanding the Dpp gradient, since, we argue, the region distal to the Spalt boundary acts as a sink for Dpp, but Dpp is still consumed with a relatively small rate constant within the Spalt domain (which makes the pure source-sink model inadequate, as it assumes zero consumption and thus $\lambda_{exp} = \infty$). Indeed, this minimal model produces a morphogen gradient whose range increases asymptotically with increasing tissue sizes, a behavior similar to what's observed experimentally (by us and others; see main text) in the case of the Dpp gradient. We will elaborate on this behavior in the next section.

6 Two-domain model

While the previous model is a good starting point for understanding the Dpp gradient, it falls short in that, in the wing disc, the region distal to the Spalt boundary doesn't act as a perfect sink (as the model assumes). Furthermore, the solution of the previous model only extends up to the location of the sink ($x = L$), and thus isn't adequate for fitting to our experimental data (which also includes data beyond the Spalt boundary).

Thus, to model the Dpp gradient, we consider two domains – the "proximal domain" and the "distal domain" – with the morphogen being allowed to take different properties in each of the domains. Let $x = x_B$ be the location of the interface boundary between the two domains, so that $x < x_B$ constitutes the proximal domain and $x > x_B$ constitutes the distal domain. We proceed by specifying a reaction-diffusion equation for each domain:

$$\begin{aligned}\frac{\partial m_p(x,t)}{\partial t} &= D_p\frac{\partial^2 m_p(x,t)}{\partial x^2} - \alpha_p m_p(x,t)\ , \quad 0 \le x \le x_B \\ \frac{\partial m_d(x,t)}{\partial t} &= D_d\frac{\partial^2 m_d(x,t)}{\partial x^2} - \alpha_d m_d(x,t)\ , \quad x > x_B\end{aligned} \tag{11}$$

where subscripts $_p$ and $_d$ represent the proximal and distal domains, respectively.

We use identical boundary conditions to those used to solve for the uniform consumption gradient (Section 4): flux j specified at the left boundary ($x = 0$) and the right boundary put at infinity to avoid boundary effects ($L = \infty$).

To solve this system, we also specify two continuity conditions at the interface boundary $x = x_B$. Specifically, we specify that the gradient should be smooth at $x = x_B$; that is, that both the value and slope of the proximal gradient should match those of the distal gradient at $x = x_B$:

$$\begin{aligned} m_p(x = x_B, t) &= m_d(x = x_B, t) \\ \left.\frac{\partial m_p(x,t)}{\partial x}\right|_{x=x_B} &= \left.\frac{\partial m_d(x,t)}{\partial x}\right|_{x=x_B}\end{aligned} \tag{12}$$

We proceed as before and solve for the steady-state solution (see Section 3 for our argument for why we should be mainly concerned with the steady-state). The steady-state solution is (Fig 5A; see attached Mathematica code for calculations):

$$m(x) = \begin{cases} m_0\frac{\lambda_d\cosh\left(\frac{x-x_B}{\lambda_p}\right)-\lambda_p\sinh\left(\frac{x-x_B}{\lambda_p}\right)}{\lambda_d\cosh\left(\frac{x_B}{\lambda_p}\right)+\lambda_p\sinh\left(\frac{x_B}{\lambda_p}\right)}, & 0 \le x \le x_B \\ m_B e^{-\frac{x-x_B}{\lambda_d}}, & x > x_B \end{cases} \tag{13}$$

where $\lambda_p \equiv \sqrt{D_p/\alpha_p}$ and $\lambda_d \equiv \sqrt{D_d/\alpha_d}$ are the intrinsic decay lengths (by resemblance to the decay length of the steady-state solution of the uniform consumption model, $\lambda_{exp} \equiv \sqrt{D/\alpha}$; see Eqn. 9) of the morphogen gradient in the proximal and distal domains, respectively. And, m_0 and m_B are the concentrations of the morphogen at the morphogen source ($x = 0$) and interface boundary ($x = x_B$), respectively:

$$\begin{aligned} m_0 &\equiv m(x=0) \\ &= j\lambda_p \frac{\lambda_d \cosh\left(\frac{x_B}{\lambda_p}\right) + \lambda_p \sinh\left(\frac{x_B}{\lambda_p}\right)}{\lambda_d \sinh\left(\frac{x_B}{\lambda_p}\right) + \lambda_p \cosh\left(\frac{x_B}{\lambda_p}\right)} \\ m_B &\equiv m(x=x_B) \\ &= \frac{j\lambda_d\lambda_p}{\lambda_d \sinh\left(\frac{x_B}{\lambda_p}\right) + \lambda_p \cosh\left(\frac{x_B}{\lambda_p}\right)} \end{aligned} \tag{14}$$

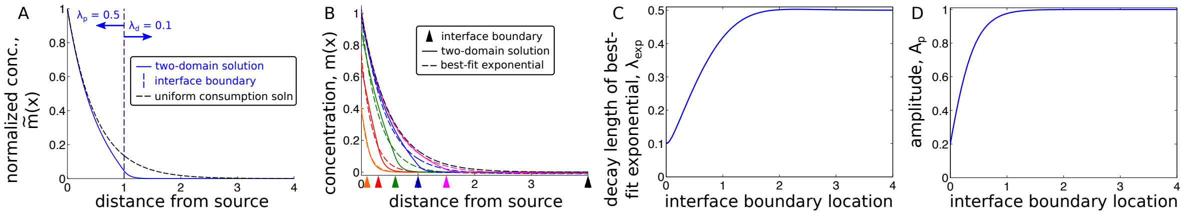


Figure 5: (A) A comparison of the steady-state solutions of the uniform consumption and two-domain models. Both gradients have been normalized to their amplitude, in order to allow comparisons of their shapes. The distal sink pulls down the concentration of the gradient. Due to a "boundary effect" which propagates a distance of approximately $\lambda_p = 0.5$ into the proximal domain, this causes the gradient in the proximal domain to become more linear, as well as a slight kinking of the gradient near the interface boundary. **(B)** The solution of the two-domain model for different locations of the interface boundary. The dashed-curves represent the best-fit decaying exponentials, $y = m_0 e^{-\frac{x}{\lambda_{exp}}} + b$ (where b is a offset term added to account for the fact that we also include this term when fitting experimental data, due to the presence of background in the signal). Note that the two-domain and uniform consumption solutions appear indistinguishable when the interface boundary is either close or far from the source, whereas intermediate locations of interface boundary produce the characteristic downwards kinking of the solution described in (A). **(C,D)** Both the range and amplitude of the gradient increase monotonically and assymptotically as the interface boundary is moved farther away from the source. The range is taken to be the decay length (λ_{exp}) of the best-fit exponential, and the amplitude is taken to be $m(x = 0)$. Parameters for all panels for two-domain model: $j = 2$, $\lambda_p = 0.5$, $\lambda_d = 0.1$. Parameters for Panel A for uniform consumption model: $j = 2$, $\lambda_{exp} = 0.5$.

Several interesting observations can be made regarding the steady-state solutions of the two-domain model (given by Eqns. 13 and 14):

- As in the case of the steady-state solution of the uniform consumption model (Eqn. 9), the solution doesn't depend on the diffusivities D's and consumption rate constants α's per se, but instead on the ratios of those two parameters, λ_p and λ_d.
- From Eqn. 13, we see that the shape of the of the morphogen gradient in the proximal domain ($x < x_B$) depends not only on its intrinsic decay length there (λ_p), but also on its intrinsic decay length in the distal domain (λ_d) and the location of the interface boundary (x_B). However, the shape of the morphogen gradient in the distal domain depends only on its intrinsic decay length there. In fact, within the distal domain, the morphogen gradient decays as a simple exponential function, with decay length λ_d (compare the solution for $x > x_B$ to Eqn. 9).
- Both m_0 (the amplitude of the morphogen gradient; $m(x = 0)$) and m_B (the concentration of the morphogen at the interface boundary and the amplitude of the decaying exponential governing the shape of the morphogen gradient in the distal domain; $m(x = x_B)$) depend on all three of these parameters (λ_p, λ_d, and x_B), in addition to the production rate j.

