Phys209: Mathematical Methods in Physics I

Homework 10

Soner Albayrak[†]

[†]*Middle East Technical University, Ankara 06800, Turkey*

Policies

- Please adhere to the *academic integrity* rules: see my explanations here for further details!
- For the overall grading scheme or any other course-related details, see the syllabus.
- Non-graded question(s) are for your own practice!
- Unless stated otherwise, you are expected to show your derivation of the results.
- The homework is due December 15th 2023, 23:59 TSI.

(1) Complex conjugation *(3 points)*

We have discussed this week in class about the concept of unitarity and how it is very important in expanding a given function on a basis of other functions, an important point of a branch of mathematics called Harmonic Analysis. We'll review some basics of these concepts in this homework.

(1.1) (1pt)

We introduce the complex conjugation as

$$\begin{aligned} \epsilon &:: \mathbb{C} \to \mathbb{C} \\ \epsilon &= z \to z^* = \mathrm{Re}(z) - i\ \mathrm{Im}(z) \end{aligned}$$

Please compute the values of $f(z^*)$ for $z = e^{i\pi\theta}$ at $\theta = 0$, $\frac{1}{4}$, $\frac{1}{2}$, and $\frac{3}{4}$ if

$$\begin{aligned} f &:: \mathbb{C} \to \mathbb{C} \\ f &= z \to \sin(z) + \cos(z) \end{aligned}$$

(1.2) (1pt)

For the function defined in the previous subsection, prove that $(f(z))^* = f(z^*)$.
Hint: Expand sin and cos in terms of exponentials using the relation $e^{ix} = \cos(x) + i\sin(x)$.

(1.3) (1pt)

Consider the functions g and h defined as

$$\begin{aligned} g &:: \mathbb{C} \to \mathbb{C} \\ g &= z \to \cos(iz) \end{aligned}$$

and

$$h ::\mathbb{C} \to \mathbb{C}$$
$$h = z \to \sin(iz)$$

Show that $(g(z))^* = g(z^*)$ but $(h(z))^* \neq h(z^*)$.

(1.4) Bonus question *(not graded)*

Read about Schwarz reflection principle. In summary,

$$f(x) \in \mathbb{R}\ \forall x \in \mathbb{R}$$

is a necessary condition for

$$(f(z))^* = f(z^*)$$

(2) Problem Two *(3 points)*

In class, we defined the transpose function with (co)domain of matrices as

$$T :: \mathfrak{M}_{n\times m}(\mathbb{C}) \to \mathfrak{M}_{m\times n}(\mathbb{C})$$
$$T = \left[x = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & & & \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} \right] \to \left[x^T = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \dots & & & \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{pmatrix} \right]$$

and introduced *hermitian conjugation* denoted by † as a function

$$\dagger :: \mathfrak{M}_{n\times m}(\mathbb{C}) \to \mathfrak{M}_{m\times n}(\mathbb{C})$$
$$\dagger = x \to (x^T)^*$$

(2.1) Hermitian conjugation (3pt)

Hermitian conjugation is generalization of complex conjugation to matrices. For instance, we call a number z real if $z^* = z$; likewise, we call a matrix H Hermitian if it satisfies $H^\dagger = H$. For example, most general Hermitian 2×2 matrix can be written as

$$A = \begin{pmatrix} a & b + ic \\ b - ic & d \end{pmatrix} \qquad \text{for} \quad a, b, c, d \in \mathbb{R}$$

Derive the most general 3×3 Hermitian matrix!
Hint: Start with the most general possible form for a 3×3 matrix and solve the constraints: you will solve 9 equations for 18 unknowns, leaving you with a matrix with 9 real paramenters. More generally, an $n \times n$ matrix M depends on n^2 complex numbers, hence it has $2n^2$ real degrees of freedom. On the contrary, a hermitian matrix also needs to satisfy n^2 constraints, so it will depend only on n^2 real degrees of freedom.

(2.2) Bonus question *(not graded)*

The question above can be solved with the following Mathematica code:

```
With[{
        matrixWithComplexEntries = Array[a, {3, 3}],
        translateToRealEntries = a[i\_, j\_] :> b[i, j] + I c[i, j]
},
With[{
        hermiticityCondition =
        ConjugateTranspose[matrixWithComplexEntries] —
        matrixWithComplexEntries
},
With[{
        conditionsOnRealEntries =
        Solve[Flatten[
        hermiticityCondition /. translateToRealEntries /.
        Conjugate —> Identity] == 0][[1]]
},
matrixWithComplexEntries /. translateToRealEntries /.
conditionsOnRealEntries /. {b[i_, j_] :> Subscript[b, i, j],
        c[i_, j_] :> Subscript[c, i, j]} // TraditionalForm
]]]
```