0 WB2630-T1 & T2 Math review

0.1 Vectors and Matrices

Vectors are lists of numbers which can graphically be represented as arrows in space. They can be denoted as either row or column vectors:

$$\text{Column: } \vec{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$

$$\text{Row: } \vec{r} = \begin{pmatrix} r_1 & \cdots & r_n \end{pmatrix}^T$$

Matrices are just multi-dimensional arrays. Or just think of them as a vector of vectors. They can graphivally be represented as some linear transformation acting on a vector. Matrices will usually be denoted as:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

0.2 Properties of Vectors

Vectors can be added and subtracted together if they are expressed in terms of the same basis vectors. Vectors can also be scaled by some scalar value $\alpha \in \mathbb{R}^n$. Multiplication of vectors takes 2 forms. The dot or scalar product and the cross or vector product.

The Dot product of 2 vectors outputs a scalar value and represents roughly how much 2 values are in the same direction. The dot product is given as:

$$c = \vec{a} \cdot \vec{b} \equiv |\vec{a}||\vec{b}| \cos(\theta) \quad c \in \mathbb{R},\ \vec{a}, \vec{b} \in \mathbb{R}^n$$

The dot productive is communicative:

$$\vec{a} \cdot \vec{b} = \vec{b} \cdot \vec{a}$$

There are several other ways to represent the dot product. Some examples are:

$$\vec{a} \cdot \vec{b} = \vec{a}^T \vec{b} = \sum_{i=1}^{n} a_i b_i$$

The cross product outputs a vector that is perpendicular to the plane spanned by the other 2 vectors. The magnitude of this 3rd vector represents the area of the plane spanned by the first 2 vectors. The cross product is given as:

$$\vec{c} = \vec{a} \times \vec{b}$$

$$|\vec{c}| = |\vec{a}||\vec{b}| \sin(\theta) \quad \theta \in [0, \pi]$$

Let $\vec{a}, \vec{b}$ and $\vec{c}$ be:

$$\vec{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \quad \vec{b} = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} \quad \vec{c} = \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix}$$

if $\vec{c} = \vec{a} \times \vec{b}$, then:

$$\begin{aligned} c_x &= a_y b_z - b_y a_z \\ c_y &= b_x a_z - a_x b_z \\ c_z &= a_x b_y - b_x a_y \end{aligned}$$

The cross product is anti-communicative:

$$\vec{a} \times \vec{b} = -(\vec{b} \times \vec{a})$$

0.3 Properties of Matrices

Some matrix A can have any $m \times n$ shape. Matrices can be multiplied with eachother to create new matrices. They can also be multiplied with vectors to map them to different vectors. Let A be an $m \times n$ matrix:

$$A\vec{x} = \vec{y} \quad \vec{x} \in \mathbb{R}^n, \ \vec{y} \in \mathbb{R}^m$$

Matrices can be transposed. The transpose operation interchanges the rows and columns of a matrix and is dentoted with T. If a matrix is symmetric then:

$$A^T = A$$

Square matrices with a non-zero determinant can be inverted. Multiplying by the inverse of a matrix is analogous to multiplying by the inverse of a number:

$$AA^{-1} = I_n$$

Which can then be used to solve vector equations:

$$A\vec{x} = \vec{y} \Rightarrow \vec{x} = A^{-1}\vec{y}$$

0.4 Eigenvalues and eigenvectors

For some special cases the following relation holds for matrix-vector multiplication:

$$A\vec{x} = \lambda\vec{x}$$

Where λ denotes an eigenvalue of A. This expression can then be used to find the eigenvalues of A:

$$(A - \lambda I)\vec{x} = \vec{0}$$

This form can then be used to find eigenvalues of A with the following:

$$\det(A - \lambda I) = 0$$

0.5 Differentiation

Time derrivatives of variables will be denoted with a dot on top of the variable name.

$$\dot{r} = \frac{\mathrm{d}r}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{r(t + \Delta t) - r(t)}{\Delta t}$$

The basic rules for differentiation are:

$$
\begin{aligned}
(c \cdot f(x))' &= c \cdot f'(x) \\
(f(g(x)))' &= f'(g(x)) \cdot g'(x) = \frac{\mathrm{d}f}{\mathrm{d}g}\frac{\mathrm{d}g}{\mathrm{d}x} && \text{(Chain Rule)} \\
(f(x) \cdot g(x))' &= f'(x)g(x) + f(x)g'(x) && \text{(Product Rule)} \\
\left(\frac{f(x)}{g(x)}\right)' &= \frac{f'(x)g(x) - f(x)g'(x)}{(g(x))^2} && \text{(Quotient Rule)}
\end{aligned}
$$

For multivaribale functions the derrivative is described in terms of partial derrivatives:

$$
\mathrm{d}f(x,y,z) = \frac{\partial f}{\partial x}\mathrm{d}x + \frac{\partial f}{\partial y}\mathrm{d}y + \frac{\partial f}{\partial z}\mathrm{d}z
$$

This can also more compactly be written as:

$$
\mathrm{d}f(x_1 \cdots x_N) = \sum_{i=1}^{N} \frac{\partial f}{\partial x_i}\mathrm{d}x_i
$$

In some cases $x = x(t)$. The total derrivative is then found applying the chain rule:

$$
\frac{\mathrm{d}f(x_1 \cdots x_N)}{\mathrm{d}t} = \sum_{i=1}^{N} \frac{\partial f}{\partial x_i}\frac{\mathrm{d}x_i}{\mathrm{d}t}
$$

Partial derrivatives are also used to express the gradient vector, which is the multi-variable analogy of the derrivative of some function.

$$
\nabla f(x,y,z) = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}
$$

0.6 Integration

Integration is less elegant and more difficult then differentiation, thus the tools for analytically solving integrals are limited. Integrals will be denoted as follows:

$$
r(t) = \int_{t_1}^{t_2} \dot{r}\,\mathrm{d}t
$$

Some general rules for analytical integration are:

$$
\begin{aligned}
\int c \cdot f(x)\,\mathrm{d}x &= c \cdot \int f(x)\,\mathrm{d}x \\
\int f(x)g'(x)\,\mathrm{d}x &= f(x)g(x) - \int f'(x)g(x)\,\mathrm{d}x
\end{aligned}
$$

Sometimes a u-subsitution can be used to analitally solve the integral, however in alot of cases no solution exists. In this case numerical methods are usually applied. One such example is the Euler method of numerical intgeration.[1]. The euler method uses the following principle:

$$
\begin{gathered}
y = y(t) \quad v = v(t) \quad a = C \\
\frac{\mathrm{d}y}{\mathrm{d}t} = v(t) \\
y_{n+1} \approx y_n + v_n \Delta t \\
v_{n+1} \approx v_n + a\Delta t
\end{gathered}
$$

The method becomes more exact for smaller choice of Δt time step.

[1]There are many more numerical methods of integrating which are more efficient, however the Euler method is easiest to implement in Python for example.