To make intuitive sense of these results, consider the case when the intrinsic decay length of the morphogen gradient is larger in the proximal domain than in the distal domain ($\lambda_p > \lambda_d$), which we argue to be the case for the Dpp gradient in the *Drosophila* wing disc. Note that $\lambda_p > \lambda_d$ means that there must be differences between the two domains in the consumption rate constants (α's) and/or the diffusivities (D's), such that ${}^{D_p}\!/\!{}_{\alpha_p} > {}^{D_d}\!/\!{}_{\alpha_d}$. When $\lambda_p > \lambda_d$, the distal domain effectively acts as a sink for the proximal domain, albeit an imperfect one (a perfect sink requires that $\lambda_d = 0$).

Even an imperfect sink creates a boundary effect similar to the one observed in the solution of Section 5, with the effect propagating into the proximal domain a distance approximately equal to the morphogen gradient's intrinsic decay length within the proximal domain (λ_p).

Fig 5B illustrates how the behavior of the morphogen gradient changes as the location of the interface boundary ($x = x_B$) is increased (e.g., due to growth), while keeping all other parameters constant, and with the intrinsic decay length in the proximal domain being larger than that in the distal domain ($\lambda_p > \lambda_d$). When the interface boundary is located close to the source (x_B is small), the gradient appears indistinguishible from a short-range exponential governed by the distal intrinsic decay length, λ_d. As the location of the interface boundary is increased, we start to observe the mixed exponential/linear gradient. Note again (see Section 5) that the amount by which the distal sink is able to pull down the concentration of the morphogen is, of course, limited by the concentration of the morphogen which would be present without the effect of the sink. Thus, when the interface boundary is far from the source (x_B is large), the sink doesn't have much of an effect on the morphogen gradient, and the gradient again appears indistinguishible from a pure exponential (but this time the exponential is long-range, being governed by the proximal intrinsic decay length, λ_p).

Recall from Section 3 that a desirable property of source-sink gradients is that their shapes automatically scale with the size of the tissue, whereas uniform consumption gradients exhibit zero scaling (Section 4). Again, with $\lambda_p > \lambda_d$, the two-domain model exhibits a mix of these behaviors. The range of the two-domain gradient automatically increases as the tissue grows, but only up to the point where the effect of the sink on the morphogen gradient becomes negligible (as x_B becomes large) (Fig 5B,C). A similar effect occurs for the amplitude of the gradient: as the location of the sink is moved farther away from the source, the amplitude of the gradient increases until, again, the effect of the sink on the morphogen gradient becomes negligible (Fig 5B,C).

To determine how the amplitude of the morphogen gradient, m_0, changes with respect to x_B, we can calculate the partial derivative of m_0 with respect to x_B (see attached Mathematica code for calculation):

$$\frac{\partial m_0}{\partial x_B} = \frac{j\left(\lambda_p^2 - \lambda_d^2\right)}{\left(\lambda_d \sinh\left(\frac{x_B}{\lambda_p}\right) + \lambda_p \cosh\left(\frac{x_B}{\lambda_p}\right)\right)^2} \tag{15}$$

Note that the denominator is always positive (because of the squaring). Thus, the sign of ${}^{\partial m_0}\!/\!{}_{\partial x_B}$ depends only on the numerator, which is always positive if $\lambda_p > \lambda_d$ and always negative if $\lambda_p < \lambda_d$. We see that, for two-domain gradients, $\lambda_p > \lambda_d$ is both a necessary and sufficient condition for the gradient's amplitude to increase with increasing locations of the interface boundary (and vice versa for the case of $\lambda_p < \lambda_d$). Thus, in the case of the Dpp gradient (which we argue has $\lambda_p > \lambda_d$), we expect the amplitude of the gradient (m_0) to increase monotonically with the location of the interface boundary (x_B).

The fact that, according to this two-domain model, the shape of the morphogen gradient in the proximal domain depends on both λ_p and λ_d is convenient from an experimental point of view, since it allows us to use the data proximal to the Spalt boundary (where our signal-over-background is high) in the estimation of the values for both λ_p and λ_d.

Concluding remarks on the two-domain model

The two-domain model assumes that the morphogen gradient has two domains, each having distinct intrinsic decay lengths. One way by which the two domains can have different decay lengths is if the morphogen were consumed with different rate constants in the two domains (e.g., if the concentration of the morphogen receptors (which are involved in consuming the morphogen) were different in the two domains, which we find to be the case for the Dpp gradient in the wing disc). When the proximal intrinsic decay length is greater than the distal intrinsic decay length ($\lambda_p > \lambda_d$), the distal domain acts as a sink for Dpp, and the ensuing morphogen gradient behaves similarly to the mixed source-sink/uniform consumption gradients described in Section 4. When the interface boundary $x = x_B$ is close enough to the source, the shape of the gradient is largely determined by the location of the interface boundary. But, as the interface boundary is moved farther away, the effect of the distal sink on the gradient becomes lesser, until eventually the gradient appears indistinguishable from that predicted from the uniform consumption model – a pure exponential governed by the proximal intrinsic decay length, λ_p.

While this two-domain model is certainly not an exact model for the Dpp gradient, it is a simple model which (as in Section 4) reproduces behaviors observed (by us and others) for the Dpp gradient in growing wing discs. In particular, it produces a Dpp morphogen gradient whose range increases assymptotically as the size of the tissue increases. However, whereas others have attributed this behavior to feedback processes which modulate the decay length of the gradient, our model produces these behaviors in open-loop fashion. In our model, these behaviors occur automatically as the location of the interface boundary, $x = x_B$, is moved farther away from the Dpp source (which occurs naturally during growth). Since our experimental data further suggest that the interface boundary is the Spalt pattern boundary, our study raises the intriguing possibility that the range of the Dpp morphogen gradient is regulated downstream of the growth of the patterns, in stark contrast to the typical assumption that it is modulation of the range of the morphogen gradient which then regulates the size of patterns.

Sensitivity analysis of the two-domain model

To perform sensitivity analyses, we first solved the two-domain (TD) and uniform consumption (UC) models for the location of the pattern boundary, $x = x_B$, noting that, in the case of the two-domain model, this boundary also corresponds to an interface boundary. We follow the convention in morphogen gradient modeling in assuming that boundaries are determined according to distinct morphogen concentration thresholds; thus, we assume that the boundary ($x = x_B$) occurs where the concentration of the morphogen is $m = m_B$.

Then, we can solve for the location of the boundary as a function of the decay lengths (λ_{exp}, λ_p, and λ_d), the amplitude of the gradient (m_0), and the morphogen threshold (m_B):

$$\begin{aligned}&\text{UC: } x_B = \lambda_{exp} \ln\left(\frac{m_0}{m_B}\right)\\ &\text{TD: } x_B = \lambda_p \left(\ln\left(\sqrt{\lambda_d^2(m_0 - m_B)(m_0 + m_B) + \lambda_p^2 m_B^2} + \lambda_d m_0\right) - \ln(m_B(\lambda_d + \lambda_p))\right)\end{aligned} \tag{16}$$

From here, it is pretty straightforward to determine how the location of the pattern boundary (x_B) varies as a result of variations in either λ_{exp}, λ_p, λ_d, m_0, or m_B. To do this, we used nondimensional Sensitivity Coefficients, S. $S_{X/Y} \equiv \frac{\partial X}{\partial Y}\frac{Y}{X} = \frac{\partial \log X}{\partial \log Y}$ quantifies how X varies as a result of variations in Y (with both variations being measured on relative scales, ${}^{\partial X}/_{X}$ and ${}^{\partial Y}/_{Y}$).

7 Two-domain-gradual-sink model

The last model we will consider is the "two-domain-gradual-sink" model. The two-domain model which was presented in the last section assumes an abrupt, step change in the properties of the morphogen at the interface boundary; i.e., the properties of the morphogen remain constant within either the proximal or distal domains, but they are allowed to differ between the two domains. The two-domain-gradual-sink model, on the other hand, allows the consumption rate constant to increase gradually (rather than abruptly) starting from the interface boundary. The motivation for this model is the observation that, although the level of *tkv* transcription only starts to rise distal to the Spalt boundary, it continues to rise well passed the Sal boundary, suggesting that the distal sink may emerge gradually, rather than abruptly.

