What's in a Name? The Matrix as an Introduction to Mathematics

Kris H. Green

St. John Fisher College, kgreen@sjfc.edu

Follow this and additional works at: https://fisherpub.sjfc.edu/math_facpub

Part of the Mathematics Commons

How has open access to Fisher Digital Publications benefited you?

Publication Information


Please note that the Publication Information provides general citation information and may not be appropriate for your discipline. To receive help in creating a citation based on your discipline, please visit http://libguides.sjfc.edu/citations.

This document is posted at https://fisherpub.sjfc.edu/math_facpub/18 and is brought to you for free and open access by Fisher Digital Publications at St. John Fisher College. For more information, please contact fisherpub@sjfc.edu.
What's in a Name? The Matrix as an Introduction to Mathematics

Abstract

In my classes on the nature of scientific thought, I have often used the movie *The Matrix* (1999) to illustrate how evidence shapes the reality we perceive (or think we perceive). As a mathematician and self-confessed science fiction fan, I usually field questions related to the movie whenever the subject of linear algebra arises, since this field is the study of matrices and their properties. So it is natural to ask, why does the movie title reference a mathematical object?

Of course, there are many possible explanations for this, each of which probably contributed a little to the naming decision. First off, it sounds cool and mysterious. That much is clear, and it may be that this reason is the most heavily weighted of them all. However, a quick look at the definitions of the word reveals deeper possibilities for the meaning of the movie's title. Consider the following definitions related to different fields of study taken from Wikipedia on January 4, 2010:

- **Matrix (mathematics)**, a mathematical object generally represented as an array of numbers.

- **Matrix (biology)**, with numerous meanings, often referring to a biological material where specialized structures are formed or embedded.

- **Matrix (archeology)**, the soil or sediment surrounding a dig site.

- **Matrix (geology)**, the fine grains between larger grains in igneous or sedimentary rocks.

- **Matrix (chemistry)**, a continuous solid phase in which particles (atoms, molecules, ions, etc.) are embedded.

All of these point to an essential commonality: a matrix is an underlying structure in which other objects are embedded. This is to be expected, I suppose, given that the word is derived from the Latin word referring to the womb — something in which all of us are embedded at the beginning of our existence. And so mathematicians, being the Latin scholars we are, have adapted the term: a mathematical matrix has quantities (usually numbers, but they could be almost anything) embedded in it. A biological matrix has cell components embedded in it. A geological matrix has grains of rock embedded in it. And so on. So a second reason for the cool name is that we are talking, in the movie, about a computer system generating a virtual reality in which human beings are embedded (literally, since they are lying down in pods). Thus, the computer program forms a literal matrix, one that bears an intentional likeness to a womb.

However, there are other ways to connect the idea of a matrix to the film's premise. These explanations operate on a higher level and are explicitly relevant to the mathematical definition of a matrix as well as to the events in the trilogy of *Matrix* movies. They are related to computer graphics, Markov chains, and network theory. This essay will explore each of these in turn, and discuss their application to either the events in the film's story-line or to the making of the movie itself.

Disciplines

Mathematics

Comments


This book chapter is available at Fisher Digital Publications: [https://fisherpub.sjfc.edu/math_facpub/18](https://fisherpub.sjfc.edu/math_facpub/18)
What's in a Name?
The Matrix as an Introduction to Mathematics

Kris Green

In my classes on the nature of scientific thought, I have often used the movie The Matrix (1999) to illustrate how evidence shapes the reality we perceive (or think we perceive). As a mathematician and self-confessed science fiction fan, I usually field questions related to the movie whenever the subject of linear algebra arises, since this field is the study of matrices and their properties. So it is natural to ask, why does the movie title reference a mathematical object?

Of course, there are many possible explanations for this, each of which probably contributed a little to the naming decision. First off, it sounds cool and mysterious. That much is clear, and it may be that this reason is the most heavily weighted of them all. However, a quick look at the definitions of the word reveals deeper possibilities for the meaning of the movie's title. Consider the following definitions related to different fields of study taken from Wikipedia on January 4, 2010:

- Matrix (mathematics), a mathematical object generally represented as an array of numbers.
- Matrix (biology), with numerous meanings, often referring to a biological material where specialized structures are formed or embedded.
- Matrix (archaeology), the soil or sediment surrounding a dig site.
- Matrix (geology), the fine grains between larger grains in igneous or sedimentary rocks.
- Matrix (chemistry), a continuous solid phase in which particles (atoms, molecules, ions, etc.) are embedded.

