Functions as Infinitely Long Column Vectors

- The Switch from Fixed Vectors to Functions
- Solutions to Differential Equations are Graphs
- Numerical analysis, Graphs and Fixed Vectors
- Trouble Storing Sampled Functions
- Column Vectors with Infinitely Many Components
- From Sampled Data to Infinitely Long Column Vectors
- The Algebra of Infinitely Long Column Vectors

The Switch from Fixed Vectors to Functions as Vectors

Linear algebra presents fixed vectors as replacements for arrows used in physics and engineering courses. Then the reader is asked to make another leap: treat a solution $\boldsymbol{y(x)}$ of a differential equation as a vector $\vec{\boldsymbol{y}}$.

This short account of how to effect the switch was written in response to the following Rookie question, which gets quickly to the heart of the matter.

If a solution $\boldsymbol{y(x)}$ of a DE is a vector, then why can't I burst it into components as

$$\vec{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \\ \vdots \\ \boldsymbol{y_n} \end{pmatrix}?$$

Solutions to Differential Equations are Graphs

The default storage system used for applications involving ordinary or partial differential equations is a *function space*. The data item packages for differential equations are their solutions, which are *functions*.

A wider viewpoint is that a function $\boldsymbol{y(x)}$ is a package $\vec{\boldsymbol{y}}$, which when burst to reveal the contents shows an algebraic equation like $\boldsymbol{y = 1 + x^2}$, or a graphic with implied graph window, or a photo or scanned JPG of a graphic. In all these examples:

Solutions to differential equations are ***not*** *column vectors of numbers.*

Numerical Analysis, Graphs and Fixed Vectors

An alternative view, adopted by researchers in numerical solutions of differential equations, is that a solution is a table of numbers, consisting of pairs of $\boldsymbol{x}$ and $\boldsymbol{y}$ values.

These researchers might view a function as being a fixed vector. Their unique intuitive viewpoint is that a function is a **graph** and a graph is determined by so many **dots**, which are practically obtained by **sampling** the function $\boldsymbol{y(x)}$ at a reasonably dense set of $\boldsymbol{x}$-values. The approximation is

$$\text{Graph of y(x)} \approx \vec{y} = \begin{pmatrix} y(x_1) \\ y(x_2) \\ \vdots \\ y(x_n) \end{pmatrix}$$

where $\boldsymbol{x_1}$, …, $\boldsymbol{x_n}$ are the **samples** and $\boldsymbol{y(x_1)}$, …, $\boldsymbol{y(x_n)}$ are the **sampled values** of function $\boldsymbol{y}$.

Trouble Storing Sampled Functions

The trouble with the approximation

$$\text{Graph of y(x)} \approx \vec{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y}(\boldsymbol{x}_1) \\ \boldsymbol{y}(\boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{y}(\boldsymbol{x}_n) \end{pmatrix}$$

is that two different functions may need different sampling rates to properly represent their graphic. The result is that the two functions might need data storage systems of different dimensions, e.g., $\boldsymbol{f}$ needs its sampled values in $\boldsymbol{R}^{200}$ and $\boldsymbol{g}$ needs its sampled values in $\boldsymbol{R}^{400}$.

The absence of a universal fixed vector storage system for sampled functions explains the appeal of a storage system like the set of all functions.

Column Vectors with Infinitely Many Components

Novices suggest a work-around for the lack of a universal numerical data storage system for sampled functions:

Develop a theory of column vectors with infinitely many components.

It may help you to think of any function $\boldsymbol{f}$ as an infinitely long column vector, with one entry $\boldsymbol{f}(\boldsymbol{x})$ for each possible sample $\boldsymbol{x}$, e.g.,

$$\vec{\mathrm{f}} = \begin{pmatrix} \vdots \\ \boldsymbol{f}(\boldsymbol{x}) \\ \vdots \end{pmatrix} \quad \text{level } \boldsymbol{x}$$

It is not clear how to order or address the entries of such a column vector: at algebraic stages it hinders. Can computers store infinitely long column vectors? The easiest path through the algebra is to deal exactly with functions and function notation.

From Sampled Data to Infinitely Long Column Vectors

There is something attractive about the change from sampled approximations to a single column vector with infinite extent:

$$\vec{\mathrm{f}} \approx \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix} \to \begin{pmatrix} \vdots \\ f(x) \\ \vdots \end{pmatrix} \quad \text{level } x$$

The thinking behind the *level* x annotation is that x stands for one of the infinite possibilities for an invented sample. Alternatively, with a rich set of invented samples $x_1, \ldots, x_n$, value $f(x)$ equals approximately $f(x_j)$, where x is closest to some sample x_j.

The Algebra of Infinitely Long Column Vectors

How to add and scalar multiply infinitely long column vectors?

While the correct way to proceed is with function notation, here's what you can use for definitions in your own work, to develop intuition.

$$\vec{f} + \vec{g} = \begin{pmatrix} \vdots \\ f(x) \\ \vdots \end{pmatrix} + \begin{pmatrix} \vdots \\ g(x) \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ f(x) + g(x) \\ \vdots \end{pmatrix} \text{ level } x$$

$$c\vec{f} = c \begin{pmatrix} \vdots \\ f(x) \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ cf(x) \\ \vdots \end{pmatrix} \text{ level } x$$

The thinking here is numerical, that the column vectors represent sampled data.

The **fundamental ideas of Dirichlet** appear directly from the previous display by not bothering to write parentheses or vertical dots: $f + g$ is a graph defined by the relation $(f + g)(x) = f(x) + g(x)$ and cf is a graph defined by the relation $(cf)(x) = c\,f(x)$. This is Dirichlet's *function notation*, universally adopted throughout all of modern mathematics.