To model this, we modify the consumption rate constant within the distal domain to a function that starts from α_p (the consumption rate constant within the proximal domain; the "basal" rate constant) and increases linearly with a slope α_s; i.e., $\alpha_d \equiv \alpha_p + \alpha_s(x - x_B)$:

$$\begin{aligned} \frac{\partial m_p(x,t)}{\partial t} &= D\frac{\partial^2 m_p(x,t)}{\partial x^2} - \alpha_p m_p(x,t) \ , \qquad 0 \le x \le x_B \\ \frac{\partial m_d(x,t)}{\partial t} &= D\frac{\partial^2 m_d(x,t)}{\partial x^2} - (\alpha_p + \alpha_s(x - x_B))m_d(x,t) \ , \qquad x > x_B \end{aligned} \tag{17}$$

We define the following two lumped parameters: (1) the intrinsic decay length of the morphogen in the proximal domain, $\lambda_p \equiv \sqrt{D/\alpha_p}$; and (2) the rate of increase of the consumption rate constant relative to the basal consumption rate constant and to the proximal intrinsic decay length, $q_s \equiv \alpha_s/(\lambda_p^2\alpha_p)$.

We were able to obtain an analytical solution assuming $q_s > 0$ (which means $\alpha_s > 0$). (The case of $q_s < 0$ is more complicated because this would mean the consumption rate constant starts gradually decreasing from the interface boundary, which could result in a negative consumption rate constant.) We refrain from stating the analytical solution here as it is quite complicated; we advise the interested reader to consult the attached Mathematica code, where they can find the solution there.

Fig 6 shows the analytical solution of this model for various values of q_s. Positive values of q_s still have the effect of pulling the morphogen's concentration downwards. However, for small values of q_s, this effect appears to be shifted downfield from the interface boundary, and the kink in the pMad gradient due to the presence of the sink doesn't appear to be as sharp.

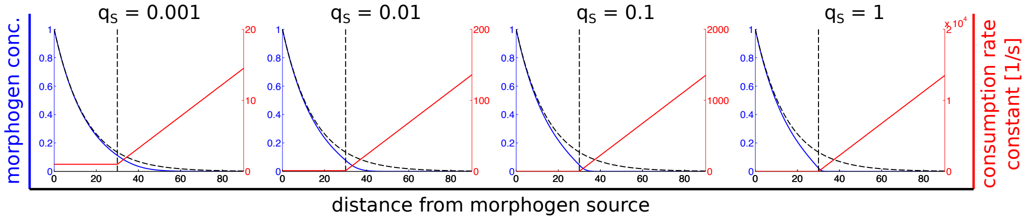


Figure 6: Solutions to the two-domain-gradual-sink model for various values of $q_s \equiv \alpha_s/(\lambda_p^2\alpha_p)$. Besides q_s, all other parameter values are kept constant between the panels; they are: $m_0 = 1$, $x_B = 30$, $\lambda_p = 15$. The vertical dashed line indicates the location of the interface boundary. The curved dashed line indicates the uniform consumption solution ($y = e^{-x/15}$), to make clear how much the sink pulls the concentration of the morphogen down. The red curve indicates the consumption rate constant (α) at any given location. Note that, although the consumption rate constants reach very large values by the time they reach $x = 90$, similar morphogen profiles can be obtained if the consumption rate constant were instead to flatten out some distance downfield of the interface boundary.

7.1 Concluding remarks on the two-domain-gradual-sink model

With the sink emerging only gradually after the interface boundary, the two-domain-gradual-sink model allows greater flexibility than the simpler two-domain model that was presented in the previous section. In particular, by changing the rate by which the sink emerges, we can control where within the distal domain to pull down the morphogen's concentration. Although the analytical solution to this model is much more complicated than that of the simpler two-domain model, the fact that we got an analytical solution means that we can fit this model to our data; see the Supplemental Curve Fitting document to see the results of these fits.

8 Mathematica code

(See next page.)

Dpp gradient modeling in *Mathematica*

Clear the workspace:

```
ClearAll["Global`*"]
```

Diffusion with no consumption (“source-sink” model)

Solving the diffusion equation, $\partial_t m == \delta\, \partial_{x,x} m$, for steady-state by setting $\partial_t m = 0$, we turn the problem into an ODE, which is easily solved by the DSolve command.

The steady-state equation, with diffusion coefficient δ:

```
eqn = {0 == δ m''[x]}
```

```
{0 == δ m′′[x]}
```

Specify the boundary conditions. We will specify a constant flux (j) at the left boundary and a sink at the right boundary:

```
bc = {m'[0] == -j, m[L] == 0}
```

```
{m′[0] == -j, m[L] == 0}
```

Combine the differential equation with the boundary conditions:

```
fullsystem = Join[eqn, bc]
```

```
{0 == δ m′′[x], m′[0] == -j, m[L] == 0}
```

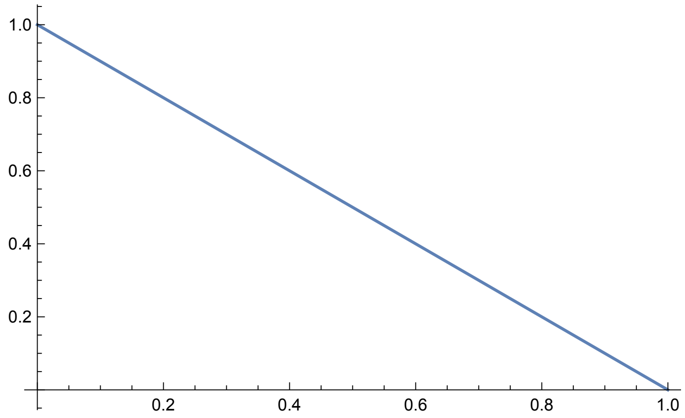
Compute the steady-state solution:

```
sol = DSolve[fullsystem, m[x], {x}] // Flatten // FullSimplify
```

```
{m[x] → j (L - x)}
```

Note that the steady-state solution is just a line (as shown in the plot below), and doesn’t depend on the diffusion coefficient δ. δ only affects the time to reach steady-state (which we are not computing here).

```
Plot[m[x] /. sol /. {j → 1, L → 1}, {x, 0, 1}]
```



Note that the amplitude of the gradient (x == 0) is j*L ; i.e., it is both proportional to j and L.

Diffusion with uniform consumption

Now, let's calculate the steady-state solution for a morphogen-gradient subjected to uniform consumption (with rate constant α) throughout the field:

```
eqn = {0 == δ m''[x] - α m[x]}
```

$\{0 == -\alpha\, m[x] + \delta\, m''[x]\}$

Specify the boundary conditions. Later, we will set L -> ∞ so that the boundary condition at x == L doesn't matter (in order to avoid boundary effects):

```
bc = {m'[0] == -j, m[L] == 0}
```

$\{m'[0] == -j,\ m[L] == 0\}$

```
fullsystem = Join[eqn, bc]
```

$\{0 == -\alpha\, m[x] + \delta\, m''[x],\ m'[0] == -j,\ m[L] == 0\}$

```
sol = DSolve[fullsystem, m[x], {x}] // Flatten // FullSimplify
```

$$\left\{m[x] \to \frac{j \sqrt{\delta}\ \mathrm{Sech}\left[\frac{L\sqrt{\alpha}}{\sqrt{\delta}}\right]\ \mathrm{Sinh}\left[\frac{(L-x)\sqrt{\alpha}}{\sqrt{\delta}}\right]}{\sqrt{\alpha}}\right\}$$

Define $\lambda = \sqrt{\delta/\alpha}$

```
sol = {m[x] → (m[x] /. sol /. δ → λ^2 α)} // FullSimplify
```

$$\left\{m[x] \to j\,\lambda\ \mathrm{Sech}\left[\frac{L}{\lambda}\right]\ \mathrm{Sinh}\left[\frac{L-x}{\lambda}\right]\right\}$$

Specify that λ and L are always positive:

```
parameterConstraints = {λ > 0, L > 0}
```

