A Brief Introduction to *Mathematica*

I. B. Aron

Temple Math Club, September 18

Table of Contents

Table of Contents

Why use *Mathematica*?

Mathematica is a great tool to use that does not require too much coding knowledge.

Why use *Mathematica*?

Mathematica is a great tool to use that does not require too much coding knowledge.

- Checking work.

Why use *Mathematica*?

Mathematica is a great tool to use that does not require too much coding knowledge.

- Checking work.
- Solving difficult computational problems (tough integrations, checking identities, solving equations/ systems of equations, etc.).

Why use *Mathematica*?

Mathematica is a great tool to use that does not require too much coding knowledge.

- Checking work.
- Solving difficult computational problems (tough integrations, checking identities, solving equations/ systems of equations, etc.).
- Plotting and tool to visualize mathematics.

Why use *Mathematica*?

Mathematica is a great tool to use that does not require too much coding knowledge.

- Checking work.
- Solving difficult computational problems (tough integrations, checking identities, solving equations/ systems of equations, etc.).
- Plotting and tool to visualize mathematics.
- Lots of online resources and great internal documentation with examples.

Why use *Mathematica*?

Mathematica is a great tool to use that does not require too much coding knowledge.

- Checking work.
- Solving difficult computational problems (tough integrations, checking identities, solving equations/ systems of equations, etc.).
- Plotting and tool to visualize mathematics.
- Lots of online resources and great internal documentation with examples.
- Wide range of abilities!

Why use *Mathematica*?

Mathematica is a great tool to use that does not require too much coding knowledge.

- Checking work.
- Solving difficult computational problems (tough integrations, checking identities, solving equations/ systems of equations, etc.).
- Plotting and tool to visualize mathematics.
- Lots of online resources and great internal documentation with examples.
- Wide range of abilities!
- Take it anywhere!

Fun Fact: Dr. Igor Rivin helped create *Mathematica* before he started teaching at Temple.

Table of Contents

Downloading/Online Version

All relevant information can be found at
https://its.temple.edu/mathematica-site-licensed-software. (Link in chat!)

Downloading/Online Version

All relevant information can be found at
https://its.temple.edu/mathematica-site-licensed-software. (Link in chat!)

- Create a Wolfram account; go to user.wolfram.com

Downloading/Online Version

All relevant information can be found at
https://its.temple.edu/mathematica-site-licensed-software. (Link in chat!)

- Create a Wolfram account; go to user.wolfram.com
- Use your Temple email and then create Wolfram ID.
- They should send you an email to confirm registration.

All relevant information can be found at
https://its.temple.edu/mathematica-site-licensed-software. (Link in chat!)

- Create a Wolfram account; go to user.wolfram.com
- Use your Temple email and then create Wolfram ID.
- They should send you an email to confirm registration.

Two Options:

1. Download *Mathematica* for your computer.

Downloading/Online Version

All relevant information can be found at
https://its.temple.edu/mathematica-site-licensed-software. (Link in chat!)

- Create a Wolfram account; go to user.wolfram.com
- Use your Temple email and then create Wolfram ID.
- They should send you an email to confirm registration.

Two Options:

1. Download *Mathematica* for your computer.
2. Use the online version.

Table of Contents

The Basics: The Very Basics

- Each *Mathematica* document is called a *notebook*.

The Basics: The Very Basics

- Each *Mathematica* document is called a *notebook*.
- The *kernel* is what runs in the background and is the engine that performs the commands (the compiler). This is important to know about since restarts of the kernel are necessary to "renew" variables and debug.

The Basics: The Very Basics

- Each *Mathematica* document is called a *notebook*.
- The *kernel* is what runs in the background and is the engine that performs the commands (the compiler). This is important to know about since restarts of the kernel are necessary to "renew" variables and debug.
- Built in functions always start with a capital letter and take arguments in square brackets. Example: Solve$[x + 1 == 0, x]$.

The Basics: The Very Basics

- Each *Mathematica* document is called a *notebook*.
- The *kernel* is what runs in the background and is the engine that performs the commands (the compiler). This is important to know about since restarts of the kernel are necessary to "renew" variables and debug.
- Built in functions always start with a capital letter and take arguments in square brackets. Example: Solve$[x + 1 == 0, x]$.
- Use $\{\dots\}$ for vectors, lists and matrices.

The Basics: The Very Basics

- Each *Mathematica* document is called a *notebook*.
- The *kernel* is what runs in the background and is the engine that performs the commands (the compiler). This is important to know about since restarts of the kernel are necessary to "renew" variables and debug.
- Built in functions always start with a capital letter and take arguments in square brackets. Example: Solve$[x + 1 == 0, x]$.
- Use $\{\dots\}$ for vectors, lists and matrices.
- Run commands using Shift+Enter!

The Basics: Commands and Functions

- Any "typical" calculations can easily be done.

The Basics: Commands and Functions

- Any "typical" calculations can easily be done.
- Some basic built in functions: Solve, Plot, Integrate, D.

The Basics: Commands and Functions

- Any "typical" calculations can easily be done.
- Some basic built in functions: Solve, Plot, Integrate, D.
- Learn about an unfamiliar command or function using "?". Example: ?DSolve.

The Basics: Commands and Functions

- Any "typical" calculations can easily be done.
- Some basic built in functions: Solve, Plot, Integrate, D.
- Learn about an unfamiliar command or function using "?". Example: ?DSolve.
- Use "//" to quickly execute a command or create "descriptor" of the command in question.

The Basics: Commands and Functions

- Any "typical" calculations can easily be done.
- Some basic built in functions: Solve, Plot, Integrate, D.
- Learn about an unfamiliar command or function using "?". Example: ?DSolve.
- Use "//" to quickly execute a command or create "descriptor" of the command in question.
- Examples

The Basics: Commands and Functions

- Any "typical" calculations can easily be done.
- Some basic built in functions: Solve, Plot, Integrate, D.
- Learn about an unfamiliar command or function using "?". Example: ?DSolve.
- Use "//" to quickly execute a command or create "descriptor" of the command in question.
- Examples
- Questions?

Table of Contents

Weekly Challenge Problems

- We will be posting two weekly challenge problems each week.

Weekly Challenge Problems

- We will be posting two weekly challenge problems each week.
- Following each meeting, after talk/club activity has ended, we will be giving solutions/discussing the problems from the previous week.

Weekly Challenge Problems

- We will be posting two weekly challenge problems each week.
- Following each meeting, after talk/club activity has ended, we will be giving solutions/discussing the problems from the previous week.
- Obviously, not required, just a little extra math in case you are itching for some problems!

Weekly Challenge Problems

- We will be posting two weekly challenge problems each week.
- Following each meeting, after talk/club activity has ended, we will be giving solutions/discussing the problems from the previous week.
- Obviously, not required, just a little extra math in case you are itching for some problems!
- Problems taken mostly from Project Euler (projecteuler.net), a great website for getting interesting problems! (Thanks Greg)

Weekly Challenge Problems

- We will be posting two weekly challenge problems each week.
- Following each meeting, after talk/club activity has ended, we will be giving solutions/discussing the problems from the previous week.
- Obviously, not required, just a little extra math in case you are itching for some problems!
- Problems taken mostly from Project Euler (projecteuler.net), a great website for getting interesting problems! (Thanks Greg)
- One problem will be computational and the other will be done by hand. Some of the computational problems can be done by hand but the point is to learn how to implement your ideas using code.

Weekly Challenge Problems

- We will be posting two weekly challenge problems each week.
- Following each meeting, after talk/club activity has ended, we will be giving solutions/discussing the problems from the previous week.
- Obviously, not required, just a little extra math in case you are itching for some problems!
- Problems taken mostly from Project Euler (projecteuler.net), a great website for getting interesting problems! (Thanks Greg)
- One problem will be computational and the other will be done by hand. Some of the computational problems can be done by hand but the point is to learn how to implement your ideas using code.
- I will send out Weekly Challenge Problems via email!

Next Week and Beyond

Next week, Deloitte Consulting is coming to talk about opportunities in consulting for STEM majors!

Next Week and Beyond

Next week, Deloitte Consulting is coming to talk about opportunities in consulting for STEM majors!

Beyond:

- Dr. Brian Rider, 10/9
- Dr. Vladimir Visnjic, 10/16
- Max Avener, 10/23
- Dr. Maria Lorenz, 10/30
- Dr. Charles Osborne, 11/6
- Mei Rose Connor, 11/20
- Dr. Jane Friedman, 12/4

If you would like to give a talk/presentation or would like to see something specific in math club, please email me! (tuh30929@temple.edu)