Geometric modeling in Engineering

Athanasios A. Markou, Ph.D.
University Lecturer
Department of Civil Engineering
Aalto University

Background

Background

Bachelor, Master in Aristotle University of Thessaloniki, Greece

PhD in Aristotle University, Greece and University of Catania, Italy

Postdoc in Norwegian Geotechnical Institute (NGI), Norway

Aalto since 5-2018, as University Teacher

University Lecturer, since 1-2021

Background in Earthquake Engineering, Structural Engineering, Dynamics

Responsible teacher:

Fundamentals of Structural Design (M) English

Numerical Methods in Engineering (B) English

Continuum Mechanics (B) English

Co-teacher:

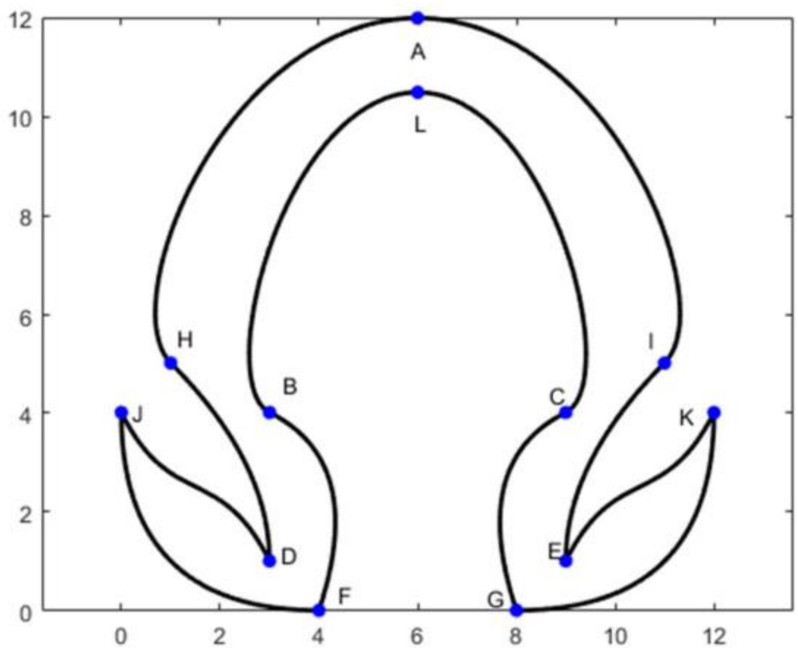
Computer-aided tools in engineering (B) English

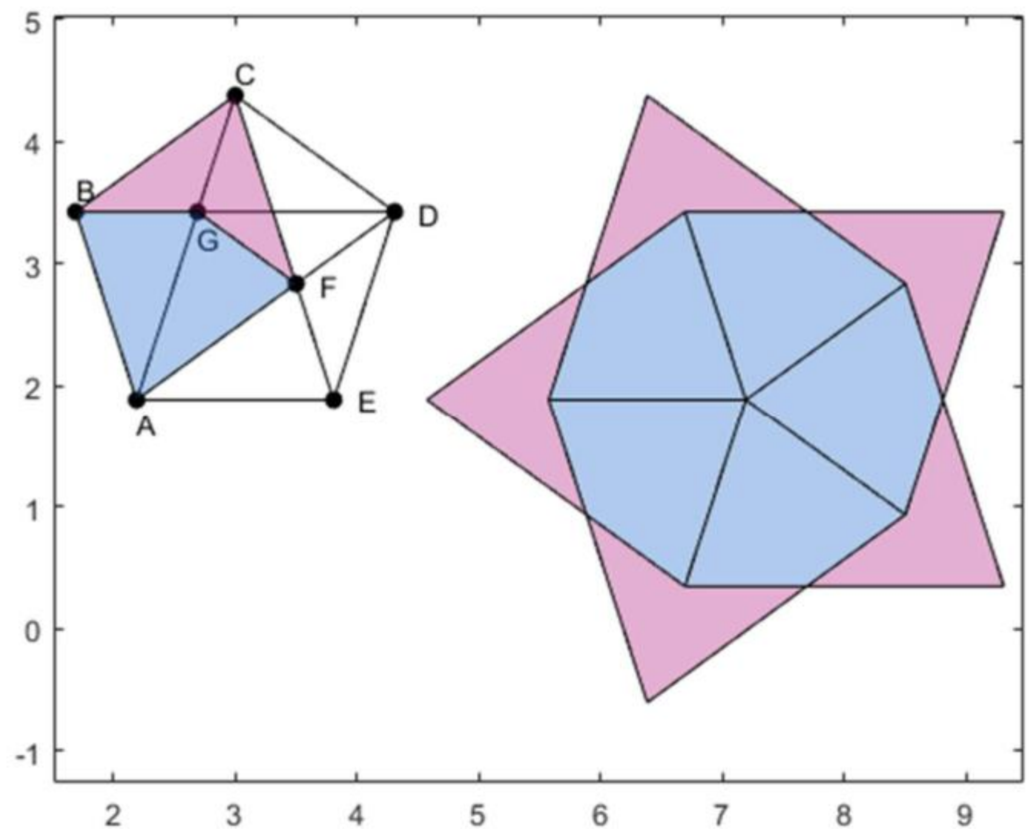
Tietokoneavusteiset työkalut insinööritieteissä (B) Finnish