$\{\lambda > 0,\ L > 0\}$

Calculate the amplitude m(x==0):

```
amplitude =
  {A → Limit[j λ Sech[L/λ] Sinh[(L - x)/λ], x → 0, Assumptions → parameterConstraints]} //
   FullSimplify
```

$$\left\{A \to j\,\lambda\,\mathrm{Tanh}\left[\frac{L}{\lambda}\right]\right\}$$

To determine how the amplitude changes as L is increased, take the derivative of the amplitude with respect to L:

```
dAdL = D[A /. amplitude, L]
```

$$j\,\mathrm{Sech}\left[\frac{L}{\lambda}\right]^2$$

Note that $\partial_L A$is always positive, since j>0 by definition, and the second term is squared. Thus, the amplitude is increases as L is increased. To show that this effect assymptotes for large values of L, show that $\partial_L A \to 0$ as L → ∞:

```
Limit[dAdL, L → Infinity, Assumptions → parameterConstraints]
```

0

Define a replacement rule to obtain a solution which includes the term ‘A’ as the amplitude:

```
amplitudeRR = Solve[A == (A /. amplitude), j] // Flatten // FullSimplify
```

$$\left\{j \to \frac{A\,\mathrm{Coth}\left[\frac{L}{\lambda}\right]}{\lambda}\right\}$$

```
sol = {m[x] → (m[x] /. sol /. amplitudeRR)} // FullSimplify
```

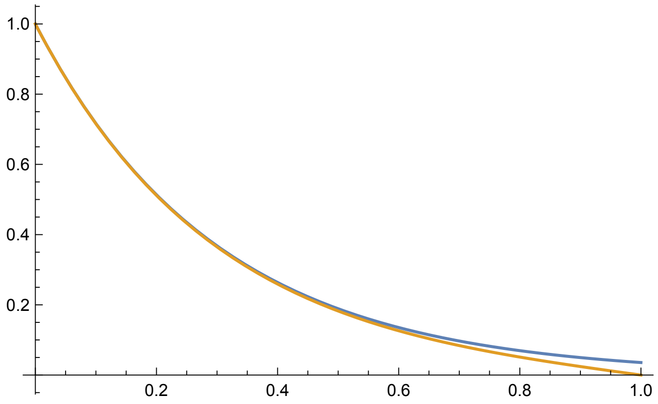
$$\left\{m[x] \to A\,\mathrm{Csch}\left[\frac{L}{\lambda}\right]\,\mathrm{Sinh}\left[\frac{L-x}{\lambda}\right]\right\}$$

Plotting this solution reveals that it looks a lot like a decaying exponential, except near the boundary (due to “boundary effects” arising from the presence of the sink there):

```
parameters = {A → 1, λ → 0.3, L → 1}
```

```
{A → 1, λ → 0.3, L → 1}
```

```
Plot[{A E^(-x/λ) /. parameters, m[x] /. sol /. parameters}, {x, 0, 1}]
```



To avoid “boundary effects” from the right boundary, set L->∞. Note that, in order to compute the limit, we must tell *Mathematica* that $\lambda > 0$

```
sol = {m[x] → Limit[m[x] /. sol, L → Infinity, Assumptions → {λ > 0}]}
```

$$\left\{m[x] \to A\, e^{-\frac{x}{\lambda}}\right\}$$

As L->∞, the solution is a decaying exponential (shown in the plot below). And, the amplitude of the gradient is j*λ:

```
Limit[A /. amplitude, L → Infinity, Assumptions → parameterConstraints]
```

$j\,\lambda$

Two-domain reaction-diffusion model

In the case of Dpp in the wing disc, Spalt inhibits the expression of Dpp’s receptor, so that λ is greater within the Spalt domain (the “proximal domain”) than beyond the Spalt boundary (the “distal domain”). Let’s model this as if each domain had distinct degradation rate constants: αp being the degradation rate constant within the proximal domain and αd being that within the distal domain.

```
eqnProx = {0 == δp m''[x] - αp m[x]}
```

```
{0 == -αp m[x] + δp m''[x]}
```

```
eqnDist = {0 == δd m''[x] - αd m[x]}
```

```
{0 == -αd m[x] + δd m''[x]}
```

Define ' xB' as the location of the "interface boundary" between the two domains.

```
bcProx = {m'[0] == -j, m[xB] == mB}
```

```
{m'[0] == -j, m[xB] == mB}
```

```
bcDist = {m[xB] == mB, m[L] == 0}
```

```
{m[xB] == mB, m[L] == 0}
```

Define λp = $\sqrt{\delta/\alpha\text{p}}$ and λd = $\sqrt{\delta/\alpha\text{d}}$

```
decayLengths = {αp → δp / λp^2 , αd → δd / λd^2}
```

$$\left\{\alpha p \to \frac{\delta p}{\lambda p^2},\ \alpha d \to \frac{\delta d}{\lambda d^2}\right\}$$

Define our parameter constraints:

```
parameterConstraints =
  {αp > 0, αd > 0, δp > 0, δd > 0, λp > 0, λd > 0, mB > 0, m0 > 0, xB > 0}
```

$$\{\alpha p > 0,\ \alpha d > 0,\ \delta p > 0,\ \delta d > 0,\ \lambda p > 0,\ \lambda d > 0,\ mB > 0,\ m0 > 0,\ xB > 0\}$$

```
eqnProx = FullSimplify[eqnProx /. decayLengths, Assumptions → parameterConstraints]
```

$$\{m[x] == \lambda p^2\, m''[x]\}$$

```
eqnDist = FullSimplify[eqnDist /. decayLengths, Assumptions → parameterConstraints]
```

$$\{m[x] == \lambda d^2\, m''[x]\}$$

```
fullsystemProx = Join[eqnProx, bcProx]
```

$$\{m[x] == \lambda p^2\, m''[x],\ m'[0] == -j,\ m[xB] == mB\}$$

```
fullsystemDist = Join[eqnDist, bcDist]
```

$$\{m[x] == \lambda d^2\, m''[x],\ m[xB] == mB,\ m[L] == 0\}$$

```
solProx = FullSimplify[DSolve[fullsystemProx, m[x], {x}],
   Assumptions → parameterConstraints] // Flatten
```

$$\left\{m[x] \to \mathrm{Sech}\left[\frac{xB}{\lambda p}\right]\left(mB\,\mathrm{Cosh}\left[\frac{x}{\lambda p}\right] - j\,\lambda p\,\mathrm{Sinh}\left[\frac{x - xB}{\lambda p}\right]\right)\right\}$$

In solving the equations for the distal domain, we want *Mathematica* to name the integration constants a different character (other than C), since these integration constants are different from the ones in the proximal domain.

```
solDist = FullSimplify[DSolve[fullsystemDist, m[x], {x}, GeneratedParameters → B],
   Assumptions → parameterConstraints] // Flatten
```

$$\left\{m[x] \to mB\,\mathrm{Csch}\left[\frac{L - xB}{\lambda d}\right]\,\mathrm{Sinh}\left[\frac{L - x}{\lambda d}\right]\right\}$$

To avoid boundary effects at x == L, set L -> Infinity. Note that the solution in the distal domain becomes a simple decaying exponential.

```
solDist =
  {m[x] → Limit[m[x] /. solDist, L → Infinity, Assumptions → parameterConstraints]}
```

$$\left\{m[x] \to e^{\frac{-x+xB}{\lambda d}}\, mB\right\}$$

Impose the “continuity condition” that the slopes of the solutions should be the same for both solProx and solDist at the interface boundary:

```
continuityCondition =
 FullSimplify[{(D[m[x] /. solProx, x] /. x → xB) == (D[m[x] /. solDist, x] /. x → xB)},
  Assumptions → parameterConstraints]
```

$$\left\{\text{j}\,\lambda\text{d}\,\lambda\text{p}\,\text{Sech}\left[\frac{\text{xB}}{\lambda\text{p}}\right] == \text{mB}\left(\lambda\text{p}+\lambda\text{d}\,\text{Tanh}\left[\frac{\text{xB}}{\lambda\text{p}}\right]\right)\right\}$$

```
mBsol = FullSimplify[Solve[continuityCondition, mB],
   Assumptions → parameterConstraints] // Flatten
```

$$\left\{\text{mB}\to\frac{\text{j}\,\lambda\text{d}\,\lambda\text{p}\,\text{Sech}\left[\frac{\text{xB}}{\lambda\text{p}}\right]}{\lambda\text{p}+\lambda\text{d}\,\text{Tanh}\left[\frac{\text{xB}}{\lambda\text{p}}\right]}\right\}$$

