

Rio de Janeiro

2024.1

CIV2802 – Sistemas Gráficos para Engenharia



Luiz Fernando Martha

André Pereira



Objetivos e escopo

Introdução a conceitos e algoritmos de **Computação Gráfica** para alunos que pretendem **desenvolver** e **customizar** software técnico-científico.

» Fundamentos de Computação Gráfica

» Ferramentas e bibliotecas

– Criação de aplicações simples usando Visual Studio e Qt.

– Programação Orientada a Objetos em C++: primitivas gráficas (pontos, curvas e polígonos).

– Sistema gráfico OpenGL com exemplos de visualização de primitivas gráficas.

– Introdução a Geometria Computacional: interseção de curvas, teste de inclusão de pontos em polígonos, representação digital de curvas.

» Modelagem Geométrica

– Motivação: modelador gráfico x modelador geométrico.

– Estruturas de dados topológicas.

– Aplicações com a estrutura de dados Half-edge.

» Geração de Malhas

– Estratégias e algoritmos.

– Uso de bibliotecas.

» Visualização Científica

– Pós-processamento de resultados de elementos finitos.

Aula			Assunto	Trabalho (aplicação)
1	08/mar	6ª f	Apresentação do ambiente de desenvolvimento – Python, Visual Studio Code e Qt. Desenvolvimento de programas simples com o ambiente: <i>hello world</i> e soma de dois números. Introdução a Computação Gráfica para Engenharia.	
2	15/mar	6ª f	Desenvolvimento de uma calculadora RPN (console) em Python. Introdução a Programação Orientada a Objetos. Modelagem Orientada a Objetos da calculadora RPN.	Trabalho 1: Calculadora RPN GUI com Qt
3	22/mar	6ª f	Visualização bidimensional com OpenGL. Programa gráfico simples com Qt e OpenGL. Transformações geométricas 2D. Transformação Window-Viewport.	
	29/mar	6ª f	FERIADO SEMANA SANTA	
4	05/abr	6ª f	Tratamento de cor. Padrão/biblioteca gráfica OpenGL. Programação em um ambiente gráfico interativo dirigido por eventos. Paradigma de signals & slots do Qt.	Trabalho 2: Visualização bidimensional com Qt e OpenGL
5	12/abr	6ª f	Eventos de mouse em canvas. Representações digitais de curvas.	
6	19/abr	6ª f	Representações digitais de curvas (continuação)	Trabalho 3: Coleta interativa de curvas
	26/abr	6ª f	AULA CANCELADA	
7	03/mai	6ª f	Introdução à Geometria Computacional. Tesselagem de regiões bidimensionais.	
8	10/mai	6ª f	Introdução à Geometria Computacional. Algoritmos de interseção de segmentos de reta. Teste de proximidade e inclusão de pontos em regiões.	Trabalho 4: Modelador geométrico com interseções de curvas e criação explícita de regiões
9	17/mai	6ª f	Geração de malhas de elementos finitos: algoritmos de mapeamento, algoritmos de avanço de fronteiras e algoritmos de triangulação Delaunay.	Trabalho 5: Geração de malhas
10	24/mai	6ª f	Transformações geométricas para visualização 3D e modelo de câmera.	
	31/mai	6ª f	RECESSO DE CORPUS CHRISTI	
11	07/jun	6ª f	Transformações geométricas para controle de visualização 3D.	Trabalho 6: Visualização 3D
12	14/jun	6ª f	Modelagem de sólidos manifold e de subdivisões planares. Estruturas de dados topológicas manifold. Estrutura de dados Half-edge.	
			Aulas subsequentes dedicadas aos projetos finais	

CIV2802 - Sistemas Gráficos para Engenharia - 2024.1



[Prof. Luiz Fernando Martha](#)

e-mail: lfm@tecgraf.puc-rio.br

[Pontifícia Universidade Católica do Rio de Janeiro - PUC-Rio - Departamento de Engenharia Civil e Ambiental](#)

[Prof. André Maués Brabo Pereira](#)

e-mail: andremaues@id.uff.br

[Universidade Federal Fluminense - UFF](#)

[[Horário](#) | [Geral](#) | [Aulas](#) | [Trabalhos](#) | [Programa](#) | [Cronograma](#) | [Aprovação](#) | [Bibliografia](#)]

Horário

- Sexta-feira: 13:00-16:00 hs - Aulas *online* no [Ambiente de Aprendizagem Online da PUC-Rio](#).
-

Geral

- [Programa da disciplina](#).
 - [Instruções para instalação do ambiente de desenvolvimentos de programas gráficos em Python: Python 3.9 e Visual Studio Code](#).
-

Aulas

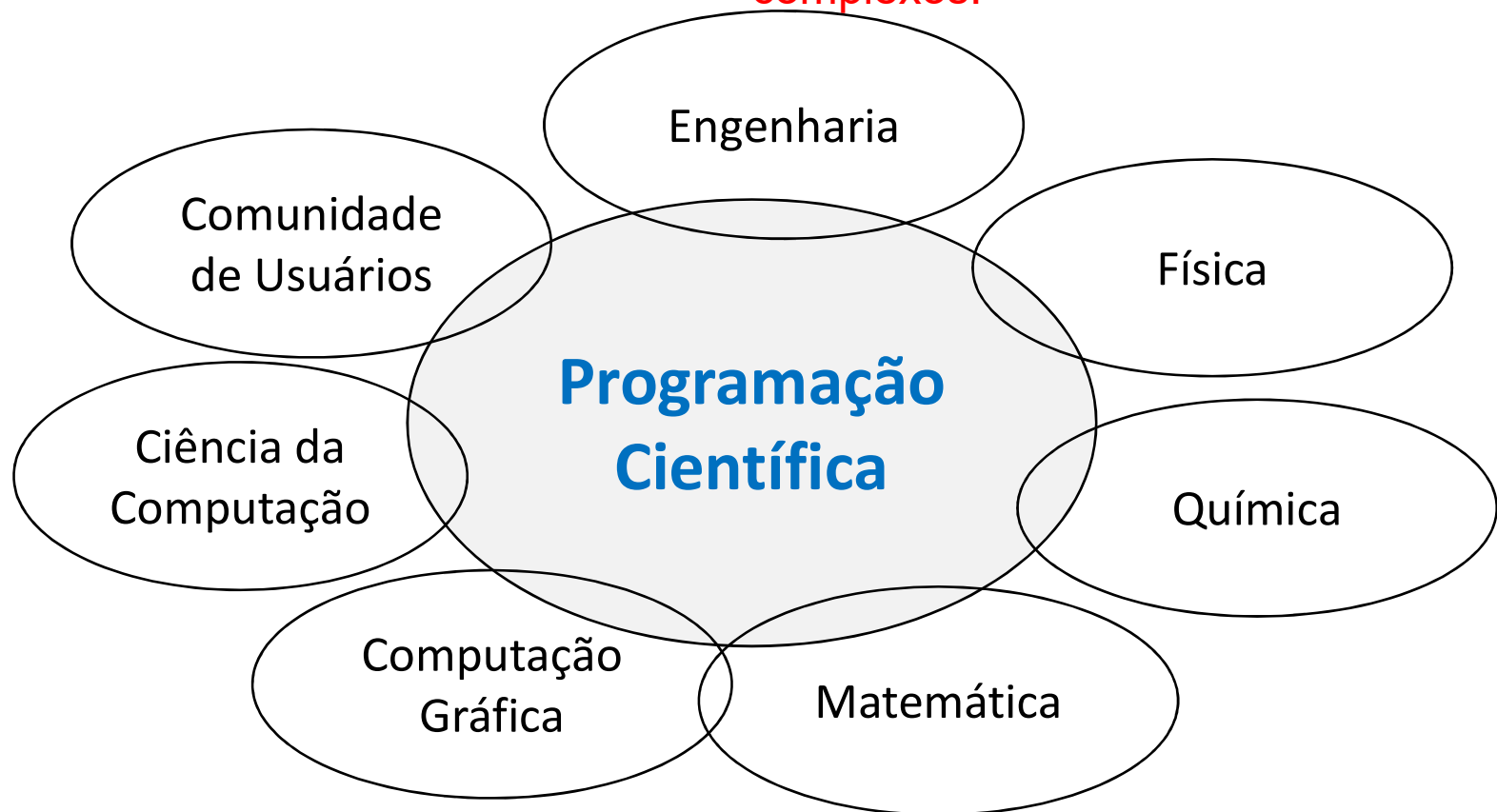
- Aula 01 - 08/mar/2024 - Apresentação do ambiente de desenvolvimento e do curso.
[Apresentação do ambiente de desenvolvimento – Python, Visual Studio Code e Qt](#).
Desenvolvimento de programas simples com o ambiente:
Programa muito simples "Hello World" em Python usando Qt 5: [Instruções para criação do programa](#).
Programa simples em Python usando Qt 5 para efetuar a soma de dois números: [Instruções para criação do programa](#).

Interesses em Modelagem Científica

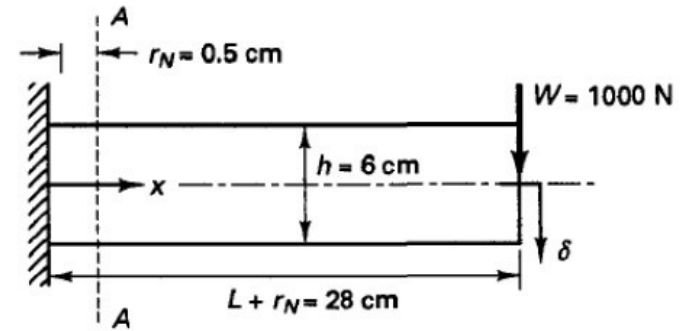
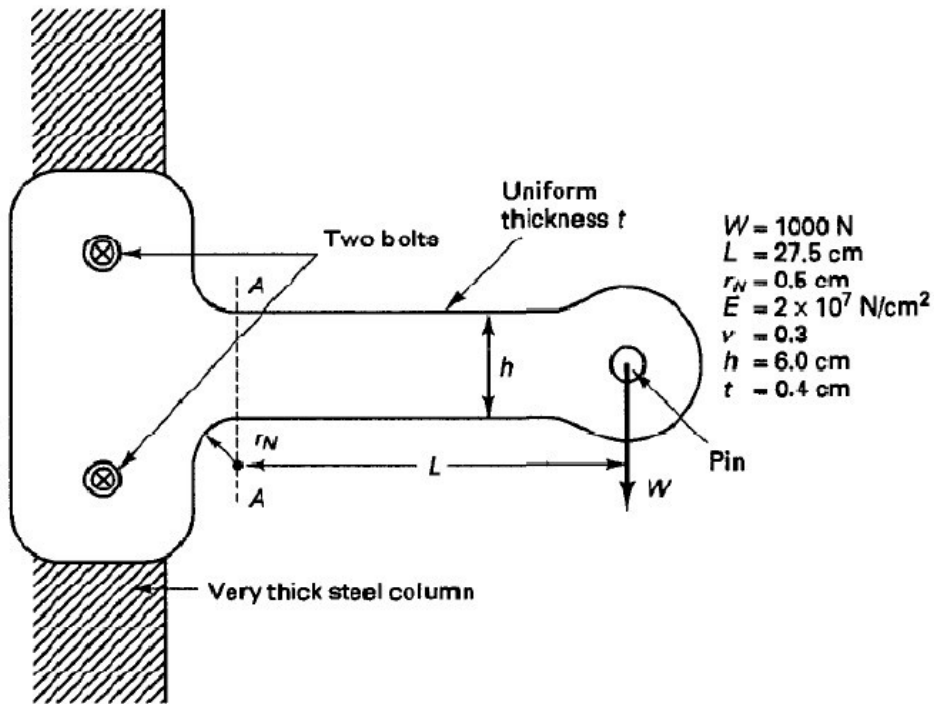
Modelagem Computacional → meio de substituir medições diretas ou experimentos em laboratórios.

&

Simulações Numéricas → meio de implementar modelos complexos.



Modelagem com Métodos Numéricos



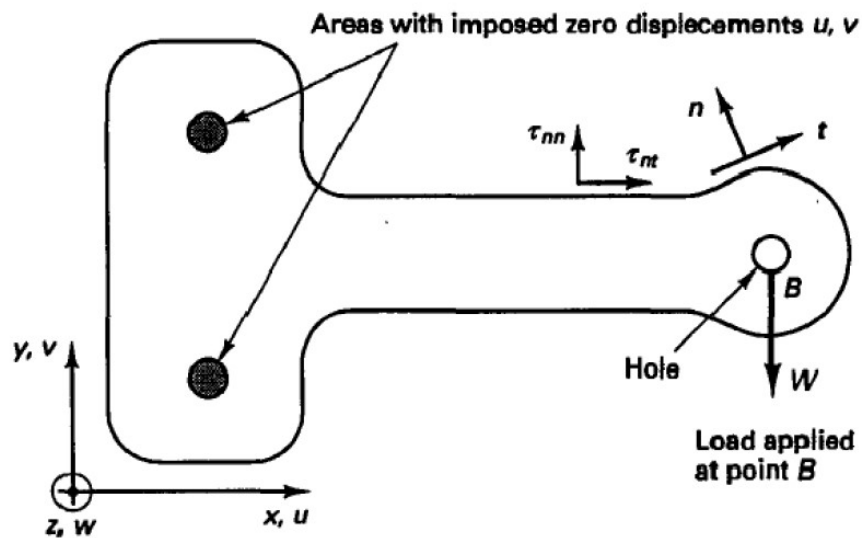
$$M = WL$$

$$= 27,500 \text{ N cm}$$

$$\delta|_{\text{at load } W} = \frac{1}{3} \frac{W(L + r_N)^3}{EI} + \frac{W(L + r_N)}{\frac{5}{8}AG}$$

$$= 0.053 \text{ cm}$$

Fonte: Bathe, 1996. Finite Element Procedures



$$\delta|_{\text{at load } W} = 0.064 \text{ cm}$$

$$M|_{x=0} = 27,500 \text{ N cm}$$

Equilibrium equations (see Example 4.2)

$$\left. \begin{aligned} \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} &= 0 \\ \frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} &= 0 \end{aligned} \right\} \text{in domain of bracket}$$

$\tau_{nn} = 0, \tau_{nt} = 0$ on surfaces except at point *B* and at imposed zero displacements

