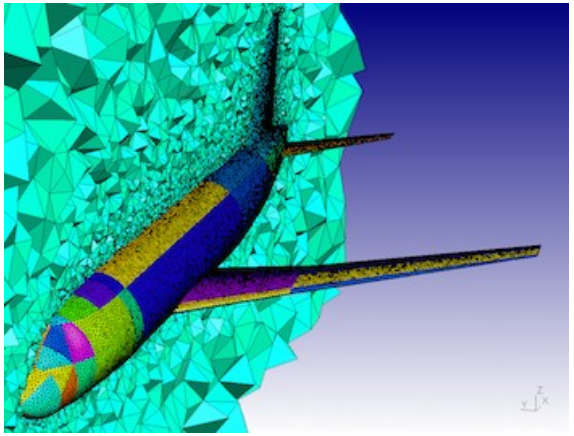
Letmý úvod do GMSH

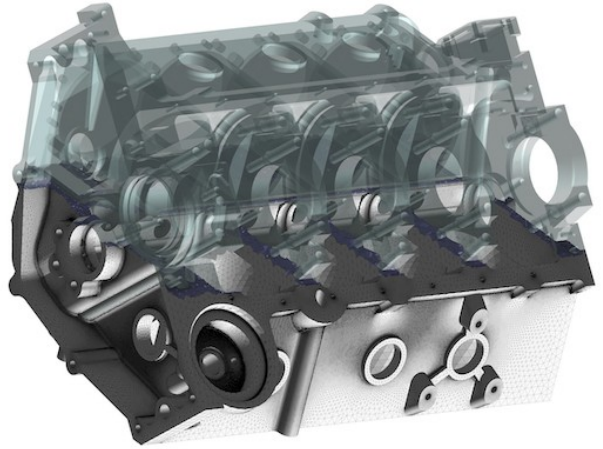
Jan Valášek

Základ https://gmsh.info/

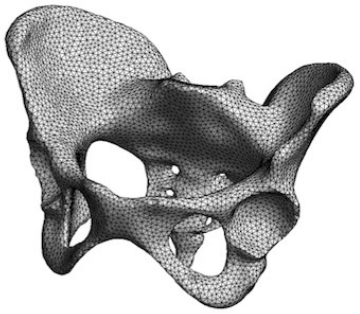


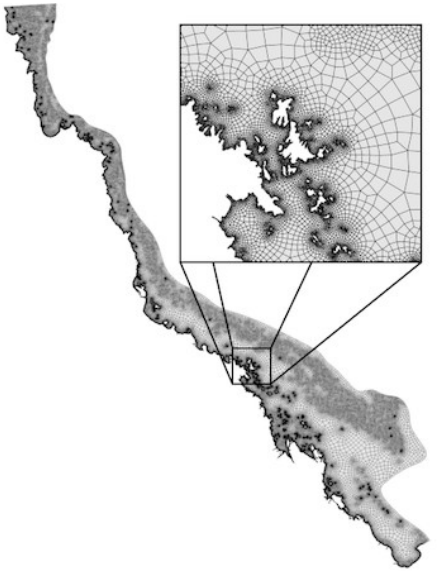


GMSH



- Populární open-source program
- Pro generování konečněprvkových sítí (meshí)
- nebo i sítí pro MKO
- Propojení s dalšími open-source programy, např. FEniCS, OpenFOAM