Substitute for ‘mB’ in our solutions:

```
solProx = FullSimplify[
  {m[x] → (m[x] /. solProx /. mBsol)}, Assumptions → parameterConstraints]
```

$$\left\{\text{m}[\text{x}]\to\frac{\text{j}\,\lambda\text{p}\,\text{Sech}\left[\frac{\text{xB}}{\lambda\text{p}}\right]\left(\lambda\text{d}\,\text{Cosh}\left[\frac{\text{x}-\text{xB}}{\lambda\text{p}}\right]-\lambda\text{p}\,\text{Sinh}\left[\frac{\text{x}-\text{xB}}{\lambda\text{p}}\right]\right)}{\lambda\text{p}+\lambda\text{d}\,\text{Tanh}\left[\frac{\text{xB}}{\lambda\text{p}}\right]}\right\}$$

```
solDist = FullSimplify[
  {m[x] → (m[x] /. solDist /. mBsol)}, Assumptions → parameterConstraints]
```

$$\left\{\text{m}[\text{x}]\to\frac{e^{\frac{-\text{x}+\text{xB}}{\lambda\text{d}}}\,\text{j}\,\lambda\text{d}\,\lambda\text{p}\,\text{Sech}\left[\frac{\text{xB}}{\lambda\text{p}}\right]}{\lambda\text{p}+\lambda\text{d}\,\text{Tanh}\left[\frac{\text{xB}}{\lambda\text{p}}\right]}\right\}$$

Note that, in the distal domain, the solution is simply a decaying exponential (similar to the solution of the model with uniform consumption) with decay length λd, since ‘x’ only appears in the exponent of the exponential. The $\frac{\text{j}\,\lambda\text{d}\,\lambda\text{p}\,\text{Sech}\left[\frac{\text{xB}}{\lambda\text{p}}\right]}{\lambda\text{p}+\lambda\text{d}\,\text{Tanh}\left[\frac{\text{xB}}{\lambda\text{p}}\right]}$ term (=mB) is a constant and constitutes the amplitude of the exponential:

```
amplitudeDist = {mB → m[x] /. solDist /. x → xB} // FullSimplify
```

$$\left\{\text{mB}\to\frac{\text{j}\,\lambda\text{d}\,\lambda\text{p}\,\text{Sech}\left[\frac{\text{xB}}{\lambda\text{p}}\right]}{\lambda\text{p}+\lambda\text{d}\,\text{Tanh}\left[\frac{\text{xB}}{\lambda\text{p}}\right]}\right\}$$

To calculate the amplitude of the gradient, consider solProx:

```
amplitudeProx =
 FullSimplify[{m0 → m[x] /. solProx /. x → 0}, Assumptions → parameterConstraints]
```

$$\left\{\text{m0}\to\frac{\text{j}\,\lambda\text{p}\left(\lambda\text{d}+\lambda\text{p}\,\text{Tanh}\left[\frac{\text{xB}}{\lambda\text{p}}\right]\right)}{\lambda\text{p}+\lambda\text{d}\,\text{Tanh}\left[\frac{\text{xB}}{\lambda\text{p}}\right]}\right\}$$

To determine whether the amplitude of the gradient is predicted to increase or decrease as the location of the interface boundary ‘xB’ is changed, calculate the derivative of the amplitude with respect to xB:

```
dm0dxB =
 FullSimplify[D[m0 /. amplitudeProx, xB], Assumptions → parameterConstraints]
```

$$\frac{j\left(-\lambda d^2+\lambda p^2\right)}{\left(\lambda p\,\mathrm{Cosh}\left[\frac{xB}{\lambda p}\right]+\lambda d\,\mathrm{Sinh}\left[\frac{xB}{\lambda p}\right]\right)^2}$$

The denominator, $\left(\lambda p\,\mathrm{Cosh}\left[\frac{xB}{\lambda p}\right]+\lambda d\,\mathrm{Sinh}\left[\frac{xB}{\lambda p}\right]\right)^2$, is always positive due to the squaring. The numerator is positve when $\lambda p > \lambda d$, and negative when $\lambda p < \lambda d$. Thus, when $\lambda p > \lambda d$, the amplitude of the gradient increases monotonically as the location of the interface boundary ‘xB’ increases. Show that the amplitude of the gradient asymptotes for large values of ‘xB’ by showing that the derivative tends to zero:

```
Limit[dApdxB, xB → Infinity, Assumptions → parameterConstraints]
```

```
dApdxB
```

Obtain some replacement rules so that we can substitute for the amplitudes in ‘solProx’ and ‘solDist’

```
amplitudeDistRR = FullSimplify[Solve[mB == (mB /. amplitudeDist), j],
   Assumptions → parameterConstraints] // Flatten
```

$$\left\{j\to\frac{mB\,\mathrm{Cosh}\left[\frac{xB}{\lambda p}\right]}{\lambda d}+\frac{mB\,\mathrm{Sinh}\left[\frac{xB}{\lambda p}\right]}{\lambda p}\right\}$$

```
amplitudeProxRR = FullSimplify[Solve[m0 == (m0 /. amplitudeProx), j],
   Assumptions → parameterConstraints] // Flatten
```

$$\left\{j\to\frac{m0\left(\lambda p+\lambda d\,\mathrm{Tanh}\left[\frac{xB}{\lambda p}\right]\right)}{\lambda p\left(\lambda d+\lambda p\,\mathrm{Tanh}\left[\frac{xB}{\lambda p}\right]\right)}\right\}$$

```
FullSimplify[amplitudeDist /. amplitudeProxRR, Assumptions → parameterConstraints]
```

$$\left\{mB\to\frac{m0\,\lambda d\,\mathrm{Sech}\left[\frac{xB}{\lambda p}\right]}{\lambda d+\lambda p\,\mathrm{Tanh}\left[\frac{xB}{\lambda p}\right]}\right\}$$

```
solProx = FullSimplify[{m[x] → (m[x] /. solProx /. amplitudeProxRR)},
  Assumptions → parameterConstraints]
```

$$\left\{m[x]\to\frac{m0\,\mathrm{Sech}\left[\frac{xB}{\lambda p}\right]\left(\lambda d\,\mathrm{Cosh}\left[\frac{x-xB}{\lambda p}\right]-\lambda p\,\mathrm{Sinh}\left[\frac{x-xB}{\lambda p}\right]\right)}{\lambda d+\lambda p\,\mathrm{Tanh}\left[\frac{xB}{\lambda p}\right]}\right\}$$

```
solDist = FullSimplify[{m[x] → (m[x] /. solDist /. amplitudeDistRR // FullSimplify)},
  Assumptions → parameterConstraints]
```

$$\left\{m[x]\to e^{\frac{-x+xB}{\lambda d}}\,mB\right\}$$

Now, combine ‘solProx’ and ‘solDist’ to make the full solution. Use the ‘UnitStep’ function to specify that ‘solProx’ should be used for x < xB, and ‘solDist’ for x > xB

```
fullsol = FullSimplify[
  {m[x] → (m[x] /. solProx) UnitStep[-(x - xB)] + (m[x] /. solDist) UnitStep[x - xB]},
  Assumptions → parameterConstraints]
```

$$\left\{\mathrm{m}[\mathrm{x}] \to e^{\frac{-\mathrm{x}+\mathrm{xB}}{\lambda\mathrm{d}}}\,\mathrm{mB}\,\mathrm{UnitStep}[\mathrm{x}-\mathrm{xB}] + \left(\mathrm{m0}\,\mathrm{Sech}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]\left(\lambda\mathrm{d}\,\mathrm{Cosh}\left[\frac{\mathrm{x}-\mathrm{xB}}{\lambda\mathrm{p}}\right] - \lambda\mathrm{p}\,\mathrm{Sinh}\left[\frac{\mathrm{x}-\mathrm{xB}}{\lambda\mathrm{p}}\right]\right)\mathrm{UnitStep}[-\mathrm{x}+\mathrm{xB}]\right) \Big/ \left(\lambda\mathrm{d}+\lambda\mathrm{p}\,\mathrm{Tanh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]\right)\right\}$$