Intro

Civil Engineering W1

MatLab



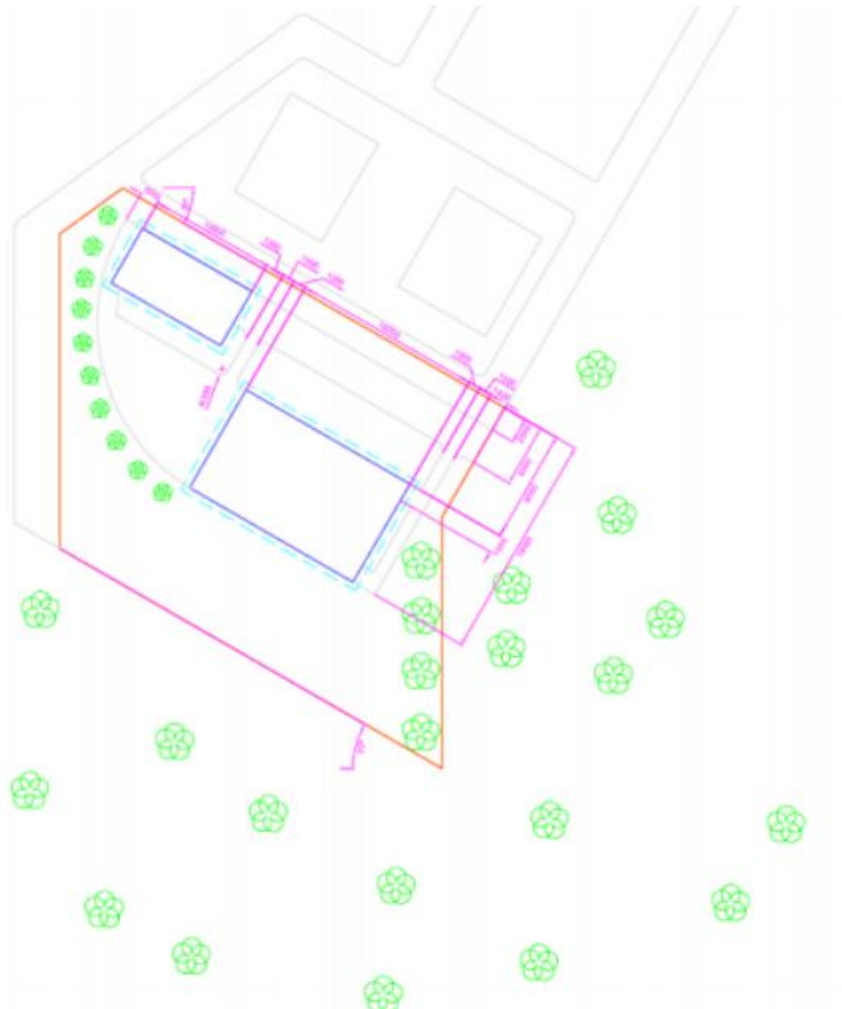


Assignments

Joona
Otto
Mark
Antton

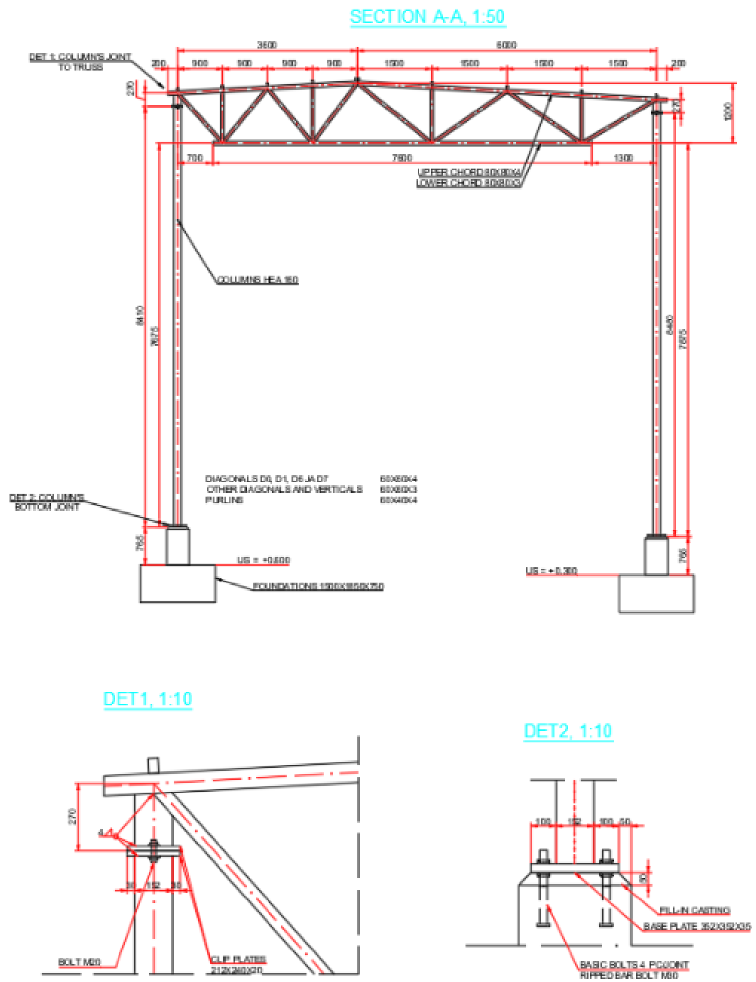
Civil Engineering W2

AutoCAD



Assignments
Joona
Otto
Mark
Antton

Civil Engineering W3

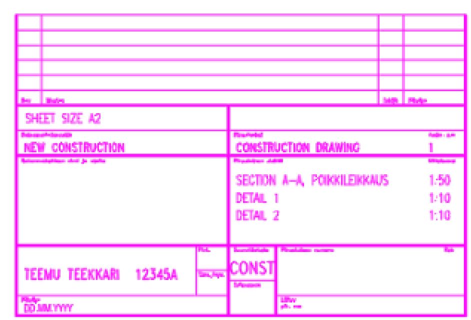


STEEL GRADE: S235H
TUBULAR BEAMS: RAUTARUUKKI

Assignments

Joona
Otto
Mark
Antton

AutoCAD

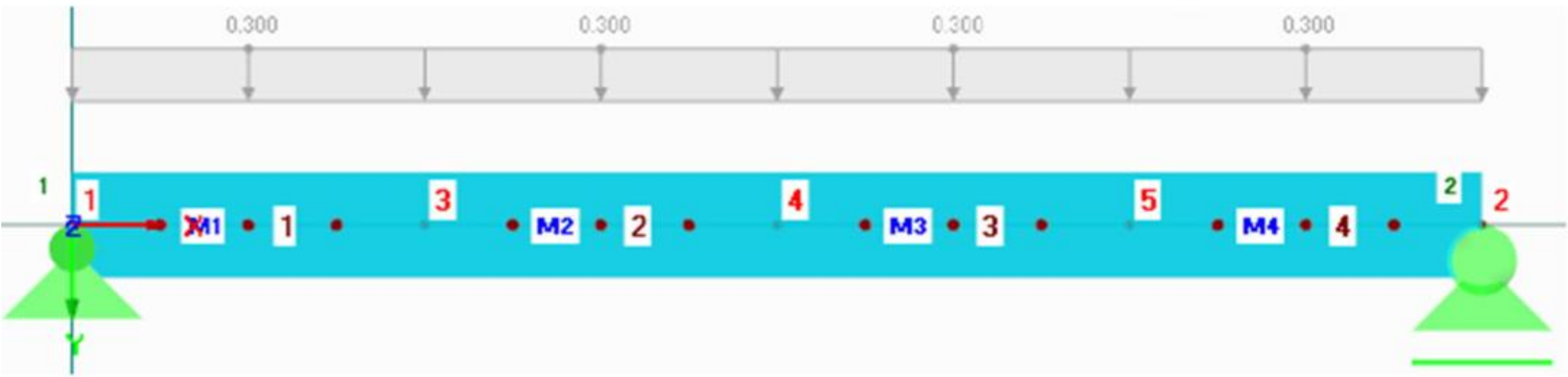


Civil Engineering W4

Assoc. Prof. Jarkko Niiranen

MatLab

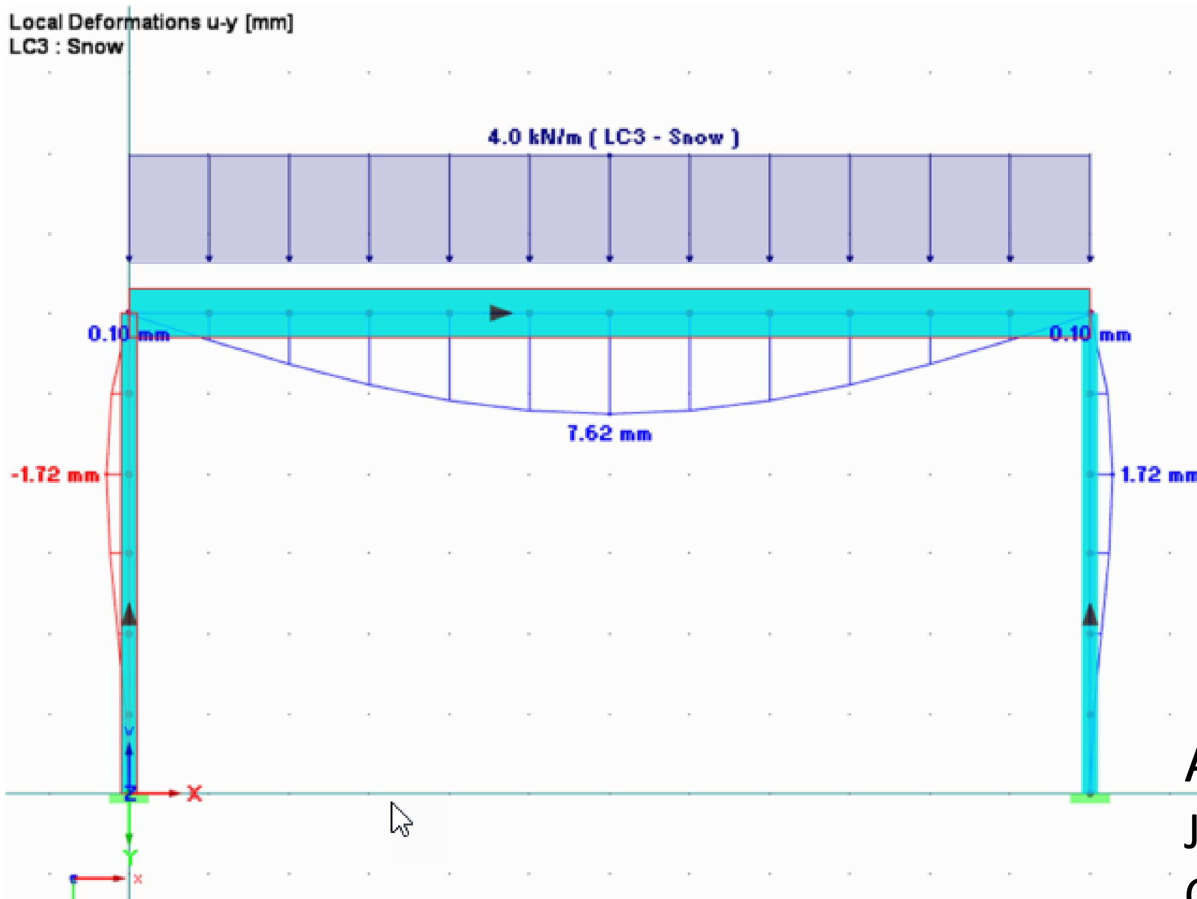
Civil Engineering W5



R-FEM

Assignments
Joona
Otto
Mark
Antton

Civil Engineering W6



R-FEM

Assignments

Joona

Otto

Mark

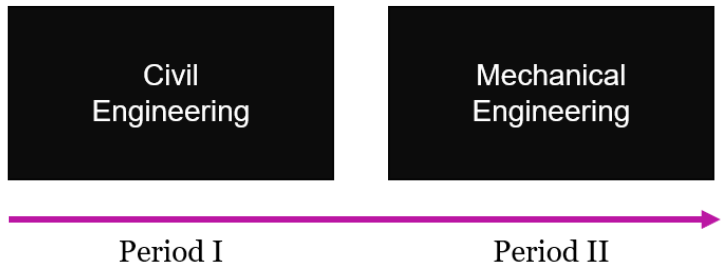
Antton

Software Installation

Mycourses front page

Structure of the course

The course is divided into two modules. First, computer-aided tools in civil engineering are presented. Second, tools in the field of mechanical engineering are presented. Both modules are needed to pass to pass the whole course.



Responsible teacher and mechanical engineering module: Kaur Jaakma

Civil engineering module: Athanasios Markou

SOFTWARE INSTALLATION
CIVIL ENGINEERING PART

Install Matlab

https://download.aalto.fi/student/index.html

Install AutoCAD

https://www.autodesk.com/company/legal-notices-trademarks/ccpa-do-not-sell

Install R-FEM (student version, NOT TRIAL)

https://www.dlubal.com/en/education/students/free-structural-analysis-software-for-students

Computers in Campus

https://wiki.aalto.fi/pages/viewpage.action?spaceKey=AaltoWin&title=Aalto+IT+Windows+Classroom+Software+list

Virtual connection

https://vdi.aalto.fi/?includeNativeClientLaunch=true

Geometric modeling in Engineering

Geometric modeling in Engineering

Content:

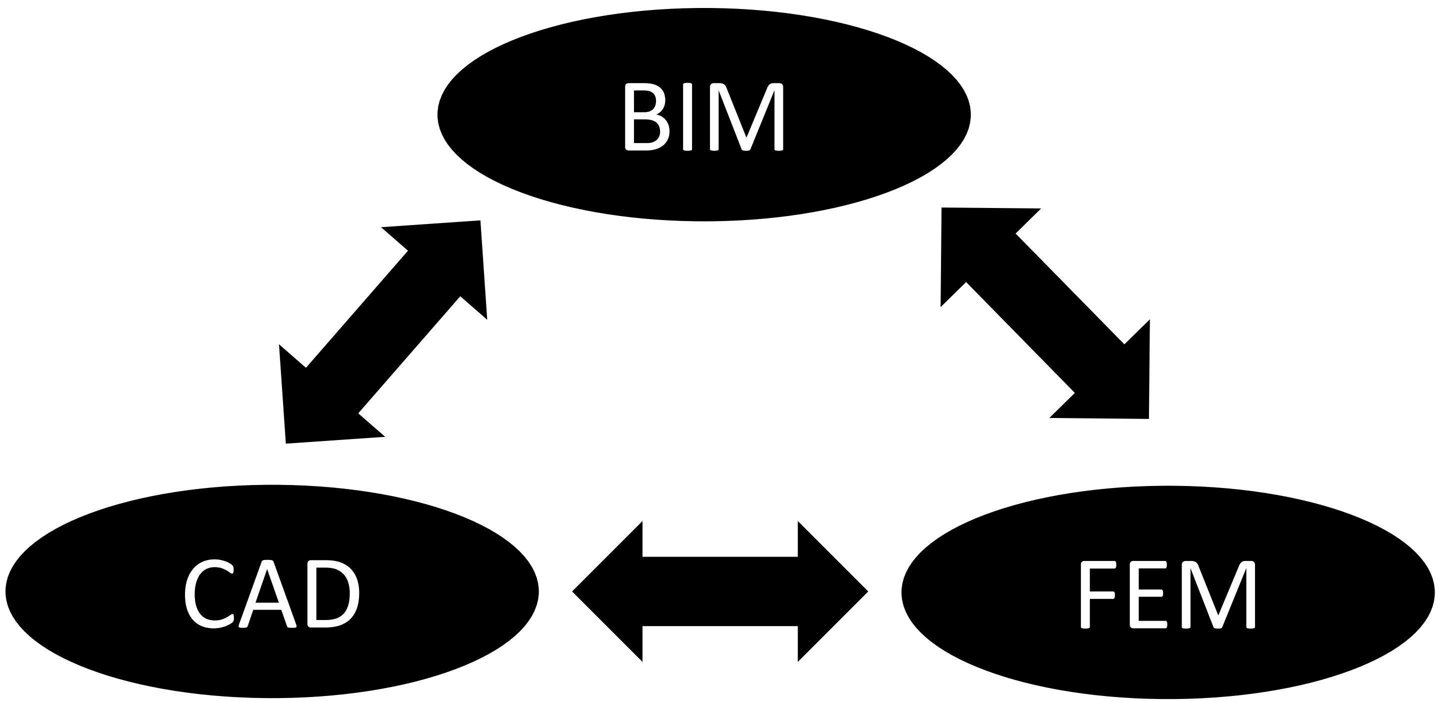
- Geometric transformations of the plane
- Homogeneous Coordinates
- Curves
- Surfaces

Geometrical modeling in Engineering

Learning outcomes:

- Understand affine transformations
- Understand homogeneous coordinates
- Implement computer execution of geometric transformations
- Describe curves
- Understand Bezier curves

Geometric modeling in Structural Engineering



Building Information Modeling (BIM)
Computer-Aided Design (CAD)
Finite Element Method (FEM)

Geometric modeling in Structural Engineering

BIM

Building Information Modeling (BIM) is a process that involves digital representations of physical characteristics of places.

Models are computer files which are networked to support decision making regarding a building.

Software is used for design, construct, operate and maintain buildings and infrastructures.