All of these point to an essential commonality: a matrix is an underlying structure in which other objects are embedded. This is to be expected, I suppose, given that the word is derived from the Latin word referring to the womb — something in which all of us are embedded at the beginning of our existence. And so mathematicians, being the Latin scholars we are, have adapted the term: a mathematical matrix has quantities (usually numbers, but they could be almost anything) embedded in it. A biological matrix has cell components embed-
ded in it. A geological matrix has grains of rock embedded in it. And so on. So a second reason for the cool name is that we are talking, in the movie, about a computer system generating a virtual reality in which human beings are embedded (literally, since they are lying down in pods). Thus, the computer program forms a literal matrix, one that bears an intentional likeness to a womb.

However, there are other ways to connect the idea of a matrix to the film's premise. These explanations operate on a higher level and are explicitly relevant to the mathematical definition of a matrix as well as to the events in the trilogy of Matrix movies. They are related to computer graphics, Markov chains, and network theory. This essay will explore each of these in turn, and discuss their application to either the events in the film's storyline or to the making of the movie itself.

Computer Graphics and The Matrix

Formally in mathematics, a matrix is an arrangement of information, usually numerical, into rows and columns. The objects within the matrix are called its entries. For instance, the following is an example of matrix with numerical entries:

$\begin{pmatrix}
2 & -1 & 3 & 4 \\
5 & 2 & 0 & -9
\end{pmatrix}$

Matrices can sometimes be added to each other and sometimes multiplied together. They have many special properties that help characterize the information they contain, and they relate to many different physical and social phenomena. One of their most common current applications is in computer graphics, and the reasons for this can be easily understood.

First of all, a computer screen is itself a matrix. It is, quite literally, a collection of rows and columns of pixels, each of which can be thought of as a single "dot" of color. In many Windows-based computers, the screen can be set to a variety of settings for both the resolution (the number of pixels in each direction) and the number of colors or states for each pixel. Consider a fairly typical arrangement of pixels into 800 rows and 1,280 columns, with 32-bit color. In this case, at any given time, each of the $800 \times 1,280 = 1,024,000$ pixels can be in any of $2^{32}$ states. To put these numbers in perspective, imagine that every particle of matter in the universe is capable of displaying just one of these pixel arrangements per second. Then, since the moment of the Big Bang, the entire universe would have displayed far less than 1 percent of the possible computer screen configurations. (Perhaps this is why we should expect more from Hollywood than remaking old movies!) Of course, the preceding counting argument ignores some important facts: many of these configurations will be related. Some will be rotations or reflections of others, and some will be the same configuration, only color-shifted. Accounting for these symmetries vastly reduces the number of distinct screens that a modern computer can display, but it's still far more than could be explored in the lifetime of the universe.

However, this approach to representing images (called a bitmap) is not very efficient. For starters, it requires a tremendous amount of memory to store many different bitmaps, such as would be needed for a movie. In addition, it treats each pixel as separate and provides no information on how pixels are grouped into objects or how those objects are organized.
spatially in relation to one another. Thus, if an object in a bitmap moves in front of another object, there is no easy way to determine which parts of each object are obscured relative to a particular perspective or to determine how the lighting should shade the object as it moves. Instead, vector-based graphics are typically used. In this method, representing an object on a computer screen (or a movie screen, or on the mind of an enslaved human embodied in the Matrix) involves defining the coordinates of each point on the object. However, in the case of simulating a three-dimensional object, different viewpoints will result in the object taking on different appearances in terms of shape, lighting, texture, and so forth. By representing these points in a matrix, we have the necessary tool for determining how the shape of an object varies with different viewpoints.

The essential idea can be understood from a two-dimensional problem. Suppose we have a triangle with its vertices (labeled A, B, and C) at the coordinates (-1,-2), (0,6), and (3,4), respectively, and we wish to know the coordinates of the shape after it is rotated counterclockwise by 30 degrees around the origin, O = (0,0). If we write each point on the triangle as a vertical column of numbers — a matrix with two rows and one column — we can perform a matrix calculation that outputs the coordinates of the point after the rotation. For example, take point C. As a matrix, point C looks like

\[
C = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.
\]

To compute the coordinates of the image, \( C' \), of point C under a counterclockwise rotation by \( \theta \) degrees, we multiply \( C \) by the following rotation matrix, \( R_\theta \):

\[
R_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.
\]