We can put the solution into a more readable form:

```
m[x] /. fullsol // TraditionalForm
```

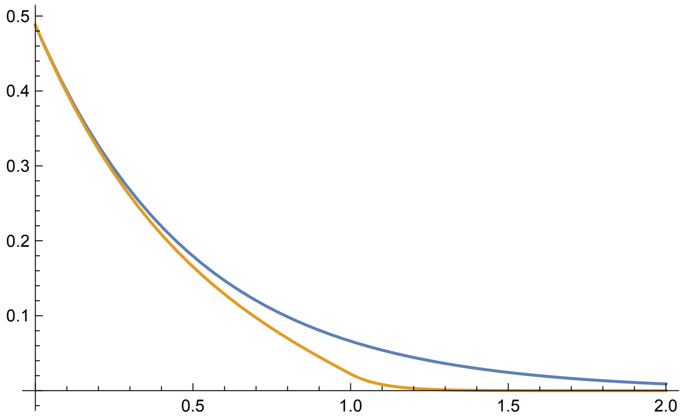
$$\frac{\mathrm{m0}\,\theta(\mathrm{xB}-x)\,\mathrm{sech}\left(\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right)\left(\lambda\mathrm{d}\cosh\left(\frac{x-\mathrm{xB}}{\lambda\mathrm{p}}\right)-\lambda\mathrm{p}\sinh\left(\frac{x-\mathrm{xB}}{\lambda\mathrm{p}}\right)\right)}{\lambda\mathrm{d}+\lambda\mathrm{p}\tanh\left(\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right)} + \mathrm{mB}\,\theta(x-\mathrm{xB})\,e^{\frac{\mathrm{xB}-x}{\lambda\mathrm{d}}}$$

Plot the solution for a certain set of parameter values:

```
parameters = {λp → 0.5, λd → 0.1, xB → 1, j → 1}
```

```
{λp → 0.5, λd → 0.1, xB → 1, j → 1}
```

```
Plot[{m0 E^(-x / λp) /. amplitudeProx /. parameters,
  m[x] /. fullsol /. amplitudeProx /. amplitudeDist /. parameters},
 {x, 0, 2}, PlotRange → Full]
```

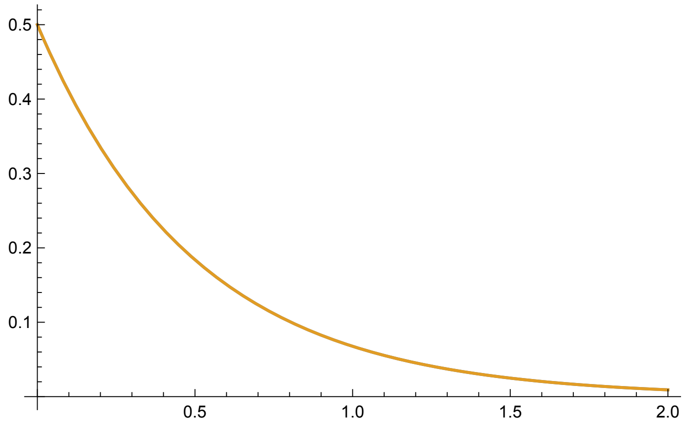


As a sanity check, the solution should match the exponential when λp == λd

```
parameters = {λp → 0.5, λd → 0.5, xB → 1, j → 1}
```

```
{λp → 0.5, λd → 0.5, xB → 1, j → 1}
```

```
Plot[{m0 E^(-x / λp) /. amplitudeProx /. parameters,
  m[x] /. fullsol /. amplitudeProx /. amplitudeDist /. parameters},
 {x, 0, 2}, PlotRange → Full]
```



Sensitivity analyses

To perform sensitivity analyses let's solve for the location 'xB' as a function of all other parameters, using the continuity condition above. First, let's replace the flux 'j' with the gradient amplitude 'm0', so that we can specify uniform consumption and two-domain models having the same amplitudes. This doesn't effect our results, since, for both the uniform consumption and two-domain gradients, 'm0' is linearly proportional 'j', and so the sensitivity of the gradient to variations in 'm0' is identical to that to variations in 'j'.

```
continuityCondition
```

$$\left\{\text{j}\ \lambda\text{d}\ \lambda\text{p}\ \text{Sech}\left[\frac{\text{xB}}{\lambda\text{p}}\right] == \text{mB}\left(\lambda\text{p} + \lambda\text{d}\ \text{Tanh}\left[\frac{\text{xB}}{\lambda\text{p}}\right]\right)\right\}$$

```
continuityConditionAmplitudeSpecified = FullSimplify[
  continuityCondition /. amplitudeProxRR, Assumptions → parameterConstraints]
```

$$\left\{\text{m0}\ \lambda\text{d}\ \text{Sech}\left[\frac{\text{xB}}{\lambda\text{p}}\right] == \text{mB}\left(\lambda\text{d} + \lambda\text{p}\ \text{Tanh}\left[\frac{\text{xB}}{\lambda\text{p}}\right]\right)\right\}$$

Now, solve for 'xB'.

```
xBsolTD = FullSimplify[Solve[continuityConditionAmplitudeSpecified, xB],
   Assumptions → parameterConstraints] // Flatten
```

$$\Big\{\text{xB} \to \text{ConditionalExpression}\Big[\lambda\text{p}\left(2\,i\,\pi\,\text{C}[1] - \text{Log}[\text{mB}\ (\lambda\text{d} + \lambda\text{p})] + \text{Log}\left[\text{m0}\ \lambda\text{d} - \sqrt{(\text{m0} - \text{mB})\ (\text{m0} + \text{mB})\ \lambda\text{d}^2 + \text{mB}^2\ \lambda\text{p}^2}\right]\right),$$
$$\text{C}[1] \in \text{Integers}\Big],\ \text{xB} \to \text{ConditionalExpression}\Big[$$
$$\lambda\text{p}\left(2\,i\,\pi\,\text{C}[1] - \text{Log}[\text{mB}\ (\lambda\text{d} + \lambda\text{p})] + \text{Log}\left[\text{m0}\ \lambda\text{d} + \sqrt{(\text{m0} - \text{mB})\ (\text{m0} + \text{mB})\ \lambda\text{d}^2 + \text{mB}^2\ \lambda\text{p}^2}\right]\right),$$
$$\text{C}[1] \in \text{Integers}\Big]\Big\}$$

```
xBsolTD = {xBsolTD /. C[1] → 0 // Last}
```

$$\left\{\mathrm{xB}\to\lambda\mathrm{p}\left(-\mathrm{Log}[\mathrm{mB}\ (\lambda\mathrm{d}+\lambda\mathrm{p})]+\mathrm{Log}\left[\mathrm{m0}\ \lambda\mathrm{d}+\sqrt{(\mathrm{m0}-\mathrm{mB})\ (\mathrm{m0}+\mathrm{mB})\ \lambda\mathrm{d}^2+\mathrm{mB}^2\ \lambda\mathrm{p}^2}\right]\right)\right\}$$

This solution corresponds to the two-domain model. Now, solve for ‘xB’ for the uniform consumption model:

```
xBsolExp = Solve[mB == m0 E^(-xB / λe), xB] // Flatten
```

$$\left\{\mathrm{xB}\to\mathrm{ConditionalExpression}\left[\lambda\mathrm{e}\left(2\,i\,\pi\,\mathrm{C}[1]+\mathrm{Log}\left[\frac{\mathrm{m0}}{\mathrm{mB}}\right]\right),\ \mathrm{C}[1]\in\mathrm{Integers}\right]\right\}$$

```
xBsolExp = {xB → (xB /. xBsolExp /. C[1] → 0)}
```

$$\left\{\mathrm{xB}\to\lambda\mathrm{e}\ \mathrm{Log}\left[\frac{\mathrm{m0}}{\mathrm{mB}}\right]\right\}$$

With these two solutions, it is relatively straightforward to calculate the sensitivity of ‘xB’ to variations in other parameters.

