

# Geometry by code in FreeFEM for conjugate heat transfer problems

AK Eswara<sup>1</sup> Pavitra Sandilya<sup>2</sup>

**FreeFEM Days 16<sup>th</sup> Edition, Paris**

**An academic and industrial research hybrid conference**

<sup>1</sup>Indian Maritime University Kolkata <sup>2</sup>Cryogenic Engineering Centre, IIT Kharagpur

13<sup>th</sup> December 2024

# Table of contents

- 1 Objectives
- 2 Geometry by code in FreeFEM
- 3 New findings - exploring the provision
- 4 Conjugate Heat Transfer
- 5 Demonstration using examples
- 6 A few issues and possibilities
- 7 References
- 8 Acknowledgements

# 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, angl*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);

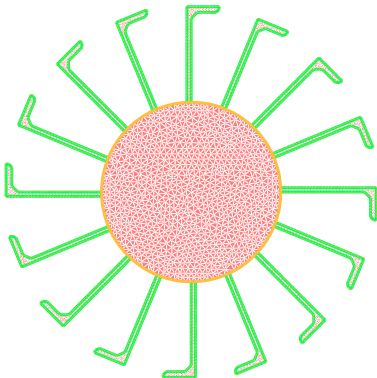
```



# Generating Geometry by looping in FreeFEM

Demonstration

Present code



# Heat transfer concepts - Transient

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

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

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 * rho_wood(Teold) * cp_wood(Teold) / dt
      + k_wood(Teold) * (dx(Te) * dx(ve) + dy(Te) * dy(ve)))
    - int2d(En)(Teold * ve * rho_wood(Teold) * cp_wood(Teold)/dt)
    + int1d(En, 110)(ksus(Tb)*(grad(Tb)'*[N.x, N.y])*ve)
    + on(110, Te=Tb);

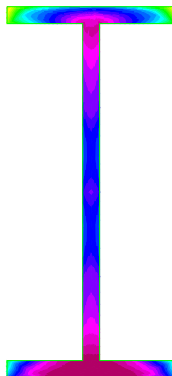
  Tbold = Tb;
  solve HeatTransferBM(Tb,vb,solver=GMRES)
    = int2d(lbm)(Tb * vb * rho_sus(Tbold) * cp_sus(Tbold) / dt
      + ksus(Tbold) * (dx(Tb) * dx(vb) + dy(Tb) * dy(vb)))
    - int2d(lbm)(Tbold * vb * rho_sus(Tbold) * cp_sus(Tbold) / dt)
    + int1d(lbm,20)(ksus(Tbold) * (grad(Tb)'* [N.x,N.y]*vb))
    - int1d(lbm, 110)(k_wood(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

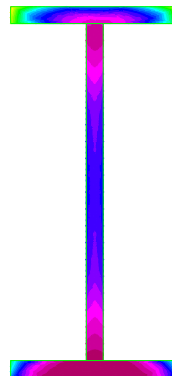
Time:4.506s Time step:0.006s



IsoValue

718
717
716
715
714
713
712
711
710
709
708
707
706
705
704
703
702
701
700
699
698
697
696
695
694
693
692
691
690
689
688
687
686
685
684
683
682
681
680
679
678
677
676
675
674
673
672
671
670
669
668
667
666
665
664
663
662
661
660
659
658
657
656
655
654
653
652
651
650
649
648
647
646
645
644
643
642
641
640
639
638
637
636
635
634
633
632
631
630
629
628
627
626
625
624
623
622
621
620
619
618
617
616
615
614
613
612
611
610
609
608
607
606
605
604
603
602
601
600
599
598
597
596
595
594
593
592
591
590
589
588
587
586
585
584
583
582
581
580
579
578
577
576
575
574
573
572
571
570
569
568
567
566
565
564
563
562
561
560
559
558
557
556
555
554
553
552
551
550
549
548
547
546
545
544
543
542
541
540
539
538
537
536
535
534
533
532
531
530
529
528
527
526
525
524
523
522
521
520
519
518
517
516
515
514
513
512
511
510
509
508
507
506
505
504
503
502
501
500
499
498
497
496
495
494
493
492
491
490
489
488
487
486
485
484
483
482
481
480
479
478
477
476
475
474
473
472
471
470
469
468
467
466
465
464
463
462
461
460
459
458
457
456
455
454
453
452
451
450
449
448
447
446
445
444
443
442
441
440
439
438
437
436
435
434
433
432
431
430
429
428
427
426
425
424
423
422
421
420
419
418
417
416
415
414
413
412
411
410
409
408
407
406
405
404
403
402
401
400
399
398
397
396
395
394
393
392
391
390
389
388
387
386
385
384
383
382
381
380
379
378
377
376
375
374
373
372
371
370
369
368
367
366
365
364
363
362
361
360
359
358
357
356
355
354
353
352
351
350
349
348
347
346
345
344
343
342
341
340
339
338
337
336
335
334
333
332
331
330
329
328
327
326
325
324
323
322
321
320
319
318
317
316
315
314
313
312
311
310
309
308
307
306
305
304
303
302
301
300
299
298
297
296
295
294
293
292
291
290
289
288
287
286
285
284
283
282
281
280
279
278
277
276
275
274
273
272
271
270
269
268
267
266
265
264
263
262
261
260
259
258
257
256
255
254
253
252
251
250
249
248
247
246
245
244
243
242
241
240
239
238
237
236
235
234
233
232
231
230
229
228
227
226
225
224
223
222
221
220
219
218
217
216
215
214
213
212
211
210
209
208
207
206
205
204
203
202
201
200
199
198
197
196
195
194
193
192
191
190
189
188
187
186
185
184
183
182
181
180
179
178
177
176
175
174
173
172
171
170
169
168
167
166
165
164
163
162
161
160
159
158
157
156
155
154
153
152
151
150
149
148
147
146
145
144
143
142
141
140
139
138
137
136
135
134
133
132
131
130
129
128
127
126
125
124
123
122
121
120
119
118
117
116
115
114
113
112
111
110
109
108
107
106
105
104
103
102
101
100
99
98
97
96
95
94
93
92
91
90
89
88
87
86
85
84
83
82
81
80
79
78
77
76
75
74
73
72
71
70
69
68
67
66
65
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0

Time:4.506s Time step:0.006s



IsoValue

718
717
716
715
714
713
712
711
710
709
708
707
706
705
704
703
702
701
700
699
698
697
696
695
694
693
692
691
690
689
688
687
686
685
684
683
682
681
680
679
678
677
676
675
674
673
672
671
670
669
668
667
666
665
664
663
662
661
660
659
658
657
656
655
654
653
652
651
650
649
648
647
646
645
644
643
642
641
640
639
638
637
636
635
634
633
632
631
630
629
628
627
626
625
624
623
622
621
620
619
618
617
616
615
614
613
612
611
610
609
608
607
606
605
604
603
602
601
600
599
598
597
596
595
594
593
592
591
590
589
588
587
586
585
584
583
582
581
580
579
578
577
576
575
574
573
572
571
570
569
568
567
566
565
564
563
562
561
560
559
558
557
556
555
554
553
552
551
550
549
548
547
546
545
544
543
542
541
540
539
538
537
536
535
534
533
532
531
530
529
528
527
526
525
524
523
522
521
520
519
518
517

# 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 University (IMU) for supporting my work