Remodeling the mathematics core at USAFA

Daniel Kaplan

USNA 02-11-2022

Introduction

- A new calculus core at USAFA
  - Core required of all cadets: Calc I/II
    - There's a third core "math" course not in the perview of this project: Statistics
  - New:
    - Enhance student engagement to increase number of students majoring in STEM disciplines.
    - Faculty discontent with traditional course: Why are we teaching this stuff?
- This talk contains my views, as a private citizen, and doesn't necessarily reflect any official view of the USAFA.

A draft textbook for the course, "MOSAIC Calculus," is available freely online at http://mosaic-web.org/MOSAIC-Calculus/

Timeline

- circa 2017: USAFA determined to replace Calc I/II
  - USAFA identified Macalester's "Applied Multivariate Calculus I" as a model, a course I developed in 2003-2012
- late 2018: I was recruited to lead the curricular development.
  - Increase connections to statistics and data science.
  - Increase connections to physics and engineering.
  - Add a meaningful computational component.
- July 2020: Project started
- August 2020: Prototype course commenced.
  - 3 faculty teaching 150 cadets.
  - Initial textbook Kilty and McAllister, *Mathematical Modeling and Applied Calculus*

Planned timeline

- August 2021: Phased expansion
  - 12 faculty teaching 400 cadets
  - Home grown textbook: *MOSAIC Calculus*
- August 2022: Anticipated full deployment:
  - approximately 25 faculty and 800 cadets
- 2022-2023: Work on statistics core course

Goals for USAFA Calculus

- Provide a useful introduction to
  - The use and analysis of multivariate functions and multi-dimensional space.
  - Develop basic competencies in the mathematics and phenomena of dynamical systems, e.g. behavior and analysis of ordinary differential equations.
  - Lay the foundations for building and understanding models based on data. (Statistical inference deferred to stats core course.)
  - Basic competencies in technical computing.
- Be accessible and meaningful to students who haven't studied calculus previously.
- Fit into 6 credit hours over two semesters.

The elephant in the room

1. Many students don't succeed with differential or integral calculus.
2. Because they don't succeed in the introductory phases, they traditionally don't reach the more "advanced" topics we set as our goals. (Previous slide.)
    - Very few fields outside of engineering, physics, and engineering have any requirements in these advanced topics.
    - Traditionally, intro- or mid-level stats does not either. But is this sustainable in an era of data-science and machine learning?

Ways to address low success rates

1. Change the goal: Don't require calculus. Substitute statistics instead.
    i. A solid idea for students in health sciences, where the calculus had no genuine connections to the downstream courses.
2. Divide students by aptitude.
    i. Remediation such as pre-calculus or "college algebra." Poor record of leading to success.
    ii. Streamlined courses such as "business calculus," but doesn't reach the topics we're interested in here.
    iii. Place high-aptitude students into advanced courses. Doesn't address the core problem of low success rates.

USAFA has a "core course" philosophy, so these approaches don't apply.

A third way

This is a metaphor.

> *The "Third Way" is used to refer to a set of political beliefs and principles that is neither extremely right-wing nor extremely left-wing. — Collins Dictionary*
>
> *Third way, in politics, a proposed alternative between two hitherto dominant models, namely left-wing and right-wing political groups. — Encyclopaedia Britannica*
>
> *A political movement in which the development of business [substitute: "skills for downstream study"] is balanced with the needs of society [substitute: "accessibility to students].* — Cambridge English dictionary*

A third way?

Note the ***?***

What evidence do we have that this will work. Some, but limited.

"Applied Calculus" at Macalester College

- 20 years of experience
- Initially, a "terminal" course, *Applied Calculus* to support students whose major requires "a single semester" of calculus.
  i. Unexpectedly turned into a gateway to advanced study for students who never knew there were genuine **applications** of math. This doubled the number of math majors and led to new majors in *Applied Math and Statistics* and, recently, *Data Science*.
  ii. The doubling occurred < 2010. Since then, enrollment has continued to soar, but due to external factors.
- Now is the first of a three-course sequence: *Applied Multivariate Calculus I/II/III*
- About half of all students take one or more of the courses.

But only half. A sort of self-selected tracking so not representative of the situations in a core course.

“Z-calculus” at USAFA

Finishing two years experience at USAFA, 500+ students.

But little data yet on downstream success.

Third way: description

How to make calculus more comprehensive yet accessible.