Two-domain-gradual-sink model

Now, we will modify the two-domain model so that the strength of the sink increases gradually starting from the interface boundary. The motivation for this model is the observation that the expression of Dpp’s cell-surface receptor Tkv doesn’t appear to increase as a step function at the Sal boundary; rather, it appears to increase gradually starting from the Sal boundary. Let’s model this by assuming that the consumption rate constant of the morphogen increases as a linear function starting from the interface boundary, and the slope of this increase is αm.

```
eqnProx = {0 == δ m''[x] - αp m[x]}
```

$$\{0=-\alpha\mathrm{p}\ \mathrm{m}[\mathrm{x}]+\delta\ \mathrm{m}''[\mathrm{x}]\}$$

```
eqnDist = {0 == δ m''[x] - (αp + (x - xB) αs) m[x]}
```

$$\{0=-(\alpha\mathrm{p}+(\mathrm{x}-\mathrm{xB})\ \alpha\mathrm{s})\ \mathrm{m}[\mathrm{x}]+\delta\ \mathrm{m}''[\mathrm{x}]\}$$

```
bcProx = {m[0] == m0, m[xB] == mB}
```

$$\{\mathrm{m}[0]=\mathrm{m0},\ \mathrm{m}[\mathrm{xB}]=\mathrm{mB}\}$$

```
bcDist = {m[xB] == mB, m[L] == 0}
```

$$\{\mathrm{m}[\mathrm{xB}]=\mathrm{mB},\ \mathrm{m}[\mathrm{L}]=0\}$$

Define λp = $\sqrt{\delta/\alpha\mathrm{p}}$ and qs = αs / (λp^2 αp) = αs / δ

```
decayLengths = {αp → δ / λp^2, αs → qs δ}
```

$$\left\{\alpha\mathrm{p}\to\frac{\delta}{\lambda\mathrm{p}^2},\ \alpha\mathrm{s}\to\mathrm{qs}\ \delta\right\}$$

Define the constraints on the parameters. Let’s assume αm (and qm) are greater than zero. (If we let them get less than zero, we risk getting a negative consumption rate constant.)

```
parameterConstraints = {δ > 0, αp > 0, λp > 0, 0 < xB < L, qs > 0, ψ >  0, ϕ + xB ψ > 0}
```

$$\{\delta > 0,\ \alpha p > 0,\ \lambda p > 0,\ 0 < xB < L,\ qs > 0,\ \psi > 0,\ \phi + xB\,\psi > 0\}$$

```
eqnProx = FullSimplify[eqnProx /. decayLengths, Assumptions → parameterConstraints]
```

$$\left\{m[x] == \lambda p^2\, m''[x]\right\}$$

```
eqnDist = FullSimplify[eqnDist /. decayLengths, Assumptions → parameterConstraints]
```

$$\left\{\left(1 + qs\ (x - xB)\ \lambda p^2\right) m[x] == \lambda p^2\, m''[x]\right\}$$

```
Solve[eqnDist, m''[x]] // FullSimplify
```

$$\left\{\left\{m''[x] \to \left(qs\ (x - xB) + \frac{1}{\lambda p^2}\right) m[x]\right\}\right\}$$

Let $\phi = \frac{1}{\lambda p^2}$-xB qs and ψ = qs

$$\texttt{lumpedParameters} = \left\{\phi \to \frac{1}{\lambda p^2} - xB\,qs,\ \psi \to qs\right\}$$

$$\left\{\phi \to -qs\ xB + \frac{1}{\lambda p^2},\ \psi \to qs\right\}$$

```
lumpedParameters2 =
  Solve[{ϕ == (ϕ /. lumpedParameters), ψ == (ψ /. lumpedParameters)}, {λp, qs}] // Last //
   FullSimplify
```

$$\left\{\lambda p \to \frac{1}{\sqrt{\phi + xB\,\psi}},\ qs \to \psi\right\}$$

```
eqnDist =
  FullSimplify[eqnDist /. lumpedParameters2, Assumptions → parameterConstraints]
```

$$\{(\phi + x\,\psi)\ m[x] == m''[x]\}$$

```
fullsystemProx = Join[eqnProx, bcProx]
```

$$\left\{m[x] == \lambda p^2\, m''[x],\ m[0] == m0,\ m[xB] == mB\right\}$$

```
fullsystemDist = Join[eqnDist, bcDist]
```

$$\{(\phi + x\,\psi)\ m[x] == m''[x],\ m[xB] == mB,\ m[L] == 0\}$$

```
solProx = FullSimplify[DSolve[fullsystemProx, m[x], {x}],
   Assumptions → parameterConstraints] // Flatten
```

$$\left\{m[x] \to \mathrm{Csch}\left[\frac{xB}{\lambda p}\right] \left(mB\ \mathrm{Sinh}\left[\frac{x}{\lambda p}\right] - m0\ \mathrm{Sinh}\left[\frac{x - xB}{\lambda p}\right]\right)\right\}$$

```
solDist = FullSimplify[DSolve[fullsystemDist, m[x], {x}],
   Assumptions → parameterConstraints] // Flatten
```

$$\left\{m[x] \to \left(\mathrm{mB}\left(\mathrm{AiryAi}\left[\frac{\phi + x\,\psi}{\psi^{2/3}}\right]\mathrm{AiryBi}\left[\frac{\phi + L\,\psi}{\psi^{2/3}}\right] - \mathrm{AiryAi}\left[\frac{\phi + L\,\psi}{\psi^{2/3}}\right]\mathrm{AiryBi}\left[\frac{\phi + x\,\psi}{\psi^{2/3}}\right]\right)\right) \Big/ \left(\mathrm{AiryAi}\left[\frac{\phi + \mathrm{xB}\,\psi}{\psi^{2/3}}\right]\mathrm{AiryBi}\left[\frac{\phi + L\,\psi}{\psi^{2/3}}\right] - \mathrm{AiryAi}\left[\frac{\phi + L\,\psi}{\psi^{2/3}}\right]\mathrm{AiryBi}\left[\frac{\phi + \mathrm{xB}\,\psi}{\psi^{2/3}}\right]\right)\right\}$$

```
solDist =
 {m[x] → Limit[m[x] /. solDist, L → Infinity, Assumptions → parameterConstraints]}
```

$$\left\{m[x] \to \frac{\mathrm{mB}\,\mathrm{AiryAi}\left[\frac{\phi+x\,\psi}{\psi^{2/3}}\right]}{\mathrm{AiryAi}\left[\frac{\phi+\mathrm{xB}\,\psi}{\psi^{2/3}}\right]}\right\}$$

As before, couple the solutions in the proximal and distal domain using the continuity condition.

```
continuityCond =
 {(D[(m[x] /. solProx), x] /. x → xB) == (D[(m[x] /. solDist), x] /. x → xB)}
```

$$\left\{\left(-\frac{\mathrm{m0}}{\lambda\mathrm{p}} + \frac{\mathrm{mB}\,\mathrm{Cosh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]}{\lambda\mathrm{p}}\right)\mathrm{Csch}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right] == \frac{\mathrm{mB}\,\psi^{1/3}\,\mathrm{AiryAiPrime}\left[\frac{\phi+\mathrm{xB}\,\psi}{\psi^{2/3}}\right]}{\mathrm{AiryAi}\left[\frac{\phi+\mathrm{xB}\,\psi}{\psi^{2/3}}\right]}\right\}$$

```
mBsol = FullSimplify[Solve[continuityCond, mB],
   Assumptions → parameterConstraints] // Flatten
```

$$\left\{\mathrm{mB} \to \frac{\mathrm{m0}}{\mathrm{Cosh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right] - \frac{\lambda\mathrm{p}\,\psi^{1/3}\,\mathrm{AiryAiPrime}\left[\frac{\phi+\mathrm{xB}\,\psi}{\psi^{2/3}}\right]\mathrm{Sinh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]}{\mathrm{AiryAi}\left[\frac{\phi+\mathrm{xB}\,\psi}{\psi^{2/3}}\right]}}\right\}$$

```
solProxCoupled = FullSimplify[solProx /. mBsol, Assumptions → parameterConstraints]
```

