STEM guide joe

Joe Loy

ZBTHS 2017, Stanford EE 2022

1 Processes

1.1 General Remarks

1. Make sure you study stuff that you enjoy and see yourself doing for a long time.

2. Be consistent and disciplined. Emphasize making small, gradual steps that appreciate over time than intense, unsustainable measures that often lead to burnout or demotivation.

3. Seek out a variety of explanations of topics, especially ones you struggle with, since there can be different explanations of the same thing. Try to find the easiest explanation for you to understand and start trying to solve problems. After you have a basic understanding of something, try to understand it a different way (if possible) or in a deeper way.

4. The single best way to get great at math and science is to do math and science. This can manifest itself in solving problems, doing experiments, and building projects. Textbooks and lectures are great, but we learn by doing. Try to allocate a large percentage of your time to active forms of learning (like the ones listed above)

5. Do not be too prideful or afraid to admit mistakes. Math and science will be filled with a variety of simple mistakes. Maintain consistency and discipline, use active forms of learning, and ask knowledgeable individuals for help when you have given something and honest effort and cannot move forward.

6. Although we want to maintain consistency in our training methods, we cannot expect results to be consistent. Often times, results can be very non-linear, with alternating bouts of seemingly little progress to bouts of immense progress. Quality standard operating procedures are meant to make results as consistent and optimal as possible, but do not allow a lack of results to make you too unhappy.

1.2 How to read a textbook

Stack Overflow : https://math.stackexchange.com/questions/279079/how-to-read-a-book-in-mathematics

Cuesta College : https://www.cuesta.edu/student/resources/ssc/study$_guides/mathematics/214_math_text.h$

UNC : https://learningcenter.unc.edu/tips-and-tools/readingmathtexts/

My advice: Keep it simple. Read chapter, understand chapter, do easy practice problems, do hard practice problems. Try to understand things from as many points of view as possible. Do a certain small batch of practice problems, check for answers, and correct the wrong ones as well as the ones you could not solve.

1.3 How to read a paper

See "How to Read and Understand a Paper" by Nicholas Higham in Princeton Companion to Applied Mathematics

1.4 How to ask for help

Make an honest effort at understanding something given the materials you have. If assigned class materials arent helping, try reading a different textbook on the subject that is generally well regarded, checking out a lecture or video on Youtube, or looking at examples of similar problems elsewhere and seeing if you can recreate the results on the problem at hand. Before asking for help, you should atleast write down all of the given variables of the problem, include relevant pictures and charts, and be able to describe what the problem is asking for. Try to have as much of an attempt of the problem written down as possible, so that the feedback given to you can be as specific as possible. Don't just find out what the answer is, but how you got the answer and try to see if you can solve a similar problem in a timely manner with minimal help.

1.5 How to learn something new

1.6 How to go from good to great

Master the fundamentals. Follow optimal training procedures. Stay up to date on the state of the art of your craft. Incrementally work on harder (but still interesting and meaningful) problems and projects.

1.7 How to create a schedule

Be realistic with yourself. Make meaningful goals. Talk to teachers, advisors, and other knowledgeable folks that know you decently well about what a good schedule / work strategy is for you.

1.8 Good Habits

1. Set aside a certain amount of time to do work for each day of the week
2. Read a certain amount of pages or chapters of a book every day
3. Solve a certain amount of problems every day
4. Write a certain number of lines of code every day
5. Set aside a certain amount of time to work on a set of problems or project without any distractions. Try to aim from anywhere from 30 minutes to a couple hours. Try to spread out sessions of intense work with short walks or something to help you calm down and prevent mental fatigue / frustration.

2 Resources

2.1 Books

1. How To Solve It by George Polya
   Reading the first section of this book will provide you with a good standard operating procedure for solving any math or science problem. Polya's 4 step process could be regarded as the math equivalent of an actstac or melcon in writing.
2. The Beginning of Infinity (BOF) and The Fabric of Reality (FOR) by David Deutsch
   The first three chapters BOF are some of the most influential I have ever read. If you want to understand why math and science are such powerful tools and why it is so damn effective, these books provide great explanations.
3. Incerto by Nassim Taleb
   Incerto is comprised of Fooled by Randomness, The Black Swan, Antifragile, and Skin in the Game. The titles of each book do a great job of encapsulating the central ideas of each book. Taleb does a great job of providing examples and elaborations of Popper's epistemology, highlighting the short comings of quantitative methods in certain domains, discussing properties/responses of complex systems, and many more interesting things.
4. Six Easy Pieces by Richard Feynman
   This book is a condensed version of his famous lectures in undergrad physics at Caltech. The chapters serve as the highlights of basic physics.
5. Princeton Companion to Mathematics (PCM)
   The second best thing to talking to a knowledgeable professor. Great for

understanding the what and why of math, to supplement your study of the how (ie using methods to solve actual problems).

6. Princeton Companion to Applied Mathematics (PCAM)
The second best thing to talking to a knowledgeable professor with skin in the game. Great for understanding the what and why of math, to supplement your study of the how (ie using methods to solve actual problems). Has additional articles that are particularly meaningful for people applying math to "real world problems".