Geometric modeling in Structural Engineering

CAD

Computer-Aided Design (CAD) is the use of computers to create, modify, analyze or optimize a design.

CAD files are computer files that used for print, machining or other operations.

CAD software uses vector-based graphics to depict objects.

Geometric modeling in Structural Engineering

FEM

Finite Element Method (FEM) is a method used for solving partial differential equations of engineering, such as structural analysis, heat transfer, fluid flow, mass transport and electromagnetic potential.

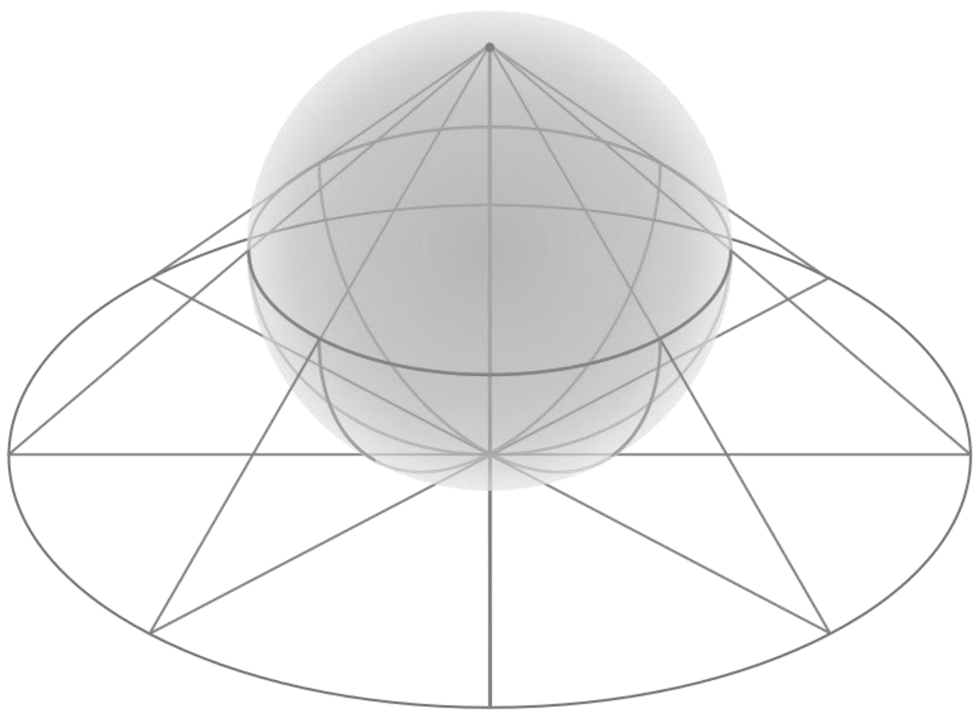
Geometric modeling in Engineering

Computer Science deals with the theoretical foundations of information, its computation and practical techniques for their application, (Wikipedia).

λ

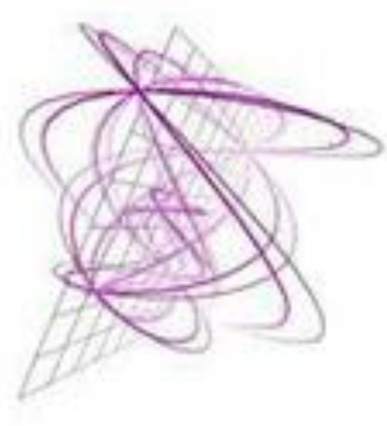
Geometric modeling in Engineering

Computational Geometry is a branch of computer science that study algorithms that can be stated in terms of geometry, (Wikipedia).



Geometric modeling in Engineering

Geometric Design (Geometric Modeling) is a branch of computer science that study algorithms that can be stated in terms of geometry, (Wikipedia).



Geometric modeling in Engineering

Geometric Design deals with free-form curves, surfaces and volumes.

Geometric **models** can be represented either parametrically or implicitly.

For their construction they use mainly polynomial methods to a set of points.

Parametric curves and surfaces (Bezier, spline) are important instruments.

Geometric **models** are distinguished from procedural and object-oriented models that define the shape implicitly through an algorithm (video games).

Geometric **models** are used in computer graphics.

Geometric Transformations of a plane

Euclidean Geometry (EG)

Alexandrian Greek mathematician **Euclid,** wrote **‘Elements’** (13 books).

‘Elements’ state results of **algebra** and **number theory** in geometric language.

EG is a mathematical system, based on axioms without coordinates (**axiomatic geometry**).

Analytic Geometry (AG)

French mathematicians: **Rene Descartes** and **Pierre de Fermat.**

Both developed independently AG, even though credits are given solely to Descartes.

AG is a mathematical system, based on coordinates.

Geometric Transformations of the plane - Notation

A point P in Euclidean plane E^2 with its coordinate vector $\mathrm{x} \in \mathbb{R}^2$ in a chosen orthogonal basis.

If $\mathrm{x}_0 \in \mathbb{R}^2$ is a point and $\mathbf{v} \neq \mathbf{0}$ then the set is a line:

$$\mathrm{L} = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = \mathbf{x}_0 + \mathrm{t}\mathbf{v}, \mathrm{t} \in \mathbb{R}\}$$

In matrix notation coordinates are represented by vectors:

$$\mathbf{x} = \begin{pmatrix} \mathrm{x} \\ \mathrm{y} \end{pmatrix} = \mathrm{x}\vec{\mathrm{i}} + \mathrm{y}\vec{\mathrm{j}}$$

Affine transformations

Affine transformation of a plane is a mapping of the form:

$$L(x, y) = (ax + by + c, ex + fy + g)$$

where a, b, c, e, f, g are constants

The point $P' = L(P)$ is the image of P

In matrix form:

$$\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{b}, \qquad \mathbf{A} = \begin{pmatrix} a & b \\ e & f \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} c \\ g \end{pmatrix}$$

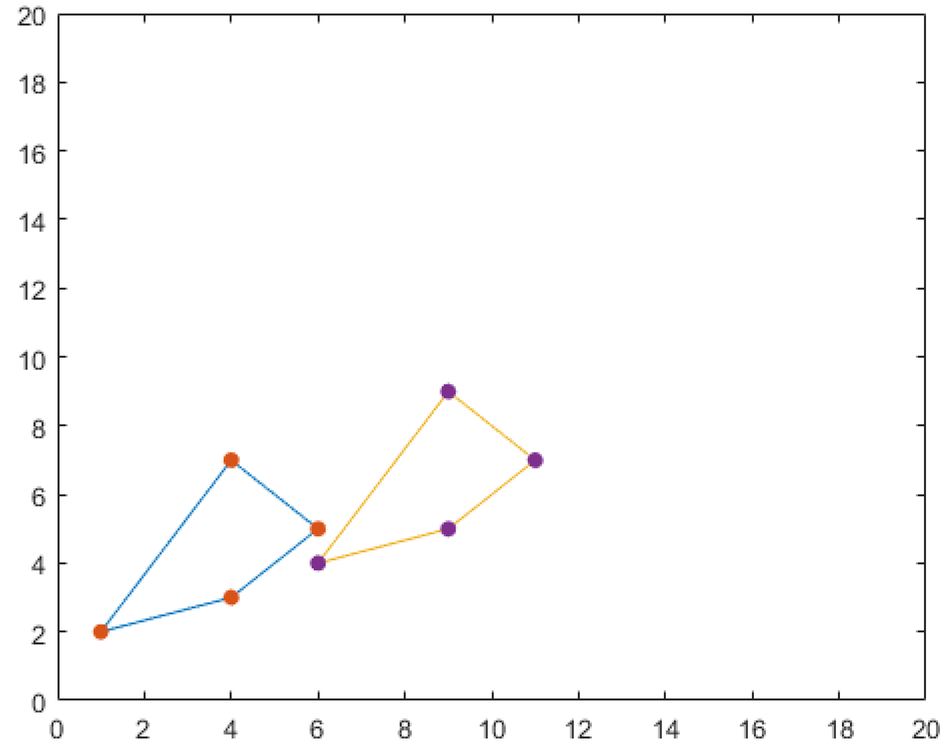
Translation

Translation can be obtained by adding a constant to each coordinate.

In matrix form:

$$\mathbf{x}' = \mathbf{x} + \mathbf{b}$$

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} c \\ g \end{pmatrix}$$



Scaling

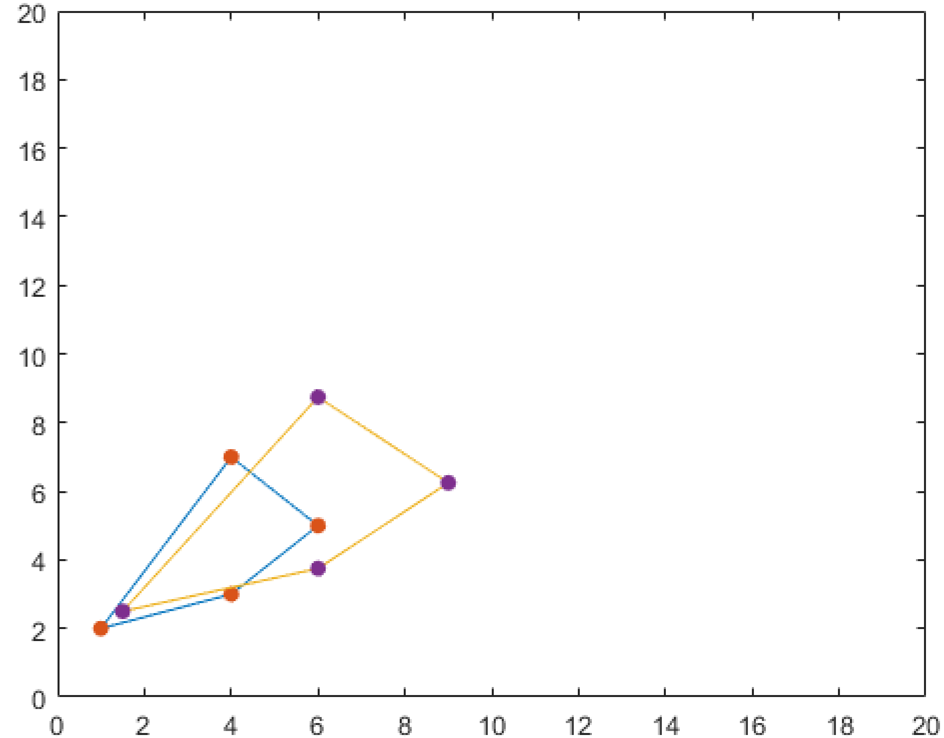
A scaling about the origin can done by using transformation matrix $\mathbf{\Lambda}$.