- Purposefully select theoretical topics and calculation techniques.
- Calculus is about describing change and relationships.
  - Infinitesimal and infinite play supporting roles, not the main characters.
- Notation and type:
  - Make it possible to determine from symbol what kind of thing is being referred to.
  - Have a close computer-language equivalent for names and operations.

Third way: description (cont.)

- **Avoid gratuitous algebra.**
  - Very little drill on differentiation and anti-differentiation of functions made up for the purpose of drill.
  - Use small set of "basic modeling functions."
- Emphasize functions of two variables.
  - No special notation for functions of one variable. (Such special notation leads to bad habits, even if it might simplify things in the short run.)

Calculus theory: Functions, not equations

- Functions take *inputs* and produce an *output*.
- The output is computed by an algorithm.
- Formulas are one form of algorithm, but there are others.
- All the calculus operations can be applied easily to any form of function.
- The word "variable" is reserved for data and statistics,

Not $y = x^2$. Instead, $f(x) \equiv x^2$ or `f <- makeFun(x^2 ~ x)`.

x^2 is an *expression* not a function.

Calculus theory: Approximation

- Approximation, not exactitude
  - low-order polynomial approximation; constant, linear, interaction, quadratic
  - Taylor polynomials (not "series") are one form, but there are others.
  - "Euler method"
- Differentiation and anti-differentiation are about relationships, not process.
  - Teach differentiation and anti-differentiation hand in hand for pattern-book functions
  - Don't pull a rabbit (fundamental theorem of calculus) out of a hat (area).

Calculus theory: Confirmation not derivation.

Example: is finite-difference derivative numerically stable as dt gets smaller?

Calculus techniques

The computational techniques used in applications of calculus

- Small, enumerated set of calculus operations: "something you do to a function."
    i. function evaluation,
    ii. differentiation,
    iii. anti-differentiation,
    iv. solving,
    v. argmax,
    vi. iteration.
- All have computational implementation.
- Most have graphical implementation.

Avoiding gratuitous algebra.

- Use almost exclusively a small set of "basic modeling functions."
- Everyone: Symbolic derivatives and anti-derivatives of "basic modeling functions" on first sight.
- Allow computing instead of algebra.
- Allow graphics instead of algebra.

Principles of notation

i. By looking at a bit of notation, you should be able to say what kind of thing it refers to.
ii. Strive to have one way of saying something and to be consistent.
iii. Notation for all functions should support functions of multiple inputs.
iv. Make **definition** explicit.
v. Strive to be compatible with computer notation and parallel it if possible. Example: hardly any computer language will let you name something `f'` or `df/dx`.
vi. Avoid ambiguous terms, even if they are traditional.

Examples of notation

Functions: A name followed by a parenthesis. So $\sin(x)$ not $\sin x$. Can use sin() to name the function. - exception: exponentials are often written with superscripts rather than parentheses: e^{kt}, x^n.

Function inputs: From back of alphabet: u, v, w, x, y, z. (These are *inputs*, not "variables." Let's leave "variables" for statistics.)

Function evaluation: $f(x = 4, t = 3)$

Function definition: - $f(x) \equiv xe^{kx}$ - `f <- makeFun(x * exp(k*x) ~ x)` - NEVER $y = 2x^2$

Examples of notation (cont.)

Parameters: Following tradition, e.g. k, P, A, ω,

Coefficients: Mostly from front of alphabet. Numerical subscripts.

Specific values of inputs: e.g. - x_0 - $x^\star$ for argmax. (But maybe I don't need to allow superscripts.)

Vectors and matrices are decorated with harpoons: $\overset{\rightharpoonup}{\mathbf{u}}$ and $\overset{\leftrightharpoons}{\mathbf{A}}$.

Derivatives:

- $\partial_t f(t)$.
- `dt_f <- D(f(t) ~ t`

Small set of core modeling functions

"Pattern-book functions" don't have parameters but do have English names. All have a single input.

- one()
- identity()
- recip()
- Exponential: e^x
- Logarithm: ln()
- Power-law: x^p
- Sinusoid: sin()
- Gaussian: dnorm()
- Sigmoid: pnorm()

Students should be able to sketch any pattern-book functions on demand, identify gaps in domain (e.g. ln() and recip()), special input/output pairs, and horizontal and vertical asymptotes.

Basic modeling functions

Parameterized versions of pattern-book functions, e.g.,

- $\sin\left(\frac{2\pi}{P}t\right)$
- $\text{dnorm}(x, \text{mean}, \text{sd})$
- e^{kt}

Must also be able to write on sight symbolic derivative and anti-derivative of every pattern-book and basic-modeling function.

One exception for anti-differentiation: Don't need to know $\int \text{dnorm}(x) dx$.

Functions of with multiple inputs

Primarily constructed from single-input basic modeling functions by

- linear combination
- products

Graphics modalities

Be able to construct and interpret

- Graph of function of one input
- Contour plot for functions with two inputs
- Vector field
- Path
- Inequality constraint
- Point plot for data

Quantities vs numbers

The inputs to basic modeling functions are ***quantities*** and have a dimension and units. The outputs are also quantities.

Functions created by differentiation or anti-differentiation (typically) have different output dimension.

Linear algebra goals

1. Understand least squares in a way that supports statistical inference in a later course.
2. Provide concrete, accessible framework to practice modeling/interpretation skills.
3. Eigenvalues/vectors (in dynamics)

Matrices are collections of (column) vectors. A matrix times a vector is a linear combination of the column vectors.

Linear algebra topics

1. Vectors: length, angle, dot product
2. Linear combinations and subspaces
   - builds on earlier construction of multi-input functions and low-order polynomial approximation
   - Matrix is a collection of (column) vectors; defines a subspace.
3. Projection of a vector $\vec{\mathbf{b}}$ onto a subspace to produce $\widehat{\mathbf{b}}$
   - *confirm* that claimed solution has the right properties
4. "Target problem" — find a linear combination of vectors that reaches $\widehat{\mathbf{b}}$.
5. Introduction to "least squares" and R^2
6. Basis set for function decomposition: Fourier

Linear algebra topics not needed

- determinants: not on topic
- inverses/singular matrices: not needed since hardly any of our matrices will be square
- gaussian elimination or similar non-trivial mechanics

Dynamics/ODE topics

- First-order linear and nonlinear DEs
- Fixed points and stability
- Sets of first-order linear and nonlinear DEs
  - Nonlinear: to reinforce modeling concepts and low-order polynomial approximations.
  - Linear: Stability and oscillation
- Eigenvalues and eigenvectors of linear flows, complex exponentials.

A calculus API

1. A minimum set of **algebra skills** needed for success in the course.
   - straight lines, slopes and y-intercept
   - recognize roots of a quadratic polynomial
   - simple exponent manipulation, distinguish between x^n and n^x
   - recognize functional notation, functions with one, two, ... inputs.

This is for students starting at Block 1.

Calculus API (cont)

2. A minimum set of **notation** that needs to be assimilated.
3. **Computing framework** to provide a non-algebra means to work with calculus concepts.
4. Small set of graphics and graphics skills.

Missing elements of an API

Components are needed for a comprehensive solution but that I could not define on my own:

- Course support software, including homework systems.
- Interactive graphics system for developing concepts.

A calculus construction kit

Design philosophy: Future users of the course should be able to design their own courses, adding new components as suits their environment:

Examples:

- A proof based component about limits
- Components closely tied to specific engineering fields, biology, etc.
- A modeling practicum

The course should be modular.

Abstract

USAFA has had a two-semester core calculus course for about 60 years. Based on a traditional Calc I/II sequence, it has changed incrementally over the years. Several years ago, USAFA determined that incremental change is insufficient and that a new core course should be developed from scratch. The new course is oriented around modeling and computation and includes topics not usually seen in Calc I/II such as dimensional analysis, linear algebra, and dynamics.

USAFA does not have tracks in its core curriculum, so the new course has to be accessible to all cadets, regardless of math background, aptitude, or intended major. In this talk, I'll briefly outline the course, focusing mainly on how the course can be accessible while being engaging and meaningful for students who have had previous calculus and who are heading toward STEM majors.

A draft textbook for the course, "MOSAIC Calculus," is available freely online at mosaic-web.org/MOSAIC-Calculus/

Bio

Danny Kaplan is distinguished visiting professor at USAFA, where for two years he has worked on the new calculus core. He is Professor Emeritus of Mathematics, Statistics, and Computer Science at Macalester College, where he introduced new curricula in calculus, scientific computing, applied math, and statistics. He has written textbooks on a variety of topics: nonlinear dynamics, scientific programming, statistical modeling, and data science. (His statistical modeling book was used briefly at USNA.) He received the Lifetime Achievement Award from the Consortium for the Advancement of Undergraduate Statistics Education. His doctorate is in biomedical engineering from Harvard University. He also holds a master's degree in Engineering-Economy Systems from Stanford.