Stress-strain relation (see Table 4.3):

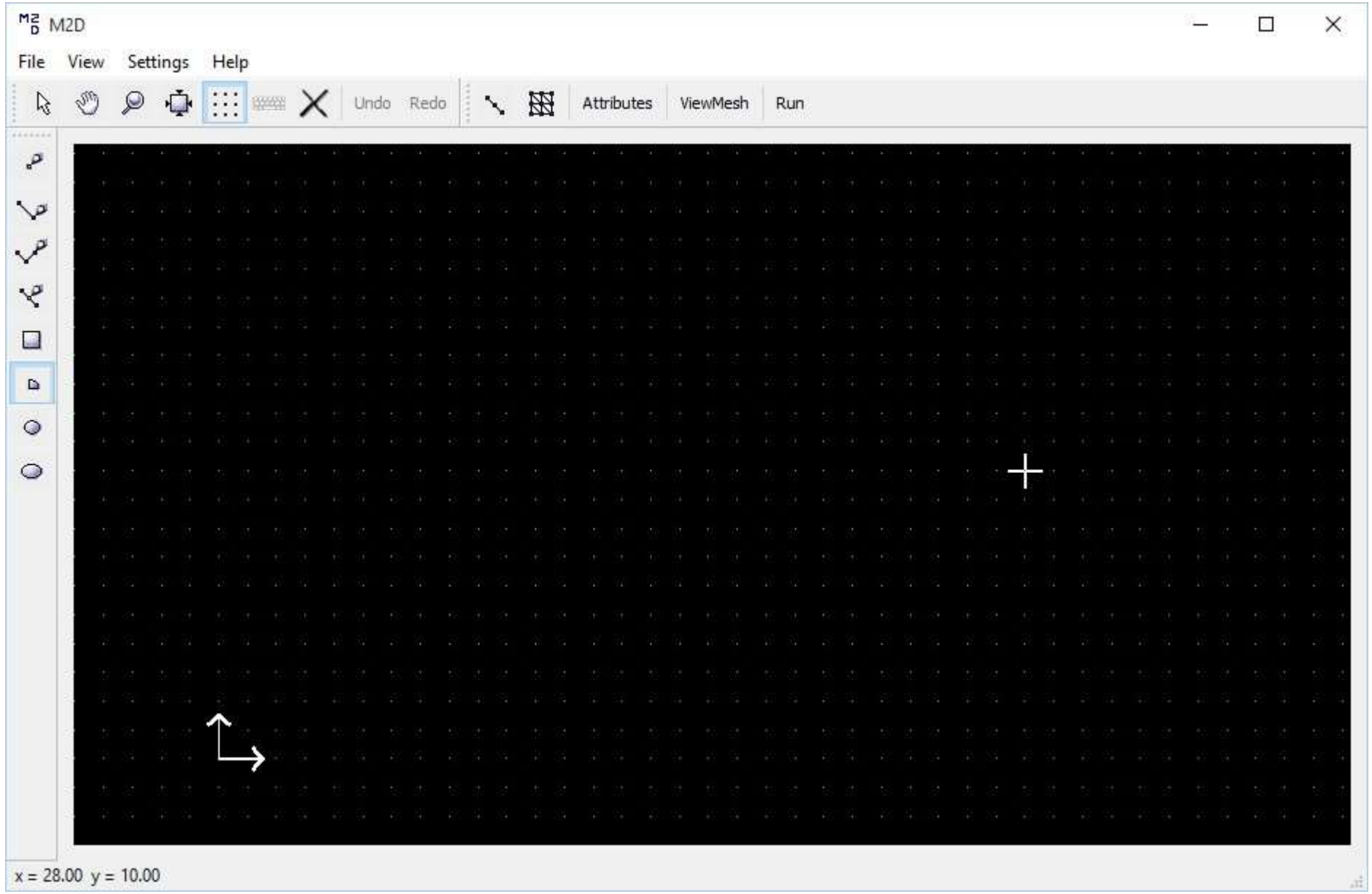
$$\begin{bmatrix} \tau_{xx} \\ \tau_{yy} \\ \tau_{xy} \end{bmatrix} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1 - \nu)/2 \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix}$$

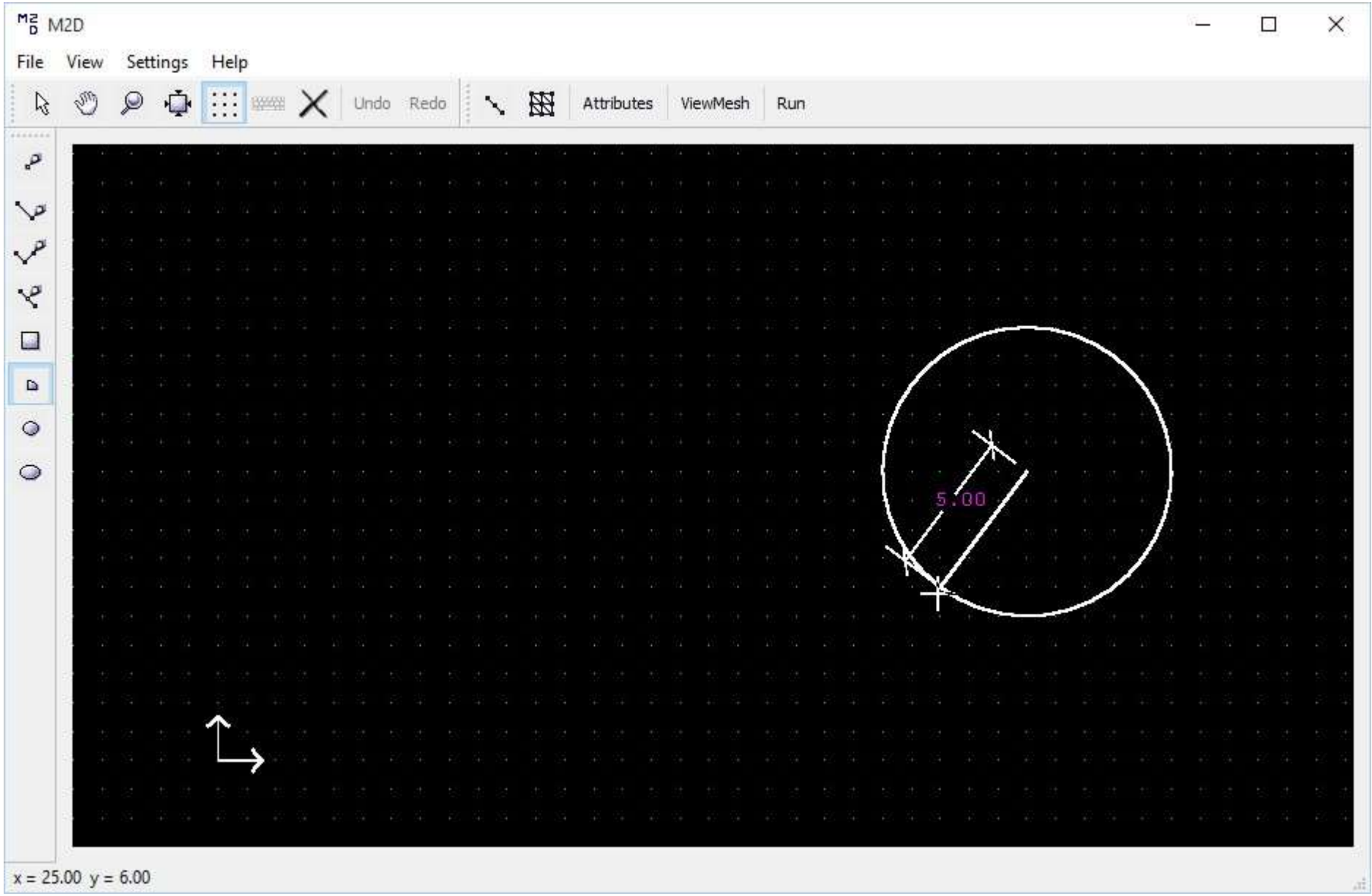
E = Young's modulus, ν = Poisson's ratio

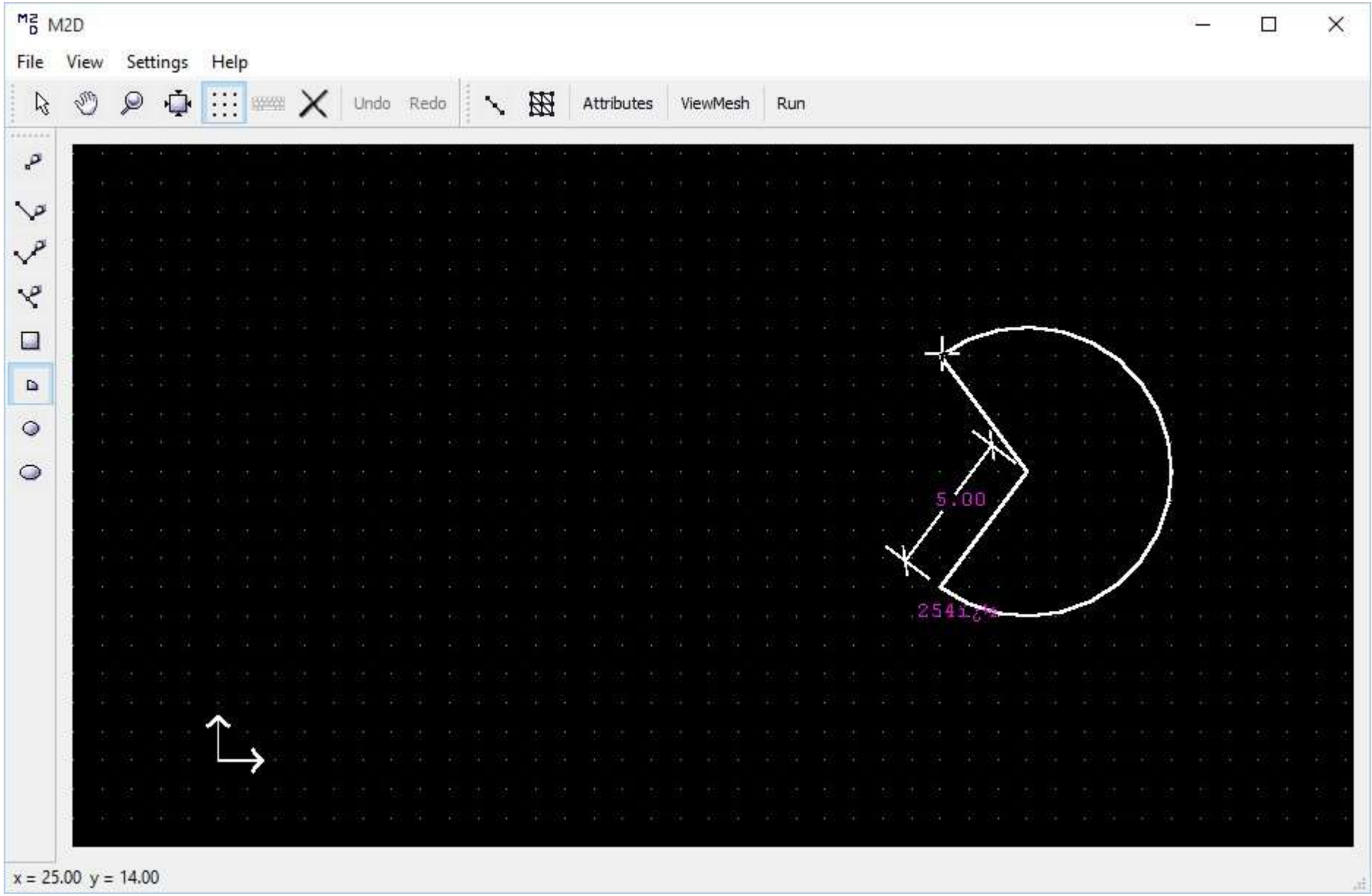
Strain-displacement relations (see Section 4.2):

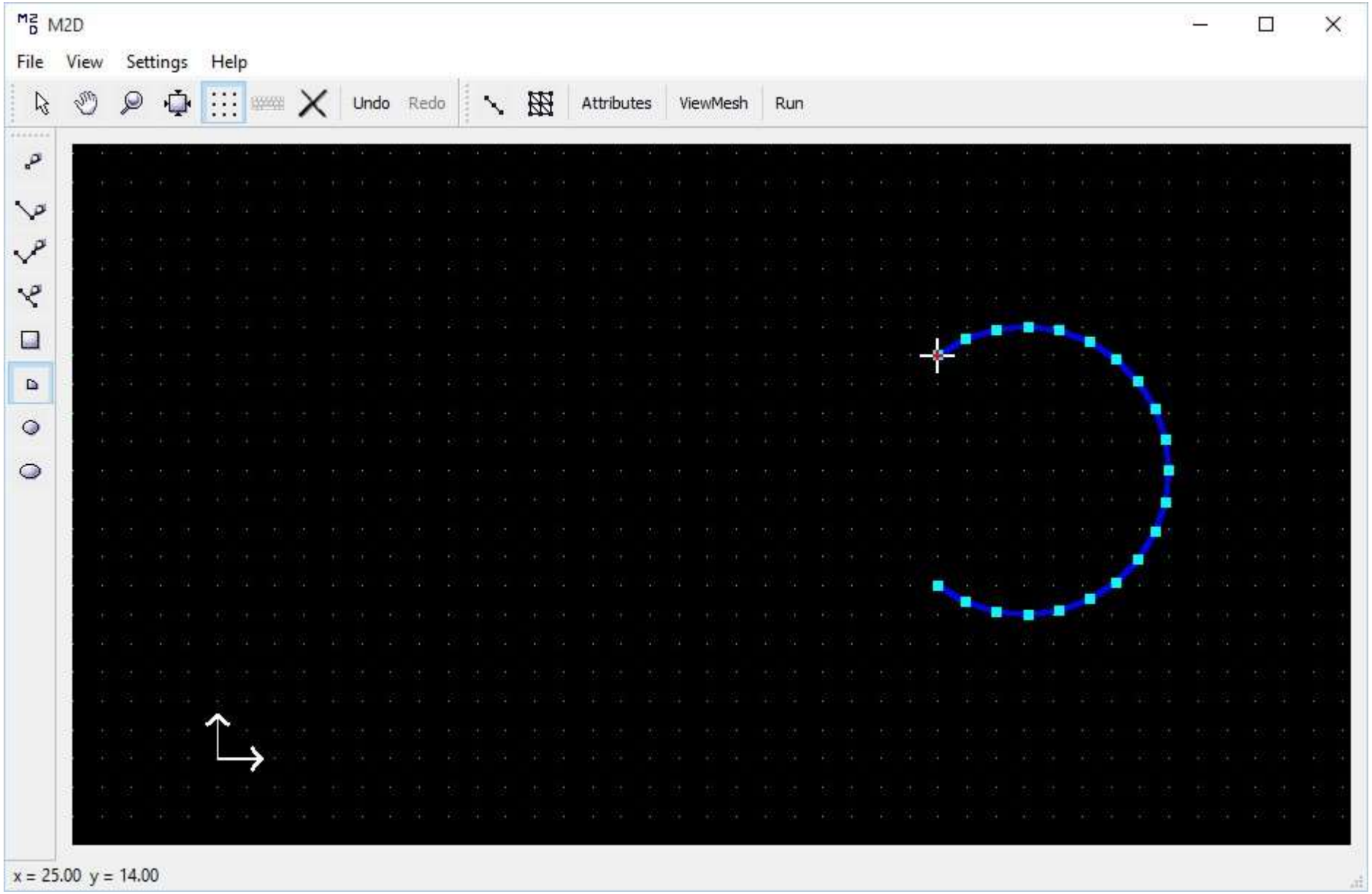
$$\epsilon_{xx} = \frac{\partial u}{\partial x}; \quad \epsilon_{yy} = \frac{\partial v}{\partial y}; \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

