# Geometry by code in FreeFEM for conjugate heat transfer problems

AK Eswara[1] Pavitra Sandilya[2]

**FreeFEM Days 16th Edition, Paris**

**An academic and industrial research hybrid conference**

[1]Indian Maritime University Kolkata [2]Cryogenic Engineering Centre, IIT Kharagpur

13th December 2024

# Table of contents

# Objectives

1. To showcase generating complex geometry using FreeFEM code
2. To present use of clear syntax while defining boundary conditions
3. To demonstrate a conjugate heat transfer problem using the new learning

# Mathematical transformations - Rotating

Listing 1: using transformation functions in FreeFEM

```
// Functions to rotate geometry
func real rotx(real xo, real yo, real theta) {
    return xo*cos(theta) - yo*sin(theta); // Rotating x
}

func real roty(real xo, real yo, real theta) {
    return xo*sin(theta) + yo*cos(theta); // Rotating y
}
```

# Mathematical transformations - Moving and scaling

Listing 2: using transformation functions in FreeFEM

```
// Functions to rotate geometry
func real shifty(real yo, real val) {
  return yo - val;
}
func real shiftx(real xo, real val) {
  return xo + val;
}
// Scaling geometry
macro xval()(1)// E.O.M
macro yval()(1) // E.O.M
```

# Mathematical transformations - Mirroring

Listing 3: using transformation functions in FreeFEM

```
// Functions for mirroring
func real mirx(real xo, real yo) {
    return xo*cos(pi) - yo*sin(pi); // Mirroring x
}

func real miry(real xo, real yo) {
    return xo*sin(pi) - yo*cos(pi); // Mirroring y
}
```

## Demonstration

Present code

# FreeFEM Geometry using "border"

Listing 4: Using border command in FreeFEM

```
// border format
border j(t=0, ang1*cos(theta)){
    xo=mirx(-t*sin(theta)/cos(theta),-t);
    yo=miry(-t*sin(theta)/cos(theta),-t);
    x = rotx(xo,yo,theta1);
    y = roty(xo,yo,theta1);
    x = shifty(x, -thick);
    y = shifty(y, v);
    x = x * xval;
    y = y * yval;
    label=labm++;}
```

# Strange behaviour of integer variable

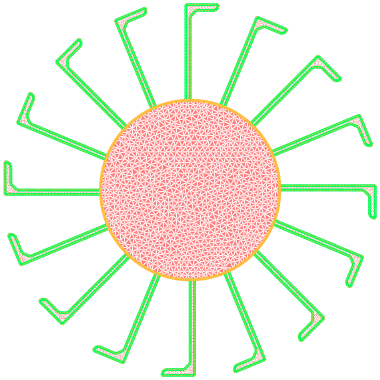
Listing 5: Generating Geometry by loops in FreeFEM

```
mesh Th;
for(int i = 0; i < 90; i += 10){
  j = i+1, k = i+2, l = i+3, lab += 50;
  border j(t=0, ang1*cos(theta)){ ------ ;}
  border k(t=0, ang1*cos(theta)){ ------ ;}
  border l(t=0,2*ang1*cos(theta)){ ------ ;}
  mesh Q = buildmesh(j(m3) + k(m3) + l(m3));
  Q = adaptmesh(Q, hmin=0.001, hmax=0.009);
  Th = Th + Q;
  u += 0.25;
 }
plot(Th, wait=true);
```

# Generating Geometry by looping in FreeFEM

## Demonstration

Present code



# Heat transfer concepts - Transient

$$\frac{\partial T_i(x,t)}{\partial t} = \alpha_i(T_i)\frac{\partial^2 T_i(x,t)}{\partial x^2} \quad \text{for} \quad i = 1,2,3,4$$

$$T_i(x, t=0) = T_0 \quad \text{for} \quad i = 1,2,3,4$$

At $x = d_1$, the boundary condition ensures continuity of temperature and heat flux:

$$T_1(d_1, t) = T_2(d_1, t)$$

$$k_1\frac{\partial T_1(d_1,t)}{\partial x} = k_2\frac{\partial T_2(d_1,t)}{\partial x}$$