In matrix form:

$$\mathbf{x}' = \mathbf{\Lambda x}$$

$$\mathbf{x} = \begin{pmatrix} \mathrm{x} \\ \mathrm{y} \end{pmatrix}$$

$$\mathbf{\Lambda}\left(\lambda_{\mathrm{x}}, \lambda_{\mathrm{y}}\right) = \begin{pmatrix} \lambda_{\mathrm{x}} & 0 \\ 0 & \lambda_{\mathrm{y}} \end{pmatrix}$$



Reflection

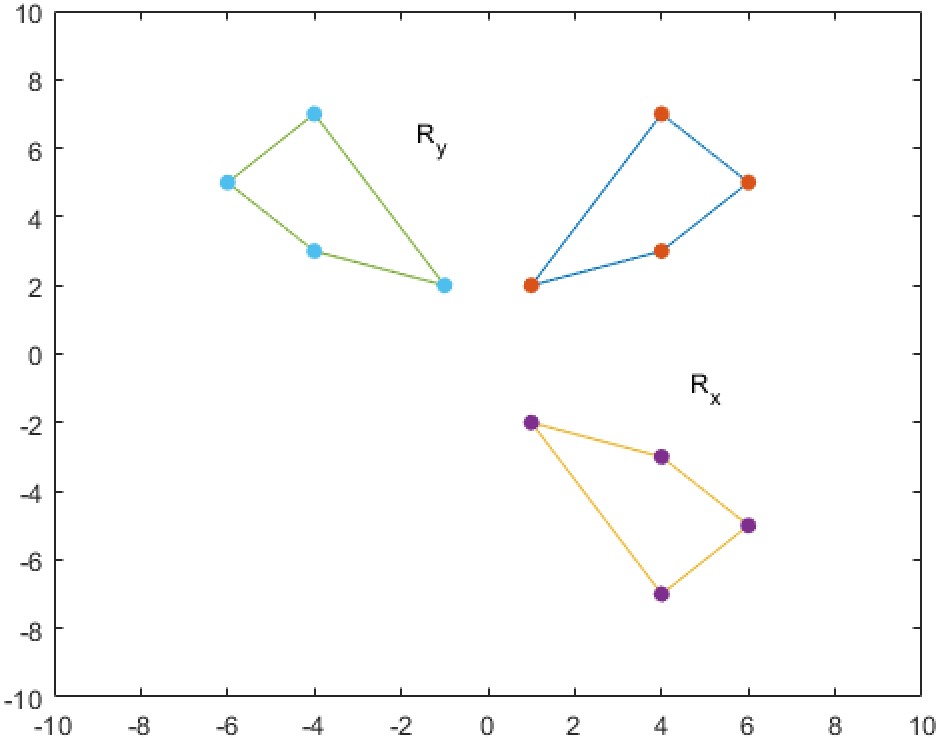
Horizontal and vertical flip can be done by about x-axis or y-axis and is called reflection.

In matrix form:

$$\mathbf{x}' = \mathbf{R}\mathbf{x}$$

$$\mathbf{R}_{\mathbf{x}} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\mathbf{R}_{\mathrm{y}} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$



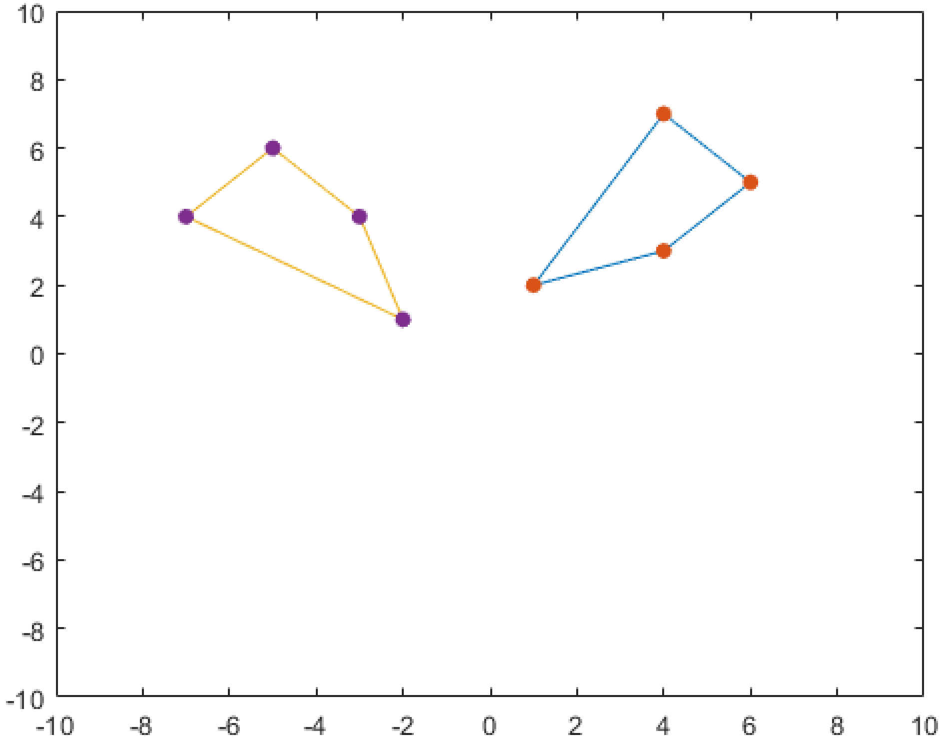
Rotations

The point $P' = L(P)$ is the image of P

In matrix form:

$$\mathbf{x}' = \mathbf{A}_\varphi \mathbf{x}$$

$$\mathbf{A}_\varphi = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$$



Rigid Body Motions

Provided that $\mathrm{L}(\mathrm{P})$ involves only rotations and translations, the transformation is called ***rigid body motion***.

In ***finite element analysis*** is essential that element formulations are able to represent rigid body motions and constant strain states (Patch tests).

Inverse Transformation

Transformation that leaves all points unchanged is the identity transformation.

The inverse transformation L, namely L^{-1}, provides:

$$\mathrm{L}^{-1}\left(\mathrm{L}\left(\mathrm{P}\right)\right) = \mathrm{P}; \qquad \mathrm{L}\left(\mathrm{L}^{-1}\left(\mathrm{P}\right)\right) = \mathrm{P}$$

The process of two simultaneous transformations is called concatenation or composition.

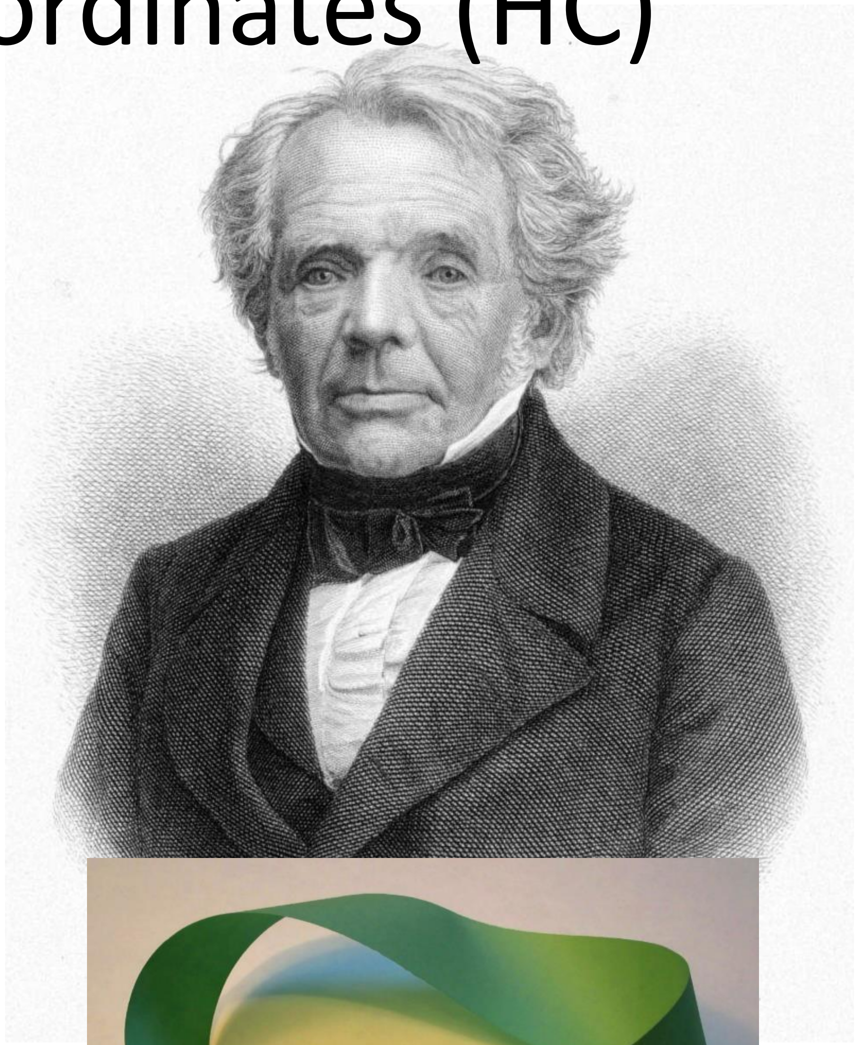
Concatenation of translation with other transformations requires matrix addition with matrix multiplication.

Homogeneous Coordinates

Homogeneous Coordinates (HC)

Introduced by German mathematician ***August Ferdinand Möbius*** in 1827 (known for ***Möbius strip***).

Problem of concatenation of a translation can be avoided by using HC system performed by *3x3 matrix multiplication* in the plane case.

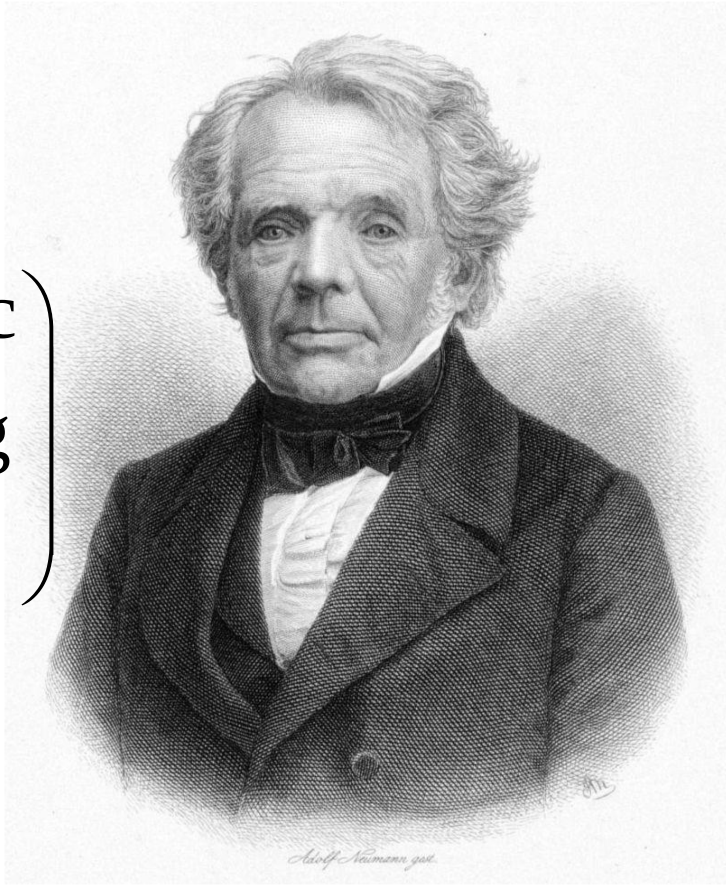


Homogeneous Coordinates (HC)

HC also called projective coordinates.

We can write:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ e & f & g \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} ax + by + c \\ ex + fy + g \\ 1 \end{pmatrix}$$



Homogeneous Coordinates (HC)

New coordinate system can be defined, where P with coordinates (x,y) is represented by HC (x,y,1) or (λx,λy,λ), with λ different from zero:

$$P = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \lambda x \\ \lambda y \\ \lambda \end{pmatrix}$$

The set of HC (x,y,w) is the projective plane.