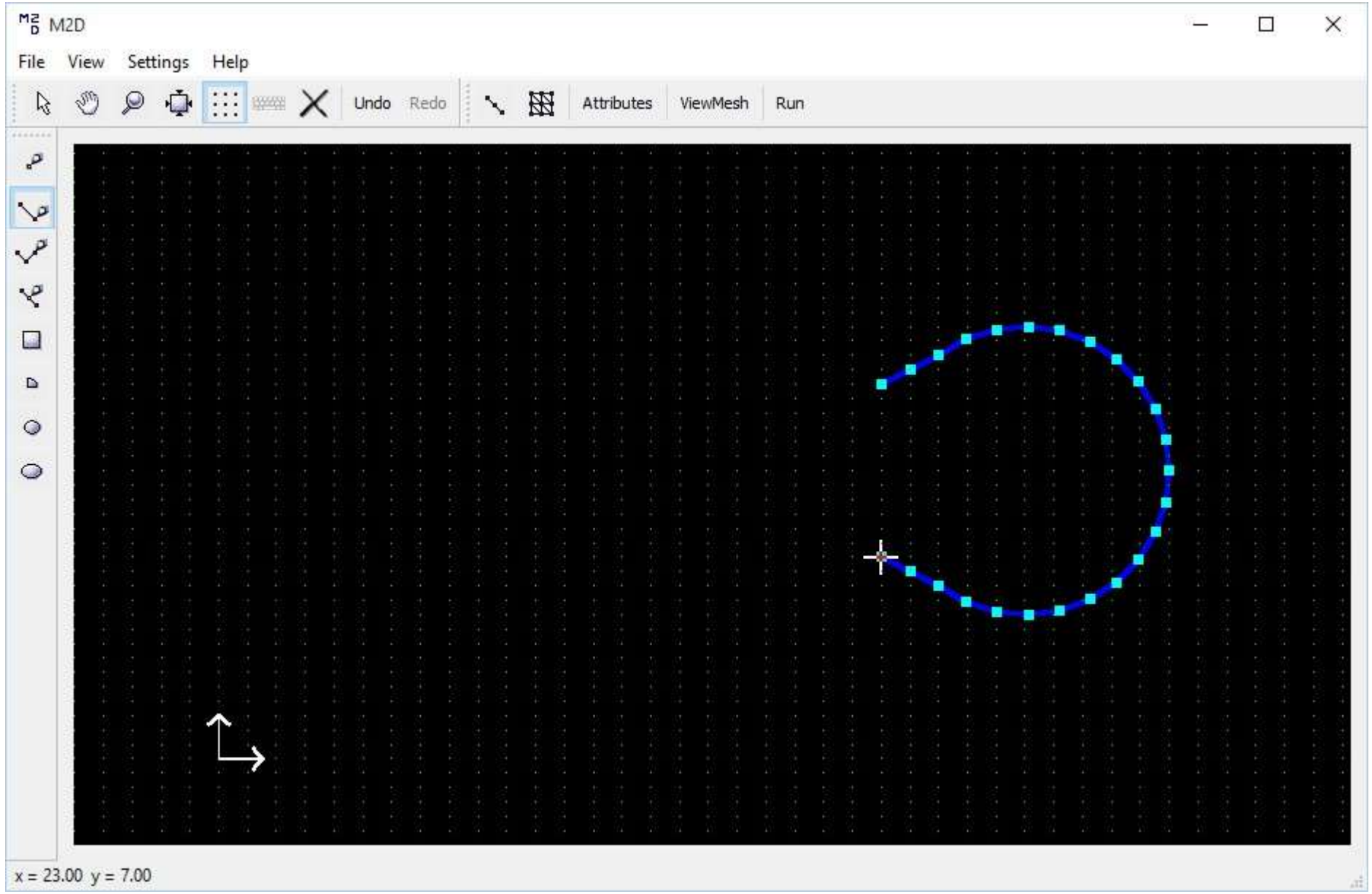
Fonte: Bathe, 1996. Finite Element Procedures

