Assignment 1 - Survival Mathematics

This assignment collects the basic mathematical concepts to help you through the entire course, as well as a few that might be new to you. You should use any resources (e.g. books, search engines, calculators, etc.) that can help you accomplish it.

Task 1: Point and Vector

A *vector* encodes information about **direction** and **magnitude**. Intuitively, a vector is an arrow rather than a specific *point* that only encodes a specific location. Given are the following statements. Determine whether the numbers in the statement can be represented as a point, or as a vector.

a) The lecture was held at 10 a.m. yesterday.

b) The exam lasts 90 minutes.

c) The metro station is 100 meters away to the south of the office.

d) The highest standing jump is 1.651 meters according to Guinness World Records.

Include your answers in a Markdown file called "task01.md".

Task 2: Space and Coordinates

A point in space is represented by coordinates. Depending on different types of reference frames, coordinates can occur in different forms.

a) Name the corresponding reference frame in *OpenGL* and *Direct3D*, and explain the difference between them.

b) Let the x-axis of a Cartesian coordinate system point to the east, the y-axis to the south, and the z-axis to the top. Assume the point $P = (3, 4, 5)$ is given the reference frame. What are the corresponding coordinates of point P in *OpenGL* if each axis lies in the same line with respect to the corresponding axis?

c) Assume a 3-dimensional spherical coordinate system and Cartesian coordinate system comply with the following rules:

   - $\mathbf{v}$ is the location vector of a point $P = (r, \theta, \phi)$
   - r is the radius, i.e. the distance of the point P from the origin, $r \geq 0$
   - θ is the angle between the positive z-axis and $\mathbf{v}$, where $\theta \in [0, \pi]$
   - ϕ is the angle between the positive x-axis and the projection of $\mathbf{v}$ into the x-y plane, where $\phi \in [0, 2\pi]$

   Calculate the cartesian coordinates of point P.

Include your answers in a Markdown file called "task02.md".

Task 3: Scalars, Vectors and Products

In a 3-dimensional *Euclidean space* $\mathbb{R}^3$, a vector contains a list of numbers that are written in a vertical form, i.e. a *column vector*:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, \mathbf{v} \in \mathbb{R}^3$$

To save space, it is common to write a column vector as a *transpose* of its horizontal form (i.e. row vector), e.g. $\mathbf{v} = (v_1, v_2, v_3)^\top$. To decode the *norm* (i.e. magnitude) $\|\mathbf{v}\|$ and the *angle* (i.e. direction) $\angle(\mathbf{v}, \mathbf{v}')$ with respect to $\mathbf{v}' = (v_1', v_2', v_3')^\top$, one can use the following formulas:

$$\|\mathbf{v}\| = \sqrt{\mathbf{v}^\top \cdot \mathbf{v}} = \sqrt{v_1^2 + v_2^2 + v_3^2}, \angle(\mathbf{v}, \mathbf{v}') = \arccos\left(\frac{\mathbf{v}^\top \cdot \mathbf{v}'}{\|\mathbf{v}\|\|\mathbf{v}'\|}\right) = \arccos\left(\frac{v_1 v_1' + v_2 v_2' + v_3 v_3'}{\|\mathbf{v}\|\|\mathbf{v}'\|}\right)$$

where $\cdot$ is the so-called *dot product* that results in a scalar.

Given the vectors: $\mathbf{v}_1 = (2, 1, 2)^\top, \mathbf{v}_2 = (1, 1, 3)^\top, \mathbf{v}_3 = (1, 2, -2)^\top$ and scalars $a = 1, b = 2, c = -3$. Calculate:

a) $a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3$

b) $\|\mathbf{v}_1\|, \|\mathbf{v}_2\|, \|\mathbf{v}_3\|$

c) $\angle(\mathbf{v}_1, \mathbf{v}_2), \angle(\mathbf{v}_2, \mathbf{v}_3), \angle(\mathbf{v}_1, \mathbf{v}_3)$

The *cross product* $\mathbf{v} \times \mathbf{v}'$ results in a vector that is orthogonal to both given vectors, which can be calculated by:

$$\mathbf{v} \times \mathbf{v}' = (v_2 v_3' - v_3 v_2', v_3 v_1' - v_1 v_3', v_1 v_2' - v_2 v_1')^\top$$