HC Example

Find the Cartesian coordinates of the following points in the projective plane: $(5,10,5),(3,6,3),(-1,-2,-1)$

The Cartesian coordinates of all points, are $(1,2)$:

$$P = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} = \frac{1}{5}\begin{pmatrix} 5 \\ 10 \\ 5 \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 3 \\ 6 \\ 3 \end{pmatrix} = -1\begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix}$$

Projective transformations

A projective transformation of the projective plane is the mapping of the form

$$L(x,y,w) = (ax+by+c, ex+fy+g, hx+ky+l)$$

with a, b, c, e, f, g, h, k, l constants.

Homogeneous transformation matrix **A**:

$$\mathbf{A} = \begin{pmatrix} a & b & c \\ e & f & g \\ h & k & l \end{pmatrix}$$

For h=k=0 and l different from 0, we have affine transformation of the Cartesian plane.

Concatenation of transformations

In HC the concatenation of transformation becomes a matrix multiplication

A rotation and a translation can be written as:

$$\mathbf{A}_{\varphi}\mathbf{T}\left(h_x, h_y\right) = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & h_x \\ 0 & 1 & h_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\varphi & -\sin\varphi & h_x\cos\varphi - h_y\sin\varphi \\ \sin\varphi & \cos\varphi & h_x\sin\varphi + h_y\cos\varphi \\ 0 & 0 & 1 \end{pmatrix}$$

Rotation about a Point

For an anticlockwise rotation through an angle ϕ about a point (x0,y0) can be obtained as follows:

1. Mapping (x0,y0) to origin
2. Rotation of angle ϕ about the origin
3. Mapping the origin to (x0,y0)

The homogeneous matrix is

$$\mathbf{A}_{\varphi}\left(x_0, y_0\right) = \mathbf{T}\left(x_0, y_0\right)\mathbf{A}_{\varphi}\mathbf{T}\left(-x_0, -y_0\right)$$

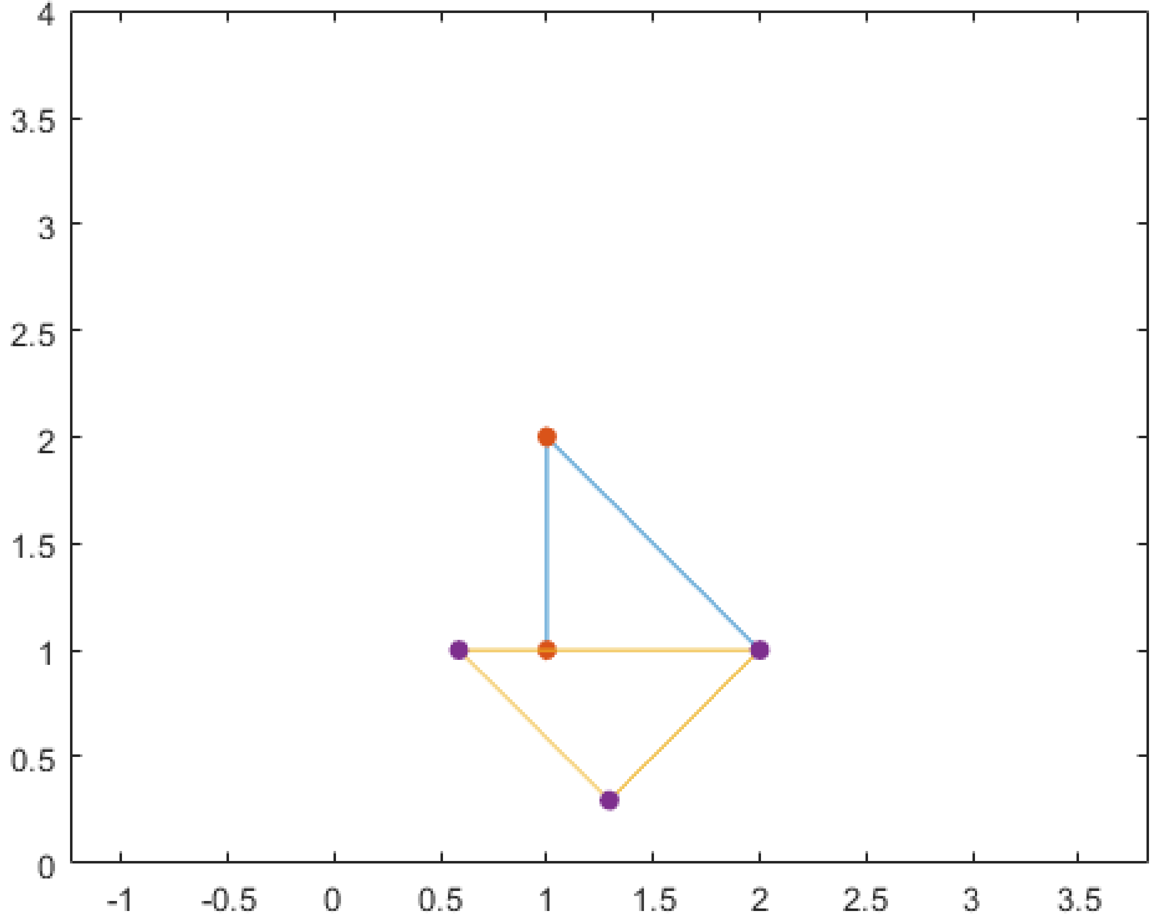
Rotation about a Point

$$\mathbf{A}_{\varphi}\left(x_0, y_0\right) = \mathbf{T}\left(x_0, y_0\right)\mathbf{A}_{\varphi}\mathbf{T}\left(-x_0, -y_0\right)$$

$$x_0 = 2$$

$$y_0 = 1$$

$$\varphi = \frac{\pi}{4}$$



Curves

Parametric and implicit forms

Two common ways to represent curves in geometric modeling: **implicit** and **parametric** forms.

Implicit form: $f(x, y) = 0$

Parametric: $C(t) = C(x(t), y(t)),\ t \in [a, b]$

where $x(t), y(t)$ are the coordinate functions

The interval $[a, b]$ can be normalized to $[0, 1]$

Example of parametric function

The first quadrant of the unit circle is defined:

$$x(t) = \cos t \qquad t \in \left[0, \pi/2\right]$$
$$y(t) = \sin t$$

Alternatively, it can be expressed as function of:

$$u(t) = \tan\frac{t}{2}$$

$$x(u) = \frac{1-u^2}{1+u^2} \qquad u \in [0,1]$$

$$y(t) = \frac{2u}{1+u^2}$$

Parametric and implicit forms

Parametric form in not unique.

The first quadrant of the unit circle in implicit form can be expressed:

$$x^2 + y^2 - 1 = 0, \qquad x, y \in [0,1]$$

A equivalent parametric representation is:

$$C(t) = C\left(t, \sqrt{1 - t^2}\right), \qquad t \in [0,1]$$

Parametric and implicit forms

Parametric form can be extended in special 3D curves by using a third coordinate function.

Bounded forms of curves can be represented directly in parametric form by restricting the variable.

In the case of unbounded curves like the straight line, the implicit form is more meaningful:

$$ax + by + c = 0$$

Parametric and implicit forms

Parametric curves have inherent direction from C(a) to C(b).

The complexity of the geometric manipulations depends on:

Estimating a point in parametric form is easy, but complicated in implicit form.

Whether a point belongs to a curve is easy with implicit form, but complicated with parametric one.

Conic Sections

Conic Sections

Plane curves can be defined by a quadratic polynomial function:

$$ax^2 + 2bxy + cy^2 + 2ex + 2fx + g = 0$$

where a, b, c, e, f, g are coefficients

In matrix form: $\mathbf{PQP}^{\mathrm{T}} = 0$

where

$$\mathbf{Q} = \begin{pmatrix} a & b & e \\ b & c & f \\ e & f & g \end{pmatrix}, \qquad \mathbf{P} = \begin{pmatrix} x & y & 1 \end{pmatrix}$$

Conic Sections

Many design and manufacturing processes are based on conical sections.

In parametric form:

$$\left(x(t), y(t)\right) = \left(\frac{a_0 + a_1 t + a_2 t^2}{c_0 + c_1 t + c_2 t^2}, \frac{b_0 + b_1 t + b_2 t^2}{c_0 + c_1 t + c_2 t^2}\right)$$

Bezier Curves

Geometric Modeling of Curves

Using specific class of functions that are rich and easy to handle.

Most widely used class of functions is the polynomials

Used for computer implementations, but cannot represent curve types as conic sections exactly.

Power Basis Curves

Using specific class of functions that are rich and easy to handle.

$$C(t) = \sum_{i=1}^{n} a_i t^i, \quad 0 \le t \le 1$$

where the n+1 functions $\{t_i\}$ are called the basis functions and $\{a_i\} = \{x_i, y_i\}$

Differentiation yields: $$a_i = \frac{C^{(i)}(0)}{i!}$$

Power Basis Curves - Example

For n=1

$$C(t) = a_0 + a_1 t, \qquad 0 \le t \le 1$$

is a line segment between points a_0 and $a_0 + a_1$

The a_1 gives the slope of the line.

For n=2

$$C(t) = a_0 + a_1 t + a_2 t^2, \qquad 0 \le t \le 1$$

is a parabolic arc between points a_0 and $a_0 + a_1 + a_2$

Power Basis Curves - Problems

Not obvious geometric shape of the curve from the coefficients.

Need for specification of end conditions at both ends of the curve in applications.

Non-Uniform Rational B-Splines (NURBS) are successful due to their ability to represent conic section and free-form shapes.

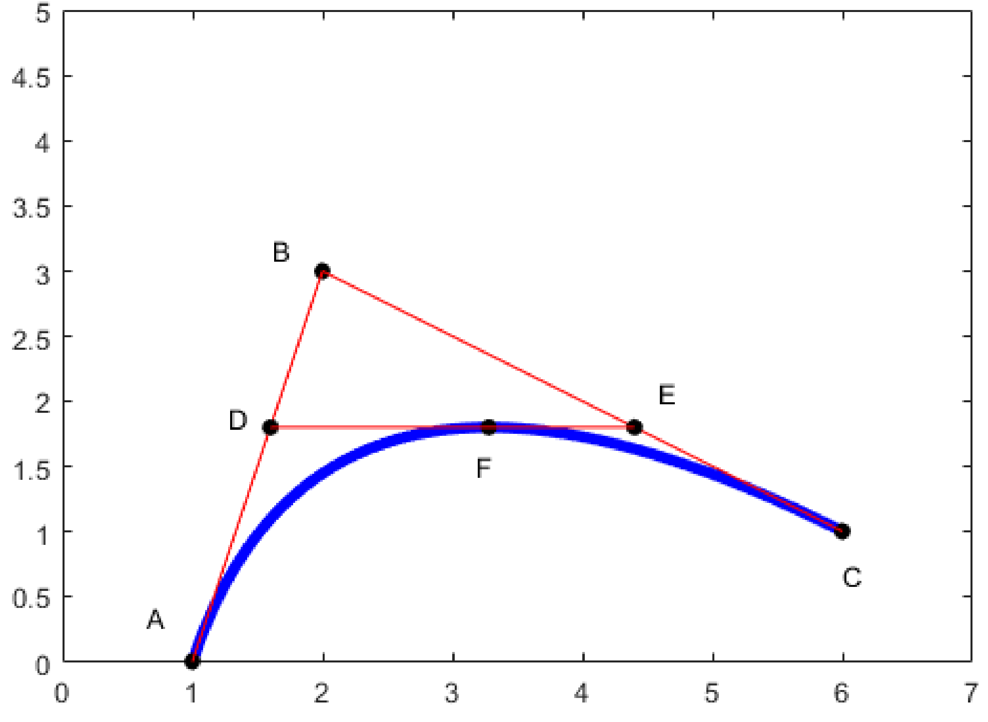
The works of Pierre Bezier (Renault) and Paul de Faget Casteljau (Citroen) are considered the origin of CAD.