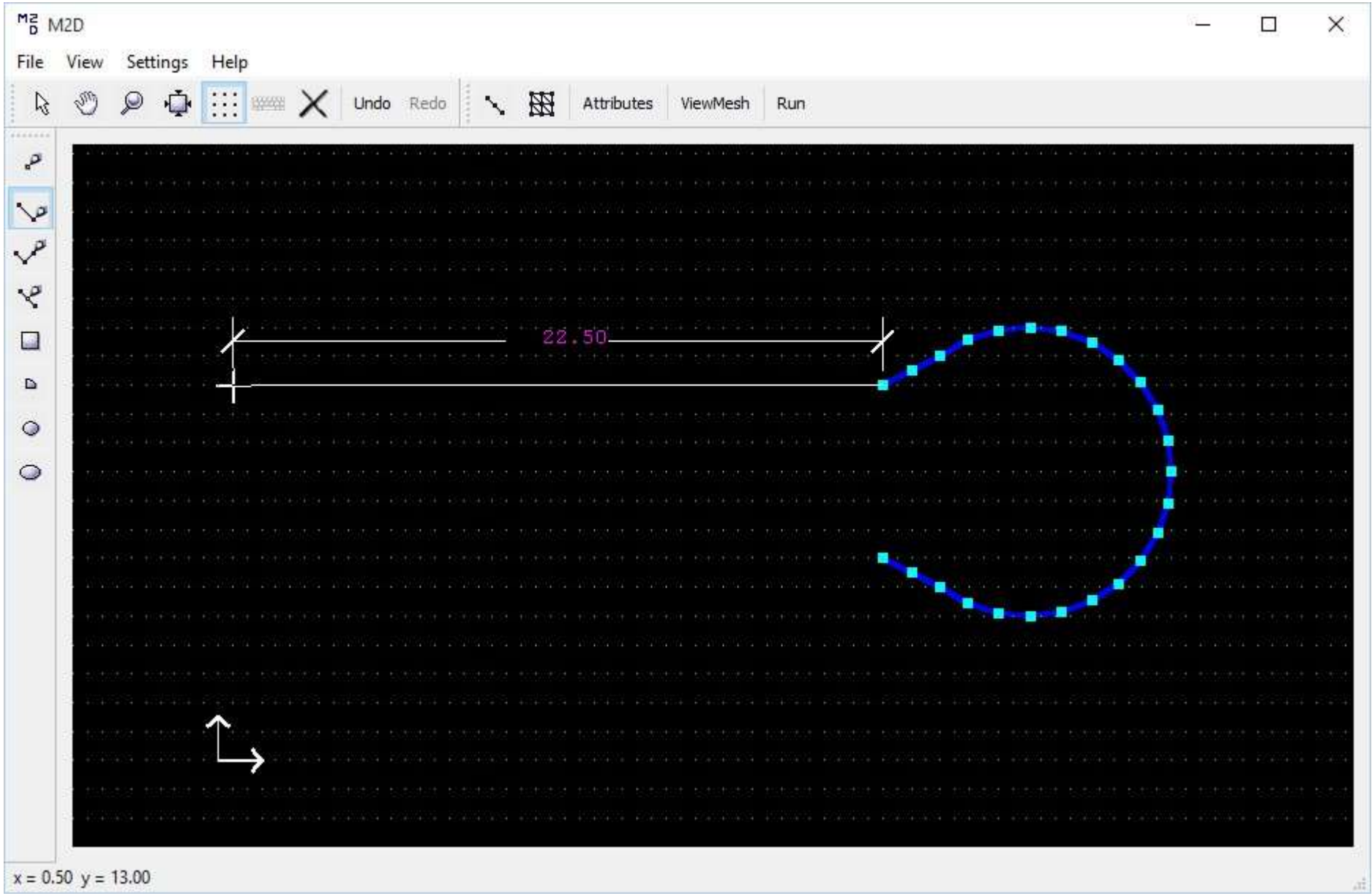


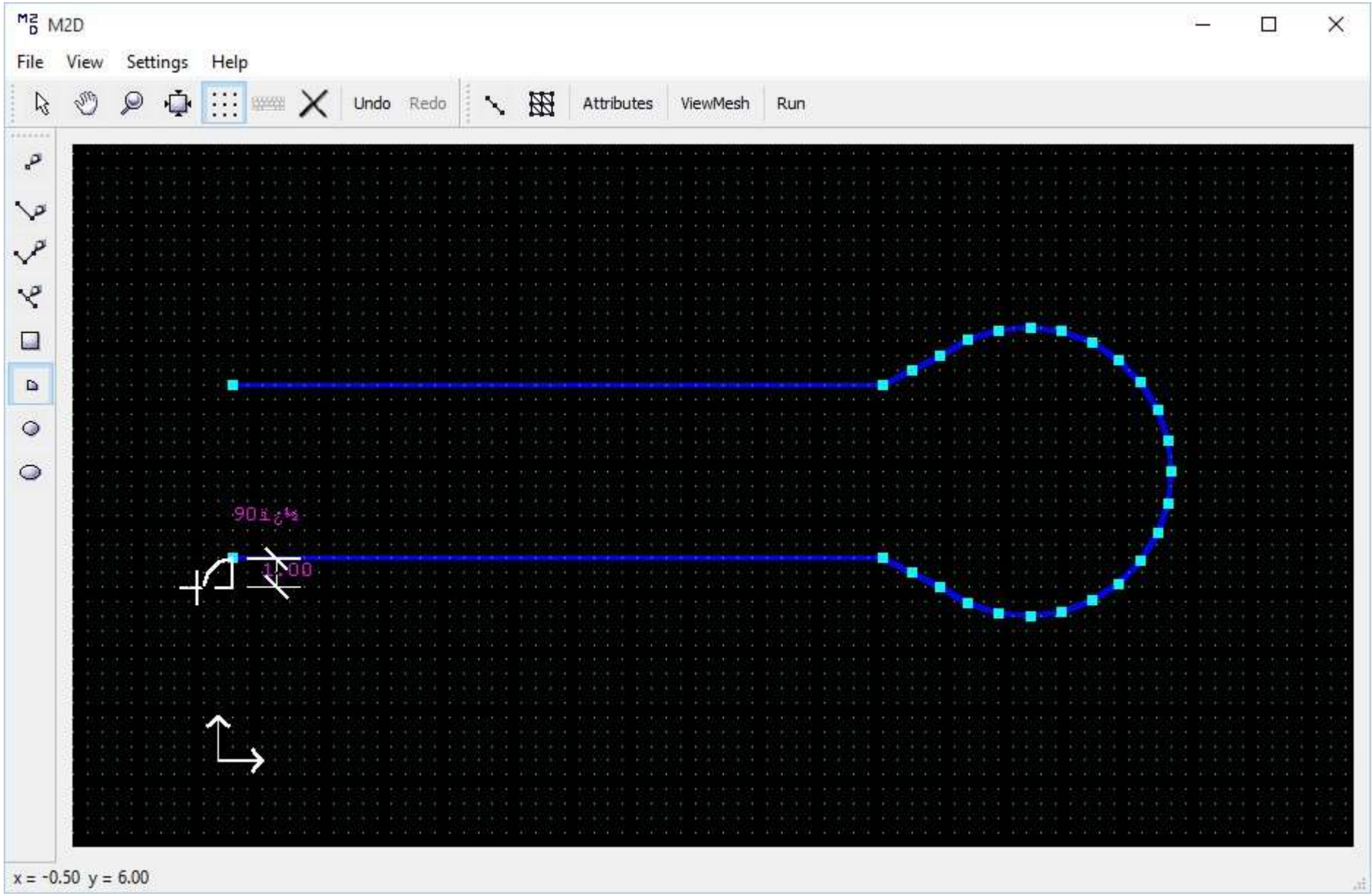


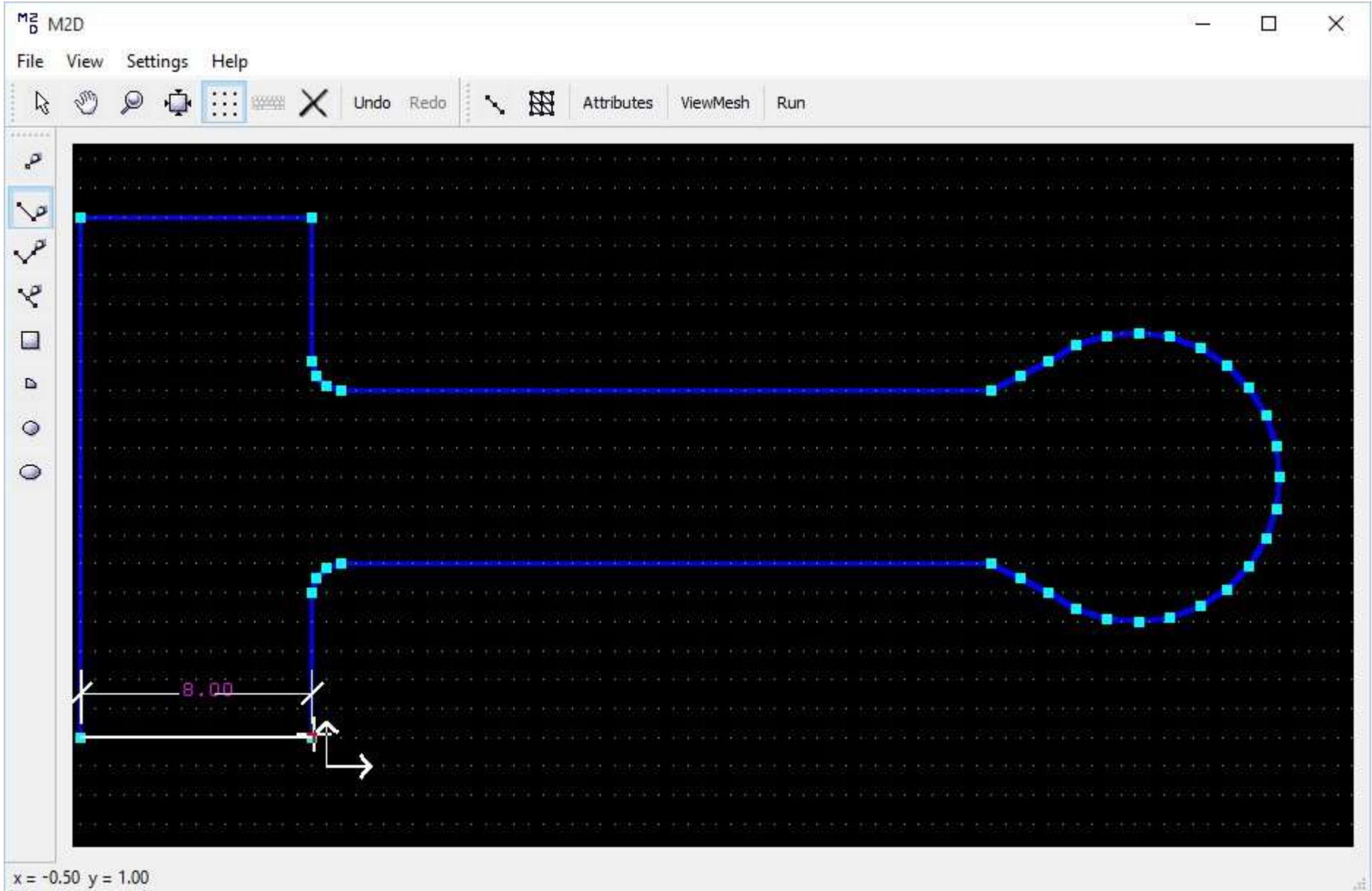


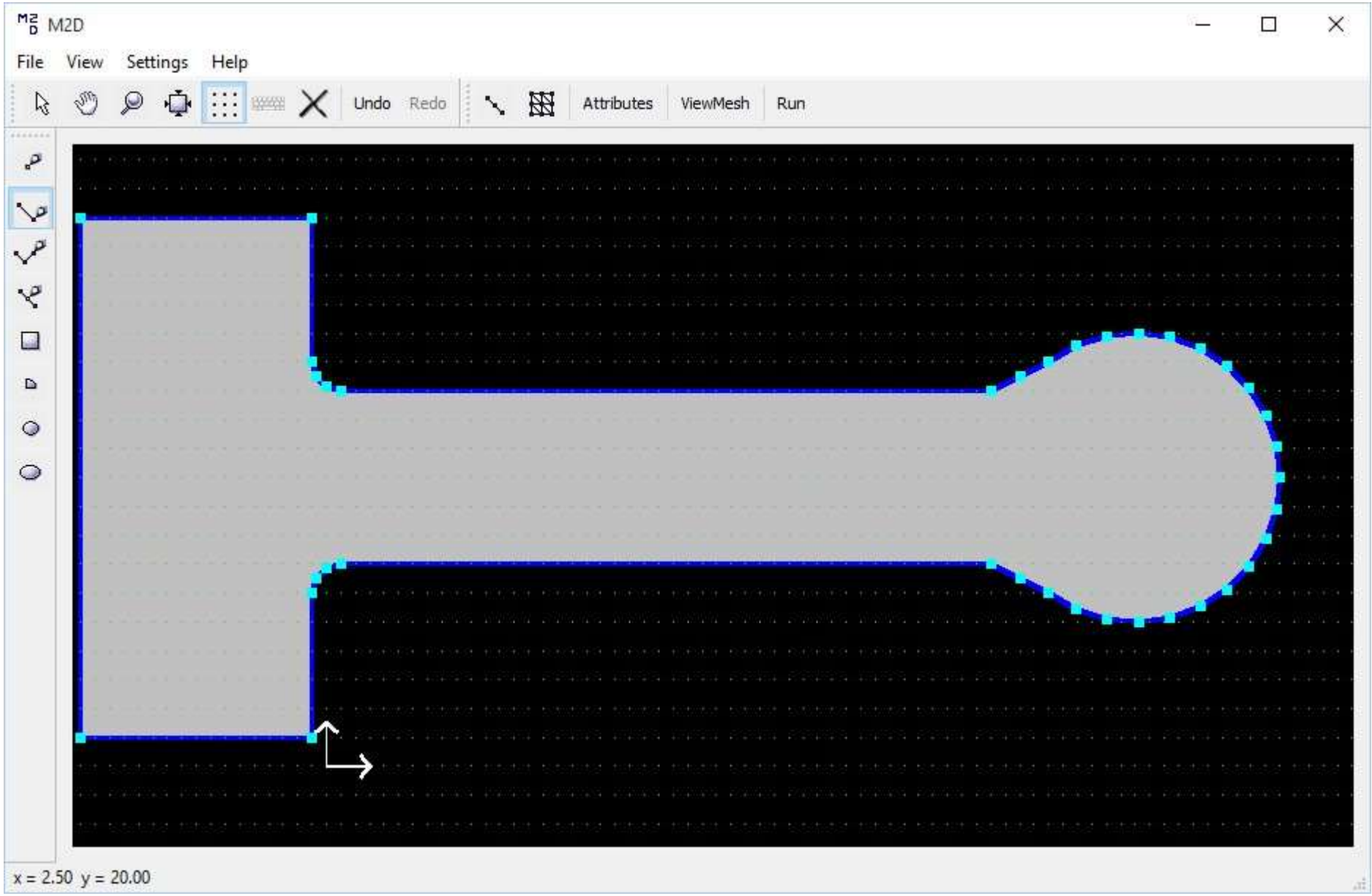


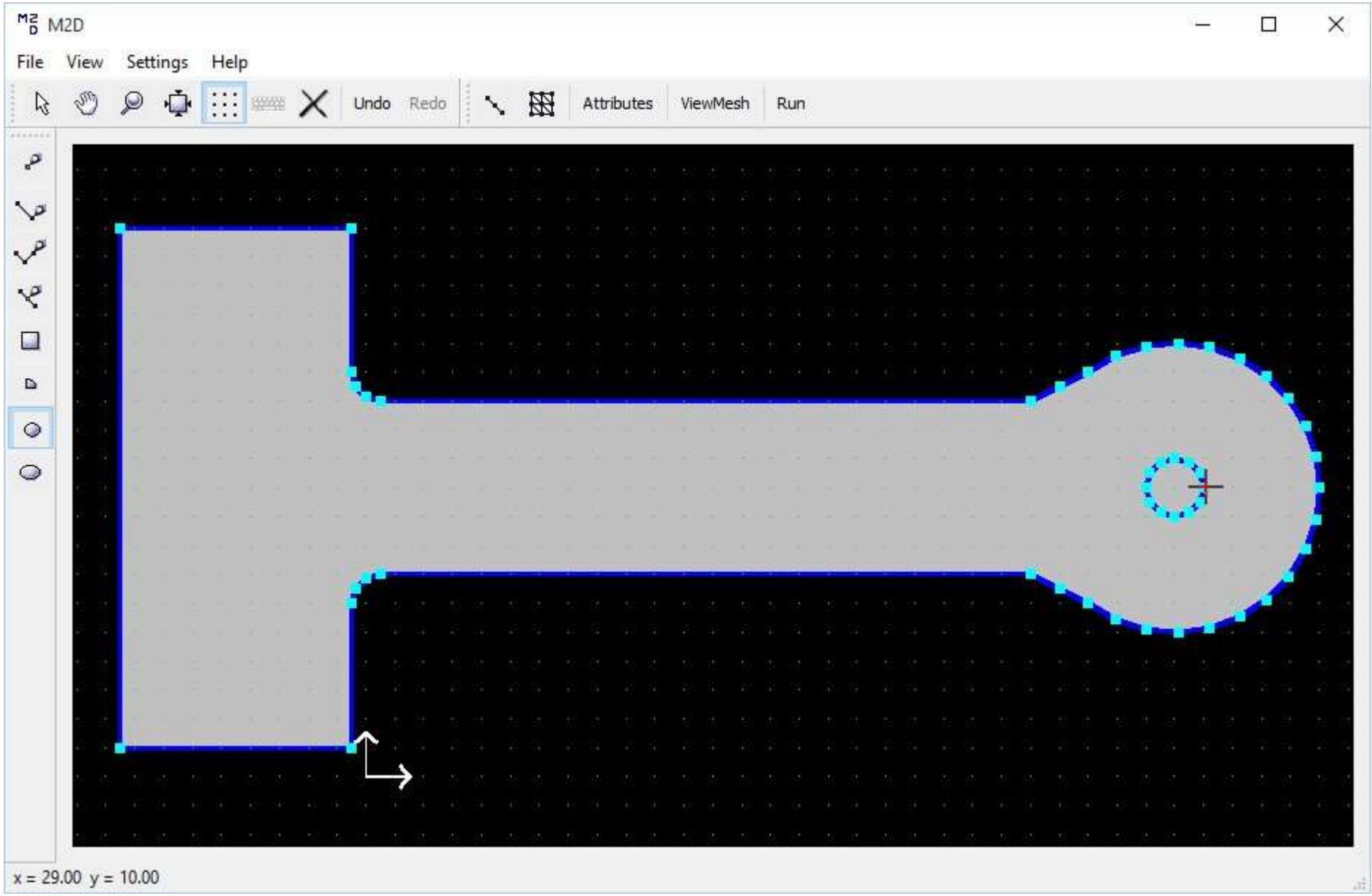