Calculate:

d) $\mathbf{v}_1 \times \mathbf{v}_2, \mathbf{v}_2 \times \mathbf{v}_1$

e) $\mathbf{v}_1 \times (\mathbf{v}_2 + \mathbf{v}_3), \mathbf{v}_1 \times \mathbf{v}_2 + \mathbf{v}_1 \times \mathbf{v}_3$

f) $\mathbf{v}_1 \times \mathbf{v}_1, \mathbf{v}_2 \times \mathbf{v}_2, \mathbf{v}_3 \times \mathbf{v}_3$

g) $\mathbf{v}_1^\top \cdot (\mathbf{v}_1 \times \mathbf{v}_2), \mathbf{v}_2^\top \cdot (\mathbf{v}_1 \times \mathbf{v}_2)$

h) $\mathbf{v}_1 \times (\mathbf{v}_2 \times \mathbf{v}_3) + \mathbf{v}_2 \times (\mathbf{v}_3 \times \mathbf{v}_1) + \mathbf{v}_3 \times (\mathbf{v}_1 \times \mathbf{v}_2)$

Include your answers in a Markdown file called "task03.md".

Task 4: Basis, Matrices and Determinants

A *span* is the set of all vectors that can be written as a *linear combination* of $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n$:

$$\text{span}(\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n) = \left\{ \mathbf{x} \middle| \mathbf{x} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i, \lambda_1, ...\lambda_n \in \mathbb{R} \right\}$$

In particular, if we have three vectors such that the $\text{span}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) = \mathbb{R}^3$, we call these vectors the *basis* of $\mathbb{R}^3$. Moreover, if our basis $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ satisfies i) unit length $\mathbf{e}_i \cdot \mathbf{e}_i = 1$ and ii) mutual orthogonality $\mathbf{e}_i \cdot \mathbf{e}_j = 0 (i \neq j)$, then we call it *orthonormal basis*.

We keep using $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$ from Task 3.

a) What is the span S with respect to $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$?

b) What is the span S' with respect to $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_1 + \mathbf{v}_2$?

c) What are the corresponding orthonormal basis to the span S and S'?

A matrix $\mathbf{M}_{m \times n}$ is a m rows by n columns arrangement of real numbers. The addition/subtraction of two matrices is trivially computed by element-wise summation/differentiation similar to vectors. However, the multiplication $\mathbf{C}_{m \times n}$ of two matrices $\mathbf{A}_{m \times p} \cdot \mathbf{B}_{p \times n}$ is defined by:

$$c_{i,j} = \sum_{k=1}^{p} a_{i,k} b_{k,j}, 1 \leq i \leq m, 1 \leq j \leq n$$

where $c_{i,j}$ is the element in i-th row, j-th column, and the meaning of $a_{i,k}, b_{k,j}$ is in the same way. In particular, a 3 dimensional vector is a 3×1 matrix, and its transpose is a 1×3 matrix.

d) Calculate $\mathbf{v}_1^\top \cdot \mathbf{v}_1$, and $\mathbf{v}_1 \cdot \mathbf{v}_1^\top$. Then tell the difference between the results.

e) Why it is **not** possible to compute $\mathbf{v}_1 \cdot \mathbf{v}_1$?

A determinant $\det(\mathbf{A})$ is a scalar value that can computed from elements in a $n \times n$matrix $\mathbf{A}_{n \times n}$. For the determinant of a 2×2 matrix $\mathbf{B}$ is computed by:

$$\det(\mathbf{B}) = \begin{vmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{vmatrix} = b_{11}b_{22} - b_{21}b_{12}$$

And the determinant of 3×3 matrix $\mathbf{C}$ is computed by:

$$\det(\mathbf{C}) = \begin{vmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{vmatrix} = c_{11} \begin{vmatrix} c_{22} & c_{23} \\ c_{32} & c_{33} \end{vmatrix} - c_{12} \begin{vmatrix} c_{21} & c_{23} \\ c_{31} & c_{33} \end{vmatrix} + c_{13} \begin{vmatrix} c_{21} & c_{22} \\ c_{31} & c_{32} \end{vmatrix}$$

f) Let matrix $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \\ 2 & 3 & -2 \end{pmatrix}$, calculate $\det(\mathbf{V})$.

g) Let matrix $\mathbf{V}' = (\mathbf{v}_1, \mathbf{v}_2, \lambda_1\mathbf{v}_1 + \lambda_2\mathbf{v}_2)$, calculate $\det(\mathbf{V}')$ where $\lambda_1, \lambda_2 \in \mathbb{R}$. Calculate $\det(\mathbf{V}')$.

h) What can be concluded from the results in f) and g)?

i) Calculate $\mathbf{v}_1^\top \cdot (\mathbf{v}_2 \times \mathbf{v}_3), \mathbf{v}_2^\top \cdot (\mathbf{v}_3 \times \mathbf{v}_1)$, and $\mathbf{v}_3^\top \cdot (\mathbf{v}_1 \times \mathbf{v}_2)$. What did you find in your results?

j) Carefully observe the computation rule of $\det(\mathbf{B})$ and $\det(\mathbf{C})$. Deduce the computation rule of the determinant for a 4×4 matrix (Hint: watch out for patterns of the multiplication factors and their plus-minus signs in the definition):

$$\begin{vmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{vmatrix}$$

Include your answers in a Markdown file called “task04.md”.

