

# *The Quantitation of Non-classical Buffering*

## *Applying the Formal and General Approach to Problems in the Biological Sciences*

**Jim Karagiannis**

A formal, axiomatic conceptualization of buffering action—generally applicable to any physical, chemical, or biological process—was first presented by B. M. Schmitt in 2005 [1, 2]. This article provides a series of tools designed to aid in the application of these concepts to both classical and non-classical buffering phenomena. To illustrate the utility of the approach in the biological sciences, an abstract measure of the magnitude of “genetic” buffering associated with an allele of the gene encoding the heat shock protein Hsp90 is determined.

### ■ Introduction

Buffering is observed when a parameter changes less than expected in response to a given disturbance. For example, if a strong acid is added to an aqueous solution, and not all of the added  $H^+$  ions remain free (unbound), then the solution is said to act as a buffer. While such phenomena have been observed in the physical, chemical, and biological sciences for centuries, a formal and general approach for their quantitation across distinct disciplines was not presented until recently [1, 2].

In [1, 2], B. M. Schmitt presents a comprehensive mathematical framework for evaluating buffering action—a framework that is applicable to any scenario in which a quantity partitions between two compartments or states. While suitable for the analysis of “classical” buffering phenomena (e.g. acid-base chemistry), the formalism also provides a simple means with which to quantitate and characterize phenomena that, at first glance, seem to be outside the buffering paradigm—at least when the term is used in its traditional sense with respect to the homeostasis of physiological parameters (see [2] for several in-depth examples of the formal and general approach applied to phenomena spanning diverse disciplines).

This article provides a series of mathematical tools designed to facilitate the application of Schmitt’s paradigm to both “classical” and “non-classical” buffering phenomena. By inputting the transfer function  $\tau(x)$  and the buffering function  $\beta(x)$ , the provided code:

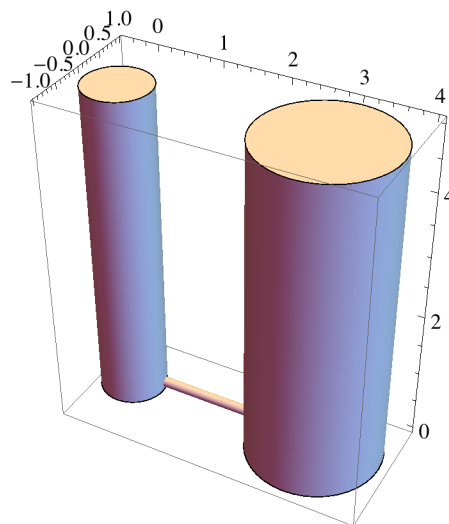
(1) defines the four buffering parameters  $t$ ,  $b$ ,  $T$ , and  $B$ ; (2) provides a series of graphical outputs describing the system; and (3) calculates the buffering angle  $\alpha$ , thereby classifying the system as being moderating/amplifying/inverting/non-inverting. Thus, the burden of computation associated with the approach is seamlessly transferred from the user to the *Mathematica* kernel.

Lastly, in order to illustrate the utility of the approach outside traditional disciplines, I use these same tools to calculate the magnitude of “genetic” buffering associated with an allele of the gene encoding the heat shock protein Hsp90. In this way, the capacity of an allele to buffer phenotypic variation is formally and quantitatively determined for the first time.

## ■ An Intuitive, Linear Example Based on the Partitioning of Fluids in Communicating Vessels

To illustrate Schmitt’s “formal and general approach,” consider a system comprised of a pair of communicating vessels (i.e. cylindrical flasks connected by a small tube). As described in [1], the total volume of liquid in the vessels,  $x$ , is the sum of the two partial volumes contained in each vessel. In the specific example described below, the rightmost of the two cylindrical vessels (the “buffering” vessel) has a radius exactly two times that of the first. Thus, any fluid added to the leftmost vessel (the “transfer” vessel) will partition in a ratio of 1:4 between the transfer and buffering vessels. The partial volume present in the transfer vessel is thus one-fifth the total volume of fluid added to the system. Likewise, the partial volume present in the buffering vessel is four-fifths the total volume of fluid added to the system.

```
Graphics3D[{
  Cylinder[{{0, 0, 0}, {0, 0, 5}}, 0.5],
  Cylinder[{{3, 0, 0}, {3, 0, 5}}, 1],
  Cylinder[{{0.1, 0.1, 0.1}, {3, 0.1, 0.1}}, 0.075]},
  Axes → True, ImageSize → {300, 300}]
```



## □ Formalizing the System

It is possible to formalize the system below by defining three functions: the transfer function  $\tau_1(x)$ , the buffering function  $\beta_1(x)$ , and the sigma function  $\sigma_1(x)$ .

The transfer function defines the amount of fluid present in the transfer vessel.

$$\tau_1[x_] := x / 5$$

Likewise, the buffering function defines the amount of fluid present in the buffering vessel.

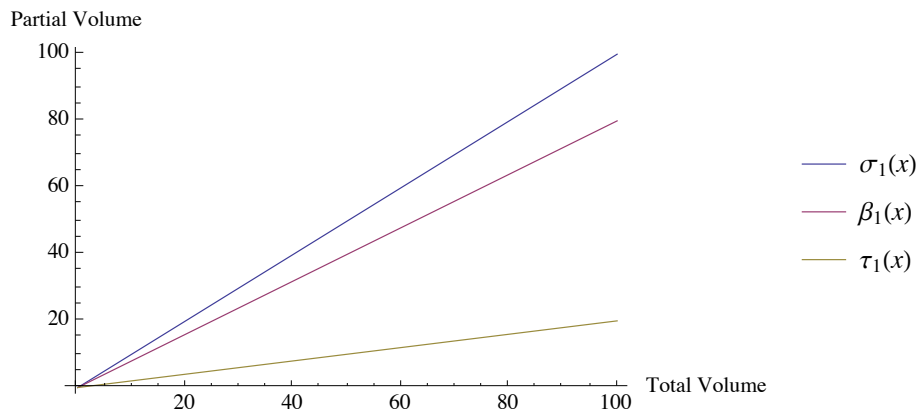
$$\beta_1[x_] := 4 x / 5$$

The sigma function defines the sum of the individual functions comprising the system.

$$\sigma_1[x_] := \tau_1[x] + \beta_1[x]$$

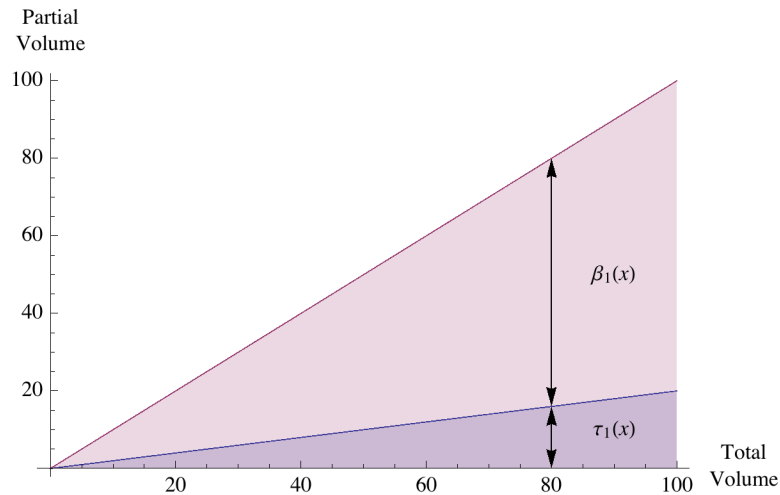
Once the functions are defined, it is possible to visualize the system by plotting the lines.

```
Plot[{ $\sigma_1[x]$ ,  $\beta_1[x]$ ,  $\tau_1[x]$ }, {x, 0, 100},  
  AxesLabel → {"Total Volume", "Partial Volume"},  
  PlotLegends → "Expressions"]
```



It is also possible to visualize the system (in a more intuitive manner) by creating an area plot in which the two partial volumes are indicated by the heights of the transfer and buffering areas at any given value of  $x$ .

```
Plot[{ $\tau_1[x]$ ,  $\sigma_1[x]$ }, {x, 0, 100},
  AxesLabel -> {"Total\nVolume", "Partial\nVolume"},
  Filling -> Bottom,
  Epilog -> {
    Text[TraditionalForm@HoldForm[ $\beta_1[x]$ ], {90, 50}],
    Text[TraditionalForm@HoldForm[ $\tau_1[x]$ ], {90, 10}],
    Arrowheads[{- .03, .03}],
    Arrow[{80, 0}, {80,  $\tau_1[80]$ }],
    Arrow[{80,  $\tau_1[80]$ }, {80,  $\sigma_1[80]$ }]
  }
]
```