2.2 Textbooks

1. **Geometry and Algebra** I would personally recommend using the OpenStax textbooks that are recommended for high school use. Combining that with Khan Academy seems like a reasonable strategy for a student not trying to spend too much for tutoring / materials. Ask a teacher to help you to navigate to a chapter in one of the textbooks that goes over what you are currently going over in class (or to navigate to the right spot on Khan Academy).

2. **Calculus**
Calculus by Gilbert Strang
Calculus by Michael Spivak
Calculus of Manifolds by Michael Spivak
Differential Equations with Applications and Historical Notes by George Simmons
Vector Calculus by Marsden and Tromba

3. **Linear Algebra**
Linear Algebra and Its Applications by Gilbert Strang
Introduction to Applied Linear Algebra by Stephen Boyd

4. **Probability and Statistics**
Introduction to Probability by Bertsekas and Tsitsiklis
Probability and Statistics for Enigineers by Sheldon Ross

5. **Proofs and Discrete Math**
Introduction to the Theory of Computation by Michael Sipser
CS103 Course Notes by Keith Schwarz
Proofwriting guides by Keith Schwarz

6. **Fourier/Laplace Transform**
Lectures on the Fourier Transform and Its Applications by Brad Osgood
Signals and Systems by Oppenheim and Willsky

Discrete-Time Signal Processing by Oppenheim and Schafer

2.3 Lectures

2.4 Websites / blogs

1. Paul's Online Math Notes
2. Hyperphysics by Georgia Tech
3. Terrance Tao blog
4. Timothy Gowers blog
5. Nicholas Higham blog

2.5 Videos

1. 3blue1brown on Youtube
2. Mathologer on Youtube
3. Dr Trefor Bazett on Youtube
4. Reducible on Youtube
5. Mathemaniac on Youtube
6. Aleph 0 on Youtube
7. Steve Brunton on Youtube
8. Brian Douglas (signals processing) on Youtube
9. Wireless Philosophy on Youtube
10. Physics with Elliot on Youtube

2.6 Tools

1. Wolfram Alpha
2. Python
3. Latex
4. Desmos
5. Geogebra
6. Jupyter Notebook
7. C++
8. C
9. SPICE Circuit Simulation
10. Matlab
11. Mathematica
12. Julia

3 Progression

1. Algebra and Geometry
   It should be the goal of your time in high school (and perhaps early parts of college) to be a master of Algebra and Geometry. Broadly speaking, this means solving for x with the elementary equations, being able to solve simple systems of equations, graphing elementary functions, and using algebraic and geometric reasoning to calculate/determine properties of functions and geometric figures.

2. Calculus
   It should be the goal of your time in college to become a master of single-variable calculus. The definition for what constitutes a master is different for mathematicians and physicists than for engineers. Essentially, if you study math or physics, a master would mean understanding the major concepts of single variable calculus, being able to solve non trivial derivatives, prove the major properties and theorems, and apply the knowledge in a meaningful way. A master for an engineer is someone who can pretty much do all of that except prove the major theorems. Even as an engineer, it only benefits you to understand the theoretical underpinnings of the math that we use.

3. Proofs and Discrete Math
   This is the one class that you do not need calculus for in order to understand deeply. Furthermore, knowledge of these subjects is required before one can be a true master of calculus. It is my opinion that every student in STEM should take at least one proof based course where proof writing is a major emphasis of the course. It is great to take a class like this as early as possible, as there aren't really any serious prereqs and helps you out so much down the road.

4. Programming
   First and foremost, I think that all people in STEM should have a basic understanding of programming. To me, this means understanding basic looping constructs, basic conditional logic, how to use libraries, how to use imperative and object-oriented paradigms, and basic event handling. I also think that people should not get to into systems programming before taking algorithms and data structures. Furthermore, one should be proficient in calculus and proof writing before taking algorithms and data structures in order to understand the material deeply.

5. Linear Algebra
   Linear algebra is a class that can be taken before mastering calculus, but mastering calculus enables you to use linear algebra in its most powerful forms. If you are planning to use linear algebra for certain applications, you may need to know calculus in order to understand the material deeply. For example, linear dynamical systems and machine learning require deep knowledge of calculus to truly understand at a theoretical level. Another helpful class to take

6. Probability and Statistics
   In an ideal world, probability and statistics is taken after mastering single variable calculus. You should be able to integrate and differentiate elementary functions and understand improper integrals. A linear algebra background before the class also helps with applying the material in projects.

7. Fourier Transforms
   Try to have proficiency in single variable calculus and linear algebra before taking fourier transforms. This is a class that can be saved for the latter half of college, since it has lots of preqeqs in order to understand in a deep level.

8. Circuits
   There should be no particular rush to take this class, and I would personally advise saving this until sophomore or junior year and make sure to be great at calculus, differential equations, and linear algebra to get the most out of the subject. Knowledge of Fourier Transforms is needed for advanced circuit analysis.

9. Machine Learning
   I would recommend being proficient in linear algebra, probability, statistics, and calculus(including multi-variable calculus) before taking this class. Do not rush to take this class unless it deeply fascinates you. Having strong fundamentals in the math prereqs will make a treatment of the material deeper and more enjoyable.