Analysis of Three-Dimensional Shapes
E. Rodolà, T. Windheuser, M. Vestner
Summer Semester 2015

Computer Vision Group
Institut für Informatik
Technische Universität München

Weekly Exercises 1

Room: 02.09.023

Wed, 22.04.2015, 14:00-16:00

Submission deadline: Tue, 21.04.2015, 23:59 to windheus@in.tum.de

Mathematics: Recap of Linear Algebra

Let us start with some definitions

Definition (Inner product)**.** Let X be a vector space. A mapping $\langle \cdot, \cdot \rangle : X \times X \to \mathbb{C}$ is called inner product, if

1. $\langle x_1 + \lambda x_2, y \rangle = \langle x_1, y \rangle + \lambda \langle x_2, y \rangle \quad \forall x_i, y \in X, \lambda \in \mathbb{C}$
2. $\langle x, y \rangle = \overline{\langle y, x \rangle} \quad \forall x, y \in X$
3. $\langle x, x \rangle \geq 0 \quad \forall x \in X$
4. $\langle x, x \rangle = 0 \Leftrightarrow x = 0$

Two elements $x, y \in X$ are called perpendicular, if $\langle x, y \rangle = 0$.

Definition (Linear operator)**.** Let X and Y be vector spaces. A mapping $T : X \to Y$ is called linear, if

$$T(x_1 + \lambda x_2) = T(x_1) + \lambda T(x_2)$$

A common notation is $Tx := T(x)$

Definition (Eigenvalues and eigenvectors)**.** Let $T : X \to X$ be a linear operator from a vector space X into itself (an endomorphism). An *eigenvector* is an element $0 \neq x \in X$ for which there exists a scalar $\lambda \in \mathbb{C}$, such that

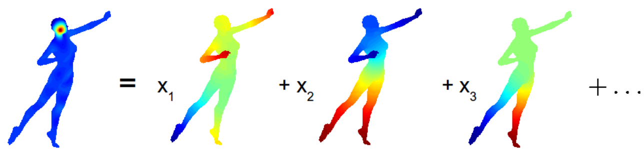
$$Tx = \lambda x$$

The scalar λ is called *eigenvalue*.

Exercise 1 (1 point). 1. Show that every matrix $\Phi \in \mathbb{C}^{n\times n}$ is representing an endomorphism on $\mathbb{C}^n$ via

$$(\Phi x)_i = \sum_{j=1}^{n} \phi_{ij} x_j \quad \forall i = 1 \dots n$$

If we denote the j-th coloumn of Φ by ϕ_j we can write $\Phi x = \sum_{j=1}^{n} \phi_j x_j$



2. Calculate the gradient $\nabla f(x) = (\partial_1 f(x), \dots, \partial_n f(x))^T$ for $f(x) = x^T A x$ ($A \in \mathbb{C}^{n\times n}$).

Solution. 1. The linearity is a consequence of the linearity of the scalar-vector multiplication:

$$\begin{aligned}
\Phi(x + \lambda y) &= \sum_{j=1}^{n} \phi_j \cdot (x_j + \lambda \cdot y_j) \\
&= \sum_{j=1}^{n} (\phi_j \cdot x_j + \phi_j \cdot \lambda \cdot y_j) \\
&= \sum_{j=1}^{n} (\phi_j \cdot x_j + \lambda \cdot \phi_j(y_j)) \\
&= \sum_{j=1}^{n} \phi_j \cdot x_j + \lambda \sum_{j=1}^{n} \phi_j \cdot y_j \\
&= \Phi x + \lambda \cdot \Phi y
\end{aligned}$$

2. First we write f as a quadratic polynomial in x:

$$\begin{aligned}
f(x) = \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix} &\begin{pmatrix} \sum_{j=1}^{n} A_{1j} x_j \\ \vdots \\ \sum_{j=1}^{n} A_{nj} x_j \end{pmatrix} \\
&= \sum_{i=1}^{n} x_i \sum_{j=1}^{n} A_{ij} x_j \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j
\end{aligned}$$

Let us highlight the dependency on x_k:

$$f(x) = A_{kk} x_k^2 + (\sum_{i\neq k} A_{ik} x_i) \cdot x_k + (\sum_{j \neq k} A_{kj} x_j) \cdot x_k + \sum_{i\neq k} \sum_{j\neq k} A_{ij} x_i x_j$$

Now we take the derivative with respect to x_k:

$$\partial_k f(x) = 2A_{kk}x_k + (\sum_{i\neq k} A_{ik}x_i) + (\sum_{j\neq k} A_{kj}x_j)$$
$$= \underbrace{\sum_{i=1}^{n} A_{ik}x_i}_{(A^Tx)_k} + \underbrace{\sum_{j=1}^{n} A_{kj}x_j}_{(Ax)_k}$$

This means the gradient is given by

$$\nabla f(x) = (A + A^T)x$$

Exercise 2 (1 point)**.** When not mentioned otherwise we consider the standard inner product on $\mathbb{C}^n$ given by

$$\langle x, y\rangle = x^T \cdot \bar{y} = \sum_{i=1}^{n} x_i\bar{y}_i$$

1. Show that this is indeed an inner product.
2. Given a matrix $A \in \mathbb{C}^{n\times n}$, find the matrix B such that
$$\forall x, y \in \mathbb{C}^n : \langle Ax, y\rangle = \langle x, By\rangle.$$
3. Show that if $A = B$ then all the eigenvalues are real.
4. Show that if $A = B$ and the two eigenvectors x^1 and x^2 are not orthogonal, it follows $\lambda_1 = \lambda_2$.