### □ Defining the Buffering Parameters

The definition of the respective functions also provides a convenient means with which to quantitate buffering action. This is accomplished in any one of four ways, using the three available differentials. The buffering coefficient  $b_1$  measures the change in the buffering compartment relative to the total change.

$$b_1 = D[\beta_1[x]] / D[\sigma_1[x]]$$

$$\frac{4}{5}$$

The buffering ratio  $B_1$  measures the change in the buffering compartment relative to the change in the transfer compartment.

$$\mathbf{B}_1 = \mathbf{D}[\beta_1[\mathbf{x}]] / \mathbf{D}[\tau_1[\mathbf{x}]]$$

4

Conversely, the transfer coefficient  $t_1$  measures the change in the transfer compartment relative to the total change.

$$\mathbf{t}_1 = \mathbf{D}[\tau_1[\mathbf{x}]] / \mathbf{D}[\sigma_1[\mathbf{x}]]$$

$\frac{1}{5}$

The transfer ratio  $T_1$  measures the change in the transfer compartment relative to the buffering compartment.

$$\mathbf{T}_1 = \mathbf{D}[\tau_1[\mathbf{x}]] / \mathbf{D}[\beta_1[\mathbf{x}]]$$

$\frac{1}{4}$

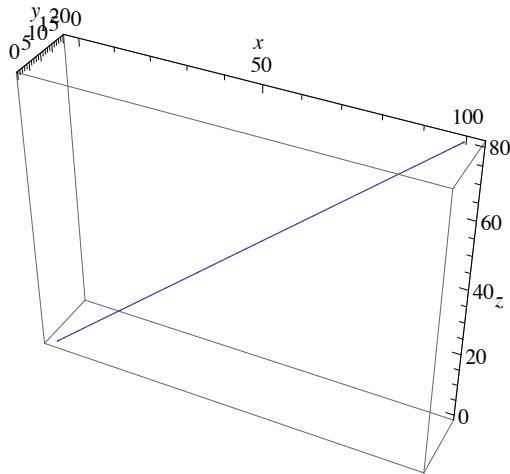
In this simple linear example—where the cross-sectional area of the buffering vessel is constant (i.e. does not vary with fluid level)—the dimensionless values of total volumes  $b$ ,  $B$ ,  $t$ , and  $T$  are invariant, equal to 0.8, 4, 0.20, and .025, respectively.

## □ Defining the Buffering Angle

While of great utility, the parameters  $B$  and  $T$  are undefined in the case of perfect buffering or perfect transfer, respectively. To address this issue, it is possible to define an alternative measure, the buffering angle  $\alpha$ , which is capable of representing any type of buffering without the introduction of discontinuities.

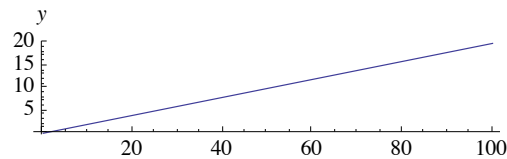
As described in [1], the buffering angle  $\alpha$  can be defined by first creating a three-dimensional space curve of the system.

```
ParametricPlot3D[{D[x], D[ $\tau_1$ [x]], D[ $\beta_1$ [x]]}, {x, 0, 100},  
AxesLabel -> {x, y, z}]
```



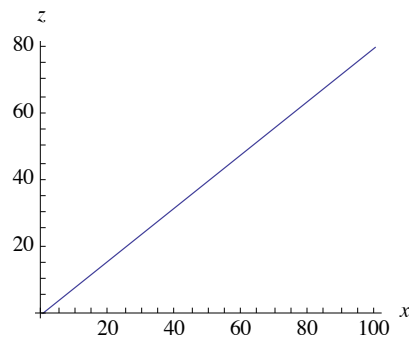
A projection of this curve onto the  $x$ - $y$  plane corresponds to the transfer function.

```
ParametricPlot[{D[x], D[ $\tau_1$ [x]]}, {x, 0, 100},  
AxesLabel -> {x, y}]
```



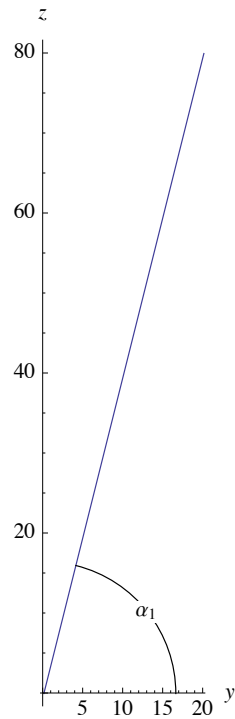
A projection onto the  $x$ - $z$  plane corresponds to the buffering function.

```
ParametricPlot[{D[x], D[ $\beta_1$ [x]]}, {x, 0, 100},  
AxesLabel -> {x, z}]
```



A projection onto the  $y$ - $z$  plane, however, generates a third curve that can be used to determine the proportions between the individual changes in the transfer and buffering functions. This is accomplished by simply measuring the angle  $\alpha_1$  between the  $y$  axis and the line formed by joining the origin to any given point on the curve.

```
Module[{x, a, r},
  x = 20;
  a = ArcTan[D[ $\tau_1$ ][x], D[ $\beta_1$ ][x]];
  r = Norm[{D[ $\tau_1$ ][x], D[ $\beta_1$ ][x]}];
  ParametricPlot[{D[ $\tau_1$ ][x], D[ $\beta_1$ ][x]}, {x, 0, 100},
    AxesLabel → {y, z}, ImageSize → 80 {1, 3},
    Epilog → {
      Circle[{0, 0}, r, {0, a}],
      Text[" $\alpha_1$ ", r {Cos[1/2 a], Sin[1/2 a]},
        Background → White]
    }
  ]
]
```



This calculates the buffering angle in degrees ([1], Supplement 5).

```

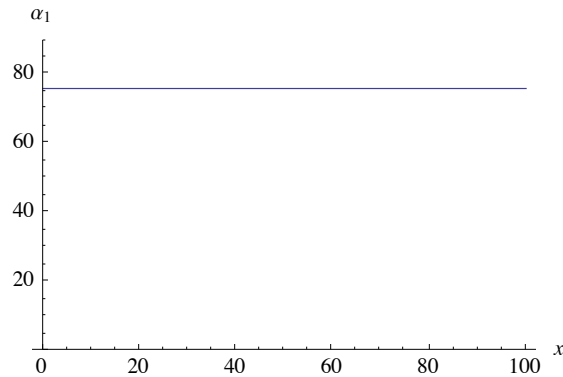
$$\alpha_1 = \text{If}[\mathbf{t}_1 \leq 1, 1, -1] \mathbf{N}\left[\text{ArcCos}\left[\left(\mathbf{t}_1 / \sqrt{\mathbf{t}_1^2 + \mathbf{b}_1^2}\right)\right]\right] / \text{Degree}$$

```

75.9638

Similar to the buffering parameters,  $\alpha_1$  is invariant in cases of linear buffering.

```
Plot [ $\alpha_1$ , {x, 0, 100}, AxesLabel → {x, " $\alpha_1$ "},  
PlotRange → {0, 90}]
```



It is evident from the above analysis that this methodology creates a system in which perfect buffering corresponds to an angle of  $90^\circ$  and perfect transfer to an angle of  $0^\circ$ . In fact, any type of buffering behavior (moderating/amplifying/inverting/non-inverting) can be represented continuously by a finite angle between  $-45^\circ$  and  $135^\circ$  ([1], Supplement 7).