$$\left\{m[x] \to \mathrm{m0}\,\mathrm{Csch}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]\left(-\mathrm{Sinh}\left[\frac{x - \mathrm{xB}}{\lambda\mathrm{p}}\right] + \frac{\mathrm{Sinh}\left[\frac{x}{\lambda\mathrm{p}}\right]}{\mathrm{Cosh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right] - \frac{\lambda\mathrm{p}\,\psi^{1/3}\,\mathrm{AiryAiPrime}\left[\frac{\phi+\mathrm{xB}\,\psi}{\psi^{2/3}}\right]\mathrm{Sinh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]}{\mathrm{AiryAi}\left[\frac{\phi+\mathrm{xB}\,\psi}{\psi^{2/3}}\right]}}\right)\right\}$$

```
solDistCoupled = FullSimplify[solDist /. mBsol, Assumptions → parameterConstraints]
```

$$\left\{m[x] \to \left(\mathrm{m0}\,\mathrm{AiryAi}\left[\frac{\phi + x\,\psi}{\psi^{2/3}}\right]\right) \Big/ \left(\mathrm{AiryAi}\left[\frac{\phi + \mathrm{xB}\,\psi}{\psi^{2/3}}\right]\mathrm{Cosh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right] - \lambda\mathrm{p}\,\psi^{1/3}\,\mathrm{AiryAiPrime}\left[\frac{\phi + \mathrm{xB}\,\psi}{\psi^{2/3}}\right]\mathrm{Sinh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]\right)\right\}$$

```
solProxCoupled = {m[x] → FullSimplify[
    m[x] /. solProxCoupled /. lumpedParameters, Assumptions → parameterConstraints]}
```

$$\left\{\mathrm{m}[\mathrm{x}] \to \mathrm{m0}\ \mathrm{Csch}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]\left(-\mathrm{Sinh}\left[\frac{\mathrm{x}-\mathrm{xB}}{\lambda\mathrm{p}}\right]+\frac{\mathrm{Sinh}\left[\frac{\mathrm{x}}{\lambda\mathrm{p}}\right]}{\mathrm{Cosh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]-\frac{\mathrm{qs}^{1/3}\,\lambda\mathrm{p}\ \mathrm{AiryAiPrime}\left[\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right]\mathrm{Sinh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]}{\mathrm{AiryAi}\left[\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right]}}\right)\right\}$$

```
solDistCoupled = {m[x] → FullSimplify[
    m[x] /. solDistCoupled /. lumpedParameters, Assumptions → parameterConstraints]}
```

$$\left\{\mathrm{m}[\mathrm{x}] \to \frac{\mathrm{m0}\ \mathrm{AiryAi}\left[\frac{\mathrm{qs}\ (\mathrm{x}-\mathrm{xB})+\frac{1}{\lambda\mathrm{p}^2}}{\mathrm{qs}^{2/3}}\right]}{\mathrm{AiryAi}\left[\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right]\mathrm{Cosh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]-\mathrm{qs}^{1/3}\,\lambda\mathrm{p}\ \mathrm{AiryAiPrime}\left[\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right]\mathrm{Sinh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]}\right\}$$

The full solution:

```
fullsol = FullSimplify[{ m[x] → (m[x] /. solProxCoupled) UnitStep[- (x - xB)] +
     (m[x] /. solDistCoupled) UnitStep[x - xB]}, Assumptions → parameterConstraints]
```

$$\Big\{\mathrm{m}[\mathrm{x}] \to \left(\mathrm{m0}\left(\mathrm{AiryAi}\left[\frac{\mathrm{qs}\ (\mathrm{x}-\mathrm{xB})+\frac{1}{\lambda\mathrm{p}^2}}{\mathrm{qs}^{2/3}}\right]\mathrm{UnitStep}[\mathrm{x}-\mathrm{xB}]+\left(\mathrm{AiryAi}\left[\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right]\mathrm{Cosh}\left[\frac{\mathrm{x}-\mathrm{xB}}{\lambda\mathrm{p}}\right]+\mathrm{qs}^{1/3}\,\lambda\mathrm{p}\ \mathrm{AiryAiPrime}\left[\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right]\mathrm{Sinh}\left[\frac{\mathrm{x}-\mathrm{xB}}{\lambda\mathrm{p}}\right]\right)\mathrm{UnitStep}[-\mathrm{x}+\mathrm{xB}]\right)\right)\Big/ \left(\mathrm{AiryAi}\left[\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right]\mathrm{Cosh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]-\mathrm{qs}^{1/3}\,\lambda\mathrm{p}\ \mathrm{AiryAiPrime}\left[\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right]\mathrm{Sinh}\left[\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right]\right)\Big\}$$

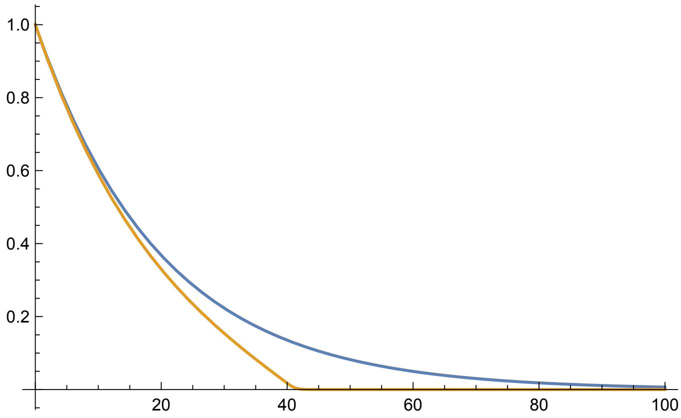
We can put the solution into a more readable form:

```
m[x] /. fullsol // TraditionalForm
```

$$\frac{\mathrm{m0}\left(\theta(x-\mathrm{xB})\ \mathrm{Ai}\left(\frac{\mathrm{qs}\,(x-\mathrm{xB})+\frac{1}{\lambda\mathrm{p}^2}}{\mathrm{qs}^{2/3}}\right)+\theta(\mathrm{xB}-x)\left(\mathrm{Ai}\left(\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right)\cosh\left(\frac{x-\mathrm{xB}}{\lambda\mathrm{p}}\right)+\lambda\mathrm{p}\ \sqrt[3]{\mathrm{qs}}\ \mathrm{Ai}'\left(\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right)\sinh\left(\frac{x-\mathrm{xB}}{\lambda\mathrm{p}}\right)\right)\right)}{\mathrm{Ai}\left(\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right)\cosh\left(\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right)-\lambda\mathrm{p}\ \sqrt[3]{\mathrm{qs}}\ \mathrm{Ai}'\left(\frac{1}{\mathrm{qs}^{2/3}\,\lambda\mathrm{p}^2}\right)\sinh\left(\frac{\mathrm{xB}}{\lambda\mathrm{p}}\right)}$$

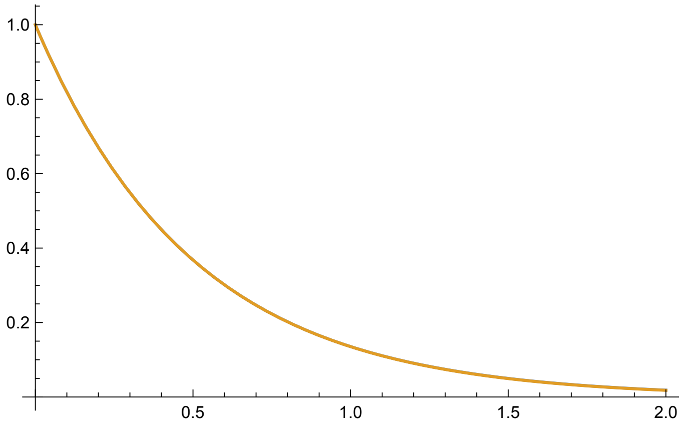
Plot the solution for a certain set of parameter values:

```
parameters = {λp → 20, qs → 1, xB → 40, m0 → 1};
Plot[{m0 E^(-x / λp) /. parameters, m[x] /. fullsol /. parameters},
 {x, 0, 100}, PlotRange → Full]
```



As a sanity check, the solution should match the exponential as qm -> 0. Note the effect of qm depends on the value of λp.

```
parameters = {λp → 0.5, qs → 0.1, xB → 1, m0 → 1};
Plot[{m0 E^(-x / λp) /. parameters, m[x] /. fullsol /. parameters},
 {x, 0, 2}, PlotRange → Full]
```


1.0
0.8
0.6
0.4
0.2
0.5
1.0
1.5
2.0