Lecture 11: Geometry and Calculus of Vectors

Reading:
Kreyszig Sections: 9.1, 9.2, 9.3, 9.4 (pages364–369, 371–374, 377–383, 384–388)

Graphical Animation: Using Time as a Dimension in Visualization

Animations can be very effective tools to illustrate time-dependent phenomena in scientific presentations. Animations are sequences of multiple images—called frames—that are written to the screen interatively at a constant rate: if one second of real time is represented by N frames, then a real-time animation would display a new image every $1/N$ seconds.

There are two important practical considerations for computer animation:

frame size An image is a an array of pixels, each of which is represented as a color. The amount of memory each color requires depends on the current image depth, but this number is typically 2-5 bytes. Typical video frames contain 1024×768 pixel images which corresponds to about 2.5 MBytes/image and shown at 30 frames per second corresponding to about 4.5 GBytes/minute. Storage and editing of video is probably done at higher spatial and temporal resolution. Each frame must be read from a source—such as a hard disk—and transfered to the graphical memory (VRAM) before the screen can be redrawn with a new image. Therefore, along with storage space the rate of memory transfer becomes a practical issue when constructing an animation.

animation rate Humans are fairly good at extrapolating action between sequential images. It depends on the difference between sequential images, but animation rates below about 10 frames per second begin to appear jerky. Older Disney-type cartoons were typically displayed at about 15 frames per second, video is displayed at 30 frames per second. Animation rates above about 75 frames per second yield no addition perceptable "smoothness." The upper bound on computer displays is typically 60 hertz.

Lecture 11 MATHEMATICA® Example 1

Introductory Animation Examples

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

Several introductory examples of animated results from equation-generated graphics are presented as model starting points. In MATHEMATICA® , the goal is produce a list of `Graphics`-objects and then display them all. The animation can be played by grouping all the individual displayed graphics objects into one super-group (i.e., a super-bracket); closing the brackets and then using the `Cell`-menu to animated the closed-and-selected group.

Two simple methods to produce animations are illustrated: 1) using `Table` with a graphics-producing argument; 2) generating a list of undisplayed graphics objects, and then displaying them using a loop structure.

1: A traveling wave is produced by iteratively plotting $\sin(kx - \omega t)$ a discrete times using `Table` together with `Plot`. `Evaluate` must be wrapped around the function, otherwise the time-variable will not be computed automatically.

2: Beats are illustrated. However, unless the `PlotRange` of each image is the same, the resulting animation would be shoddy.

3: In this case, the animation is computed first and then displayed later. The `Graphics`-objects are stored in a list using `Table` and `ParametricPlot`, but graphical display is suppressed by using `DisplayFunction→Identity`. Any list-generating function, such as `AppendTo`, can be used to produce such *Graphics-lists*.

4: Such Graphics-lists can be combined with other computed graphics objects and shown together. In this example, a list of colors is produced with `Graphics` and `CMYKColor` is computed and then used to colorize the previously computed graphics list. `Do` is used with `Show` to produce the graphics; in this case, the previously-suppressed display is forced with `DisplayFunction→ $DisplayFunction`.

5: An animation of a sequence of 3D graphics objects is produced similarly; note that `PlotRange` must be used to ensure that the z-axis remains constant between frames.

```
1
frequency = 3;
Table[
  Plot[Evaluate[Sin[x - frequency t]], {x, -Pi, Pi}], {t, 0, 2, 0.1}]

2
frequency1 = 1;
frequency2 = 2/3;
k1 = 1/2;
k2 = 4/7;
Table[Plot[Evaluate[
      Sin[k1 x - frequency1 t] + Sin[k2 x - frequency2 t]],
    {x, 0, 20 Pi}, PlotRange -> {-2, 2}], {t, 0, 10, 0.25}]

3
GraphicsList = Table[ParametricPlot[
        Evaluate[{Sin[t] + Sin[t a], Cos[t] + Cos[t a]} ],
        {t, 0, 2 Pi}, AspectRatio -> 1,
        DisplayFunction -> Identity], {a, 0, 20}];

4
Do[Show[GraphicsList[[i]],
    DisplayFunction -> $DisplayFunction],
  {i, 1, Length[GraphicsList]}]

5
RandomColors =
    Table[Graphics[CMYKColor[0.5 + 0.5 Sin[2 Pi t] ,
          0.5 + 0.5 Cos[2 Pi t], 0.5, 0]],
      {t, 0, 1, 1/Length[GraphicsList]}];
Do[Show[{RandomColors[[i]], GraphicsList[[i]]},
    AspectRatio -> 1, DisplayFunction -> $DisplayFunction],
  {i, 1, Length[GraphicsList]}]

6
Table[Plot3D[Evaluate[
      Exp[-0.01 * ((x - t)^2 + (y - t)^2)] Sin[x - t] Sin[y - t]],
    {x, -Pi, 4 Pi}, {y, -Pi, 4 Pi}, PlotRange -> {-1, 1}],
  {t, -4 Pi, 6 Pi, Pi/24}]
```

Vector Products

The concept of vectors as abstract objects representing a collection of data has already been presented. Every student at this point has already encountered vectors as representation of points, forces, and accelerations in two and three dimensions.

Review: The Inner (dot) product of two vectors and relation to projection

An inner (or dot-) project is multiplication of two vectors that produces a scalar.

$$\begin{aligned}
\vec{a}\cdot\vec{b} \equiv& \\
&\equiv a_i b_i \\
&\equiv a_i b_j \delta_{ij} \text{ where } \delta_{ij} \equiv \begin{cases} 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} \\
&\equiv (a_1, a_2, \ldots a_N) \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix} \\
&\equiv (b_1, b_2, \ldots b_N) \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix}
\end{aligned} \tag{11-1}$$

The inner product is:

linear, distributive $(k_1\vec{a} + k_2\vec{b})\cdot\vec{c} = k_1\vec{a}\cdot\vec{c} + k_2\vec{b}\cdot\vec{c}$

symmetric $\vec{a}\cdot\vec{b} = \vec{b}\cdot\vec{a}$

satifies Schwarz inequality $\|\vec{a}\cdot\vec{b}\| \leq \|\vec{b}\|\|\vec{a}\|$

satifies triangle inequality $\|\vec{a}+\vec{b}\| \leq \|\vec{b}\| + \|\vec{a}\|$

If the vector components are in a cartesian (i.e., cubic lattice) space, then there is a useful equation for the angle between two vectors:

$$\cos\alpha = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|} = \hat{n_a}\cdot\hat{n_b} \tag{11-2}$$

where $\hat{n_i}$ is the unit vector that shares a direction with i. *Caution: when working with vectors in non-cubic crystal lattices (e.g, tetragonal, hexagonal, etc.) the angle relationship above does not hold. One must convert to a cubic system first to calculate the angles.*

The projection of a vector onto a direction $\hat{n_b}$ is a scalar:

$$p = \vec{a}\cdot\hat{n_b} \tag{11-3}$$

Review: Vector (or cross-) products

The vector product (or cross $\times$) differs from the dot (or inner) product in that multiplication produces a vector from two vectors. One might have quite a few choices about how to define such a product, but the following idea proves to be useful (and standard).

normal Which way should the product vector point? Because two vectors (usually) define a plane, the product vector might as well point away from it.

The exception is when the two vectors are linearly dependent; in this case the product vector will have zero magnitude.

The product vector is normal to the plane defined by the two vectors that make up the product. A plane has two normals, which normal should be picked? By convention, the "right-hand-rule" defines which of the two normal should be picked.

magnitude Given that the product vector points away from the two vectors that make up the product, what should be its magnitude? We already have a rule that gives us the cosine of the angle between two vectors, a rule that gives the sine of the angle between the two vectors would be useful. Therefore, the cross product is defined so that its magnitude for two unit vectors is the sine of the angle between them.

This has the extra utility that the cross product is zero when two vectors are linearly-dependent (i.e., they do not define a plane).

This also has the utility, discussed below, that the triple product will be a scalar quantity equal to the volume of the parallelepiped defined by three vectors.

The triple product,

$$\begin{aligned} \vec{a} \cdot (\vec{b} \times \vec{c}) = (\vec{a} \times \vec{b}) \cdot \vec{c} = \\ \|\vec{a}\|\|\vec{b}\|\|\vec{c}\| \sin \gamma_{b-c} \cos \gamma_{a-bc} = \\ \|\vec{a}\|\|\vec{b}\|\|\vec{c}\| \sin \gamma_{a-b} \cos \gamma_{ab-c} \end{aligned} \qquad (11\text{-}4)$$

where γ_{i-j} is the angle between two vectors i and j and γ_{ij-k} is the angle between the vector k and plane spanned by i and j. is equal to the parallelepiped that has $\vec{a}$, $\vec{b}$, and $\vec{c}$, emanating from its bottom-back corner.

If the triple product is zero, the volume between three vectors is zero and therefore they must be linearly dependent.

Lecture 11 MATHEMATICA® Example 2

Cross Product Example

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

This is a simple demonstration of the vector product of two spatial vectors and comparison to the the memorization device:

$$\vec{a} \times \vec{b} = \det \begin{pmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{pmatrix}$$

1: `Cross` produces the vector product of two symbolic vectors $\vec{a}$ and $\vec{b}$ of length 3.

2: `Det` produces the same result using the memorization device.

3: `Coefficient` is used to extract each vector component.

```
1 crossab = Cross[{a1, a2, a3} , {b1, b2, b3}]

2 detab = Det[( i  j  k
               a1 a2 a3
               b1 b2 b3 )]

3 testcrossab = {Coefficient[detab, i],
      Coefficient[detab, j], Coefficient[detab, k]}

4 testcrossab == crossab
```

Derivatives Vectors

Consider a vector, $\vec{p}$, as a point in space. If that vector is a function of a real continuous parameter for instance, t, then $\vec{p}(t)$ represents the loci as a function of a parameter.

If $\vec{p}(t)$ is continuous, then it sweeps out a continuous curve as t changes continuously. It is very natural to think of t as time and $\vec{p}(t)$ as the trajectory of a particle—such a trajectory would be continuous if the particle does not disappear at one instant, t, and reappear an instant later, $t + dt$, some finite distance distance away from $\vec{p}(t)$.

If $\vec{p}(t)$ is continuous, then the limit:

$$\frac{d\vec{p}(t)}{dt} = \lim_{\Delta t \to 0} \frac{\vec{p}(t + \Delta t) - \vec{p}(t)}{\Delta t} \tag{11-5}$$

Notice that the numerator inside the limit is a vector and the denominator is a scalar; so, the derivative is also a vector. Think about the equation geometrically—it should be apparent that the vector represented by the derivative is locally tangent to the curve that is traced out by the points $\vec{p}(t - dt)$, $\vec{p}(t)$ $\vec{p}(t + dt)$, etc.

Lecture 11 MATHEMATICA® Example 3

Visualizing Time-Dependent Vectors and their Derivatives

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

Examples of $\vec{x}(t)$ and $d\vec{x}/dt$ are illustrated as curves and as animations.

1: A list of three time-dependent components for (x, y, z) is constructed and...

2: Displayed with `ParametricPlot3D`.

3: This is a second example of a curve, `DeltoidSpiral`, for which the derivative will be calculated.

6: The derivative operator `D` is a *threadable function* so it will operate on each component of its vector argument.

7: This will display the curve that is tangent to `DeltoidSpiral` at each time t. Because the z-component is linear in t, the resulting tangent curve has a constant value of z.

9: This function will produce a graphical object which is the image superposition of *all* the results of `DeltoidSpiral` for $t = 0$ up to some specified value $t = t_{lim}$—i.e., a visible curve.

11: A graphics list of the development of the curve and its derivative is constructed and...

12: subsequently animated with a `Do`.

```
1  TimeVector = {Cos[4 π t], Sin[8 π t], Sin[2 π t]}

2  ParametricPlot3D[TimeVector, {t, 0, 1}]

3  DeltoidSpiral =
     {(2 Cos[π t] + Cos[2 π t]), (2 Sin[π t] − Sin[2 π t]), t/3}

4  pp = ParametricPlot3D[DeltoidSpiral,
       {t, −3, 3}, AxesLabel → {"x", "y", "z"}]

5  Show[
     {Graphics3D[Thickness[0.01]], Graphics3D[Hue[1]], pp}]

6  dDSt = D[DeltoidSpiral, t]

7  ppdt = ParametricPlot3D[dDSt,
       {t, −3, 3}, AxesLabel → {"x", "y", "z"}]

8  Show[
     {Graphics3D[Thickness[0.01]], Graphics3D[Hue[0.3]], ppdt}]

9  ppdtlim[tl_] := {Graphics3D[Thickness[0.01]],
       Graphics3D[Hue[0.3]], ParametricPlot3D[
         0.33 * dDSt, {t, 0, tl}, AxesLabel → {"x", "y", "z"},
         Compiled → False, DisplayFunction → Identity]}

10 dtlim[tl_] := {Graphics3D[Thickness[0.01]],
       Graphics3D[Hue[1]], ParametricPlot3D[
         DeltoidSpiral, {t, 0, tl}, AxesLabel → {"x", "y", "z"},
         Compiled → False, DisplayFunction → Identity]}

11 TheGraphicsList = Table[{ppdtlim[t], dtlim[t]}, {t, .05, 3, .05}];

12 Do[Show[TheGraphicsList[[i]],
       PlotRange → {{−4.25, 4.25}, {−4.25, 4.25}, {0, 1}},
       PlotRegion → {{0, 1}, {0, 1}}, SphericalRegion → True,
       Boxed → True, BoxRatios → {1, 1, 1},
       ViewCenter → {0, 0, 0.5}, AspectRatio → 1,
       ViewPoint → {1.2, −3, 2}], {i, 1, Length[TheGraphicsList]}]
```

Review: Partial and total derivatives

One might also consider a time- and space-dependent vector field, for instance $\vec{E}(x, y, z, t) = \vec{E}(\vec{x}, t)$ could be the force on a unit charge located at $\vec{x}$ at time t.

Here, there are many different things which might be varied and give rise to a derivative. Such questions might be:

1. How does the force on a unit charge differ for two nearby unit-charge particles, say at (x, y, z) and at $(x, y + \Delta y, z)$?
2. How does the force on a unit charge located at (x, y, z) vary with time?
3. How does the the force on a particle change as the particle traverses some path $(x(t), y(t), z(t))$ in space?

Each question has the "flavor" of a derivative, but each is asking a different question. So a different kind of derivative should exist for each type of question.

The first two questions are of the nature, "How does a quantity change if only one of its variables changes and the others are held fixed?" The kind of derivative that applies is the partial derivative.

The last question is of the nature, "How does a quantity change when all of its variables depend on a single variable?" The kind of derivative that applies is the total derivative. The answers are:

1.
$$\frac{\partial \vec{E}(x,y,z,t)}{\partial y} = \left(\frac{\partial \vec{E}}{\partial y}\right)_{\text{constant}x,z,t} \tag{11-6}$$

2.
$$\frac{\partial \vec{E}(x,y,z,t)}{\partial t} = \left(\frac{\partial \vec{E}}{\partial t}\right)_{\text{constant}x,y,z} \tag{11-7}$$

3.
$$\frac{d\vec{E}(x(t),y(t),z(t),t)}{dt} = \frac{\partial \vec{E}}{\partial x}\frac{dx}{dt} + \frac{\partial \vec{E}}{\partial y}\frac{dy}{dt} + \frac{\partial \vec{E}}{\partial z}\frac{dz}{dt} + \frac{\partial \vec{E}}{\partial t}\frac{dt}{dt} = \nabla \vec{E}(\vec{x}(t),t) \cdot \frac{d\vec{x}}{dt} + \frac{\partial \vec{E}}{\partial t} \tag{11-8}$$

Time-Dependent Scalar and Vector Fields

A physical quantity that is spatially variable is often called a *spatial field*. It is a particular case of a field quantity.

Such fields can be simple scalars, such as the altitude as a function of east and west in a topographical map. Vectors can also be field quantities, such as the direction uphill and steepness on a topographical map— this is an example of how each scalar field is naturally associated with to its *gradient field*. Higher dimensional objects, such as stress and strain, can also be field quantities.

Fields that evolve in time are *time-dependent fields* and appear frequently in physical models. Because time-dependent 3D spatial fields are four-dimensional objects, animation is frequently used to visulize them.

For a working example, consider the time-evolution of "ink concentration" $c(x,y,t)$ of a very small spot of ink spilled on absorbant paper at $x = y = 0$ at time $t = 0$. This example could be modeled with Fick's first law:

$$\vec{J} = -D\nabla c(x,y,t) = -D\left(\frac{\partial c}{\partial x} + \frac{\partial c}{\partial y}\right) \tag{11-9}$$

where D is the diffusivity that determines "how fast" the ink moves for a given gradient ∇c, and $\vec{J}$ is a time-dependent vector that represents "rate of ink flow past a unit-length line segment oriented perpendicular to $\vec{J}$. which leads to the two-dimensional diffusion equation

$$\frac{\partial c}{\partial t} = D\left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2}\right) \tag{11-10}$$

For this example, the solution, $c(x,y,t)$ is given by

$$c(x,y,t) = \frac{c_o}{4\pi Dt} e^{-\frac{x^2+y^2}{4Dt}} \tag{11-11}$$

where c_o is the initial concentration of ink.