Importantly, it is also possible—using the same logic in the reverse direction—to use this single value to unambiguously define all of the buffering parameters describing the system. This can be demonstrated by defining a variable `littleb1` (corresponding to the buffering coefficient  $b$ ) that is calculated using only the buffering angle.

```

$$\text{littleb1} = \frac{\text{Sin}[\alpha_1^\circ]}{\text{Cos}[\alpha_1^\circ] + \text{Sin}[\alpha_1^\circ]}$$

```

0.8

This checks that  $b$  and `littleb` are the same.

```
b1 == littleb1
```

True



Similarly, the values of the remaining three buffering parameters can be calculated using only the buffering angle and shown to be equal to the corresponding parameters calculated using the relevant differentials.

$$\mathbf{bigB1} = \mathbf{Sin}[\alpha_1^\circ] / \mathbf{Cos}[\alpha_1^\circ]$$

4.

$$\mathbf{B_1} == \mathbf{bigB1}$$

True

$$\mathbf{littlet1} = \mathbf{Cos}[\alpha_1^\circ] / (\mathbf{Cos}[\alpha_1^\circ] + \mathbf{Sin}[\alpha_1^\circ])$$

0.2

$$\mathbf{t_1} == \mathbf{littlet1}$$

True

$$\mathbf{bigT1} = \mathbf{Cos}[\alpha_1^\circ] / \mathbf{Sin}[\alpha_1^\circ]$$

0.25

$$\mathbf{T_1} == \mathbf{bigT1}$$

True

Thus, the buffering angle provides a simple, powerful, single-value measure that is capable of capturing and communicating the underlying nature of the buffering relationships comprising the system.

## ■ A More Complex, Nonlinear Example

While the simple linear scenario described above made it possible to lay the foundation for Schmitt's "formal and general approach," the computational effort associated with its application was minimal. Thus, the utility of applying these methods within *Mathematica* was not fully exploited. For this reason, I next consider a more complex, nonlinear example (i.e. where the cross-sectional area of the buffering vessel is not constant). This serves the purpose of highlighting the advantages of exploiting *Mathematica*'s computational engine.

### □ Defining a Nonlinear Buffering System

Consider a nonlinear system defined by the functions listed below.

$$\tau_2[x_] := 0.00395 x^{11.09746} \text{Exp}[-2.30471 x]$$

$$\sigma_2[x_] := x$$

$$\beta_2[x_] := \sigma_2[x] - \tau_2[x]$$

As previously described, the buffering parameters can be determined using the relevant differentials. However, in contrast to the previous example, the computational burden associated with their calculation is no longer trivial (at least in the absence of *Mathematica*).

$$b_2 = D[\beta_2[x]] / D[\sigma_2[x]]$$

$$\frac{x - 0.00395 e^{-2.30471 x} x^{11.0975}}{x}$$

$$B_2 = D[\beta_2[x]] / D[\tau_2[x]]$$

$$\frac{253.165 e^{2.30471 x} (x - 0.00395 e^{-2.30471 x} x^{11.0975})}{x^{11.0975}}$$

$$t_2 = D[\tau_2[x]] / D[\sigma_2[x]]$$

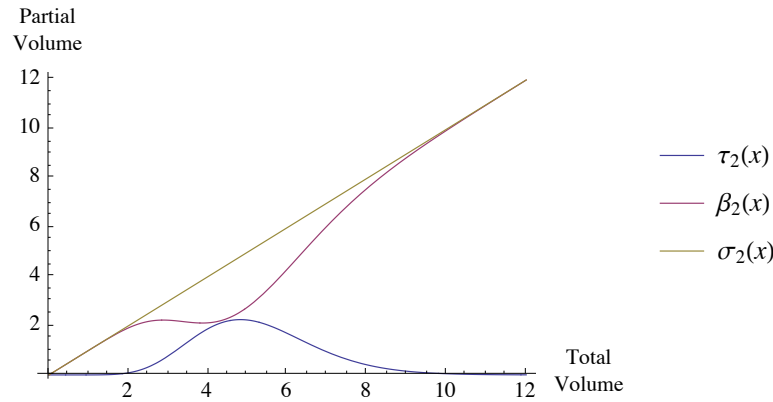
$$0.00395 e^{-2.30471 x} x^{10.0975}$$

$$T_2 = D[\tau_2[x]] / D[\beta_2[x]]$$

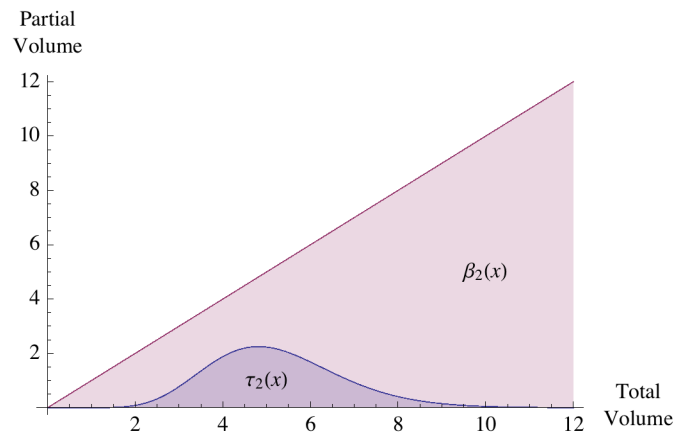
$$\frac{0.00395 e^{-2.30471 x} x^{11.0975}}{x - 0.00395 e^{-2.30471 x} x^{11.0975}}$$

Once the functions are defined and the system visualized through the creation of the corresponding line and area plots, one can clearly see that the magnitude of buffering action in this nonlinear scenario varies with fluid level (imagine a buffering vessel in the shape of an hourglass as opposed to a cylinder).

```
Plot[{ $\tau_2[x]$ ,  $\beta_2[x]$ ,  $\sigma_2[x]$ }, { $x$ , 0, 12},  
  AxesLabel → {"Total\nVolume", "Partial\nVolume"},  
  PlotLegends → "Expressions"]
```

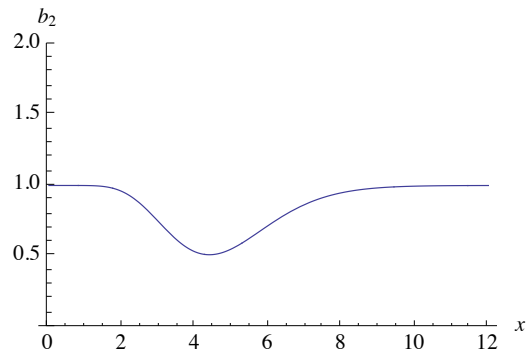


```
Plot[{ $\tau_2[x]$ ,  $\sigma_2[x]$ }, { $x$ , 0, 12}, Filling → Bottom,  
  AxesLabel → {"Total\nVolume", "Partial\nVolume"},  
  Epilog → {  
    Text[HoldForm[ $\beta_2[x]$ ], {10, 5}],  
    Text[HoldForm[ $\tau_2[x]$ ], {5, 1}]  
  }  
]
```

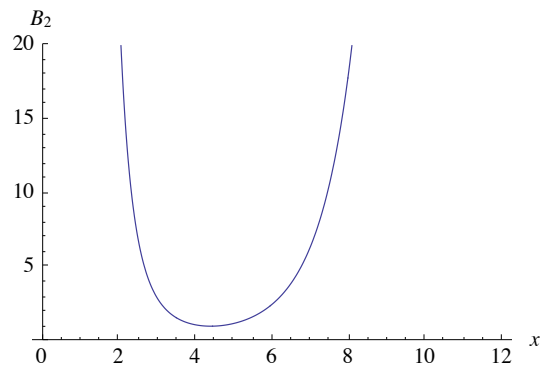


Furthermore, unlike the linear case, it is also clear that the buffering parameters vary as a function of  $x$ .

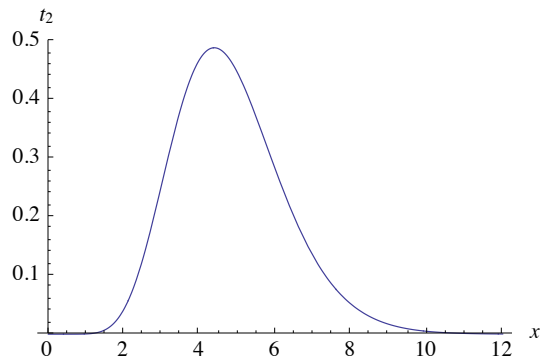
```
Plot[b2, {x, 0, 12}, PlotRange → {0, 2}, AxesLabel → {x, "b2"}]
```



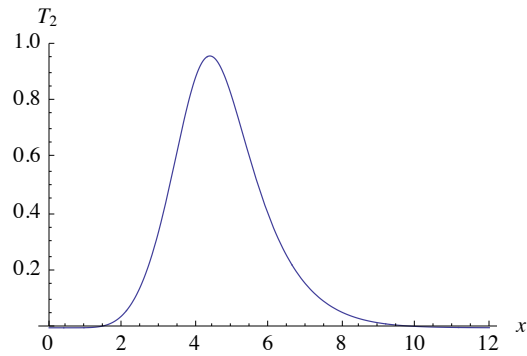
```
Plot[B2, {x, 0, 12}, PlotRange → {0, 20}, AxesLabel → {x, "B2"}]
```



```
Plot[t2, {x, 0, 12}, PlotRange → {0, 0.5},  
  AxesLabel → {x, "t2"}]
```