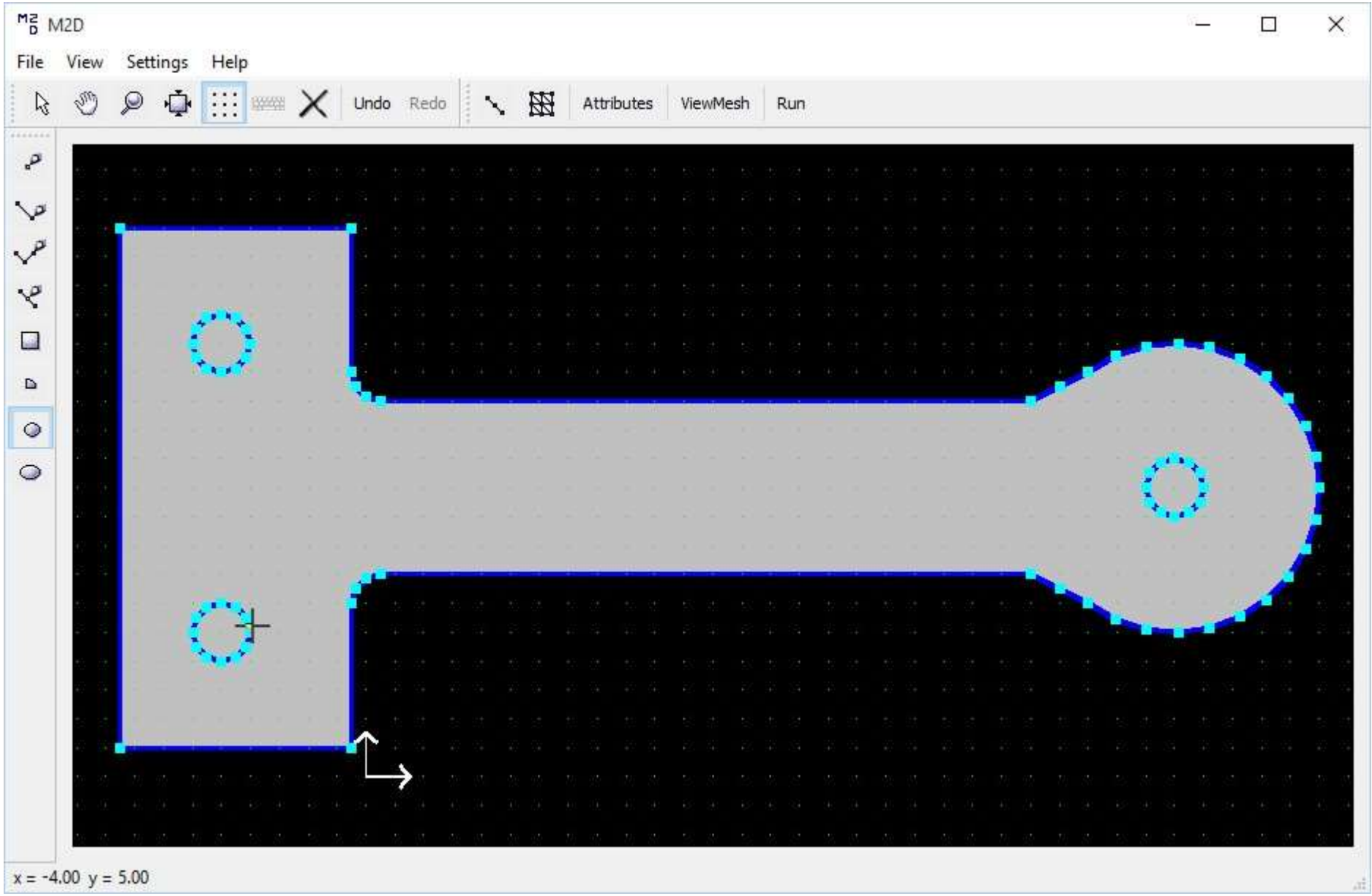


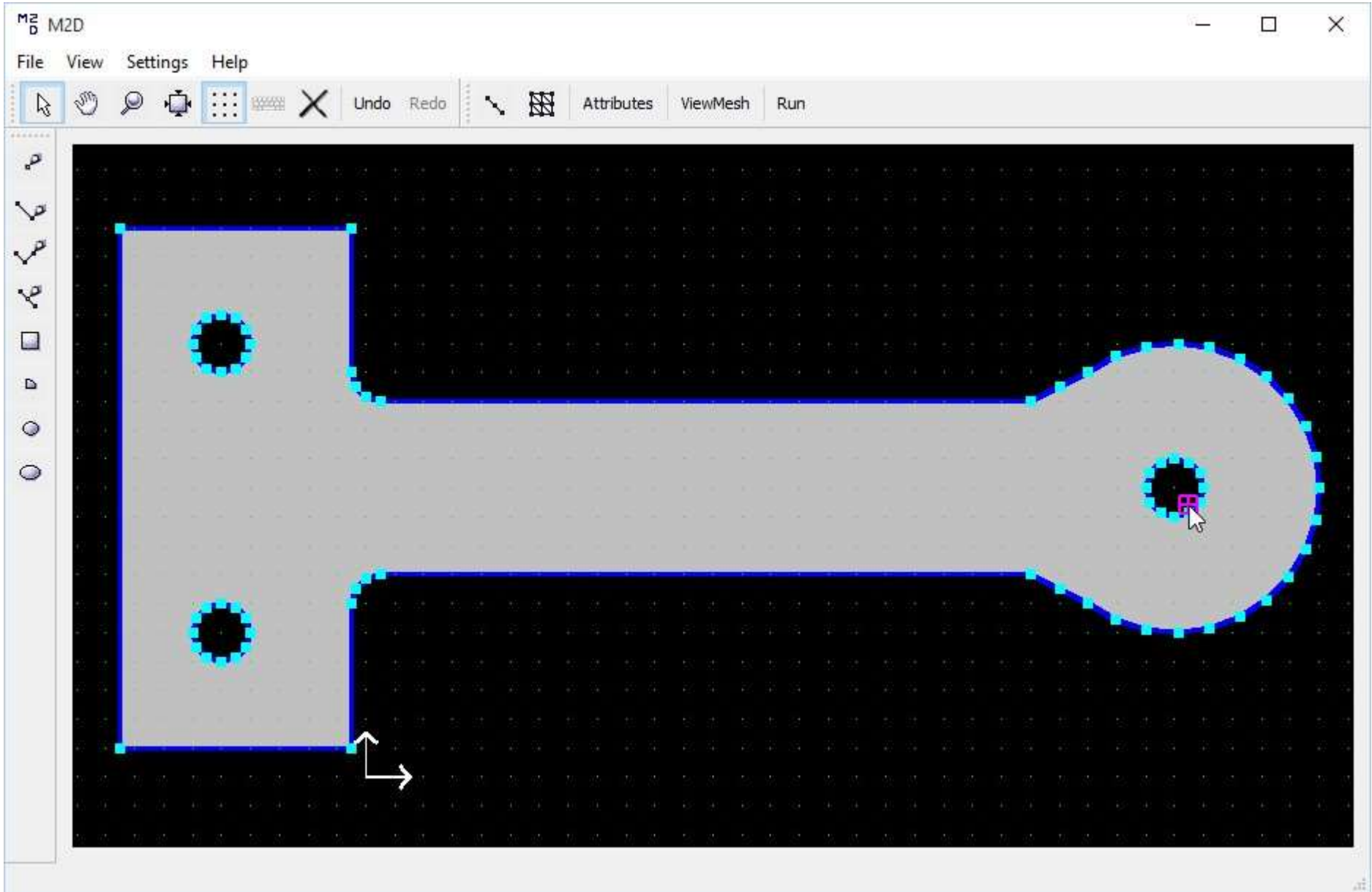


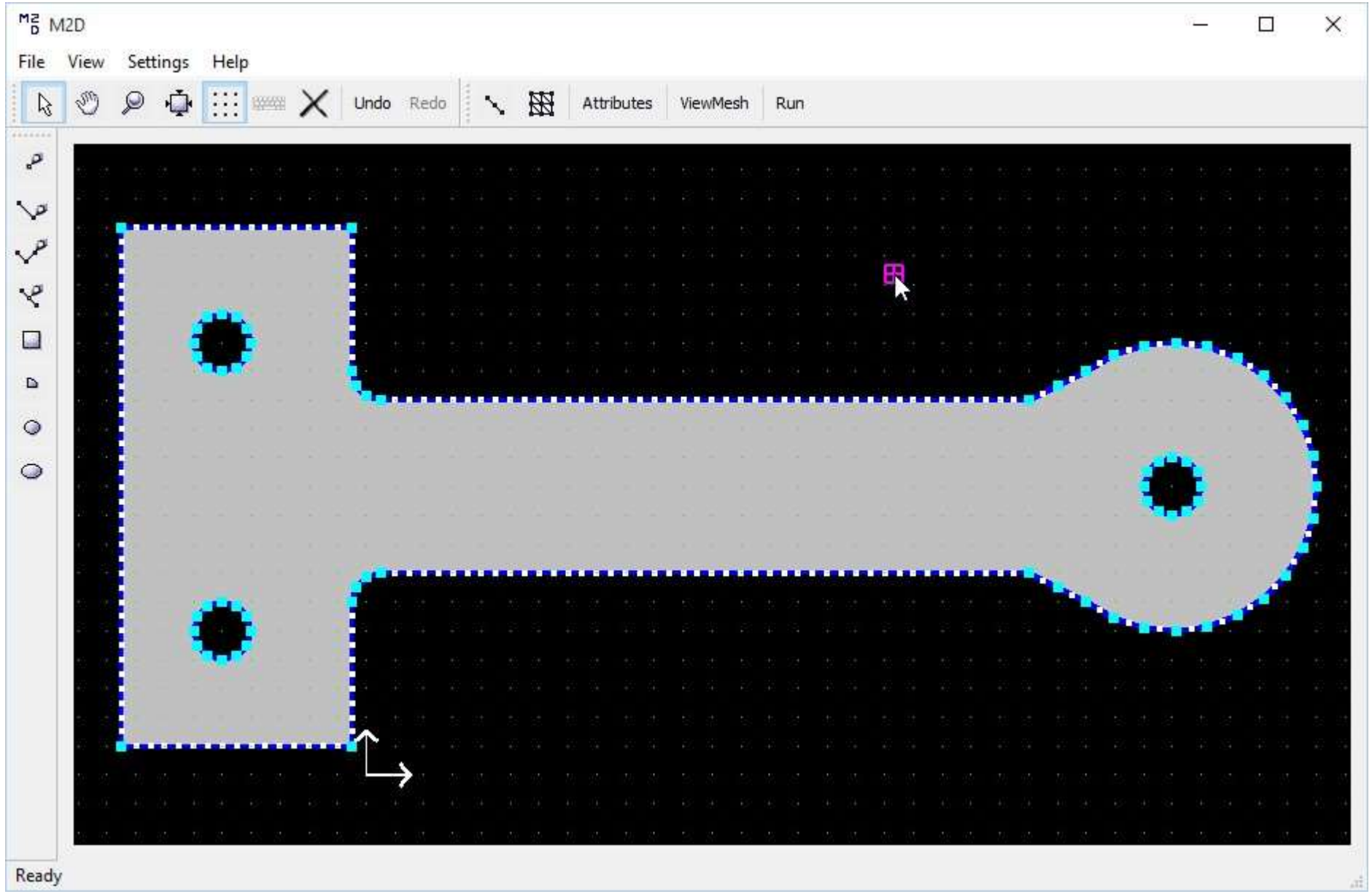


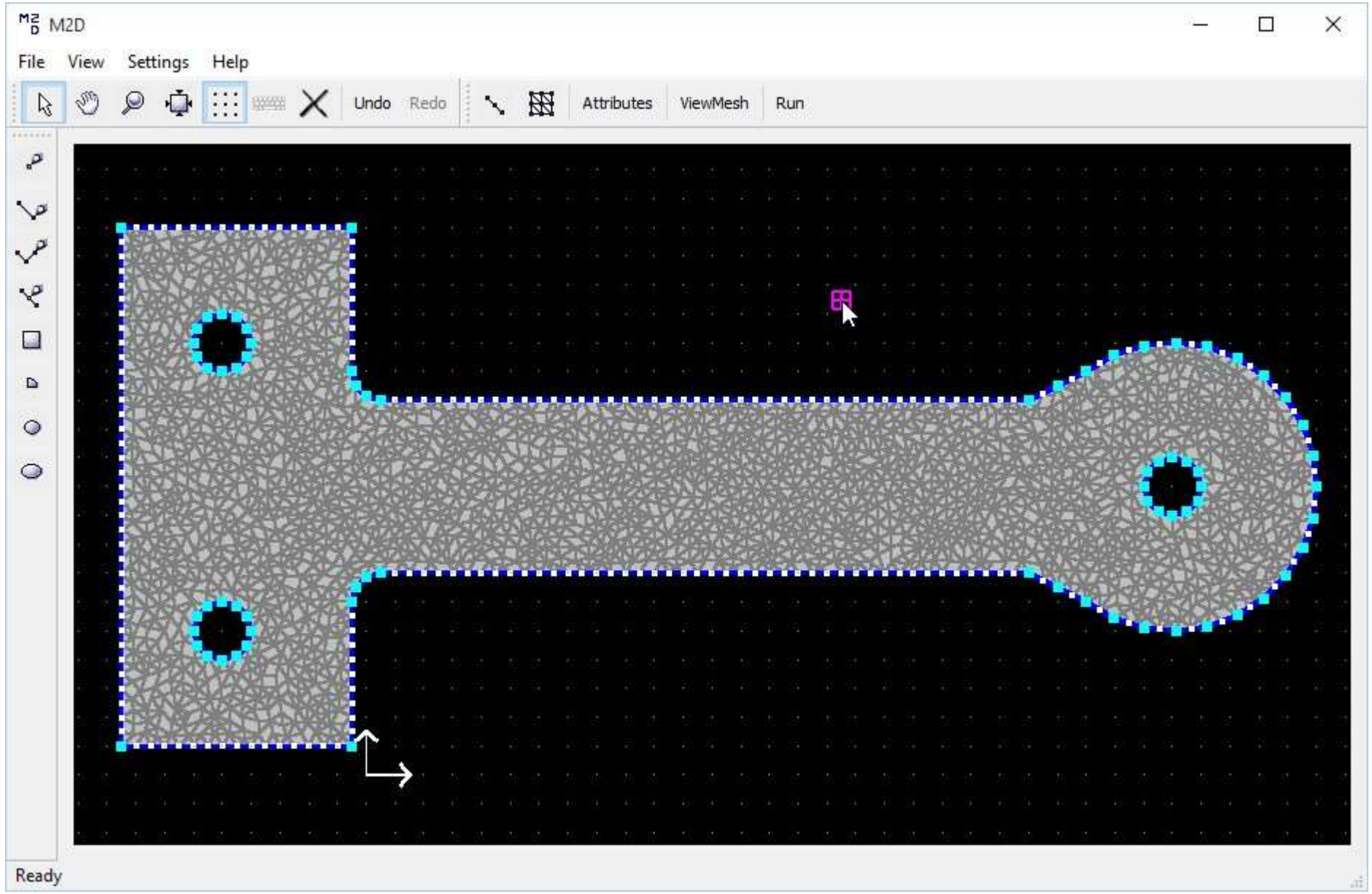


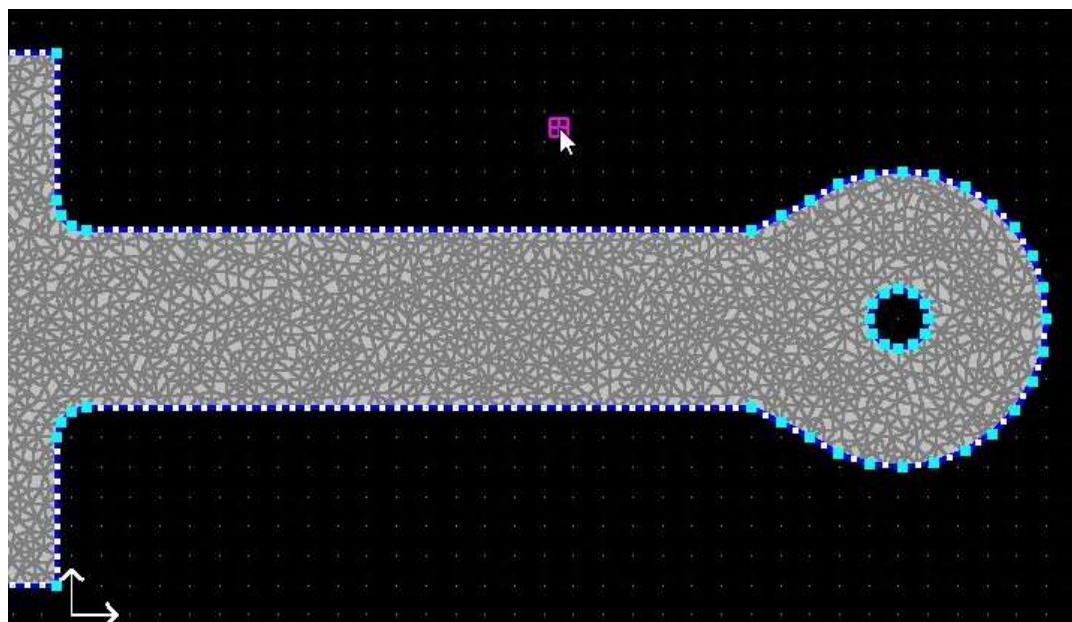
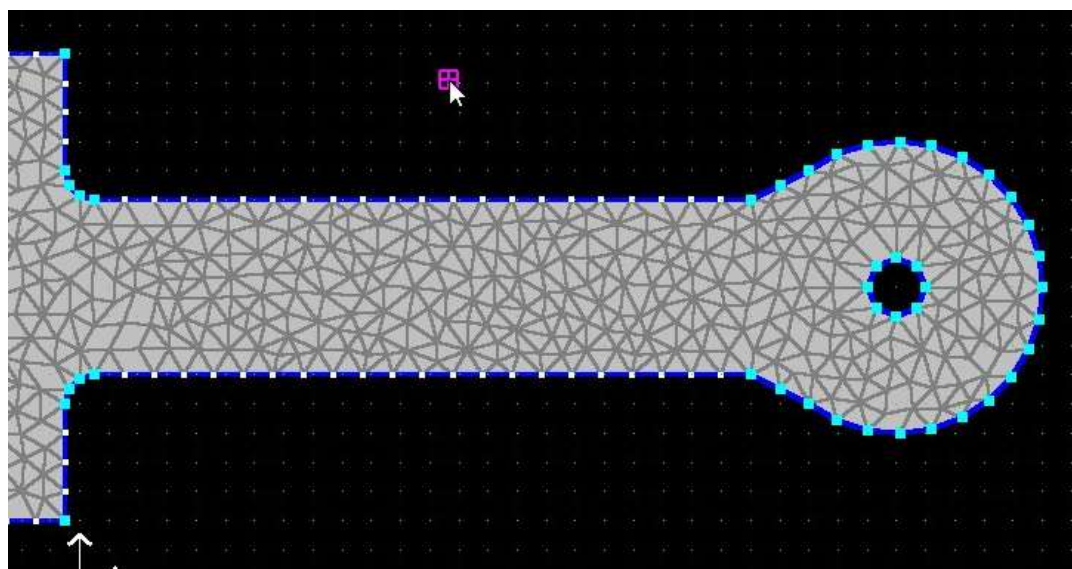


