Basics of *Mathematica*

Easy Problems

Notebook environment

1. First write three separate cells (can contain anything, e.g. 1+1). After that merge them and evaluate the resulting cell.

2. Evaluate the following cell and abort its evaluation (Alt+. or Evaluation-menu).

```
Monitor[For[i = 1, True, i = i + 1, Pause[1]], i]
```

3. It's always a good practice to organize your notebooks well. Making sections and chapters helps with this. Create a section called 'A' with a subsection called 'B' and some regular cells. Collapse cell 'A' to hide its children.

Practicing with input

4. Practice inputting expressions in different forms, e.g. type in the following expressions (preferably without using the Basic Math Assistant palette):

$$a^2 + b^2$$

$$\sqrt{a + i\, b}$$

$$\sum_{n=1}^{10} \left(n^2 + 1\right)$$

$$\int_{-\infty}^{+\infty} e^{-\lambda x^2}\, dx$$

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\partial_x \psi[x]$$

5. At the same time understand what other ways you can input these expressions: input each one of them without using 2D formatting (i.e. just type them in as text that you could potentially copy to notepad) and

in FullForm (i.e. without using any math symbols like + ^ ( ) { }, just heads with arguments, e.g. Power[x, 2], etc.).

Input form:

FullForm:

Working with Lists

6. Define a list ‘x’ with elements 1, 13, 6.1, 10 and 520.

7. Extract the third element of ‘x’.

8. Extract the last element of ‘x’ (hint: indexing a list with negative numbers counts from the end of the list).

9. Extract the elements 2 through 4 of ‘x’.

10. Define a list ‘y’ with elements Red, Green and Blue. Join the list ‘x’ and ‘y’ together to form a new list ‘z’.

11. Clear the variables ‘x’, ‘y’ and ‘z’.

12. Generate a list of 100 random integers between 0-10 and count how many times the number 5 appears in the list (hint: use RandomInteger and Count).

13. Find all subsets of the set {a,b,c}.

14. Generate a list of random integers, find the smallest and largest element. In the end, sort the list in ascending order.

15. Generate a 4×4 identity matrix and flatten it to a one-dimensional list

16. Generate a list of the squares of numbers from 1 to 100.

17. Replace the head of the List with Plus in order to find the sum of squares.

18. Construct a 4×4 Vandermonde matrix and find its determinant.

$$V = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \dots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \dots & \alpha_2^{n-1} \\ 1 & \alpha_3 & \alpha_3^2 & \dots & \alpha_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_m & \alpha_m^2 & \dots & \alpha_m^{n-1} \end{bmatrix}$$

19. Given a list of data points $\{\{x_1, y_1\}, \{x_2, y_2\}, \{x_3, y_3\}, \{x_4, y_4\}, \{x_5, y_5\}\}$, separate the x and y components to get

$\{\{x_1, x_2, x_3, x_4, x_5\}, \{y_1, y_2, y_3, y_4, y_5\}\}$.

Working with expressions (everything is an expression!)

20. Find the FullForm of x+y+z. Find also its TreeForm.

21. Find the FullForm of x^y. Find also the head of x^y. What is the zeroth part of x^y (that is, ⟦0⟧).

22. Write the expression $(1 + a + x)^{10}$ only using Plus and Power (don't use + and ^).

23. Change the head of $(1 + a + x)^{10}$ to Times (hint: use Apply or @@).

24. All expressions can be manipulated like they were lists. Define $t = 1 + x + x^2 + y^2$. Find the length of 't'. Take the first three elements of 't'.

25. Define text=Style[Text["Lorem ipsum"],Magenta]. Now evaluate text/.Magenta→Cyan. What happened? Use FullForm to explain why.

26. Plot the function Sin[x] and find its Head.

27. Expand $(1 + a + x)^{10}$, collect in terms of x and simplify each coefficient.

28. Generate the sum $\sum_{i=1}^{5} \frac{a_i}{b_i}$ and make it have a common denominator.

29. Given an expression $\text{MyLog}\left[\prod_{n=1}^{5} f[n]\right]$, convert it into a sum of MyLogs, i.e. $\sum_{n=1}^{5} \text{MyLog}[f[n]]$.

Functions and patterns

30. Make the following definitions

```
f[n_] = Sum[(1 + x)^i, {i, 1, n}];
g[n_] := Sum[(1 + x)^i, {i, 1, n}]
```

and explain why their outputs look different (e.g. f[3] and g[3]). Prove that they are actually equivalent.

32. If one wanted to create a function that yields n random numbers, one might naively define

```
randNums[n_] = Table[RandomReal[], {n}]
```

What is the problem with the above definition?

34. Use a replacement to transform $\{x^2, x^3, x^4, x^5\}$ to {2,3,4,5}.

35. Use now the same replacement on a list $\{x^1, x^2, x^3, x^4, x^5\}$. If your replacement failed to yield {1,2,3,4,5} explain why.

36. Generate a list of integers and use a replacement to transform all odd integers to their inverses.

37. Use a replacement to convert $\frac{x}{y}$ to x^y. Your replacement should also work for $\frac{a+c}{b+d}$.

38. Define a function that computes the mean of a list, i.e. MyMean[{1,2,3}] = 2.

39. Define a function that squares odd numbers and cubes even numbers.

40. Define a "function" that squares its input only when the current time in seconds is even (hint: Use AbsoluteTime, Round and EvenQ).

41. Define a function that returns Green if the input is 'a' or 'h' and Purple for other inputs. Use alternatives (?Alternatives).

42. Define a function that takes a list as input and returns a list with non-prime numbers removed.

43. Define a function that converts mathematical expressions comprised of sums, products and powers to a sum. That is, a*b*c→a+b+c and a*b^c+d→a+b+c+d etc.

44. Implement a function FirstAndLast with a pattern match, which does the following: FirstAndLast[a,b,c,d, ..., x,y,z] = {a, z}

45. Expand the expression $(1 + a + x)^{10}$ and produce a list of terms with odd powers of x.

46. Implement the factorial function recursively. Make sure it can be called only with positive integers and that it performs well.

47. Implement a function 'log' that knows the usual properties of the logarithm, that is, log[x y]=log[x]+log[y], log[x^y] = y log[x].

48. Implement the simplest case of the system function Riffle (check the documentation for what it does).

49. Create a predicate ‘NaturalQ’, that returns true if its argument is a natural number. Now create a list of random numbers between -10 and 10 and use ‘NaturalQ’ to filter out all non-natural numbers (hint: ?Select).

Plotting

50. Plot Sin[x], Sin[2x] and Sin[3x] together in a same graph.

51. Make the previous plot bigger (use arguments to Plot) and add Legends to distinguish which curve corresponds to which function.

52. Plot $e^{-(x-\mu)^2/2\sigma^2} \Big/ \sqrt{2\pi}\,\sigma$ with some parameter values, add labels for axes and a plot title containing the (μ,σ) you used.

53. Make four plots with different values for μ and σ. Arrange them in a grid (GraphicsGrid) and export (Export) the resulting figure as a PDF-file.

54. Plot a 3D parabola.

55. Plot the real and imaginary parts of the function $\sqrt{x^2 - 1}$ on the complex plane.

56. Plot Cos[x]Sin[y] where both x and y run from 0 to 4π. Make a region plot showing the (x,y)-regions where Cos[x]Sin[y]>0.

57. A point particle moves in (1+2)d spacetime along a trajectory t→ (x(t),y(t),t)=(Exp[-t/2]Sin[5t],Cos[3t],t). Make a 3D-plot representing this trajectory (hint: use ParametricPlot3D).

58. Previously you create your own implementation of the factorial function. Use AbsoluteTiming to measure the performance of your implementation with respect to the argument (that is, n for n!). Make a ListPlot of execution time versus n. Compare with the built in factorial function.

59. Generate 25 random integers between 1-10. Visualize how many times a certain number appeared by drawing a pie chart (PieChart). Rerun your code now with 1000 random numbers.

60. Consider a random walk in one dimension. The walk starts at the origin (x=0) and the walker steps to the left (-1) or right (+1) with equal probabilities. Generate a list of steps (that is, list of -1:s and 1:s). Compute the position of the walker at all time steps and plot the position of the walker w.r.t. time.

Medium and harder problems

61. Generate a random walk in 2 dimensions and visualize the path (hint: study what Graphics[Line[{{0,0},{1,0},{1,1}}]] does).

62. Create a code that plots the random walk only up to some intermediate time between 0 and the end. Use Manipulate to interactively change this intermediate time.

63. Generate a 2×2 matrix whose entries are Random numbers between 0 and 1. Compute the average of the squares of its two Eigenvalues $\Lambda = \left(\lambda_1^2 + \lambda_2^2\right)/2$.

64. Repeat the computation a few thousand times and Plot a Histogram of the Λs obtained. Compute the average of Λ.

65. Generalize the previous computations to matrices of arbitrary size M. Plot the dependence of $\langle\Lambda\rangle$ on M. Do a polynomial fit on the dependence.

66. Represent states of the oscillator by Ket[*n*], where *n* is the mode number. Implement a function that would act on a given state with a given operator. States can be linear combinations of Kets and operators can also be linear combinations of creation and annihilation operators.