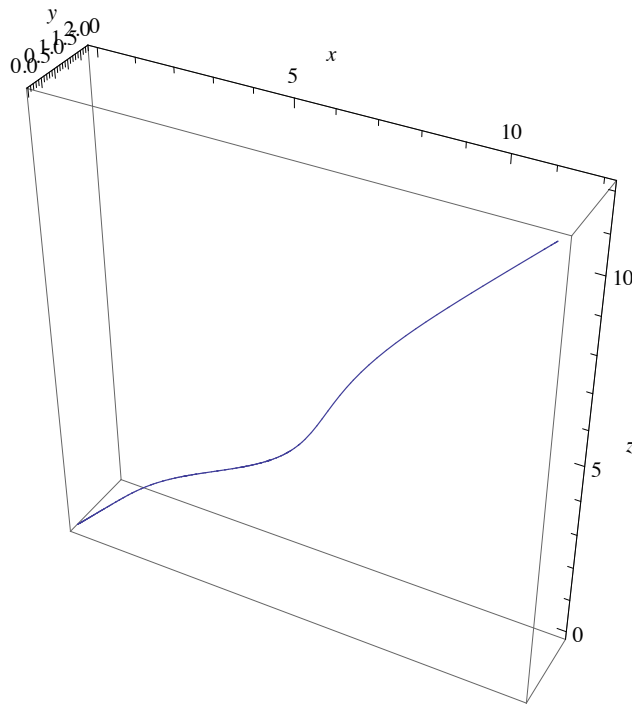
```
Plot[T2, {x, 0, 12}, PlotRange → {0, 1}, AxesLabel → {x, "T2"}]
```



### □ Defining the Buffering Angle of a Nonlinear System

As previously described, it is also possible to calculate the buffering angle  $\alpha_2$  by first creating a three-dimensional space curve of the system and then projecting this curve onto the  $y$ - $z$  plane.

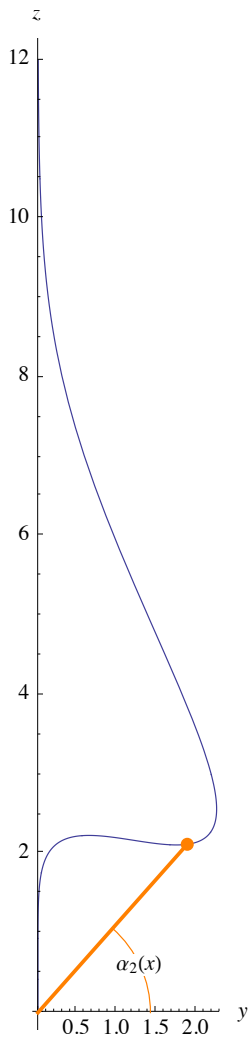
```
ParametricPlot3D[{D[x], D[ $\tau_2$ [x]], D[ $\beta_2$ [x]]}, {x, 0, 12},  
AxesLabel → {x, y, z}]
```



```

Module[{x, p, a, r},
  x = 4;
  p = {D[ $\tau_2$ [x]], D[ $\beta_2$ [x]]};
  a = ArcTan[D[ $\tau_2$ [x]], D[ $\beta_2$ [x]]];
  r = Norm[{D[ $\tau_2$ [x]], D[ $\beta_2$ [x]]}];
  ParametricPlot[{D[ $\tau_2$ [x]], D[ $\beta_2$ [x]]}, {x, 0, 12},
    AxesLabel → {y, z}, ImageSize → 100 {1.25, 5.5},
    Epilog → {
      {Orange, Circle[{0, 0}, .5 r, {0, a}]},
      Text[Row[{ $\alpha_2$ ", "(", Style["x", Italic], ")"}],
        .5 r {Cos[1 / 2 a], Sin[1 / 2 a]}, Background → White],
      Thickness[.02], Orange, Line[{0, 0}, p],
      PointSize[.07], Point[p]
    }
  ]
]

```



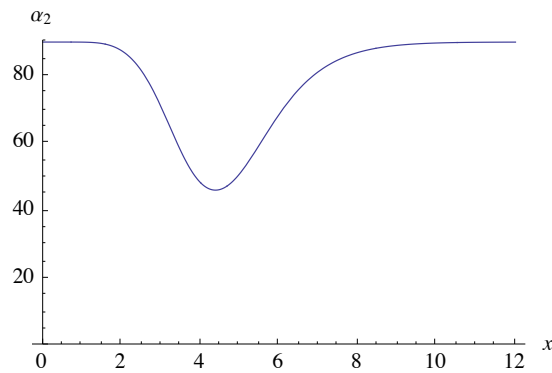
In this instance the buffering angle  $\alpha_2$  varies as a function of  $x$ , with near-perfect buffering being observed at both high and low fluid levels, and minimal buffering being observed when  $x$  attains a value of approximately 4.38.

$$\alpha_2 = \text{If}[\mathbf{t_2} \leq 1, 1, -1] \mathbf{N} \left[ \text{ArcCos} \left[ \left( \mathbf{t_2} / \sqrt{\mathbf{t_2}^2 + \mathbf{b_2}^2} \right) \right] \right] / \text{Degree}$$

$$\frac{1}{\circ} \text{ArcCos} \left[ \left( 0.00395 \times 2.71828^{-2.30471 \times x^{10.0975}} \right) / \left( \sqrt{0.0000156025 \times 2.71828^{-4.60942 \times x^{20.1949}} + \frac{\left( x - 0.00395 \times 2.71828^{-2.30471 \times x^{11.0975}} \right)^2}{x^2}} \right) \right] \right]$$

$$\text{If} \left[ 0.00395 e^{-2.30471 \times x^{10.0975}} \leq 1, 1, -1 \right]$$

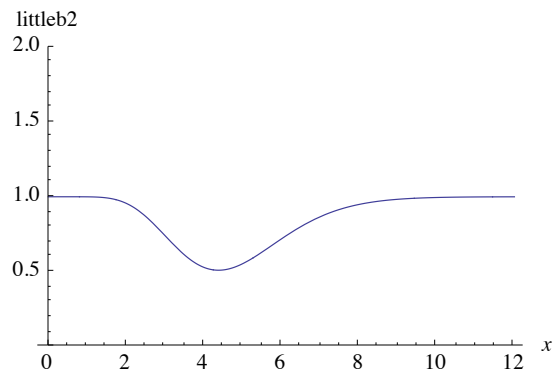
**Plot** $[\alpha_2, \{x, 0, 12\}, \text{AxesLabel} \rightarrow \{x, " \alpha_2 " \}, \text{PlotRange} \rightarrow \{0, 90\}]$



As described previously, the buffering parameters can be determined using the buffering angle and are equivalent to the values calculated using the originally defined functions (compare the plots below to the plots for  $b_1$ ,  $B_1$ ,  $t_1$ , and  $T_1$  above).