For example, if we rotate the triangle 30 degrees counterclockwise, we have

\[
R_\theta = \begin{pmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{pmatrix} = \begin{pmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{pmatrix};
\]

multiplying this matrix by the coordinates of point C gives the coordinates of \( C' \).

\[
C' = R_\theta C = \begin{pmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \end{pmatrix} \approx \begin{pmatrix} 0.598 \\ 4.964 \end{pmatrix}.
\]

If we repeat this process for each point of the object, its new coordinates can be determined and the object redrawn to appear as it would from another perspective; see Figure 1. This is surely an important function of the computer program in the movie, since it must continually construct a shared reality among many different viewpoints and allow each person to perceive the virtual world correctly as he or she interacts with it.

**Markov Chains and The Matrix**

But it is the use of matrices to represent an object called a Markov chain that allows us to see how the *Matrix* trilogy uses the idea of a mathematical matrix in order to deal
with the ages-old debate of free will versus predestination. In other words, the mathematical concept of a matrix can help us understand how, in a computer-generated world such as that depicted in the Matrix trilogy, a human mind—or any “mind”—can possibly know the future the way Neo (Keanu Reeves) appears to in the second movie of the trilogy.

The key lies in thinking about the future as a series of events, each with its own probability of occurring, and each of which is selected from a large but finite number of choices.
But it goes even further. If you think about an event that occurs as defining the current "state" of the future, the chance of transitioning into a different state may be different depending on your starting state. For example, if you go to use a photocopier at the office, it will be in one of two states, working (W) or broken (B). Now, past experience might tell you that when you need to copy something and have no prior information, you expect the copier to be in state W about 80 percent of the time and state B 20 percent of the time. But, once it is in state B, the probability of the copier working in the near future changes. The chance of a broken copier remaining broken the next day might be 55 percent, while the chance that it is really fixed (and thus back to state W) is 45 percent. Thus, we have probabilities, called transition probabilities, for how the future states of the machine will unfold, given its current and previous states. If we have a sequence of states leading up to the present, we can use these probabilities to predict the future states of the machine.

This chain of events determined by transition probabilities is called a Markov chain, and the transition probabilities between states can be written as the elements of a transition matrix, T. By raising T to the nth power we can study the likelihood of the system (in this case, the copier) being in a particular state after n time steps.

Thus, the use of a matrix in a Markov chain process can help us determine the probabilities for future events, but it cannot tell us for certain (a) how we get there (the chain of states we pass through) or (b) in which state we will definitely be at a given future time. This is physically illustrated in the final scenes of the second matrix movie (The Matrix Reloaded) as Neo sits in front of the Architect (Helmut Bakaitis) discussing the reality of the matrix. He is surrounded by hundreds of television screens, each of which depicts different possible reactions Neo might have to the information he is receiving (see Figure 2).

As Neo starts to make a decision and act, more and more of the screens depict the same resulting action, neatly illustrating the increased probability of transitioning from his current state to a particular future action.

This idea of a Markov chain also explains the insistence of the Oracle (Gloria Foster) that Neo already knows the future and has made his choice but does not understand it.

Figure 2. Neo's possible reactions. (Film still from The Matrix Reloaded, courtesy Warner Bros. Entertainment, Inc.)
Essentially, this means that the set of states and the transition probabilities Neo has been using to predict the future is no longer sufficient, since some of the realizable states of the system are not included in his current mental collection. Until he understands the radical events surrounding him and constructs a new mental picture — taking into account recent events and new states — of the possible future states, he cannot predict events past this discrepancy between what he can predict and what is actually possible: his list of current states and transitions is not up to the task. The copier example can be used to illustrate this idea. Suppose that after a week we have gone through a succession of states like $W-W-B-W-W-B-X$ and arrived at a new state $X$ that means the copier is in a permanently broken state. This new state was not part of our original list of states. It is not accounted for in our transition matrix, thus we could never have predicted this outcome. Until we revise our set of states from $\{W, B\}$ to $\{W, B, X\}$, we can never represent the chain of possible futures. Philosophically, this means that our assumptions about the copier were wrong; we assumed that it could always be fixed. Neo’s assumptions about the future fall apart after Trinity (Carrie-Anne Moss) nearly dies as a result of the new information he has received. Until he processes this and understands it, his mind cannot make further predictions about the future.