Hlavní vlastnosti

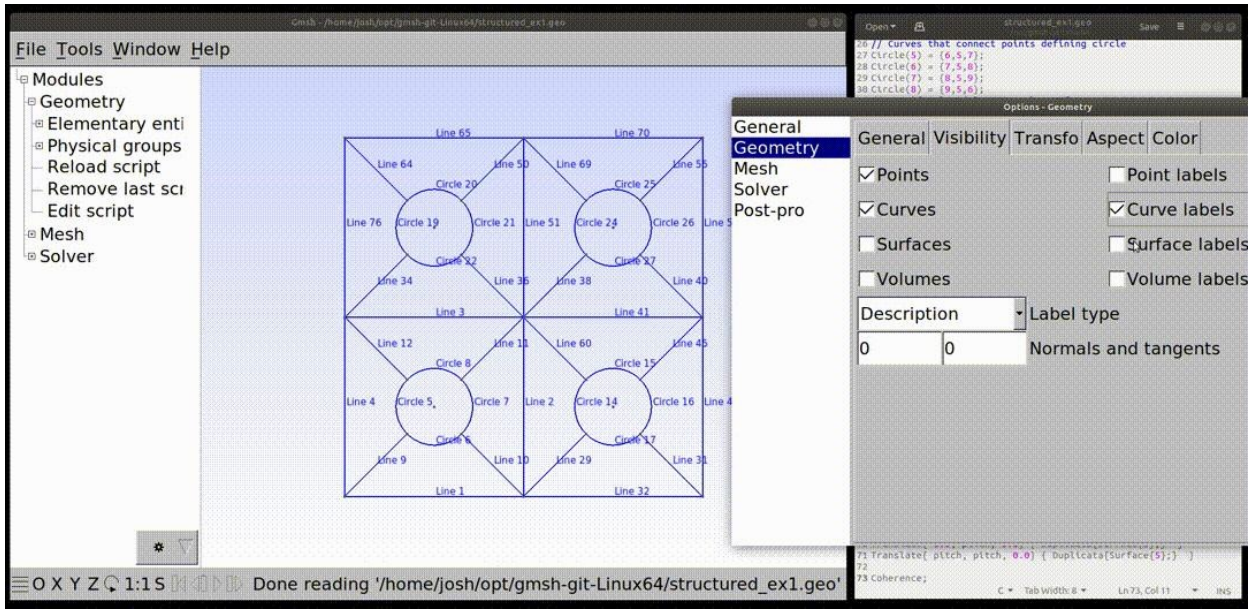
- Umožňuje vytvářet 2D i 3D geometrie
- Umí vytvářet strukturované i nestrukturované (trojúhelníkové, tetrahedrické) sítě
- Zadání geometrie graficky nebo pomocí skriptovacího jazyka (.geo) => automatizace
- Výstupy lze exportovat do formátů .msh, .vtk, .stl atd
- Program dovoluje také postprocessing pro vizualizaci výsledků

Vytváření sítě

1) Definice geometrie (geom. entit)
   - včetně určení parametrů sítě (kde zjemnit)
2) Definice fyzikálních entit
3) Uložení sítě (možné různé formáty)

Geometrie

- Síť se generuje z přesně definovaného geometrickém modelu
- hierarchie geometrických entit
  - Body
  - Křivky (úsečky)
  - Plochy
  - Objemy



Bod (Point)

- Základní stavební kámen geometrie.
- Je definován souřadnicemi (x,y,z) a char. velikostí
- Příklad v Gmsh skriptu: *Point(1) = {0, 0, 0, 0.1};*

Úsečka (Line)

- Spojuje dva body a definuje hranu. Je **orientovaná.**
- Další možnosti: kruh, elipsa, oblouk, Bézierova křivka apod.
- Příklad: *Line(1) = {1, 2}; Circle(2) = {2, 1, 3}; //{A, střed, B}*

Plocha (Surface)

- Vzniká spojením několika hran, které tvoří **uzavřený obvod.**
- Rozlišujeme Plane Surface (rovinná) a Ruled Surface (obecná)
- Příklad: *Curve Loop(1) = {1, 2, 3, 4}; Plane Surface(1) = {1};*
- Stará verze: *Line Loop(1)*
- ***Orientace!***

Objem (Volume)

- Trojrozměrný region, který je ohraničen uzavřeným systémem ploch.
- Na těchto entitách se pak generují 3D elementy (př. tetraedry).
- Příklad: *Surface Loop(1) = {1, 2, 3, 4, 5, 6}; Volume(1) = {1};*

Fyzikální domény (Physical groups)