Lecture 11 MATHEMATICA® Example 4

Visualizing a Solution to the Diffusion Equation

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

The solution to 2D diffusion equation for point source initial conditions and its resulting flux, $\vec{J}$ is illustrated with several types of animation.

1: The diffusivity it set to one, this effectively sets the length and time scales for the subsequent simulation.

2: The concentration plotted with `Plot3D` as a two-dimensional surface embedded in three-dimensions and animated as a function of time.

3: A simpler graphical representation is obtained with `ContourPlot` by plotting contours of constant concentration. The resulting animation is of a two-dimensional object.

4: To plot the flux which is vector field, the package `Graphics‘PlotField‘` is loaded for its `PlotVectorField` function.

8: An animation of `PlotVectorField` for the flux can be obtained. However, getting the sequential frames to be consistent and the size of the arrows representing the vectors requires a somewhat complicated use of `PlotVectorField`'s options such `ScaleFunction`. This may serve as a working first example for beginning users.

9: Animations of vector fields can be simplified, by converting them into a set of scalar representations. In this example, the dot-product exracts the J_x-component only which is animated with contourplots.

```
1  concentration = Exp[-(x^2 + y^2)/(4 Diffusivity t)] / (4 Pi Diffusivity t)
   Diffusivity = 1;

2  Table[Plot3D[Evaluate[concentration],
       {x, −2, 2}, {y, −2, 2}, PlotPoints → 40,
       PlotRange → {0, 4}], {t, 0.0125, .25, .0125}]

3  Table[ContourPlot[Evaluate[concentration], {x, −2, 2},
       {y, −2, 2}, PlotPoints → 40, PlotRange → {0, 0.5},
       ColorFunction → (Hue[1 − 0.75 #] &)], {t, 0.0125, 1, .025}]

4  << Graphics`PlotField`

5  flux = {−D[concentration, x], −D[concentration, y]}

6  PlotVectorField[flux /. t → 0.8, {x, −2, 2}, {y, −2, 2},
     PlotPoints → 20, ColorFunction → (Hue[1 − 0.75 #] &)]

7  PlotVectorField[flux /. {t → 0.2}, {x, −2, 2},
     {y, −2, 2}, PlotPoints → 21, Frame −> True,
     ScaleFunction −> (10.0 # &), MaxArrowLength −> 50,
     ScaleFactor −> None, ColorFunction → (Hue[1 − 0.75 #] &)]

8  Table[PlotVectorField[flux, {x, −2, 2}, {y, −2, 2},
       PlotRange −> {{−3, 3}, {−3, 3}}, Frame −> True,
       PlotPoints → 19, ScaleFunction −> (100.0 # &),
       MaxArrowLength −> 10, ScaleFactor −> None,
       ColorFunction → (Hue[1 − 0.75 #] &)], {t, 0.01, 3.01, .05}]

9  Table[ContourPlot[flux.{1, 0}, {x, −2, 2},
       {y, −2, 2}, PlotPoints → 40, PlotRange → {0, 0.5},
       ColorFunction → (Hue[1 − 0.75 #] &)], {t, 0.01, 1.01, .05}]

10 Table[ContourPlot[
       Max[{flux.{1, 1}, flux.{−1, 1}, flux.{1, −1}, flux.{−1, −1}}]/
         √2, {x, −2, 2}, {y, −2, 2},
       PlotPoints → 40, PlotRange → {0, 0.5},
       ColorFunction → (Hue[1 − 0.75 #] &)], {t, 0.01, 1.01, .05}]
```

All vectors are not spatial

It is useful to think of vectors as spatial objects when learning about them—but one shouldn't get stuck with the idea that all vectors are points in two- or three-dimensional space. The spatial vectors serve as a good analogy to generalize an idea.

For example, consider the following chemical reaction:

Reaction:	H_2	$\frac{1}{2}O_2$	$\rightleftharpoons$	H_2O	
Initial:	1	1	$\rightleftharpoons$	0	The composition could be written as a vector:
During Rx.:	$1-\xi$	$1-\frac{1}{2}\xi$	$\rightleftharpoons$	ξ	

$$\vec{N} = \begin{pmatrix} \text{moles } H_2 \\ \text{moles } O_2 \\ \text{moles } H_2O \end{pmatrix} = \begin{pmatrix} 1-\xi \\ 1-\frac{1}{2}\xi \\ \xi \end{pmatrix} \tag{11-12}$$

and the variable ξ plays the role of the "extent" of the reaction—so the composition variable $\vec{N}$ lives in a reaction-extent (ξ) space of chemical species.