A further application of Markov chains connects the movie trilogy’s exciting fight scenes with mathematics. When one is involved in a physical fight, predicting the actions of the opponent will allow you to react more quickly and to respond more effectively. This is why, in the opening moments of many fights — boxing, martial arts, wrestling — there is a period of “feeling each other out” while the opponents get a sense of how each other moves and what signs will give away his or her actions. But one can go much deeper. Each style of fighting has distinct characteristics. A taekwondo stylist tends to use a lot of fast, far-reaching kicks. Kung fu stylists use circular motions. Karateka tend to rely on more linear movements with powerful punches. And it goes even deeper than this. Regardless of training, everyone has a unique physique and a unique training history. This contributes to some movements and some combinations of movements feeling more natural. And the more natural a sequence of moves, the faster and more accurately one can reproduce them.

This means that each fighter will tend to have “favorite combinations” of techniques that he or she uses more often than others. And if you can identify those combinations, you are more likely to be able to accurately predict your opponent’s actions. One way to do this — a way that lends itself nicely to creating computerized opponents — is to use transition matrices. If each of the states is a fighting technique (uppercut, reverse punch, back-leg roundhouse kick, etc.) then one can analyze a fighter by counting the number of times each technique is followed by each other technique. After a little data collection and analysis, one will have a pretty good idea of what moves are more likely to follow each of your opponent’s moves.

Clearly, this would require collecting a large amount of data. And clearly during a fight one is not going to be computing transition probabilities. But over the course of many fights, one will naturally adjust and “get comfortable” with an opponent. In the Matrix movies, however, rather than generating a matrix from an opponent’s moves, the fictional computer system controlling the Agents and the environment needs to create multiple and varied opponents to deal with the rebels. By creating slightly different transition matrices, it can develop different fighting styles to challenge Neo and his cohorts. And since we are talking about a massively parallel computer system, the Matrix itself can analyze Neo,
Trinity, and Morpheus (Laurence Fishburne) to determine their likely strategies and develop optimal opponents to confront them. This gives the computer a powerful tool to deal with human intuition.

This type of analysis is used frequently in computer games to generate opponents with different styles. For example, a chess game could create different opponents by having a different likelihood of using a particular opening, then having slightly different likelihoods for responding to each type of move. Of course, in doing this, one must carefully determine the allowable states in the system. Simply using "punch," "kick," and "move" as options in a fight gives one very little predictive ability since there are many different punches, kicks and movements possible. But it allows one to build a rough idea of style quickly, because the data is coarser grained. On the other hand, defining states by every possible stance, body part, and technique combination (like "left-hand reverse uppercut from a right-foot-forward Seisan stance") results in too few opportunities to observe and collect data, but allows, after sufficient data is available, very precise predictions. Thus, one might choose to adopt a mixture of analysis tools beginning with a coarse-grained approach and refining each category as further information becomes available.

Network Theory and The Matrix

A final connection between the movies and linear algebra comes from the emerging science of network theory. This is the study of how interconnected agents (in this case, we are referring to the people trapped in the Matrix, not the Agents in the movies) share and pass information along through a network. One can represent the interconnections with a matrix and by exploring the properties of the matrix come to understand how various phenomena (usually called "epiphenomena" or "emergent phenomena") come about purely as a result of these interactions. A typical example is the way crickets in a field will initially chirp independently, but as a result of the feedback from hearing each other, will quickly synchronize their chirps. The emergent phenomena — synchronized chirping — is not the result of a central dictatorial cricket demanding harmony, but rather an emergent property of the way each individual cricket responds to its surroundings. For more details on this field of study — albeit without the technical features — one should consult Duncan Watts's volume *Six Degrees* or *Nexus* by Mark Buchanan.

Basically, in a network model, one can represent the connections as a graph or in the form of one of several different matrices, one of which is called the adjacency matrix. Visualizing a matrix as a two-dimensional array of numbers, the adjacency matrix simply lists all the individual components next to the left side of the matrix and above the top row of the matrix. If two components of the network are connected, then there is a 1 in the corresponding entry of the matrix. If they are not connected, there is a 0 in that entry. Using this approach, one-way connections can also be represented. For example, if our network of interest is a food web, the individual components (agents, nodes, or whatever term is your favorite) are the types of organisms in the food web. The connections might represent who eats whom. Thus, in the matrix entry representing the interaction between the row containing a shark and the column containing a small fish, there would be a 1. However, since the fish does not prey upon the shark, the entry in the row containing the fish and
the column containing the shark would be a 0. Such matrices lack the property of symmetry: that is, we may have $T_{ij} \neq T_{ji}$, where $T_{ij}$ indicates the element in row $i$ and column $j$ of the matrix $T$. In this way, one can study various properties of the food web by computing quantities related to the adjacency matrix representing this web.