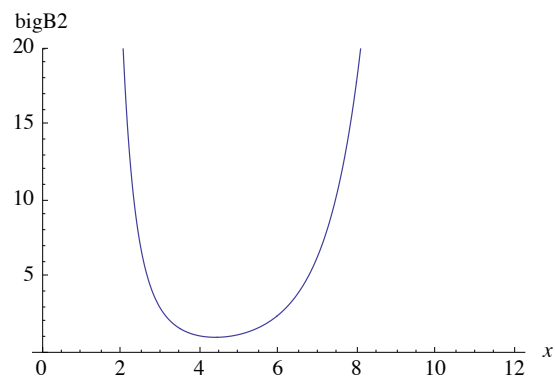
$$\text{littleb2} = \text{Sin}[\alpha_2^\circ] / (\text{Cos}[\alpha_2^\circ] + \text{Sin}[\alpha_2^\circ]);$$

```
Plot[littleb2, {x, 0, 12}, PlotRange -> {0, 2},
  AxesLabel -> {x, "littleb2"}]
```



```
bigB2 = (Sin[α2 °] / Cos[α2 °]);
```

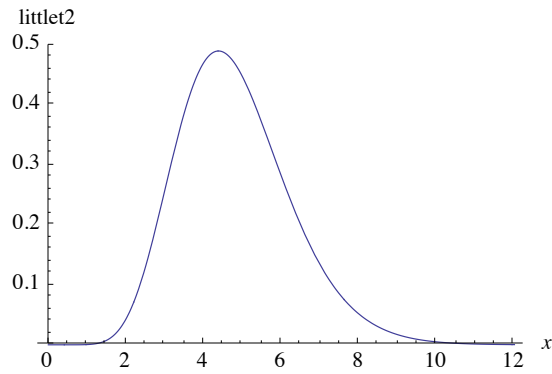
```
Plot[bigB2, {x, 0, 12}, PlotRange -> {0, 20},
  AxesLabel -> {x, "bigB2"}]
```



```
littlet2 = Cos[α2 °] / (Cos[α2 °] + Sin[α2 °]);
```

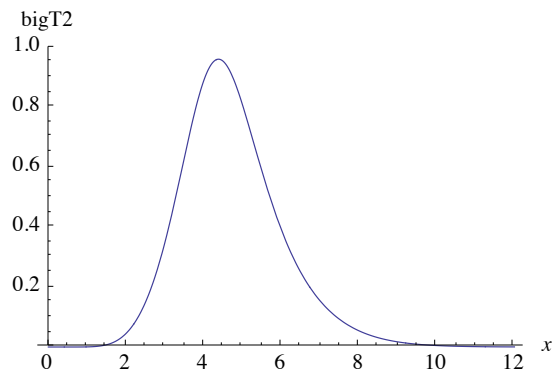


```
Plot[littlet2, {x, 0, 12}, PlotRange -> {0, 0.5},
  AxesLabel -> {x, "littlet2"}]
```



```
bigT2 = (Cos[α2 °] / Sin[α2 °]);
```

```
Plot[bigT2, {x, 0, 12}, PlotRange -> {0, 1},
  AxesLabel -> {x, "bigT2"}]
```



Thus, just as in the case of a linear system, plotting  $\alpha_2$  over the range of the disturbance (i.e. the addition of fluid) provides a simple, single-value measure with which to grasp the inherent buffering properties of the system under consideration.

## □ Differentials versus Derivatives

At this point it is important to note that, up to now, only the overall changes in the buffering and transfer compartments have been considered (by comparing differentials). However, in the case of nonlinear systems, it is also informative to consider the *rate* at which these values change relative to one another (by comparing derivatives). For example, in the same way that it would be important for an investor to be aware of the current price of gold, and how it has changed in previous years, months, or weeks, it would also be important for that same investor to be cognizant of the rate at which the price is changing at any given instant in time (i.e. increasing, decreasing, or stable). This is to say, the choice to sell at a given price at a given moment would be greatly influenced by knowing if the price was stable or on either an upward or a downward trend. In other words, it is not only of interest to know how a quantity *has changed*, but how that quantity *is changing*.

To determine the instantaneous rates of change in a buffered system, simply calculate and plot the first derivatives of the functions describing the system.

$$D[\beta_2'[x]]$$

$$1 - 0.043835 e^{-2.30471 x} x^{10.0975} + 0.0091036 e^{-2.30471 x} x^{11.0975}$$

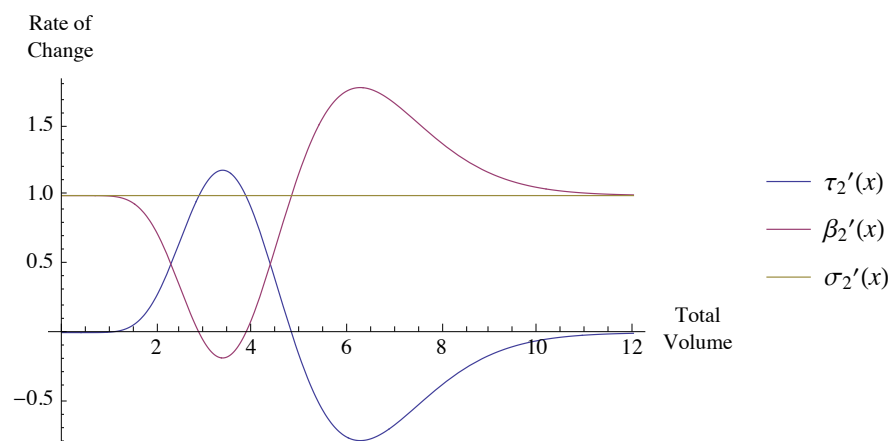
$$D[\tau_2'[x]]$$

$$0.043835 e^{-2.30471 x} x^{10.0975} - 0.0091036 e^{-2.30471 x} x^{11.0975}$$

$$D[\sigma_2'[x]]$$

$$1$$

```
Plot[{ $\tau_2'[x]$ ,  $\beta_2'[x]$ ,  $\sigma_2'[x]$ }, {x, 0, 12},
  AxesLabel -> {"Total\nVolume", "Rate of\nChange"},
  PlotLegends -> "Expressions"]
```



Once the functions are defined, the instantaneous buffering parameters (denoted by adding an “i” to the respective variable names) can also be calculated.

$$\mathbf{b2i} = \mathbf{D}[\beta_2'[\mathbf{x}]] / \mathbf{D}[\sigma_2'[\mathbf{x}]]$$

$$1 - 0.043835 e^{-2.30471 \times x^{10.0975}} + 0.0091036 e^{-2.30471 \times x^{11.0975}}$$

$$\mathbf{B2i} = \mathbf{D}[\beta_2'[\mathbf{x}]] / \mathbf{D}[\tau_2'[\mathbf{x}]]$$

$$\frac{(1 - 0.043835 e^{-2.30471 \times x^{10.0975}} + 0.0091036 e^{-2.30471 \times x^{11.0975}})}{(0.043835 e^{-2.30471 \times x^{10.0975}} - 0.0091036 e^{-2.30471 \times x^{11.0975}})}$$

$$\mathbf{t2i} = \mathbf{D}[\tau_2'[\mathbf{x}]] / \mathbf{D}[\sigma_2'[\mathbf{x}]]$$

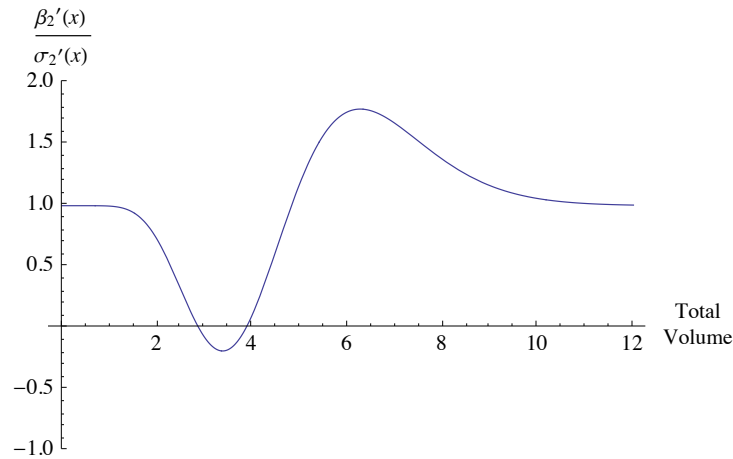
$$0.043835 e^{-2.30471 \times x^{10.0975}} - 0.0091036 e^{-2.30471 \times x^{11.0975}}$$

$$\mathbf{T2i} = \mathbf{D}[\tau_2'[\mathbf{x}]] / \mathbf{D}[\beta_2'[\mathbf{x}]]$$

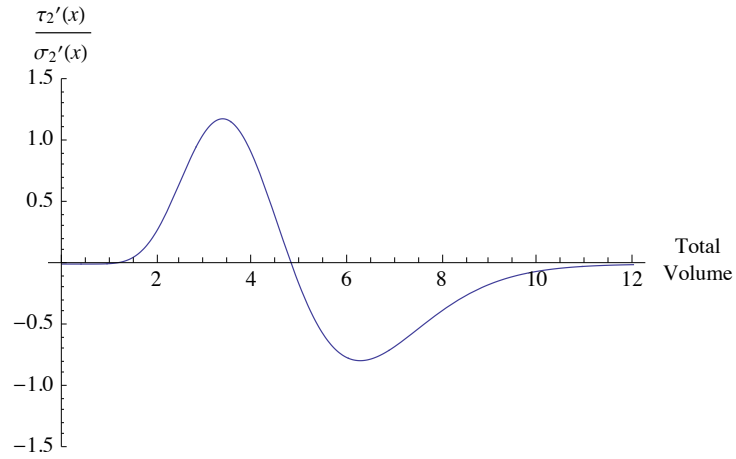
$$\frac{(0.043835 e^{-2.30471 \times x^{10.0975}} - 0.0091036 e^{-2.30471 \times x^{11.0975}})}{(1 - 0.043835 e^{-2.30471 \times x^{10.0975}} + 0.0091036 e^{-2.30471 \times x^{11.0975}})}$$

As is evident from viewing the respective plots of these functions, this analysis allows one to clearly identify the points of maximum and minimum buffering/transfer rate, as well as the points at which the buffering/transfer rates are zero.