Task 5: Getting started with JavaScript

Before you walk into computer graphics, you also need to understand at least a programming language. This task helps you get familiar with the basics of JavaScript (JS) which we will use for the programming tasks in the subsequent tutorials.

Note that we use JS for building prototypes to get rid of most of the complexity from hardware/OS platforms, programming language details, dependency managements, etc.[1]. If you have no prior experience with JS, don’t be afraid, the whole language was created in 10 days. You should be able to learn it very quickly. Find a tutorial[2] and get familiar with these concepts: *constant, variable, object, function, class, string, array, condition/loop statement, error.*

In this coding task, you will implement the following vector and matrix operations that you have calculated in the previous tasks:

a) Vector sum: $\mathbf{v} + \mathbf{w}$

b) Vector scalar multiplication: $s\mathbf{v}$

c) Vector dot product: $\mathbf{v}^\top \cdot \mathbf{w}$

d) Vector norm: $\|\mathbf{v}\|$

e) Vector cross product: $\mathbf{v} \times \mathbf{w}$

f) Angle of two vectors: $\angle(\mathbf{v}, \mathbf{w})$

[1] If you found that computer graphics is attractive to you, it is highly recommended to use C++ as a serious programming language after you have learned the basics of computer graphics.

[2] For example: https://www.w3schools.com/js/default.asp

g) Matrix multiplication: $\mathbf{A}_{m\times p} \cdot \mathbf{B}_{p\times n}$

h) Determinants: $\det(\mathbf{A}_{2\times 2})$ and $\det(\mathbf{A}_{3\times 3})$

We provided a coding skeleton for you, checkout our GitHub repository (`https://github.com/mimuc/cg1-ss20`), you should see three files: `mat.js`, `mat_test.js`, and `package.json`. Look for the `// TODO:` in the `mat.js`. To verify your implementation, install NodeJS[3], then use **`npm run test`** in the `1-math` folder. You should see a `PASS` if you implemented them properly, otherwise you get a **`FAIL`** with a specific test sample.

```
$ npm run test

> cg1-1-math@ test ~/cg1/code/1-math
> node mat_test.js

PASS
```

```
$ npm run test

> cg1-1-math@ test ~/cg1/code/1-math
> node mat_test.js

{
  type: 'sum',
  test: {
    v1: Vector3 { x1: 0, x2: 0, x3: 0 },
    v2: Vector3 { x1: 1, x2: 2, x3: 3 },
    want: Vector3 { x1: 1, x2: 2, x3: 3 }
  },
  got: Vector3 { x1: 0, x2: 0, x3: 0 }
}
FAIL
```

Include your implementation in a folder called “task05”, submit it if and only if you get a `PASS`.

[3] `https://nodejs.org/en/download/package-manager/`

Submission

- Participation in the exercises and submission of the weekly exercise sheets is voluntary and not a prerequisite for participation in the exam. However, participation in an exercise is a good preparation for the exam (the content is the content of the lecture and the exercise).
- For non-coding tasks, write your answers in a Markdown file. Markdown is a simple mark-up language that can be learned within a minute. A recommended the Markdown GUI parser is typora (`https://typora.io/`), it supports parsing embedded formula in a Markdown file. You can find the syntax reference in its Help menu.
- **Please submit your solution as a ZIP file** via Uni2Work (`https://uni2work.ifi.lmu.de/`) before the deadline. We do not accept group submissions.
- Your solution will be corrected before the discussion. Comment your code properly, organize the code well, and make sure your submission is clear because this helps us to provide the best possible feedback.
- If we discover cheating behavior or any kind of fraud in solving the assignments, you will be withdrawn for the entire course! If that happens, you can only rejoin the course next year.
- If you have any questions, please discuss them with your fellow students first. If the problem cannot be resolved, please contact your tutorial tutor or discuss it in our Slack channel.