A critical component of making the network aspect function, though, is that each agent in the system (e.g., the people in the pods) can influence each other and cause real changes in each other’s behavior. This is illustrated at several points in the trilogy. One of the most graphic illustrations of this is the use of food. The Merovingian (Lambert Wilson) claims to have written a program —in the form of a piece of cake—which influences a young woman’s behavior. The Oracle gives Neo a cookie she baked herself, claiming that he will feel “right as rain” when he finishes it. And at their second meeting, Neo comes away full of candy. After each of these food-related encounters, Neo undergoes a significant change. Without this ability to explicitly affect each other, information could not pass through the connected network of individuals, as it would exist only within the computer-generated construct and not within the individual minds connected to the system.

The third movie of the trilogy provides a superb example of network theory and emergent phenomena in action. And in this example, it applies to both the fictional action on screen and the methods for creating the imagery of the movie. In the climactic battle at the docks, millions of sentinel-machines emerge from the tunnels to swarm the defenses of Zion. These objects are all independent “entities” but seem to cohere into huge “meta-creatures” that attack the dock. This behavior is seen in swarms of bees, colonies of ants, flocks of birds, and schools of fish: some sort of local behavioral rules lead to an emergent behavior of the collective. Although the machines seem capable of tremendous computational complexity, trying to create a central unit to coordinate the movements of millions of sentinels

Image redacted at the request of the publisher.
would be nearly impossible, given the number of variables involved. For example, if one sentinel is destroyed, which one takes its place? What we have learned recently, through simulations of bird flocking behavior (see Craig Reynolds’s *Boids*), is that no central coordinator is needed. By simply defining the relationships — usually by nearest neighbor, which frequently changes — and local rules for acting, one can generate large-scale behavior. The sentinels utilize this very effectively. Notice how easily the defenders hold the dock before the collective coheres. And then take note of how quickly the defenders lose control after the collective coheres.

In the same way, movie special effects houses would be overwhelmed trying to produce such epic battles if they had to rely on programming and moving each object by itself. Instead, they create agents and rules of behavior and simulate the movements of the objects. This allows efficient creation of huge armies, such as those in the final battles of the *Matrix* films and the *Lord of the Rings* trilogy. Such particle simulations can be used at a variety of scales, from generating armies to schooling fish to clouds of vapor moving across the screen. By giving each agent or particle rules for how to behave when a collision occurs (like when a smoke particle runs into a wall or when a soldier from one large army meets a soldier from another) each agent can behave independently according to reasonable rules (like the laws of Newtonian physics) and the appropriate large-scale behavior emerges from tweaking certain parameters that are shared by all agents of a particular type.

**Further Thoughts**

On a related note, the sequels to *The Matrix* (*The Matrix Reloaded* and *The Matrix Revolutions*) take the nature of network connections to the next logical step. Starting in the second movie, Agent Smith (Hugo Weaving) begins to download his code into the brains of humans plugged into the system, effectively copying himself. At one point, he even copies himself into the brain of an individual who has been freed from the system, thereby releasing his program from the system. Eventually, Smith copies himself onto every individual plugged into the Matrix (an excellent example of exponential growth, I might add). Although this may seem impossible, it is merely an extension of the mechanisms already at work in the computer system. For example, the rebels can easily download information from cartridges into a person’s mind in order to teach them kung fu or how to fly a helicopter. This surely involves changes in the real brain of the individual, for they must remember, at the very least, that they have this knowledge in the computer world in order to make use of it the next time they “plug in.” Thus, programming in the computer world can have effects in the real world. Without these reciprocal effects, information — about people and places and abilities — would never pass through the system of interconnected people, and the rebellion would be doomed to failure.

If one thinks of a human brain as a network of electrical connections, then the matrix of an individual’s brain is surely changed by his or her interactions with the computer-simulated reality of the Matrix. And so each individual represents a matrix as well. So we have a more complicated structure: a computer-simulated reality supported by a matrix of people connected together, each of whom is him- or herself a complex matrix of ideas. It really seems that the machines must have some other purpose for humanity than simply extracting
power from them, if they are willing to put up with all the complexity and hassle of such a system, rather than simply snipping a few nerves in the brain and having vegetative bio-batteries without a need for all these shenanigans.