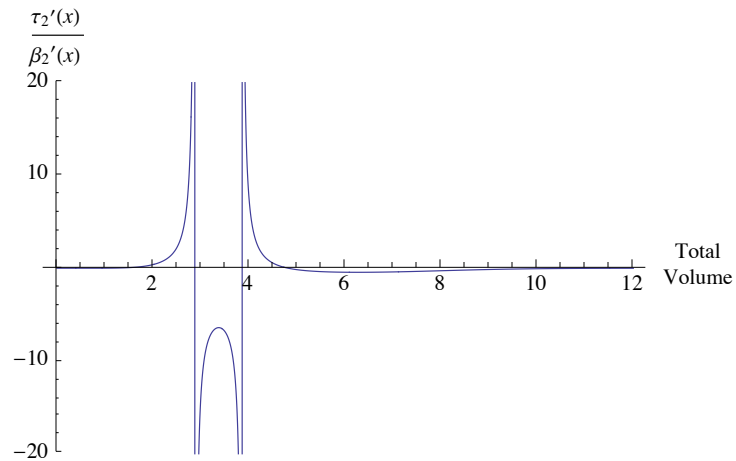
```
Plot[b2i, {x, 0, 12}, PlotRange -> {-1, 2},  
AxesLabel -> {"Total\nVolume", HoldForm[beta_2' [x] / sigma_2' [x]] }]
```



```
Plot[t2i, {x, 0, 12}, PlotRange → {-1.5, 1.5},
  AxesLabel → {"Total\nVolume", HoldForm[ $\tau_2'[x] / \sigma_2'[x]$ ]}]
```



```
Plot[T2i, {x, 0, 12}, PlotRange → {-20, 20},
  AxesLabel → {"Total\nVolume", HoldForm[ $\tau_2'[x] / \beta_2'[x]$ ]}]
```



In this way, the most critical junctures in the development of the system (over the range of the disturbance) can be quickly and easily identified.

## ■ From the Abstract to the Practical

The analysis presented in the previous sections concludes my efforts to establish the theoretical foundation of the formal and general approach, as well as its application within *Mathematica*. Next, in order to illustrate how these abstractions can be directly applied to the real world, I consider an example of non-classical buffering—namely, that of the “genetic buffering” of phenotypic variation. In this way, the ability to exactly and formally quantitate phenomena that at first glance might be thought of as being outside the buffering paradigm is practically demonstrated.

### □ Genetic Buffering

The axiomatic foundation and abstract nature of the “formal and general approach” allows it to be readily applied to any scenario in which an arbitrary quantity partitions itself between two compartments or states. While suitable for the analysis of “classical” buffering phenomena (the homeostasis of physiological parameters), the approach is also germane to a variety of “non-classical” buffering phenomena. One such phenomenon, which up to now has been considered in only a qualitative manner, is that of “genetic” buffering. While many instances of genetic buffering have been described and characterized, a well-known and classic example of the phenomenon—revolving around the gene encoding the heat shock protein Hsp90—will be used to illustrate the real-world application of Schmitt’s paradigm.

### ■ The Heat Shock Protein Hsp90

The Hsp90 protein is an ATP-dependent molecular “chaperone” that is extensively expressed in organisms ranging from bacteria to humans. It functions to promote the proper folding of a specific subset of molecular targets referred to as its “client” proteins. Interestingly, many of Hsp90’s client proteins are involved in the process of signal transduction and modulate developmental processes [3]. For example, in the fruit fly, *Drosophila melanogaster*, mutations in the gene encoding Hsp90 result in morphological abnormalities affecting the development of the eye, legs, wings, thorax, and bristles [4].

Interestingly, the proportion of Hsp90 mutant flies exhibiting eye defects (in a line prone to such abnormalities) increases dramatically as the temperature rises. In contrast, the proportion of control flies (expressing the normal or “wild-type” version of the Hsp90 protein) also increases in response to increased temperature, but not nearly to the same degree as in the mutant flies [4]. The gene encoding the wild-type version of the Hsp90 protein is thus said to “buffer” the appearance of the defective eye trait.

### ■ Formally Quantitating Hsp90-Mediated Buffering of Phenotypic Variation

It is possible to analyze these experimental results in more detail by importing the raw data (see Figure 8 in [4]) from the original publication in the form of two arrays. These two arrays, corresponding to the wild-type and mutant fly lines, relate temperature to the penetrance of the mutant phenotype (i.e. the percentage of flies in the line displaying the eye defect).

```
wildtypeHsp90 = {{18, 0}, {21, 0}, {25, 0}, {26, 1},
  {27, 3}, {30, 19}, {32, 16}}
```

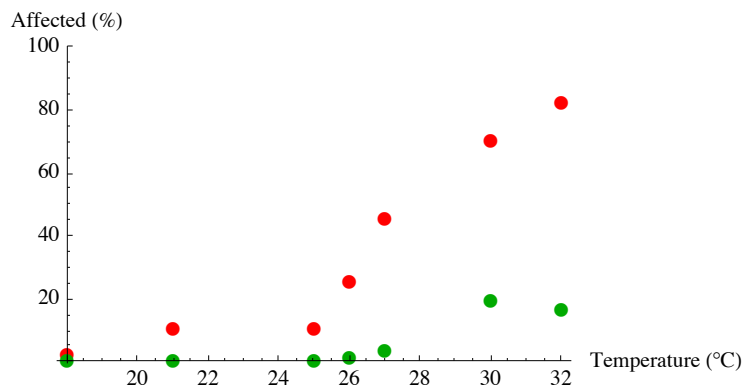
```
{{18, 0}, {21, 0}, {25, 0},
  {26, 1}, {27, 3}, {30, 19}, {32, 16}}
```

```
mutantHsp90 = {{18, 2}, {21, 10}, {25, 10}, {26, 25},
  {27, 45}, {30, 70}, {32, 82}}
```

```
{{18, 2}, {21, 10}, {25, 10},
  {26, 25}, {27, 45}, {30, 70}, {32, 82}}
```

Plotting the data illustrates that the mutant line is indeed more sensitive to increasing temperature than the wild-type line.

```
Show[ListPlot[mutantHsp90, PlotStyle → Red,
  PlotMarkers → Automatic,
  AxesLabel → {"Temperature (°C)", "Affected (%)"},
  PlotRange → {-1, 100}],
  ListPlot[wildtypeHsp90, PlotStyle → Darker@Green,
  PlotMarkers → Automatic]]
```



To proceed with the application of the formal and general approach, we now create two curves that fit the data using the `FindFit` function.

```
FindFit[wildtypeHsp90, j Exp[-k X], {j, k}, X,  
Method → NMinimize]
```

```
{j → 0.0031044, k → -0.272652}
```

```
FindFit[mutantHsp90, j Exp[-k X], {j, k}, X,  
Method → NMinimize]
```

```
{j → 0.193965, k → -0.191242}
```

With the above constants, the curves describing the behavior of the wild-type and mutant lines can be generated.

```
wildtypecurve[x_] :=  
j Exp[-k x] /. FindFit[wildtypeHsp90, j Exp[-k X],  
{j, k}, X, Method → NMinimize]
```

```
mutantcurve[x_] :=  
j Exp[-k x] /. FindFit[mutantHsp90, j Exp[-k X], {j, k},  
X, Method → NMinimize]
```

```
wildtypecurve[x]
```