Bezier curves

$$\frac{AD}{AB} = \frac{BE}{BC} = \frac{DF}{DE} = t_i, \qquad 0 \le t_i \le 1$$

$$A(x_A, y_A)$$
$$B(x_B, y_B)$$
$$C(x_C, y_C)$$

$$x_D = x_A + t_i(x_B - x_A)$$
$$y_D = y_A + t_i(y_B - y_A)$$
$$x_E = x_B + t_i(x_C - x_B)$$
$$y_E = y_B + t_i(y_C - y_B)$$
$$x_F = x_D + t_i(x_E - x_D)$$
$$y_F = y_D + t_i(y_E - y_D)$$



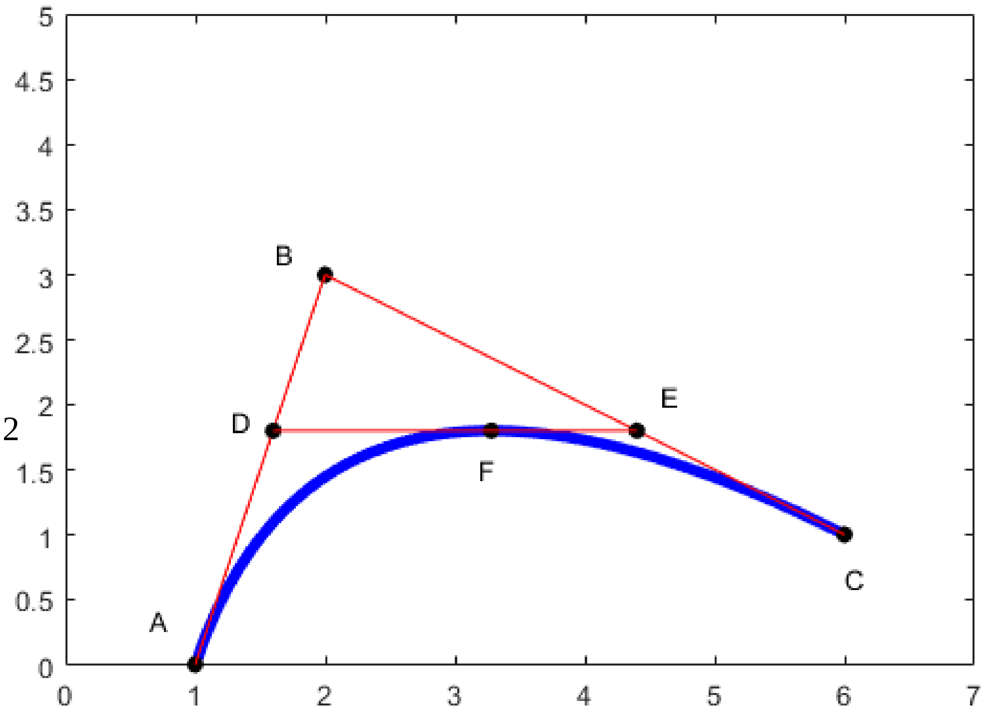
Bezier curves

$$x_F = (1-t_i)^2 x_A + 2t_i(1-t_i)x_B + t_i^2 x_C \qquad x(t) = (1-t)^2 x_0 + 2t(1-t)x_1 + t^2 x_2$$

$$y_F = (1-t_i)^2 y_A + 2t_i(1-t_i)y_B + t_i^2 y_C \qquad y(t) = (1-t)^2 y_0 + 2t(1-t)y_1 + t^2 y_2$$

In vector form:

$$\mathbf{p}(t) = (1-t)^2 \mathbf{p}_0 + 2t(1-t)\mathbf{p}_1 + t^2 \mathbf{p}_2$$



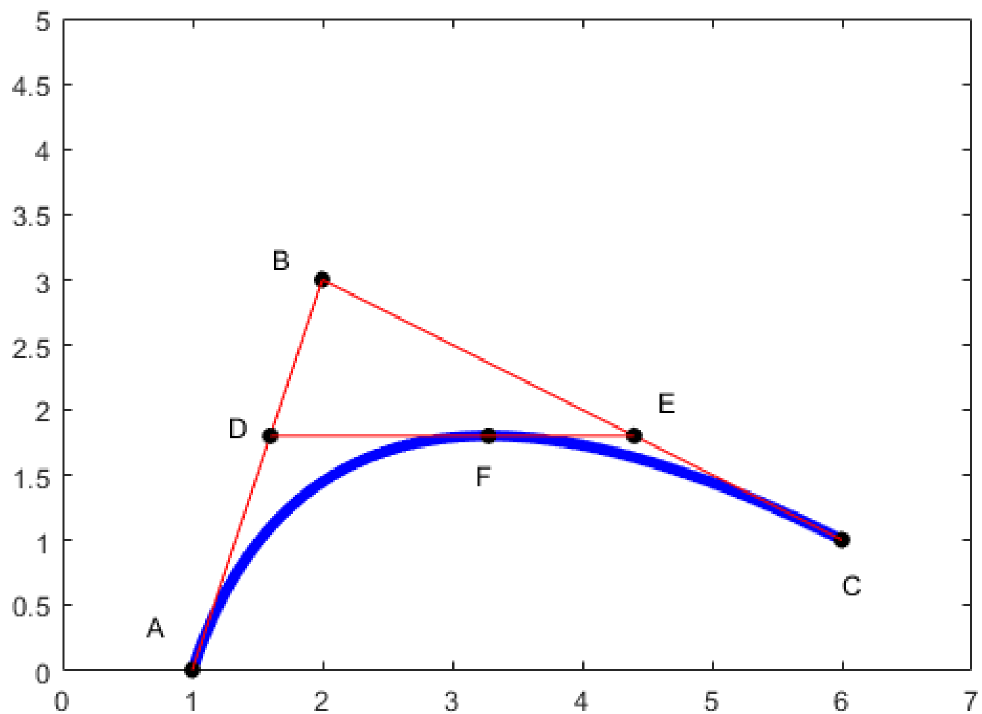
Bezier curves

In matrix form: $\mathbf{p}(t) = \mathbf{UMP}$

$$\mathbf{P} = \begin{pmatrix} \mathbf{p_0} \\ \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix}; \quad \mathbf{p}_i = \begin{pmatrix} p_i(x) \\ p_i(y) \end{pmatrix}, \quad i = 0,1,2$$

$$\mathbf{U} = \begin{pmatrix} t^2 & t & 1 \end{pmatrix}$$

$$\mathbf{M} = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$



Bezier curves

For n+1 control points, p0, p1, …pn, the Bezier curve of degree n is:

$$\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{p_i} B_{i,n}(t); \quad 0 \le t \le 1$$

where $B_{i,n}(t)$ are the Bernstein basis polynomials

$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}$$

and $\frac{n!}{i!(n-i)!}$ are the binomial polynomials.

Bezier curves

The binomial polynomials $$\frac{n!}{i!(n-i)!}$$

n=0

.

.

.

.

.

n=7

$$
\begin{array}{ccccccccccccccc}
 & & & & & & & 1 & & & & & & & \\
 & & & & & & 1 & & 1 & & & & & & \\
 & & & & & 1 & & 2 & & 1 & & & & & \\
 & & & & 1 & & 3 & & 3 & & 1 & & & & \\
 & & & 1 & & 4 & & 6 & & 4 & & 1 & & & \\
 & & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & & \\
 & 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 & \\
1 & & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & & 1
\end{array}
$$

Bezier curves

For the linear case the $\mathbf{p}(t)=\mathbf{p_0}(1-t)+\mathbf{p}_1 t$

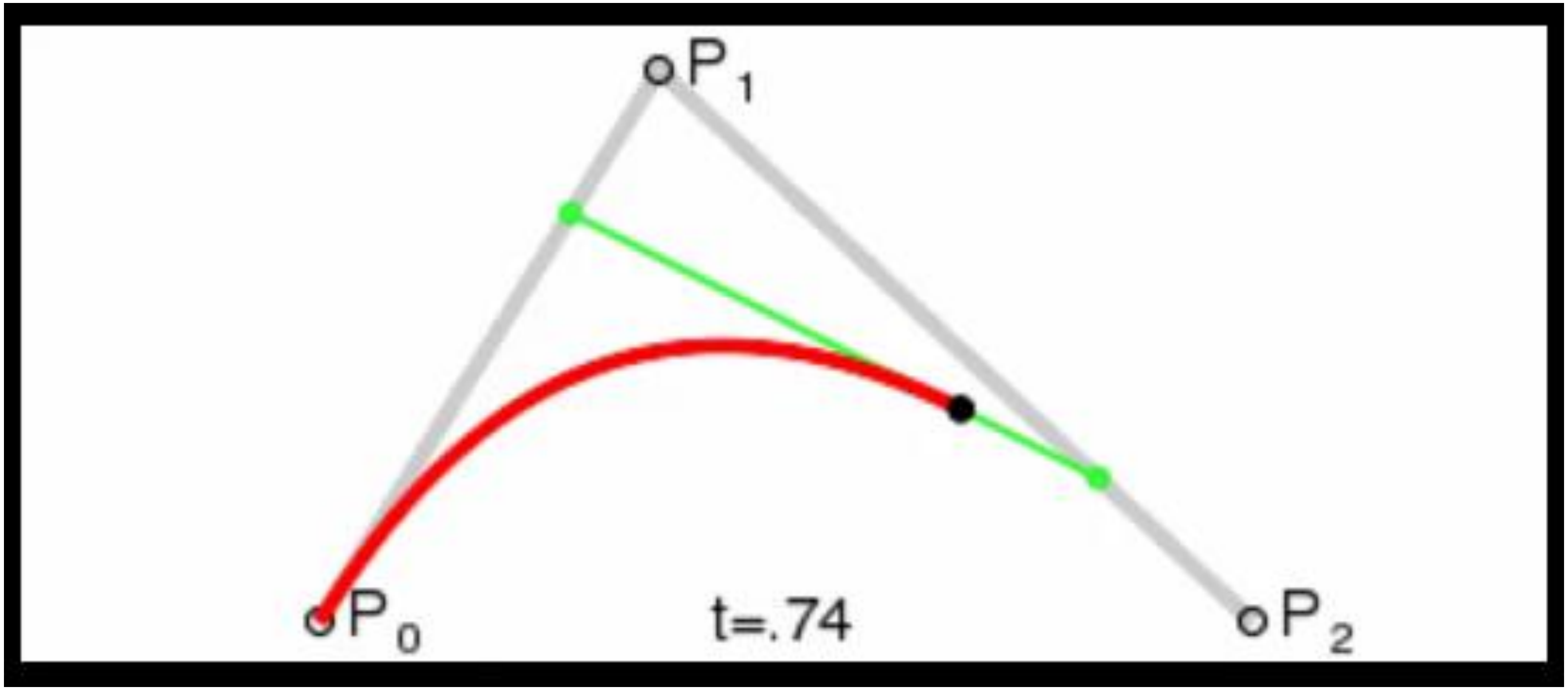
Due to $\mathbf{p}(0)=\mathbf{p_0}$ and $\mathbf{p}(1)=\mathbf{p_1}$ the curve interpolates first and last control points.