If we are willing to stretch our analogy a little beyond the obvious, we can find still other examples of the relationship between mathematical matrices and the movies. One, in particular, harkens back to the situation leading to the trilogy’s climax. Agent Smith has copied himself onto every mind that is plugged in, and is close to crashing the system. At the same time, humanity is in desperate trouble in their last city, Zion, as the machines physically assault them with an unbelievably massive army. But how did this come to pass?

Earlier I mentioned that one can multiply a square matrix of probabilities by itself to determine likelihoods of events happening later. If one multiplies a typical transition matrix by itself sufficiently many times, though, an interesting thing happens. For instance, if we start with the following simple transition matrix

\[
M = \begin{pmatrix}
0.2 & 0.9 \\
0.8 & 0.1
\end{pmatrix}
\]

and multiply it by itself repeatedly, we obtain, for instance, the following matrices as its fourth and sixty-fourth powers:

\[
M^4 = \begin{pmatrix}
0.6424 & 0.4023 \\
0.3576 & 0.5977
\end{pmatrix}, \quad M^{64} \approx \begin{pmatrix}
0.52941 & 0.52941 \\
0.47058 & 0.47058
\end{pmatrix}.
\]

Notice that as \(M\) is multiplied by itself many times, the entries converge so that each row’s entries all look the same after a relatively small number of iterations. Technically speaking, a normal transition matrix for a Markov chain is a matrix with a dominant eigenvalue of 1. Multiplying the matrix by itself many times eventually leads to a matrix with the same column repeated over and over; the column is the eigenvector associated with the dominant eigenvalue and scaled so that the elements of the vector add to a total of 1. This vector then defines the steady-state distribution of the Markov chain, showing the likelihood that the long-term state of the system will be in each of the possible states. The matrix, \(M\), above, results in a steady-state distribution with approximately a 52.941 percent chance of being in the first state and a 47.058 percent chance of being in the second state.

In the case of the network in the \textit{Matrix} movies, the Architect explains that there is an inevitability to the matrix: eventually it will crash and lead to potentially disastrous consequences for humanity. In the movie, this crash is related to the copying of Agent Smith over and over until every person in the matrix is a copy of Smith. Thus, there must be something about the particulars of the matrix — and the connections between the elements of the network — such that after many iterations, the Markov chain is almost 100 percent guaranteed to be in a particular state — one with all individuals replaced by Agent Smith.

\textbf{Conclusion}

In the end, we are left with not only popular reasons for the movie title, but also mathematical reasons. The notational power of matrices and the mathematical sophistication of the related field of linear algebra provide tools that can be used to model transformations
of images, to examine the relationships among interconnected people in a network, and to analyze the likelihood of various future events. Thus, the title of the film opens our eyes to ideas that are critical to the movie’s plot, themes, and special effects.

Shakespeare’s Juliet asks, “What’s in a name?” She concludes that the object to which the name refers would be identical, regardless of its label. But in the case of a movie, which interacts with us on intellectual and emotional levels, its name has power. It suggests relationships and concepts. It provides us a framework for attaching events and ideas. It provides a lens through which we view its plot and characters. To see how important the words used to describe a narrative can be, consider the thirty other titles Hemingway considered for his novel *A Farewell to Arms* (Oldsey 14–16). Imagine how differently *The Matrix* might have been received had it been called “Our Machine Masters” or “Ghosts in the Machine.” The former suggests a rebellion against oppression, while the latter hints at mystical significance related to the soul; the chosen title, however, evokes an enigma and connections to mathematics. The alternate titles fail to capture the mathematical and scientific connections that fascinate many viewers. As it stands, we’ll be watching and re-watching the film for many years, looking for the instances of mathematics hidden within.

**Notes**

1. A shorter version of this essay originally appeared in *Math Horizons* Sept. 2008: 18–21; I thank the editor(s) of *Horizons* for permission to reprint it in its present form.

2. Specifically, if $A$ and $B$ are matrices, then we can form the sum $A+B$ exactly when $A$ and $B$ share the same number of rows and the same number of columns, and the product $AB$ exactly when the number of columns of matrix $A$ is the same as the number of rows of matrix $B$.

**Works Consulted**