- Pro použití v simulaci, musíme jednotlivým částem přiřadit fyzikální význam.
- To se dělá pomocí tzv. Physical groups a přidání námi zvoleného indexu (tagu) nebo další varianta je přiřažení stringu (případně definicí: Inlet = 105;)
- Příklady:
  - Physical Point - například bod se známou hodnotou (měření, uzel síly) *Physical Point("ForcePoint") = {3};*
  - Physical Curve - okraj v 2D úloze, například pevné upnutí nebo okraj s předepsaným tokem: *Physical Curve("Inlet") = {1};*

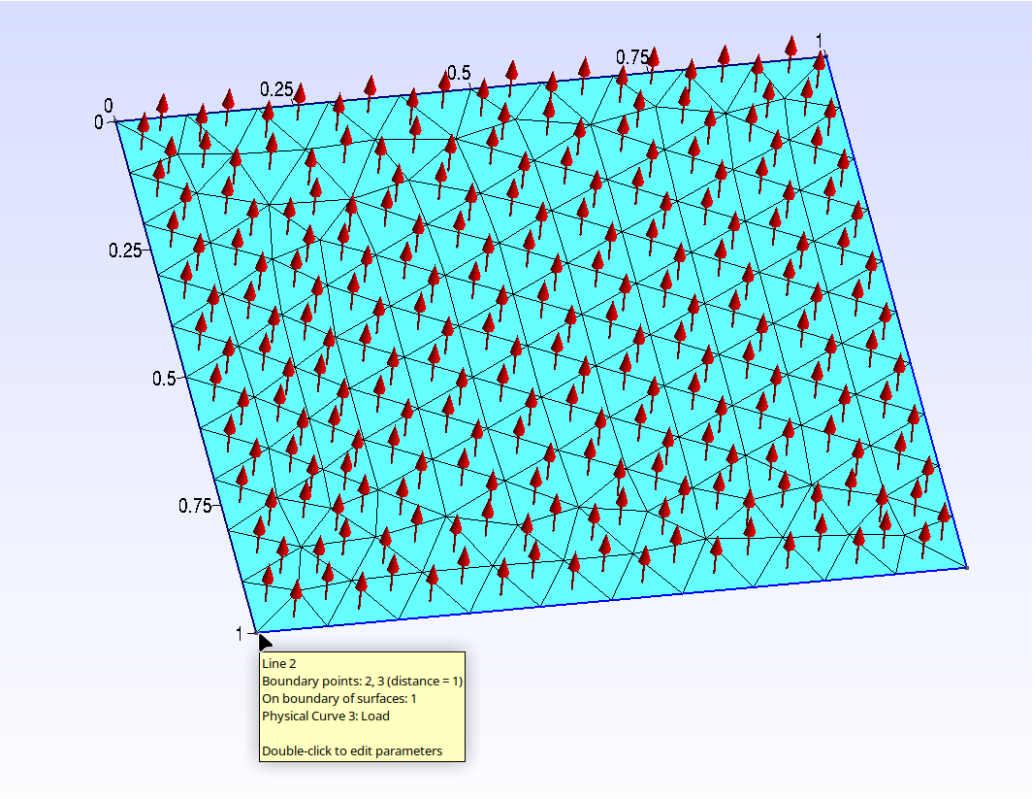
    (staré: Physical Line)
  - Physical Surface - oblast v 2D nebo stěna v 3D
    Physical Surface("Wall") = {5};
  - Physical Volume - objemová doména (např. těleso, tekutina)
    Physical Volume("Fluid") = {1};