# Tutorial - Conjugate Heat transfer in FreeFEM

## Demonstration

Present code and Explain

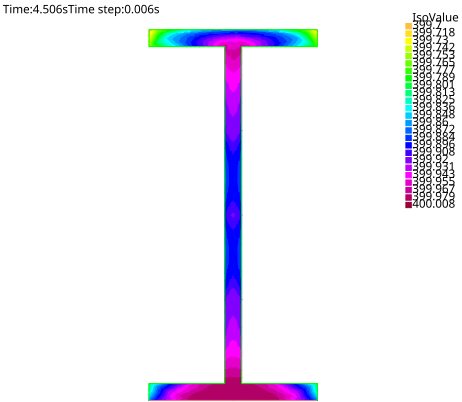
# FreeFEM - Solution for conjugate heat transfer problem

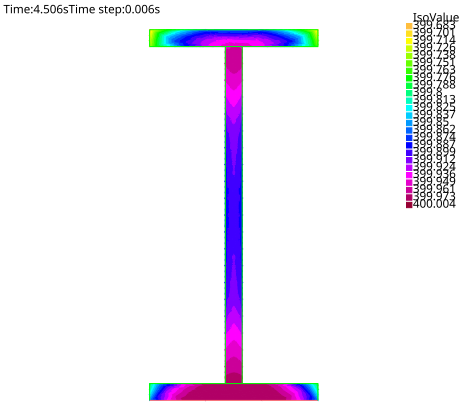
```
while (tyme < totaltime)
{
  Teold = Te;
  solve HeatTransferDOM(Te,ve,solver=GMRES)
    = int2d(En)(Te * ve * rhowood(Teold) * cpwood(Teold) / dt
                + kwood(Teold) * (dx(Te) * dx(ve) + dy(Te) * dy(ve)))
    - int2d(En)(Teold * ve * rhowood(Teold) * cpwood(Teold)/dt)
    + int1d(En, 110)(ksus(Tb)*(grad(Tb)'*[N.x, N.y])*ve)
    + on(110, Te=Tb);

  Tbold = Tb;
  solve HeatTransferBM(Tb,vb,solver=GMRES)
    = int2d(Ibm)(Tb * vb * rhosus(Tbold) * cpsus(Tbold) / dt
                + ksus(Tbold) * (dx(Tb) * dx(vb) + dy(Tb) * dy(vb)))
    - int2d(Ibm)(Tbold * vb * rhosus(Tbold) * cpsus(Tbold) / dt)
    + int1d(Ibm,20)(ksus(Tbold) * (grad(Tb)'* [N.x,N.y]*vb))
    - int1d(Ibm, 110)(kwood(Te)*(grad(Te)'*[N.x, N.y])*vb)
    ;
  plot(Tb,value=true, fill=true, nbiso=25, cmm = "Time:"+ tyme +"s" + "Time step:" + dt + "s");
  tyme += dt;
}
```

# Tutorial - Conjugate Heat transfer in FreeFEM

## Results with continuous mesh and joined mesh



## Unkown

1. How to initialize the scalar value over the finite element space based on coordinates ?
2. It is possible to generate many complex geometries using code in FreeFEM
3. It is possible to generate geometry by AI, need to try

# References

1 Hecht, F. (2012). New development in FreeFem++. Journal of numerical mathematics, 20(3-4), 251-266.

2 Hecht, F. (2024). FreeFEM Documentation - Release 4.13

3 FreeFEM examples on Github

- FreeFEM team and Prof. Frederic Hecht for the great package, support and providing me with this opportunity
- Prof. Pavitra Sandilya, Cryogenic Engineering Centre, IIT Kharagpur
- IIT Kharagpur for facilitating collaboration
- Indian Maritime Unversity (IMU) for supporting my work