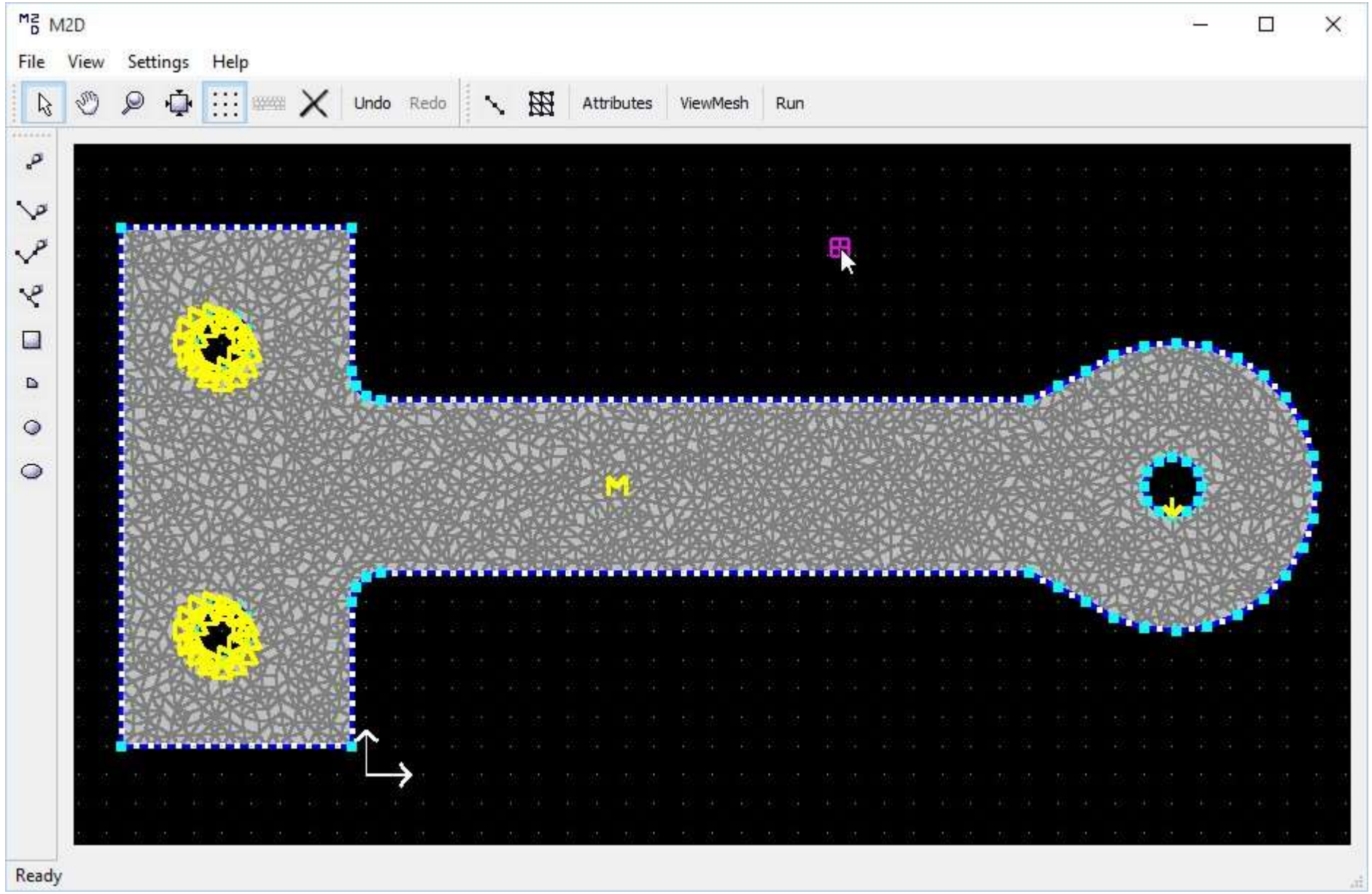


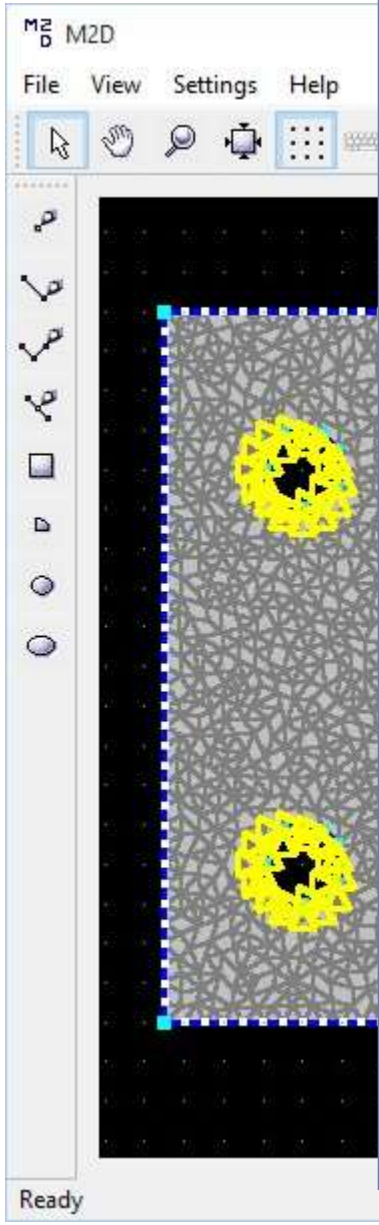












M2D Attribute Manager

Prototypes
Material Create

Attributes
Material_1
Force_1
Restrain_1

Property Editor

Property	Value
1 Conductivity	0
2 PoissonsRatio	0.3
Thickness	0.4
YoungsModulus	20000000

Scale symbol size

Apply Remove Close

M2D Attribute Manager

Prototypes
Material Create

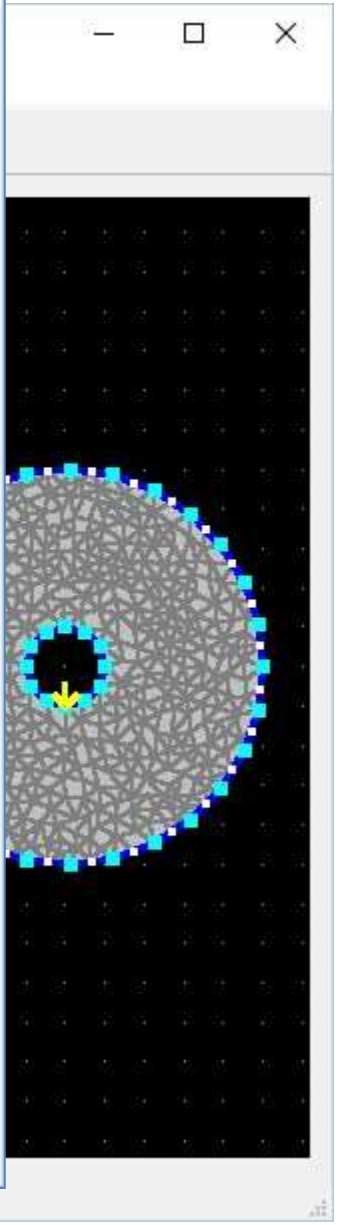
Attributes
Material_1
Force_1
Restrain_1