Příklad: jednoduchý 2D model

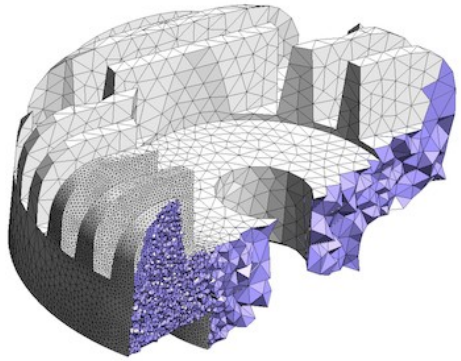
```
// Geometrie
Point(1) = {0, 0, 0, 0.1};
Point(2) = {1, 0, 0, 0.1};
Point(3) = {1, 1, 0, 0.1};
Point(4) = {0, 1, 0, 0.1};
Line(1) = {1, 2};
Line(2) = {2, 3};
Line(3) = {3, 4};
Line(4) = {4, 1};

Line Loop(1) = {1, 2, 3, 4};
Plane Surface(1) = {1};

// Fyzikální domény
Physical Surface("Plate") = {1};     // oblast, kde se počítá
Physical Line("Fixed") = {4};        // spodní okraj - třeba Dirichletova podmínka
Physical Line("Load") = {2};         // horní okraj - zatížení
```

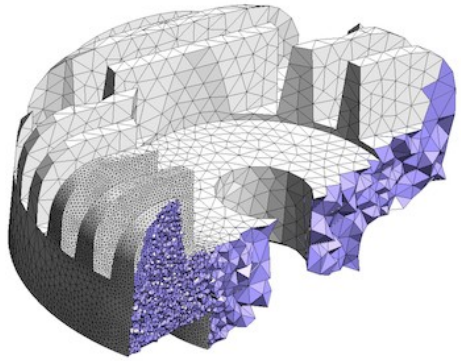


Použití skriptování



- Náročné udržovat očíslování všech entit (průšvih při úpravách)
- `Výrazně lepší použít syntaxi s automatickým číslováním:`
  - `x = 0; y = 0.75; z = 0; r = 0.09;`
  - p1 = newp; Point(p1) = {x, y, z, cl};
  - p2 = newp; Point(p2) = {x + r, y, z, cl};
- `Podobně:`
  - `c1 = newc; Circle(c1) = {p2,p1,p7};`
  - `l1 = newcl; Curve Loop(l1) = {c5,c10,c4};`
  - `s1 = news; Surface(s1) = {l1};`

Použití skriptování



- Nejsilnější je pak spojení seznamy:
  - `theloops[t] = newsl; Surface Loop(theloops[t]) = {s1:s8};`
  - `thehole = newv; Volume(thehole) = theloops[t];`

- A také for cykly:
  - `For t In {1:5}`
  - `  x += 0.166;`
  - `  z += 0.166;`
  - `p[i] = newp; Point(p[i]) = {x,  y,  z,  lc};`
  - `EndFor`

- Pozor na další funkce: správně Cos(), Sin()