Solution. 1. We need to check four requirements

(a)

$$\langle x_1 + \lambda x_2, y\rangle = \sum_{i=1}^{n}(x_{1i} + \lambda x_{2i})\bar{y}_i$$
$$= \sum_{i=1}^{n} x_{1i}\bar{y}_i + \lambda \sum_{i=1}^{n} x_{2i}\bar{y}_i$$
$$= \langle x_1, y\rangle + \lambda\langle x_2, y\rangle$$

(b)

$$\langle x, y\rangle = \sum_{i=1}^{n} x_i\bar{y}_i = \sum_{i=1}^{n} \overline{\bar{x}_i y_i}$$
$$= \overline{\sum_{i=1}^{n} \bar{x}_i y_i} = \overline{\langle y, x\rangle}$$

(c) $\langle x, x\rangle = \sum_{i=1}^{n} x_i \bar{x}_i = \sum_{i=1}^{n} |x_i| \geq 0 \quad \forall x \in X$ since $|x_i| \geq 0$

(d) $\langle x, x\rangle = 0 \Leftrightarrow |x_i| = 0 \forall i \Leftrightarrow x = 0$

2.

$$\begin{aligned} \langle Ax, y\rangle = \langle x, By\rangle \quad \forall x, y \in \mathbb{C}^n \\ \Leftrightarrow (Ax)^T \bar{y} = x^T \overline{By} \quad \forall x, y \in \mathbb{C}^n \\ \Leftrightarrow x^T A^T \bar{y} = x^T \overline{B} \overline{y} \quad \forall x, y \in \mathbb{C}^n \\ \Leftrightarrow B = \overline{A^T} \end{aligned}$$

3. Let λ be an eigenvalue and x its corresponding eigenvector

$$\begin{aligned} \lambda\langle x, x\rangle &= \langle Ax, x\rangle \\ &= \langle x, Ax\rangle \\ &= \langle x, \lambda x\rangle \\ &= \bar{\lambda}\langle x, x\rangle \Rightarrow \lambda = \bar{\lambda} \end{aligned}$$

4. Let x^1, x^2 be two non-orthogonal eigenvectors with corresponding eigenvalues λ_1 and λ_2

$$\begin{aligned} \lambda_1\langle x^1, x^2\rangle &= \langle Ax^1, x^2\rangle \\ &= \langle x^1, Ax^2\rangle \\ &= \langle x^1, \lambda_2 x^2\rangle \\ &= \lambda_2\langle x^1, x^2\rangle \Rightarrow \lambda_1 = \lambda_2 \end{aligned}$$

since $\langle x^1, x^2\rangle \neq 0$.

Programming: Working with Matlab and triangle meshes

A *triangle mesh* $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ is a discrete surface embedded into $\mathbb{R}^3$. It consists of a *vertex* set $\mathcal{V} = \{v_1, \ldots, v_n\}$, and a set of *triangles* $\mathcal{F} \subset \mathcal{V} \times \mathcal{V} \times \mathcal{V}$ (also called *faces* in a more general setting). The coordinates of the vertices embedded in $\mathbb{R}^3$ are denoted by $\mathbf{x}(v_1), \ldots, \mathbf{x}(v_n) \in \mathbb{R}^3$. Note that the vertices of a triangle $t = (u, v, w) \in \mathcal{F}$ are ordered. We define triangles to be identical if they can be transformed into each other by a cyclic permutation, i.e. $(u, v, w) = (v, w, u) = (w, u, v)$ but $(u, v, w) \neq (w, v, u)$.

This programming exercises will introduce you to working with meshes in Matlab.

Exercise 3 (One point for 1./2. and one point for 3./4./5.)**.** Download and expand the file `exercise1.zip` from the lecture website. Modifiy the files `adjacency.m`, `incidence.m`, `facearea.m`, `meshvolume.m` and `gaussiancurvature.m` to implement the functions as explained below. You can run the script `exercise.m` to test and visualize your solutions.

1. Given a triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ with n vertices the *adjacency matrix* $A \in \mathbb{R}^{n \times n}$ is defined by

$$A_{i,j} = \begin{cases} 1 & \exists t \in \mathcal{F} : v_i, v_j \in t \\ 0 & \text{otherwise.} \end{cases}$$

   Implement function *adjacency* that returns the adjacency matrix in sparse format for a given triangle mesh.

2. A mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ with n vertices induces a set of halfedges $\mathcal{H} = \{(u, v) \in \mathcal{V} \times \mathcal{V} : \exists w \in \mathcal{V} : (u, v, w) \in \mathcal{F}\}$. We assume some ordering on the halfedges, i.e. $\mathcal{H} = \{h_1, \ldots, h_m\}$. The *vertex-to-halfedge incidence matrix* $I \in \mathbb{R}^{m \times n}$ is defined by

$$I_{i,j} = \begin{cases} -1 & h_i = (v_j, \cdot) \\ 1 & h_i = (\cdot, v_j) \\ 0 & \text{otherwise.} \end{cases}$$

   Implement function *incidence* that returns the vertex-to-halfedge incidence matrix in sparse format for a given triangle mesh.

3. Implement function *facearea* that returns the area of each triangle as an $\mathbb{R}^{|\mathcal{F}|}$ vector for a given triangle mesh.

4. Implement function *meshvolume* that returns the volume of a given triangle mesh. Hint: Think about the volume of a tetrahedron constructed from the vertices of a mesh triangle and the origin.

5. The *Gaussian curvature* at vertex v is given by $\kappa_v = 2\pi - \sum_{t \in \mathcal{F}: v \in t} \theta_{t,v}$, where $\theta_{t,v}$ is the angle of triangle t at vertex v. Implement function *gaussiancurvature* that returns the Gaussian curvature of each vertex as an $\mathbb{R}^{|\mathcal{V}|}$ vector for a given triangle mesh.