Property Editor

Property	Value
1 Fx	0
2 Fy	-1000

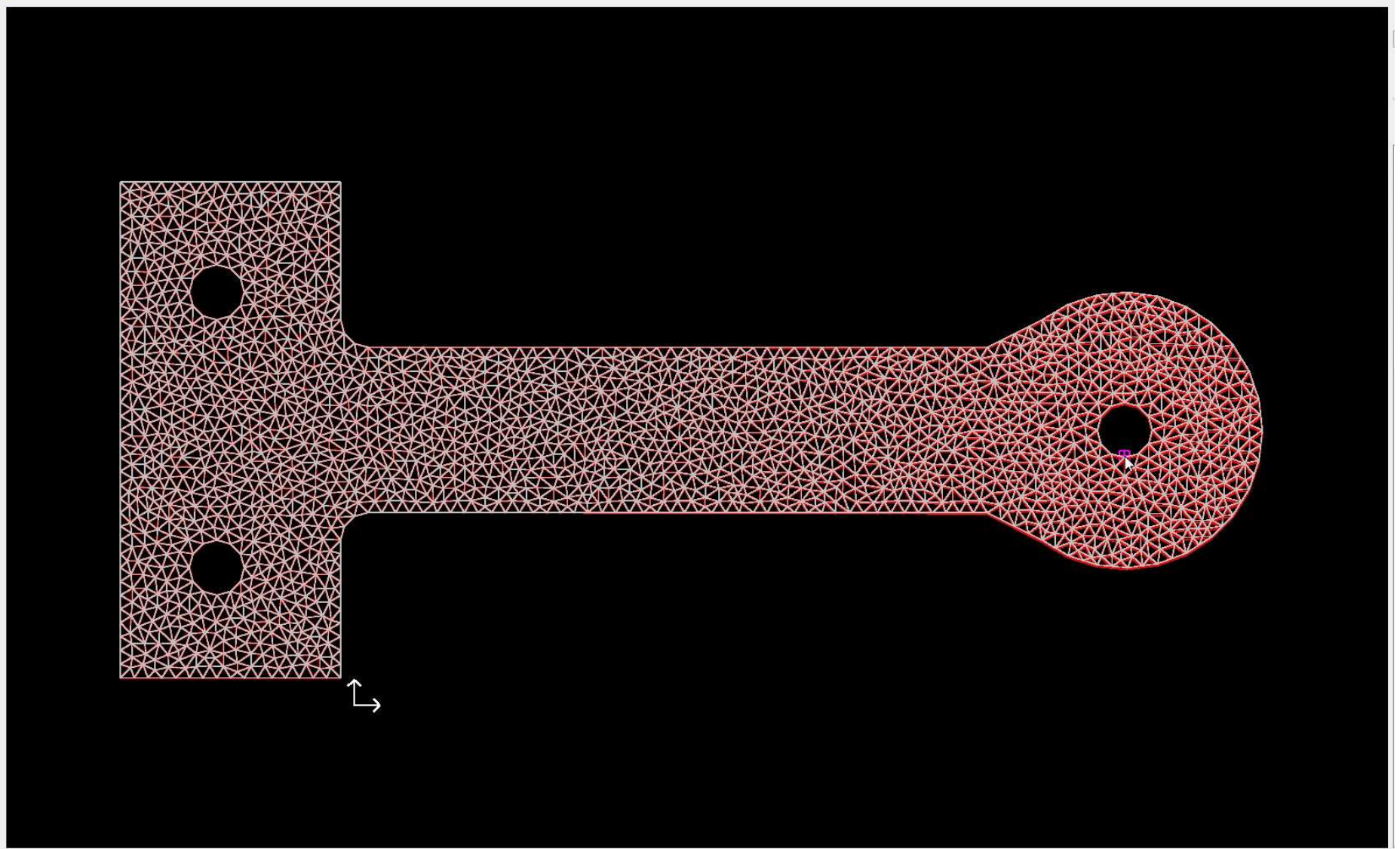
Scale symbol size

Apply Remove Close



Deformada 1x

M² Visualization

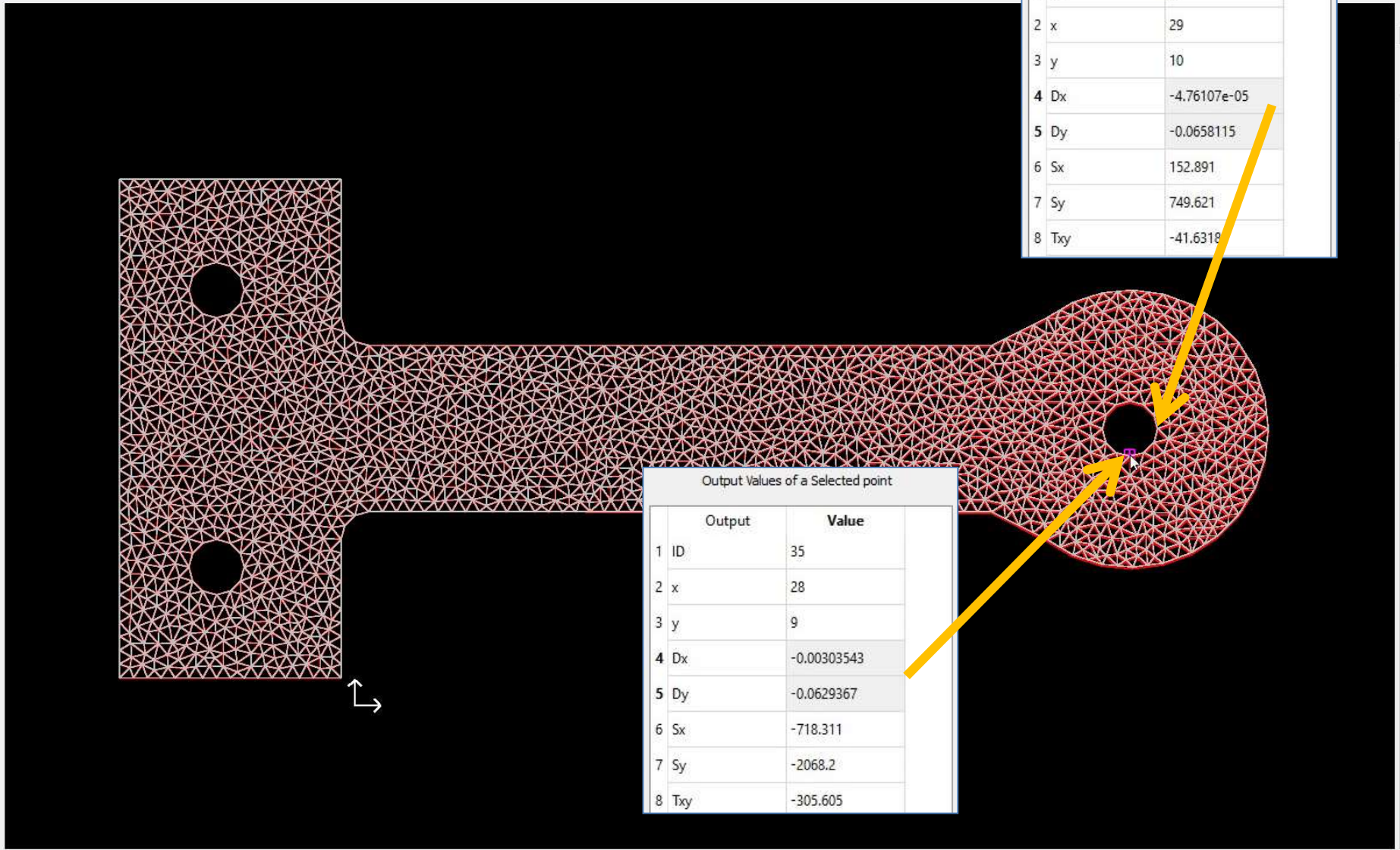


Deformada 1x

M² Visualization

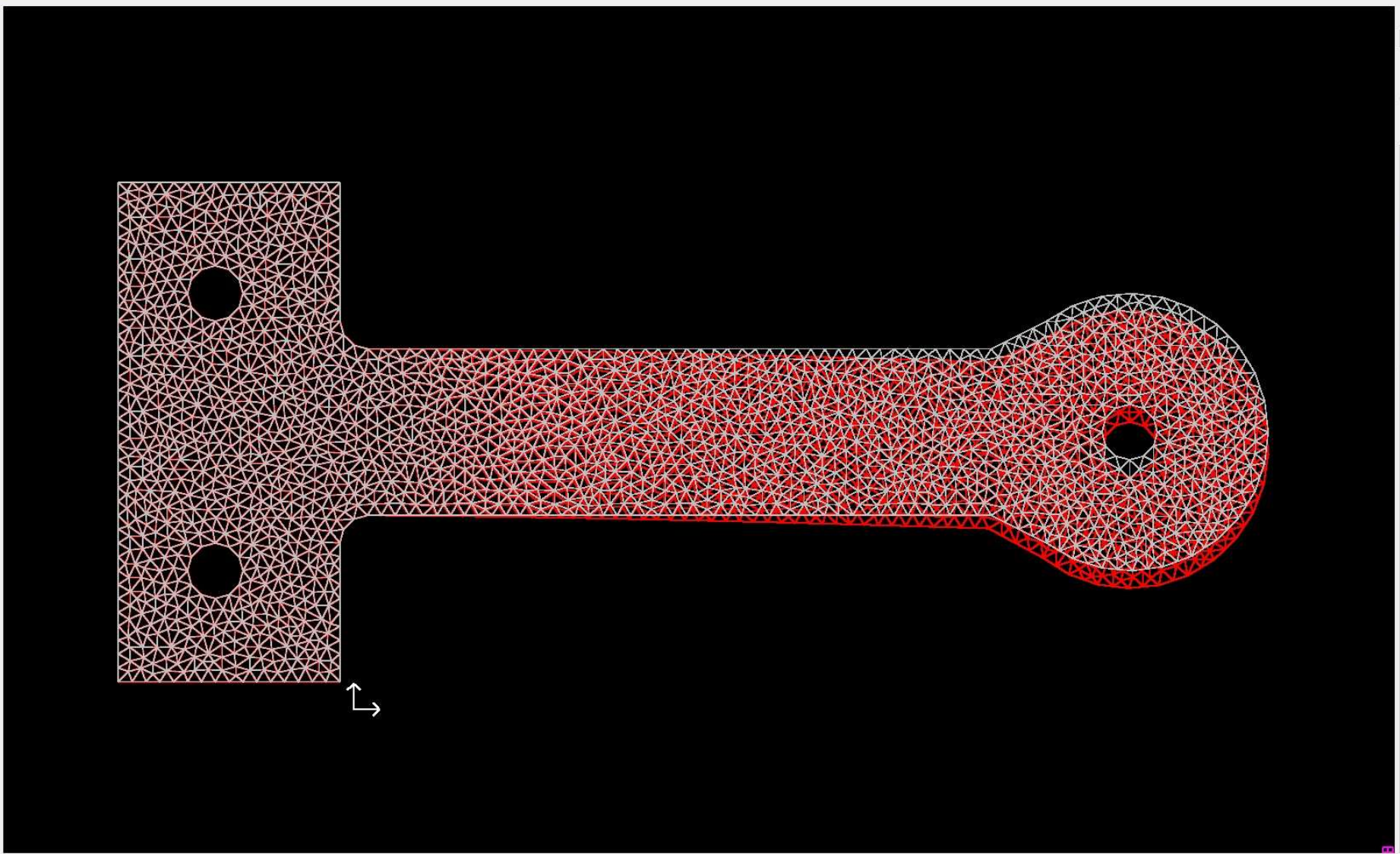
Output Values of a Selected point	
Output	Value
1 ID	26
2 x	29
3 y	10
4 Dx	-4.76107e-05
5 Dy	-0.0658115
6 Sx	152.891
7 Sy	749.621
8 Txy	-41.6318

Output Values of a Selected point	
Output	Value
1 ID	35
2 x	28
3 y	9
4 Dx	-0.00303543
5 Dy	-0.0629367
6 Sx	-718.311
7 Sy	-2068.2
8 Txy	-305.605



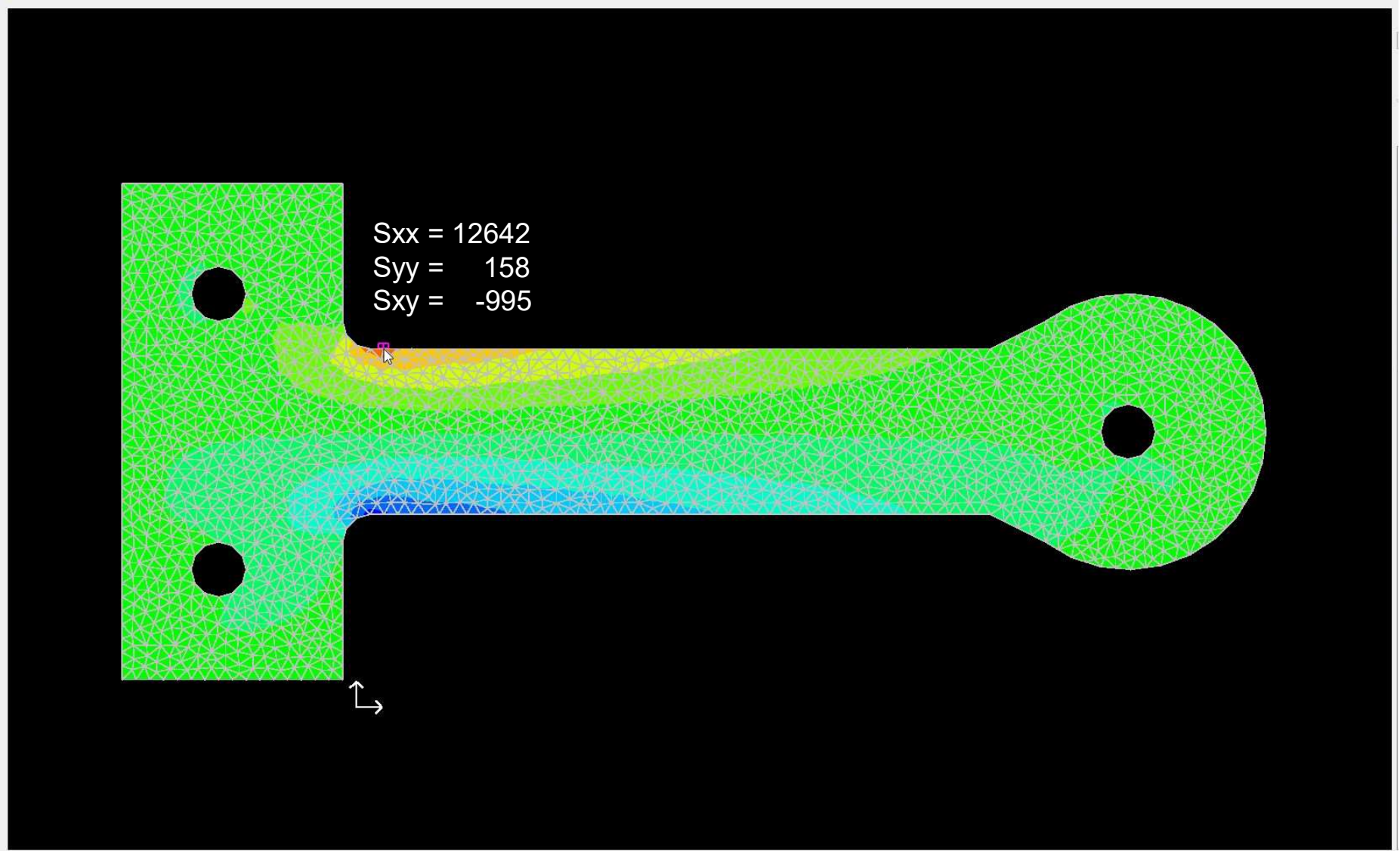
Deformada 10x

M² Visualization



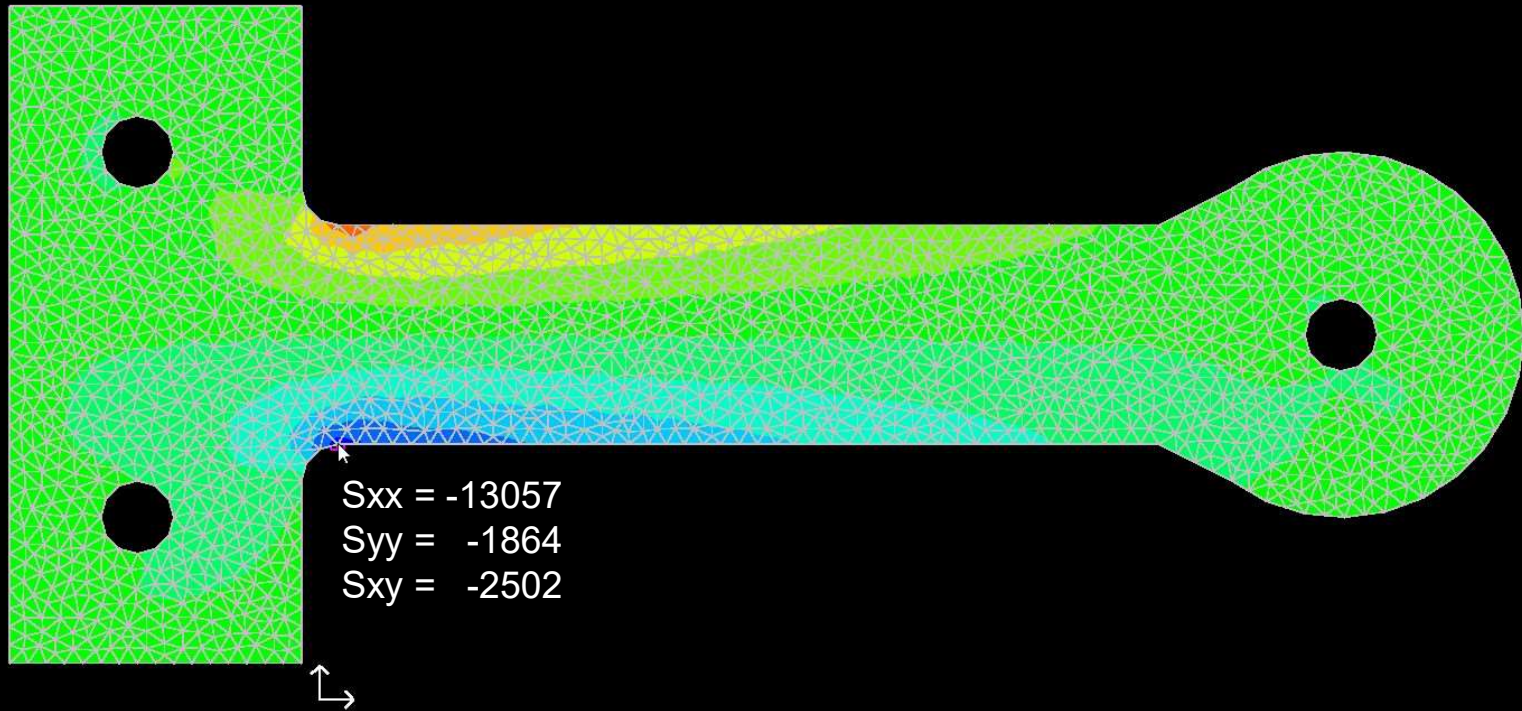
Tensões Sxx

Visualization



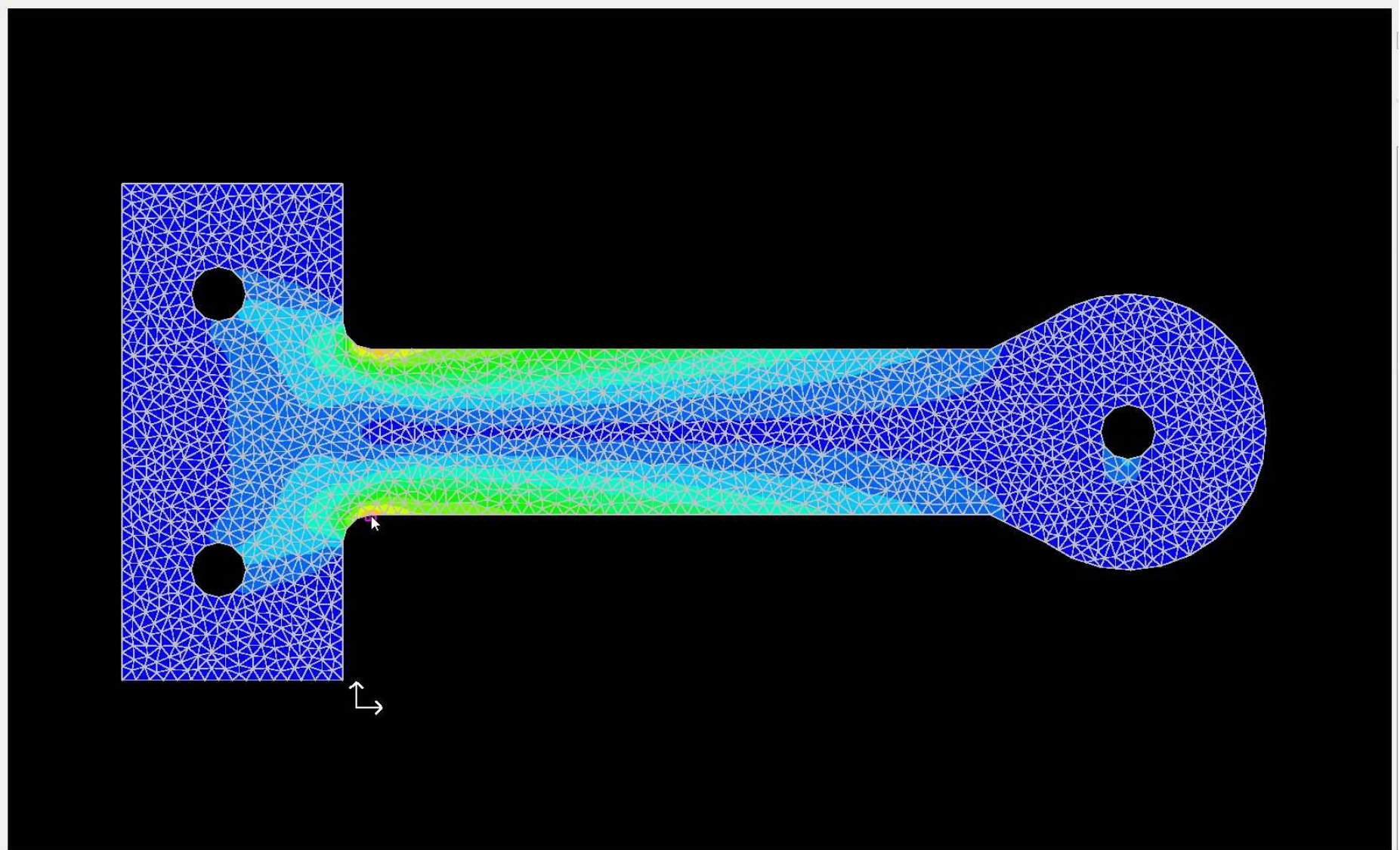
Tensões Sxx

Visualization



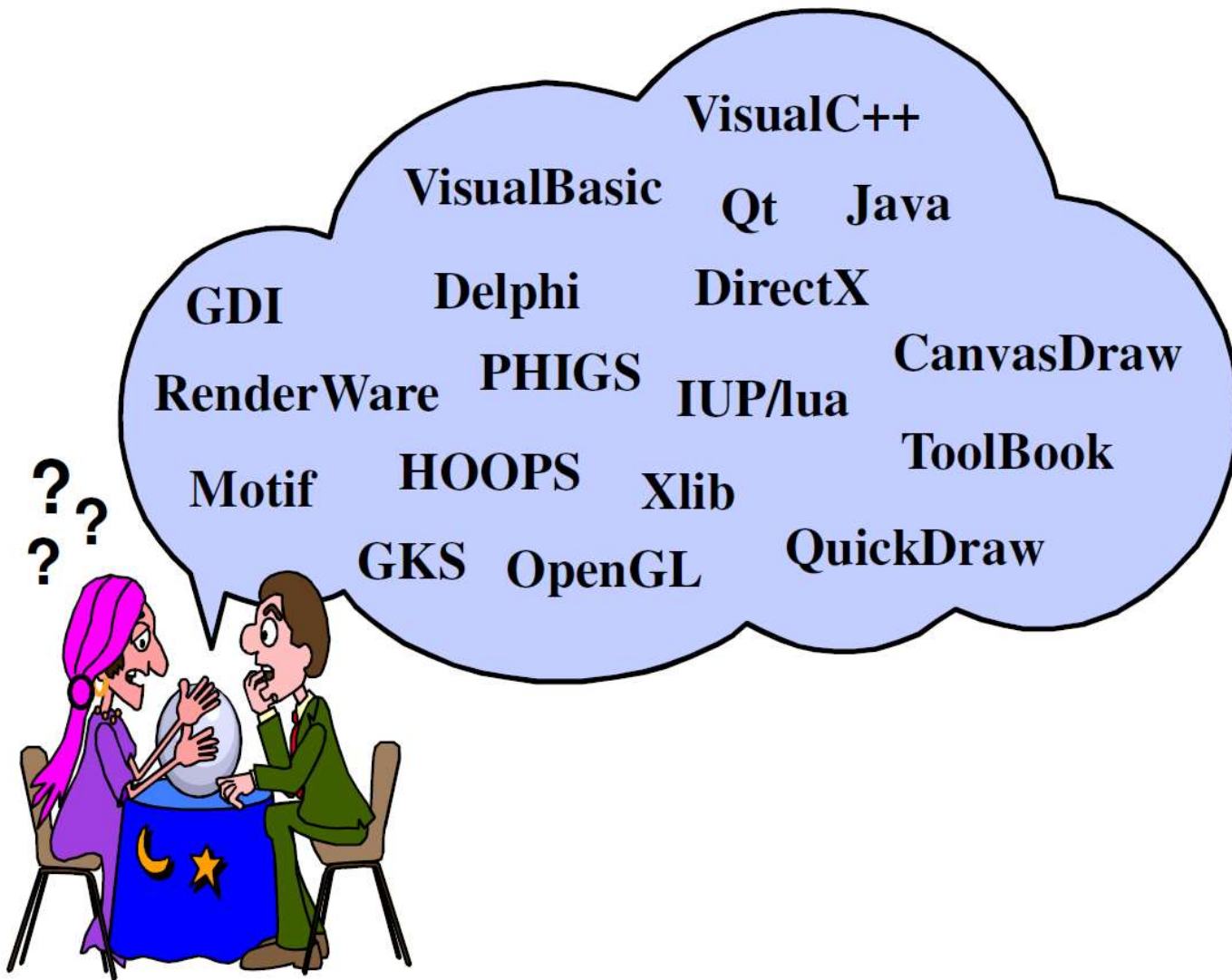
Tensões de Von Mises

M² Visualization



Como desenvolver em Computação Gráfica?

IDEs, Compiladores, Bibliotecas e
Ferramentas



VisualC++

VisualBasic

Qt

Java

GDI

Delphi

DirectX

RenderWare

PHIGS

IUP/lua

CanvasDraw

Motif

HOOPS

Xlib

ToolBook

GKS

OpenGL

QuickDraw

??
??

Ambiente de Desenvolvimento



Ambiente de Desenvolvimento



**Python is powerful... and fast;
plays well with others;
runs everywhere;
is friendly & easy to learn;
is Open.**

These are some of the reasons people who use Python would rather not use anything else.



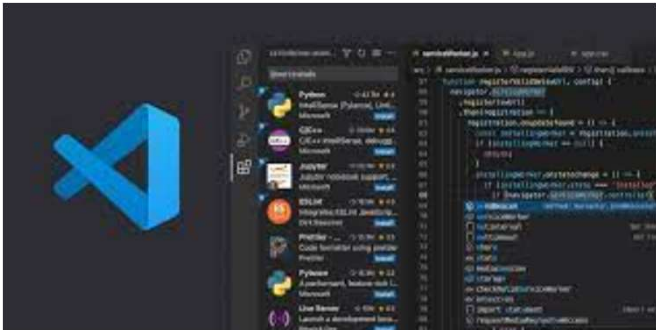
Python is a programming language that lets you work more quickly and integrate your systems more effectively.

You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Ambiente de Desenvolvimento



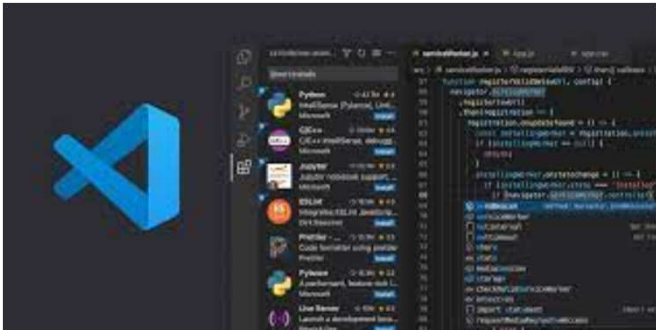
Visual Studio Code



Ambiente de Desenvolvimento



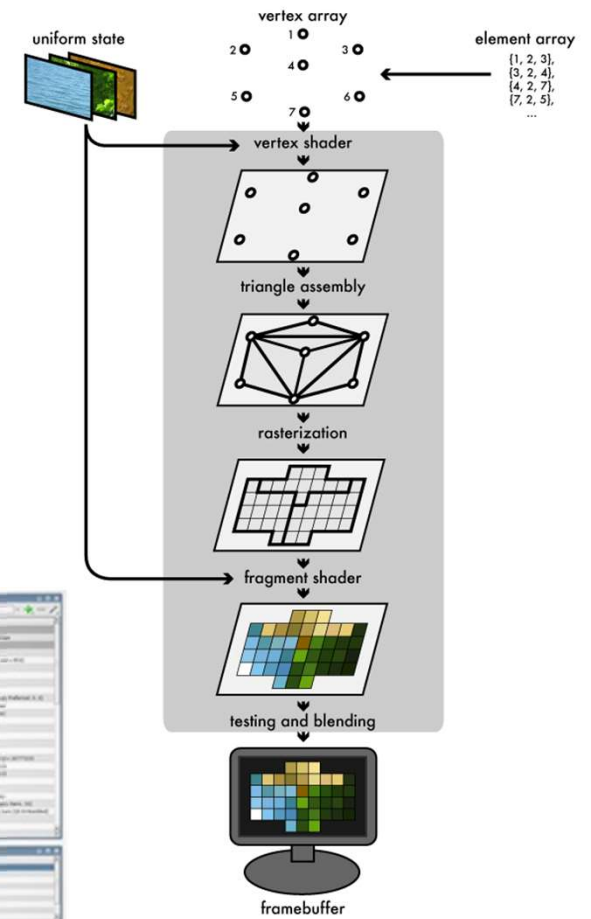
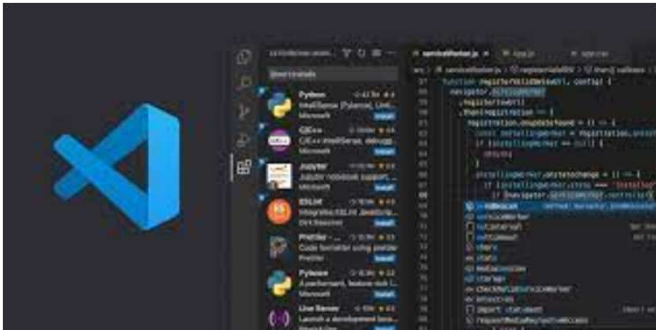
Visual Studio Code



Ambiente de Desenvolvimento



Visual Studio Code



Bibliotecas Gráficas

Sistemas operacionais e interface com usuário:

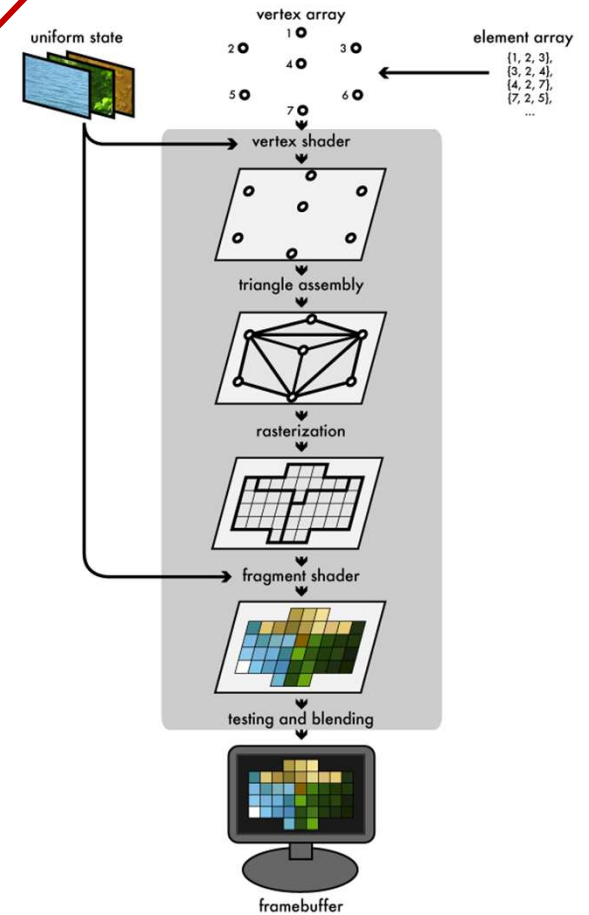
- 1984-Macintosh
- 1987-XWindow v.11
- 1988-Motif
- 1990-Windows 3.0
- 1993-Visual Basic v.3
- 1995-Delphi
- 1995-Windows 95
- 1996-Java
- 1999-Windows NT

...
2008-Qt



Gráficas:

- 1978-core
- 1985-GKS
- 1988-PHIGS
- 1990-PEX
- 1991-GL
- 1993-OpenGL
- 1996-Direct 3D





Perguntas?