Vylepšený příklad

```
h = 0.1; // Velikost síťového prvku

// Počáteční ID pro body a křivky
p1 = newp; Point(p1) = {0, 0, 0, h};
p_prev = p1;

// Definice obvodu pomocí for cyklu (čtverec 1x1)
coords_x = {1, 1, 0, 0};
coords_y = {0, 1, 1, 0};

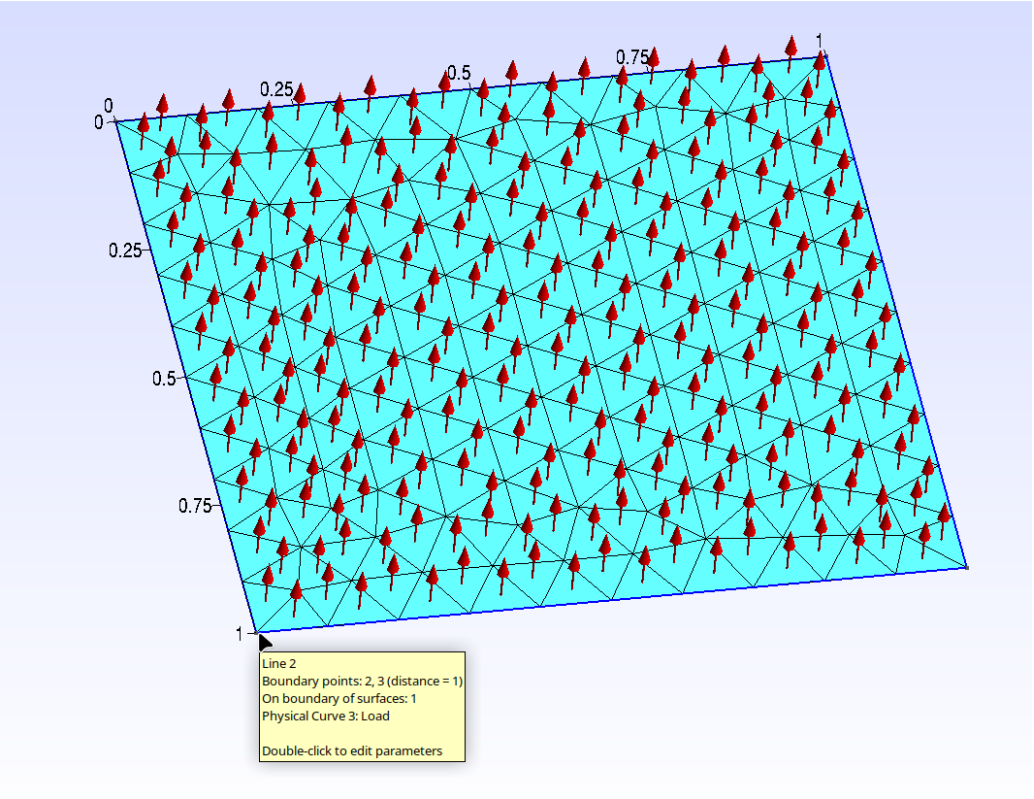
for i = 1 : 4
    p_next = newp; Point(p_next) = {coords_x(i-1), coords_y(i-1), 0,
    c(i) = newc;   Line(c(i)) = {p_prev, p_next};
    p_prev = p_next;
end

// Uzavření smyčky křivek zpět k prvnímu bodu
c(5) = newc; Line(c(5)) = {p_prev, p1};

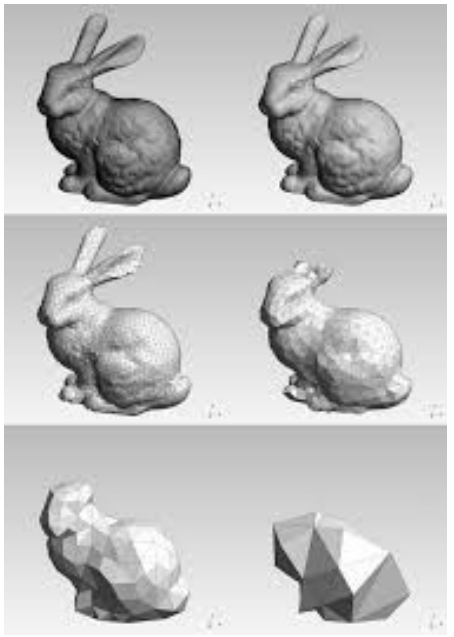
// Geometrická plocha
ll = newll; Line Loop(ll) = {c(1), c(2), c(3), c(4), c(5)};

s = news;  Plane Surface(s) = {ll}; // nebo moje_id = newreg;


// Fyzikální domény (používají dynamicky přiřazená ID)
Physical Surface("Plate") = {s};      // oblast, kde se počítá
Physical Line("Fixed") = {c(5)};      // spodní okraj (původní Line 4)
Physical Line("Load") = {c(2)};       // horní okraj (původní Line 2)
```