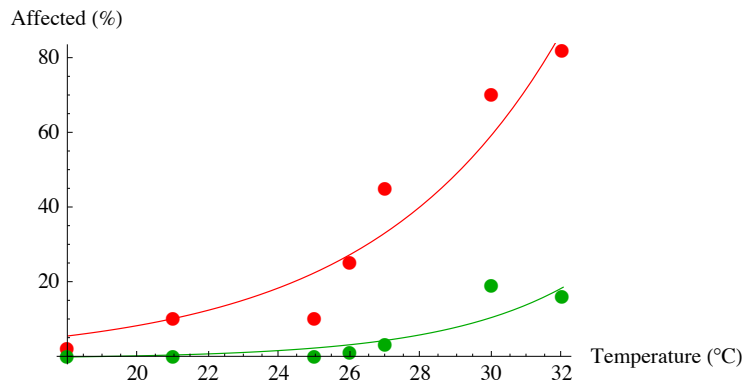
```
0.0031044 e0.272652 x
```

```
mutantcurve[x]
```

```
0.193965 e0.191242 x
```

The appropriateness of the fit can then be inspected by plotting the curves together with the raw data.

```
Show[
  ListPlot[mutantHsp90, PlotStyle -> Red,
    PlotMarkers -> Automatic,
    AxesLabel -> {"Temperature (°C)", "Affected (%)"}],
  ListPlot[wildtypeHsp90, PlotStyle -> Darker@Green,
    PlotMarkers -> Automatic],
  Plot[Evaluate[{mutantcurve[x], wildtypecurve[x]}],
    {x, 18, 32}, PlotStyle -> {Red, Darker@Green}]
]
```



With these results in hand, it is now possible to define both the buffering and transfer functions and then to visualize the system using an area plot.

```
 $\tau_3[x_] := \text{Evaluate}[\text{wildtypecurve}[x]]$ 
```

```
 $\beta_3[x_] := \text{Evaluate}[\text{mutantcurve}[x]]$ 
```

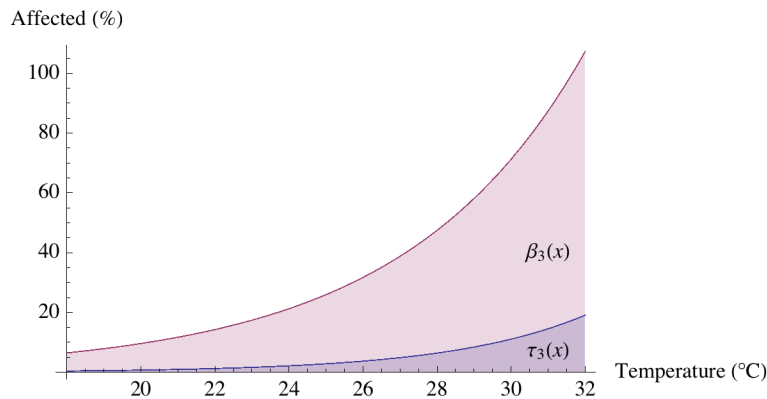
```
 $\sigma_3[x_] := \tau_3[x] + \beta_3[x]$ 
```



```

Plot[{ $\tau_3[x]$ ,  $\sigma_3[x]$ }, {x, 18, 32}, Filling -> Bottom,
  AxesLabel -> {"Temperature (°C)", "Affected (%)"},
  Epilog -> {
    Text[HoldForm[ $\beta_3[x]$ ], {31, 40}],
    Text[HoldForm[ $\tau_3[x]$ ], {31, 7}]
  }
]

```

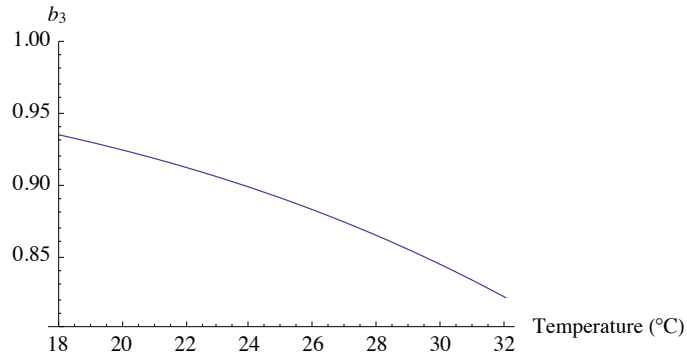


As is evident from inspecting this plot, an increase in temperature translates into only a small increase in the proportion of flies with eye defects in the wild-type line. However, in the mutant line, a much more pronounced increase is observed. We can thus infer that the expression of the wild-type Hsp90 protein buffers the effect of temperature on the penetrance of the eye defect. In other words, since the “transfer” of the eye defect phenotype is greater in the mutant line, the wild-type Hsp90 protein must possess the abstract capacity of diverting or “soaking up” this disturbance (into an equally abstract buffering compartment) so that the “transfer” of the eye defect is diminished. Importantly, using the formal and general approach, we can now provide an exact quantitative measure of the buffering capacity of the wild-type allele by calculating and plotting the four buffering parameters together with the buffering angle (below).

$$b_3 = D[\beta_3[x]] / D[\sigma_3[x]]$$

$$\frac{0.193965 e^{0.191242 x}}{0.193965 e^{0.191242 x} + 0.0031044 e^{0.272652 x}}$$

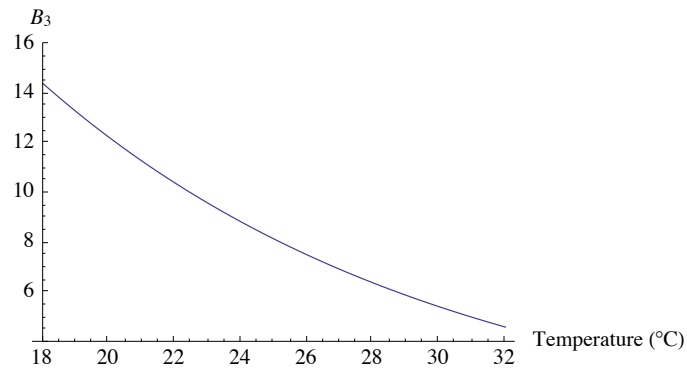
```
Plot[b3, {x, 18, 32}, PlotRange → {.8, 1},  
  AxesLabel → {"Temperature (°C)", "b3"}]
```



```
B3 = D[β3[x]] / D[τ3[x]]
```

```
62.4808 e-0.0814101 x
```

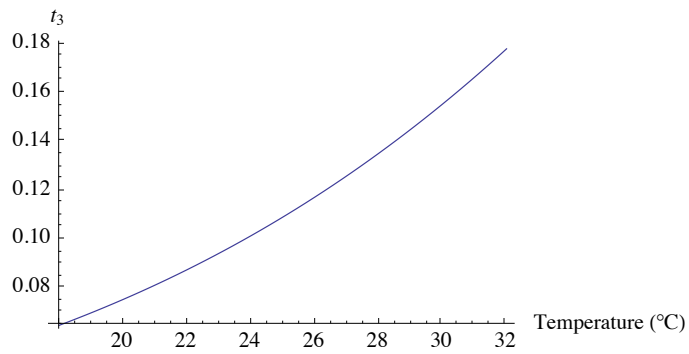
```
Plot[B3, {x, 18, 32}, PlotRange → {4, 16},  
  AxesLabel → {"Temperature (°C)", "B3"}]
```



```
t3 = D[τ3[x]] / D[σ3[x]]
```

```
0.0031044 e0.272652 x  
-----  
0.193965 e0.191242 x + 0.0031044 e0.272652 x
```

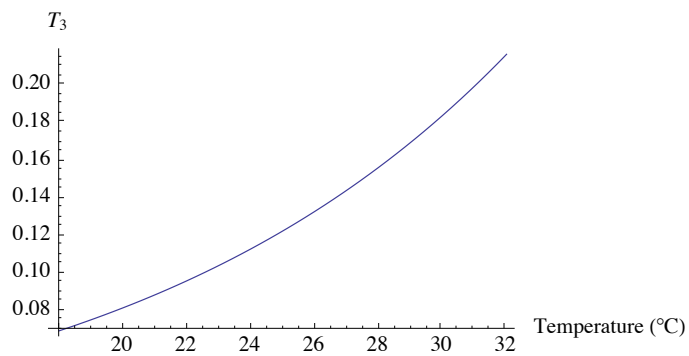
```
Plot[t3, {x, 18, 32}, AxesLabel → {"Temperature (°C)", "t3"}]
```



```
T3 = D[τ3[x]] / D[β3[x]]
```

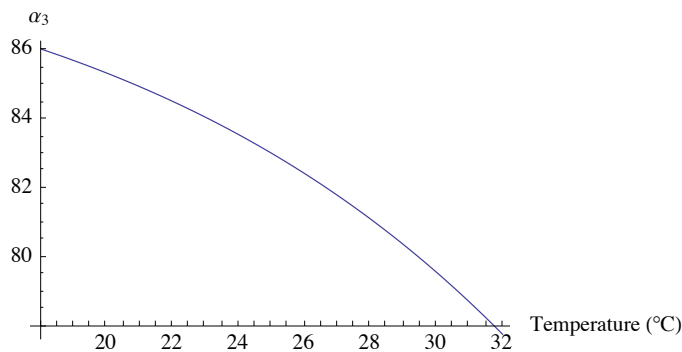
```
0.0160049 e0.0814101 x
```

```
Plot[T3, {x, 18, 32}, AxesLabel → {"Temperature (°C)", "T3"}]
```



```
α3 = If[t3 ≤ 1, 1, -1] N[ArcCos[(t3 / √(t32 + b32))] / Degree;
```

```
Plot[ $\alpha_3$ , {x, 18, 32}, AxesLabel → {"Temperature (°C)", " $\alpha_3$ "}]
```



As shown above, the buffering coefficient  $b_3$  can be calculated to range from  $\sim 0.935$  at 18 °C to  $\sim 0.822$  at 32 °C. While these calculations are now trivial, the critical biological question of how to interpret these calculations is more challenging. As discussed by Schmitt ([1], Supplements 6 and 7), one way to proceed is to consider the conceptual overlap between the formal and general approach and probability theory. In essence, the buffering coefficient  $b_3$  is analogous to a probability. In the same way that a probability measures the proportion of a part to a whole (i.e. the number of successful events to the total number of events), so does the buffering coefficient (by measuring the fractional change in the buffering compartment relative to the whole compartment). The critical realization is that both quantities are measured on a relative scale that is normalized to 1. Interpreting the graph in this way, one can say that the “probability” of buffering decreases from  $\sim 0.935$  to  $\sim 0.822$  from 18 °C to 32 °C.

Using similar logic, one can say that, in essence, the buffering ratio  $B_3$  is analogous to the “odds” of buffering. In the same way that odds measure the proportion between two parts of a whole (by comparing events to non-events), so does  $B_3$  (by comparing the fractional change in the buffering compartment to the fractional change in the transfer compartment). Again the critical realization is that both quantities are measured on a scale with equal intervals and an absolute zero. Interpreting the data in this way, one can say that the “odds” of buffering decrease from  $\sim 14.4$  to  $\sim 4.6$  from 18 °C to 32 °C. Similarly, one can consider  $t_3$  and  $T_3$  to be analogous to the “probability” of transfer and the “odds” of transfer, respectively.

Lastly, the buffering angle provides information regarding not only the magnitude of buffering but also the class of buffering behavior. As discussed by Schmitt ([1], Supplement 7), angles between 90° and 0° correspond to non-inverting moderation, with angles of 90° corresponding to perfect buffering and 0° to zero buffering. Thus, the simple inspection of the plot of  $\alpha_3$  reveals that the buffering capacity of wild-type Hsp90 is very strong (near perfect at  $\sim 86^\circ$ ) but decreases slightly with increasing temperature (to a value of  $\sim 78^\circ$ ). In any event—and however the data is interpreted—the analysis presented above clearly demonstrates the ease with which the buffering capacity of a genetic determinant can be formally and exactly quantified.

## ■ Conclusion

A formal conceptualization of genetic buffering was first put forward by famed geneticist and Nobel laureate Leland Hartwell [5, 6]. In these publications, Hartwell highlights the importance of buffering relationships in determining phenotype in outbred organisms. Subsequent research both in yeast and in the roundworm, *Caenorhabditis elegans*, has strongly supported his assumptions [7, 8]. Thus, in addition to increasing our basic understanding of genetic networks, these results also raised the theoretical question of whether or not it might be possible to provide an exact quantitative measure of a genetic determinant's buffering activity.

As described for the first time above, Schmitt's "formal and general approach" provides a powerful means with which to accomplish this goal. Through the calculation of the four buffering parameters  $b$ ,  $B$ ,  $t$ , and  $T$  and the buffering angle  $\alpha$ , it is indeed possible to provide an exact quantitative measure of the buffering activity or "power" of any genetic determinant in response to an environmental disturbance. Furthermore, the activity can be measured using either a relative scale normalized to one ( $b$  and  $t$ ) or an absolute scale with equal intervals ( $B$  and  $T$ ), or distilled to a single value measure  $\alpha$ , capable of describing the system in its entirety.

In conclusion, it must be noted that the application of the formal and general approach becomes practical only when used in conjunction with a sophisticated computational engine that is capable of readily analyzing the relevant data. As demonstrated in this article, *Mathematica* provides such an engine. The application of the formal and general approach using the Wolfram Language thus provides a powerful tool with which to quantitatively analyze both classical and non-classical buffering phenomena, irrespective of discipline.

## ■ References

- [1] B. M. Schmitt, "The Quantitation of Buffering Action I. A Formal & General Approach," *Theoretical Biology and Medical Modelling*, **2**(8), 2005. doi:10.1186/1742-4682-2-8.
- [2] B. M. Schmitt, "The Quantitation of Buffering Action II. Applications of the Formal & General Approach," *Theoretical Biology and Medical Modelling*, **2**(9), 2005. doi:10.1186/1742-4682-2-9.
- [3] M. Taipale, D. F. Jarosz, and S. Lindquist, "HSP90 at the Hub of Protein Homeostasis: Emerging Mechanistic Insights," *Nature Reviews Molecular Cell Biology*, **11**(7), 2010 pp. 515–528. doi:10.1038/nrm2918.
- [4] S. L. Rutherford and S. Lindquist, "Hsp90 as a Capacitor for Morphological Evolution," *Nature*, **396**, 1998 pp. 336–342. doi:10.1038/24550.
- [5] J. L. Hartman, B. Garvik, and L. Hartwell, "Principles for the Buffering of Genetic Variation," *Science*, **291**(5506), 2001 pp. 1001–1004. [www.sciencemag.org/content/291/5506/1001.abstract](http://www.sciencemag.org/content/291/5506/1001.abstract).
- [6] L. Hartwell, "Robust Interactions," *Science*, **303**(5659), 2004 pp. 774–775. doi:10.1126/science.1094731.

- [7] A. H. Tong, G. Lesage, G. D. Bader, H. Ding, H. Xu, X. Xin, J. Young, G. F. Berriz, R. L. Brost, M. Chang, Y. Chen, X. Cheng, G. Chua, H. Friesen, D. S. Goldberg, J. Haynes, C. Humphries, G. He, S. Hussein, L. Ke, N. Krogan, Z. Li, J. N. Levinson, H. Lu, P. Ménard, C. Munyana, A. B. Parsons, O. Ryan, R. Tonikian, T. Roberts, A. M. Sdicu, J. Shapiro, B. Sheikh, B. Suter, S. L. Wong, L. V. Zhang, H. Zhu, C. G. Burd, S. Munro, C. Sander, J. Rine, J. Greenblatt, M. Peter, A. Bretscher, G. Bell, F. P. Roth, G. W. Brown, B. Andrews, H. Bussey, and C. Boone, "Global Mapping of the Yeast Genetic Interaction Network," *Science*, **303**(5659), 2004 pp. 808–813. doi:10.1126/science.1091317.
- [8] B. Lehner, C. Crombie, J. Tischler, A. Fortunato, and A. G. Fraser, "Systematic Mapping of Genetic Interactions in *Caenorhabditis Elegans* Identifies Common Modifiers of Diverse Signaling Pathways," *Nature Genetics*, **38**(8), 2006 pp. 896–903. doi:10.1038/ng1844.

J. Karagiannis, "The Quantitation of Non-classical Buffering," *The Mathematica Journal*, 2015. dx.doi.org/doi:10.3888/tmj.17-3.

## About the Author

Dr. Jim Karagiannis is an associate professor at the University of Western Ontario in London, Ontario, Canada. His research makes use of the model eukaryote *Schizosaccharomyces pombe* and explores the complex post-translational modifications that take place on the carboxy-terminal domain (CTD) of the largest subunit of RNA polymerase II. Through an empirical examination of the informational properties and regulatory potential of the CTD, Dr. Karagiannis hopes to decipher the "programming language" used by eukaryotes to control aspects of gene expression.

### Jim Karagiannis

*Department of Biology  
The University of Western Ontario  
London, ON  
Canada, N6A-5B7  
jkaragia@uwo.ca*