In the quadratic case:

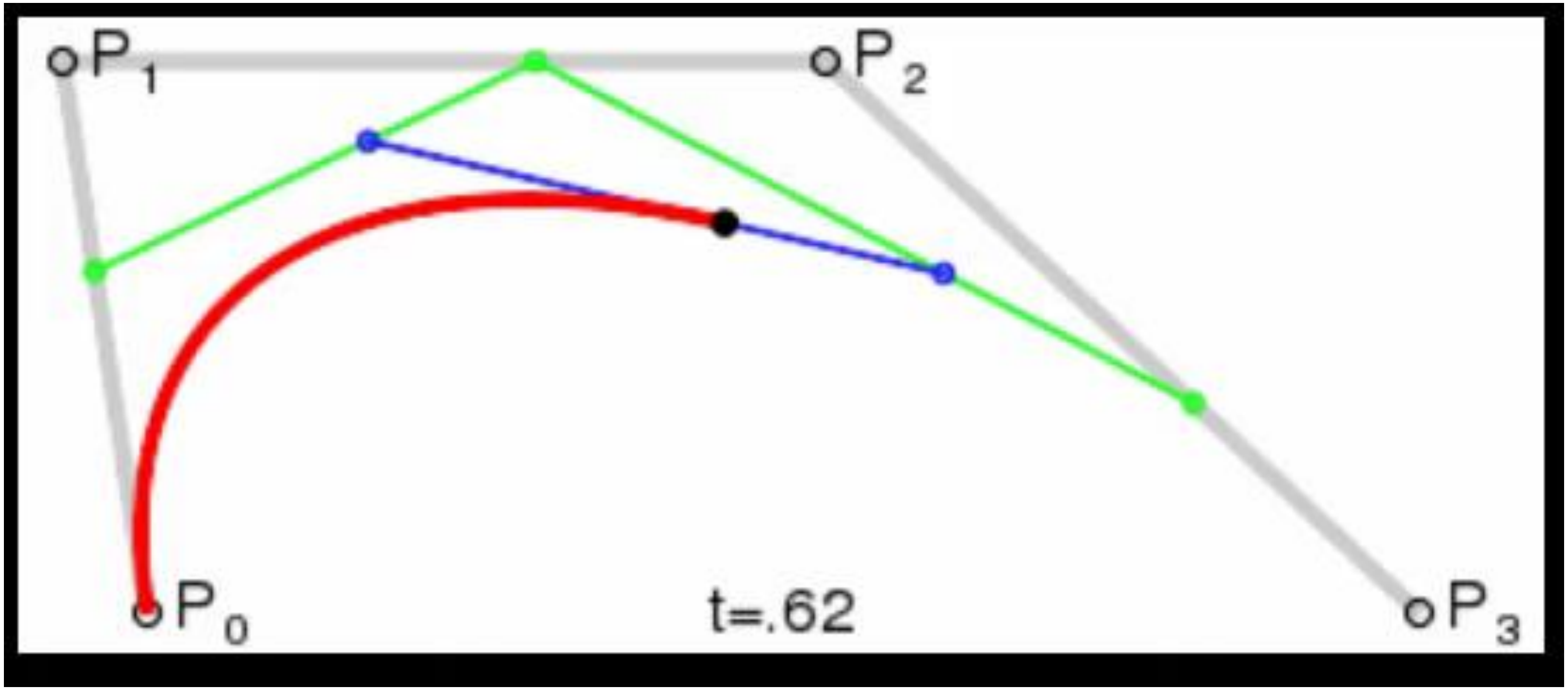
$$\mathbf{p}(t)=\mathbf{p_0}(1-t)^2+\mathbf{p}_1 2t(1-t)+\mathbf{p}_2 t^2$$

The polygon $\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2$ is the control polygon.

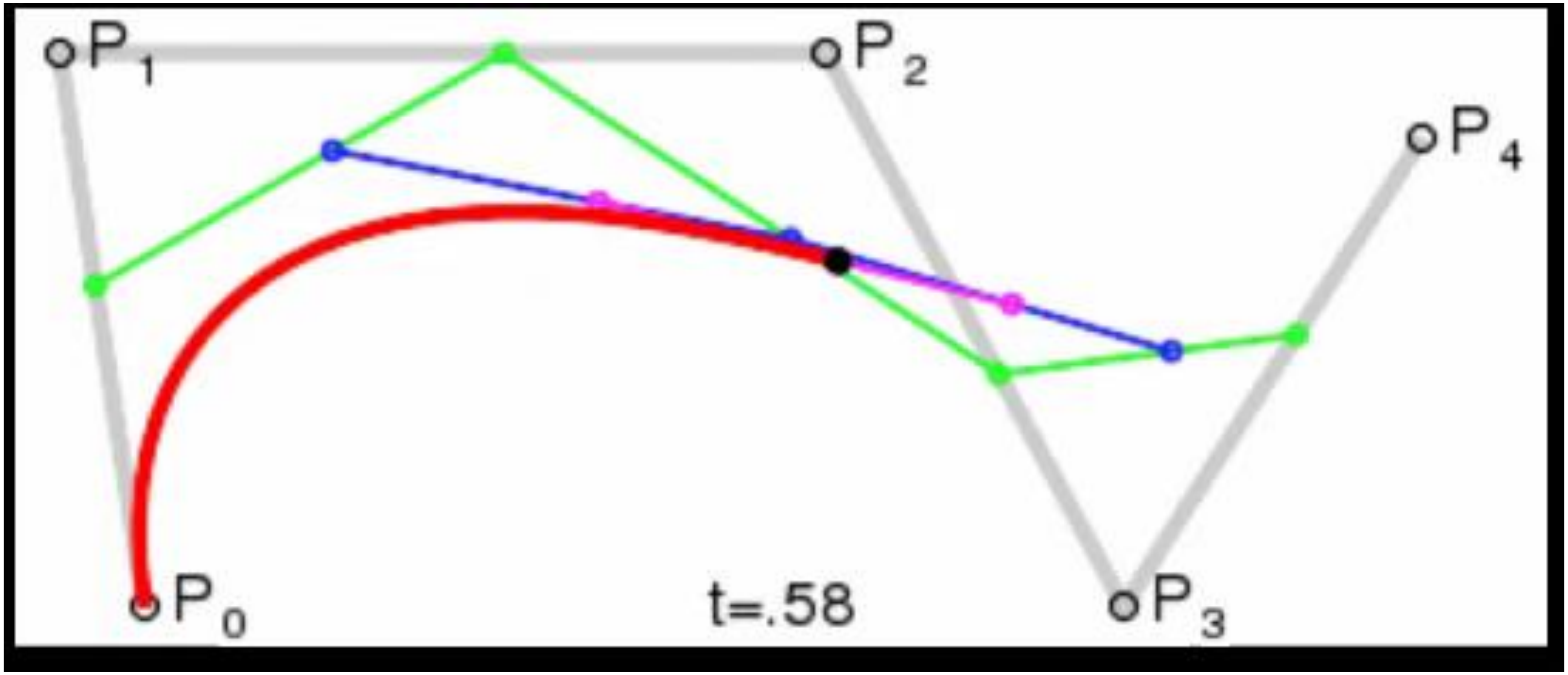
Bezier curves



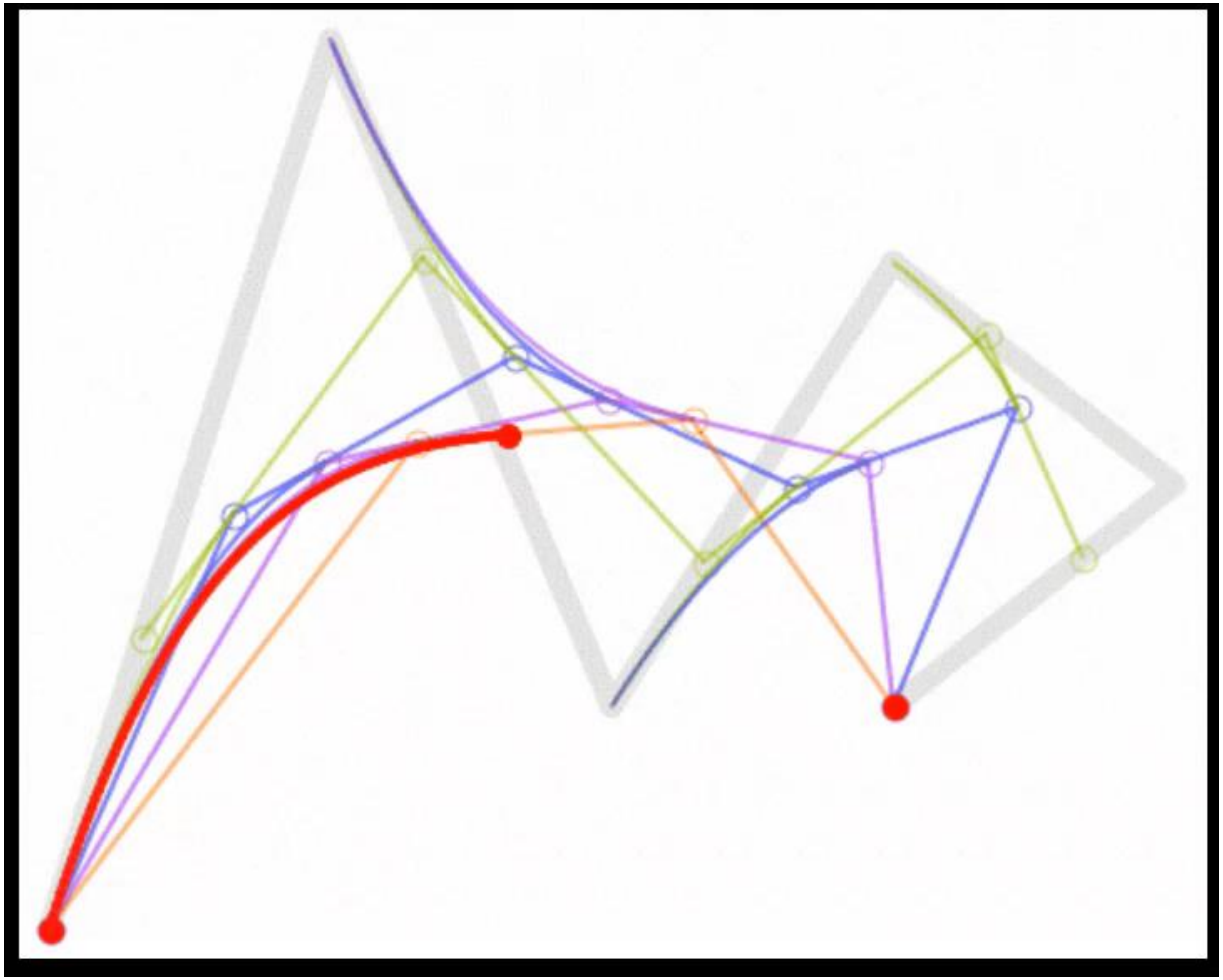
Bezier curves



Bezier curves

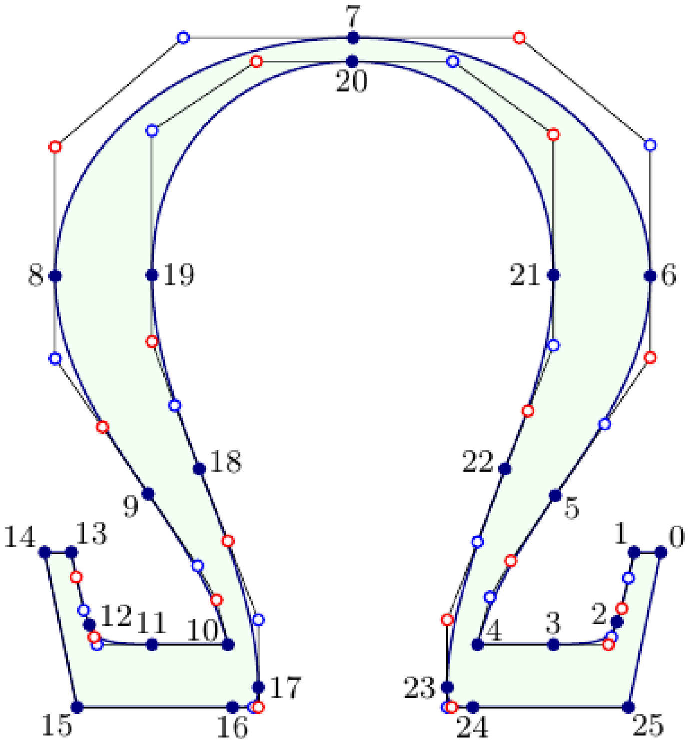


Bezier curves



Bezier curves - Applications

- Design
- Animation
- Computer graphics
- Fonts



B-splines, NURBS, Surfaces

B-splines, NURBS, Surfaces

https://www.youtube.com/watch?v=qhQrRCJ-mVg

https://www.youtube.com/watch?v=2WUKLkNLXlI

Assignments Week 1

Rotating a vector (angle known)

$$\vec{O}P = \vec{O}Q + \vec{Q}P$$

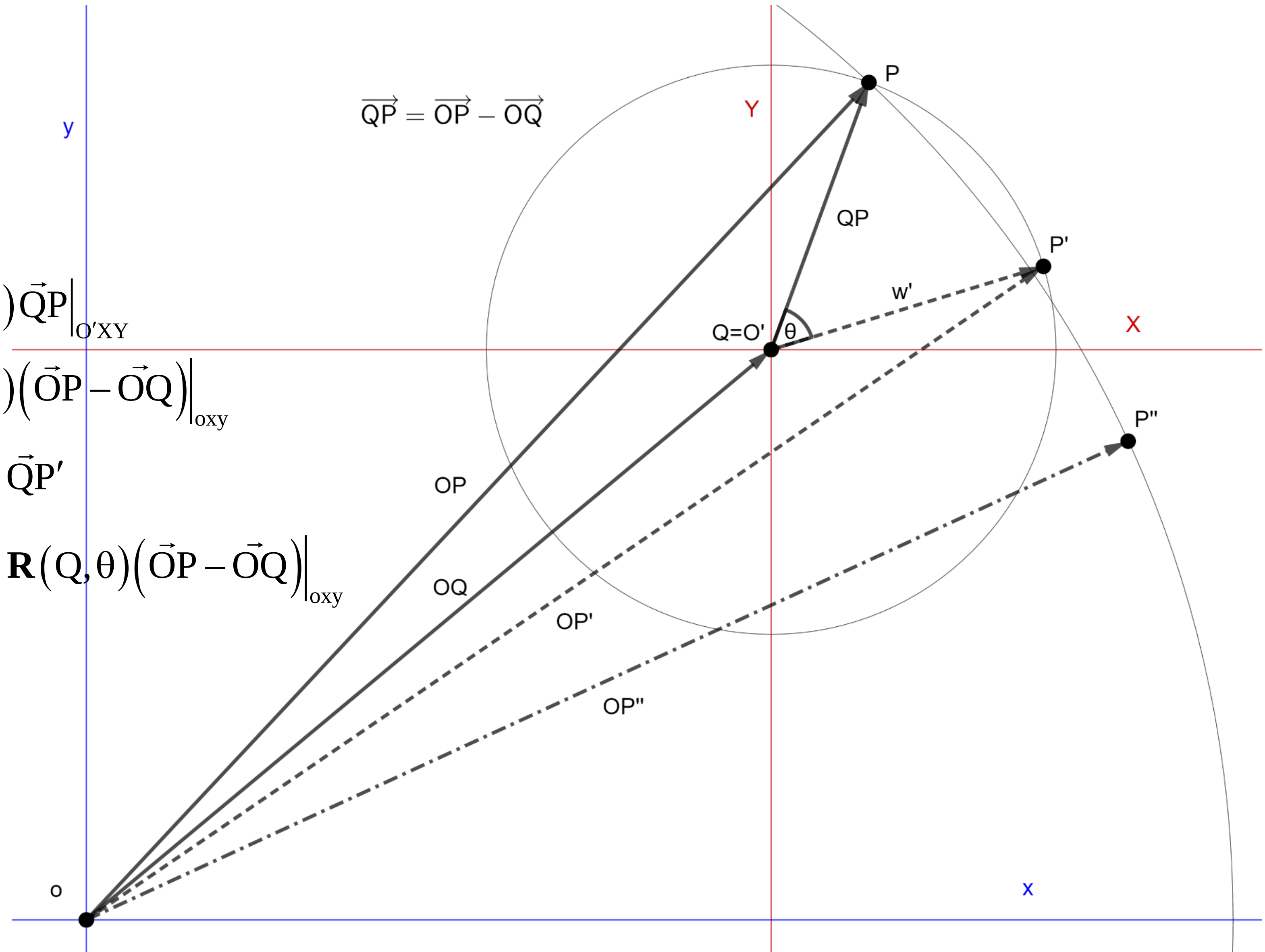
$$\vec{Q}P = \vec{O}P - \vec{O}Q$$

$$\vec{Q}P'\Big|_{O'XY} = \mathbf{R}(Q,\theta)\vec{Q}P\Big|_{O'XY}$$

$$\vec{Q}P'\Big|_{O'XY} = \mathbf{R}(Q,\theta)\left(\vec{O}P - \vec{O}Q\right)\Big|_{oxy}$$

$$\vec{O}P'\Big|_{oxy} = \vec{O}Q\Big|_{oxy} + \vec{Q}P'$$

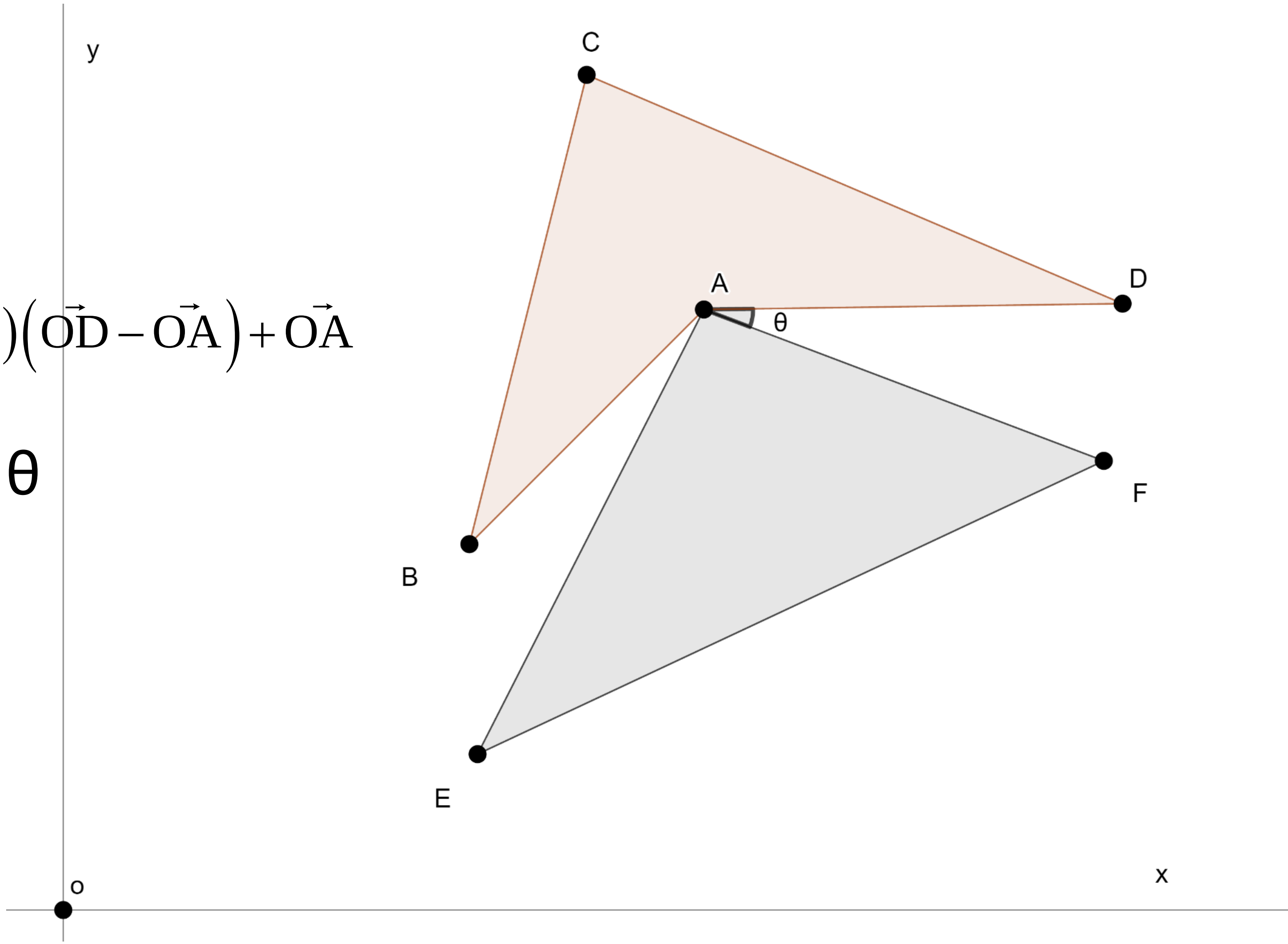
$$\vec{O}P'\Big|_{oxy} = \vec{O}Q\Big|_{oxy} + \mathbf{R}(Q,\theta)\left(\vec{O}P - \vec{O}Q\right)\Big|_{oxy}$$



Rotating a vector (angle unknown)

$$\vec{OF} = \mathbf{R}(A,\theta)\left(\vec{OD} - \vec{OA}\right) + \vec{OA}$$

Solve for θ



Bezier curve in matrix form

Second order

$$\mathbf{P} = \begin{pmatrix} \mathbf{p_0} \\ \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix}; \quad \mathbf{p}_\mathrm{i} = \begin{pmatrix} \mathrm{p_i}(\mathrm{x}) \\ \mathrm{p_i}(\mathrm{y}) \end{pmatrix}, \quad \mathrm{i} = 0,1,2$$

$$\mathbf{U} = \begin{pmatrix} \mathrm{t}^2 & \mathrm{t} & 1 \end{pmatrix}$$

$$\mathbf{M} = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Write for third and fourth order

Material

Mathematics for Computer Graphics Applications, M.E. Mortenson

Geometric modeling in Engineering, Antti H. Niemi.