Shrnutíčko

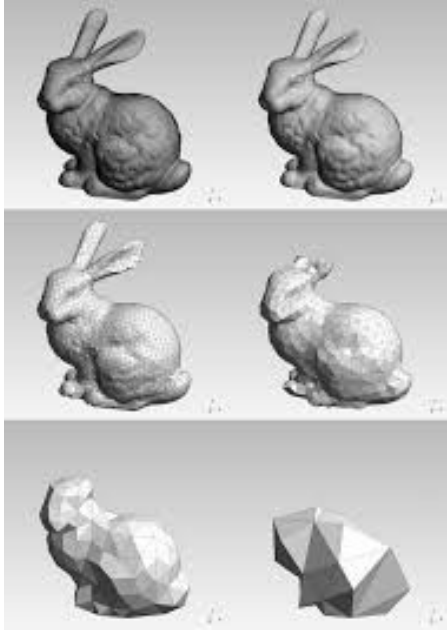


- Geometrické entity = tvar modelu.
  - Každá vyšší entita je složená z entit nižší dimenze.
  - Tj. Body → Hrany → Plochy → Objem (0D → 1D → 2D → 3D)
  - Po definici geometrie lze ji síťovat (meshovat)

- Fyzikální domény (Physical groups) = význam jednotlivých částí pro řešení.
  - **Exportovaná síť (.msh) obsahuje právě tyto Physical groups**
  - Naše indexy se použíjí pro přiřazení materiálů, okrajových podmínek a rovnic.

- Tento přístup dává plnou kontrolu nad přesností, topologií i kvalitou výsledné sítě.
- Výborné skriptování, podobné syntaxi jazyka C

Formát .msh



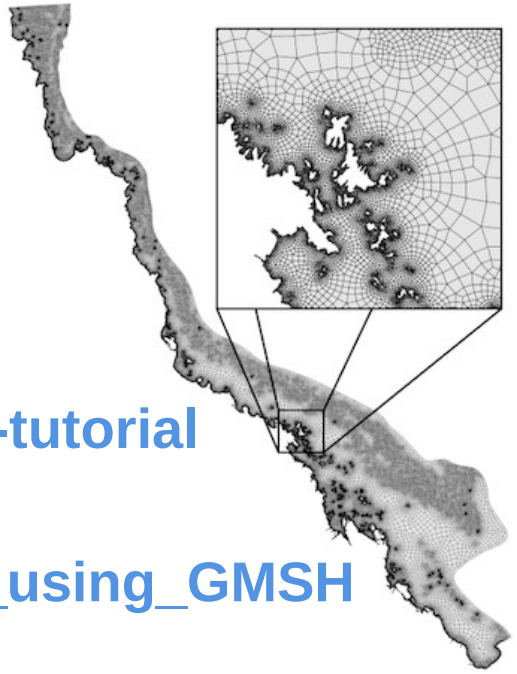
- **MSH Verze 2.3**
  - Klasický a široce rozšířený standard
  - Jednoduchá textová struktura bloků
  - Snadné ruční parsování skripty
  - Nativně podporovaný staršími solvery

```
$MeshFormat
2.3 0 8
$EndMeshFormat
$Nodes
1
1 0.0 0.0 0.0
$EndNodes
```

- **MSH Novější verze (4.x)**
  - Optimalizováno pro velké sítě
  - Hierarchická entitní struktura
  - Podpora binárního uložení dat
  - Efektivní správa paralelního I/O

```
$MeshFormat
4.1 0 8
$EndMeshFormat
$Entities
1 1 0 0
$EndEntities
$Nodes
1 1 0 1
1 0 1 0.0 0.0 0.0
$EndNodes
```

Druhý závěr



- **Jasné = http://gmsh.info/doc/texinfo/gmsh.html#Gmsh-tutorial**
- **Hezké = https://openfoamwiki.net/index.php/2D_Mesh_Tutorial_using_GMSH**
- **Výstupní formát msh = různé verze!!!**
- Zobrazení orientace sítě (pozor při napojování více částí)
- Příkazy vhodné znát:
  - Mesh.MshFileVersion = 2.2; Mesh.Format = 10;
  - Spline, Include, chl, Cos
  - Transfinite Line, Surface; Background field; Recombined;
  - Extrude, Symmetry {0, 1, 0, 0}, Duplicata, Translate, Rotate
- Možnost importu .stl
- Možnost použití dalších funkcí z knihovny *OpenCASCADE*

Sestavte geometrii:

