Mesh Transformations in PETSc

Matt Knepley

Computer Science and Engineering
University at Buffalo

PETSc 2024
Universität zu Köln, Köln, DE
May 24, 2024

University at Buffalo
Center for Hybrid Rocket
Exascale Simulation Technology

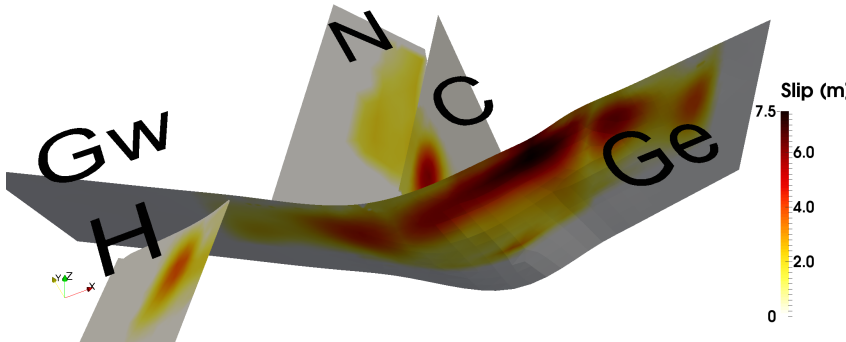
Never believe anything

until you run it.

DMPlex Applications

Plex has been used in many large-scale applications:

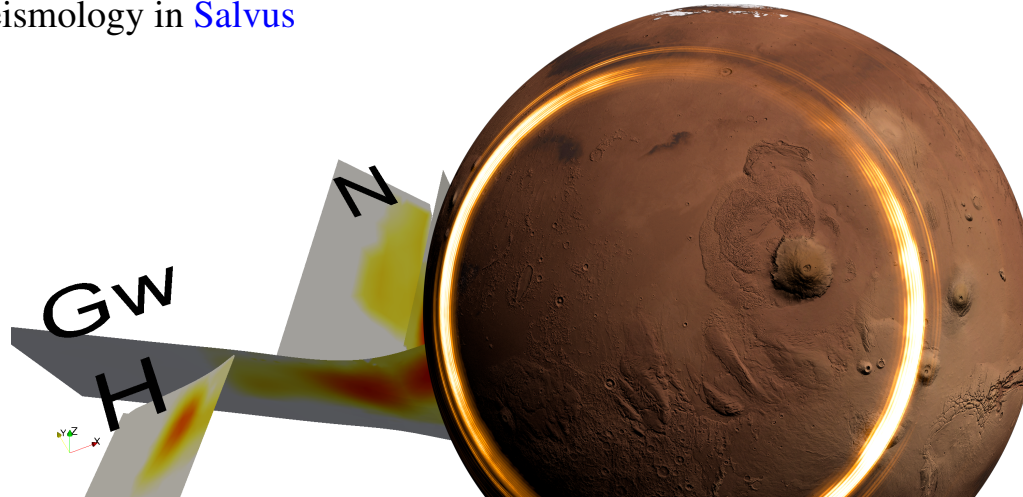
- Crustal deformation in PyLith



DMPlex Applications

Plex has been used in many large-scale applications:

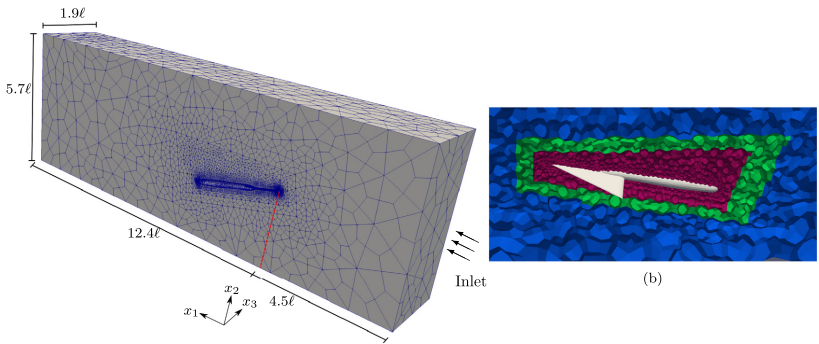
- Crustal deformation in PyLith
- Seismology in Salvus



DMPlex Applications

Plex has been used in many large-scale applications:

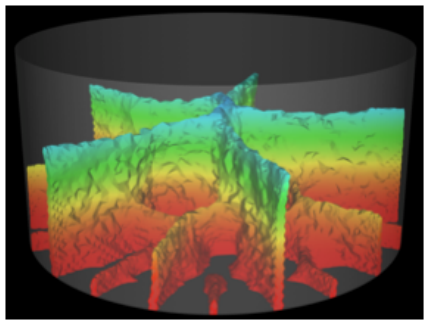
- Crustal deformation in PyLith
- Seismology in Salvus
- Compressible Navier-Stokes flow in SSDC

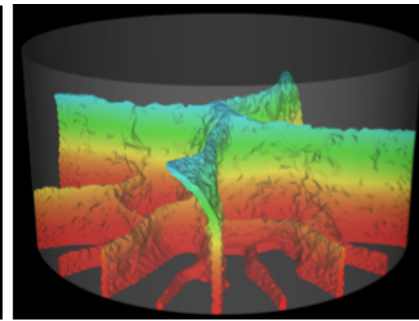


DMPlex Applications

Plex has been used in many large-scale applications:

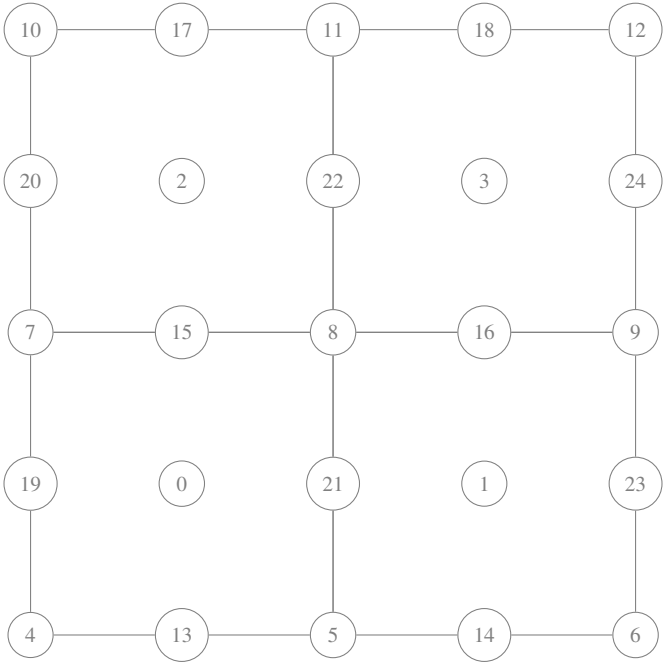
- Crustal deformation in PyLith
- Seismology in Salvus
- Compressible Navier-Stokes flow in SSDC
- Fracture mechanics in MEF90

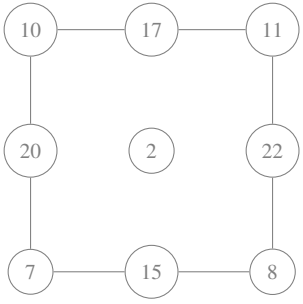


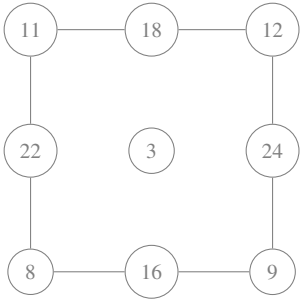


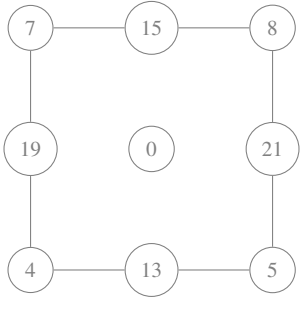
Outline

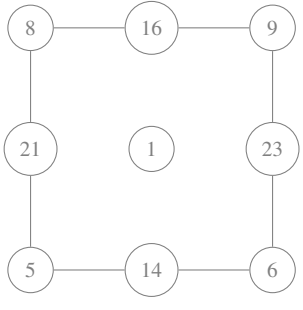
Outline

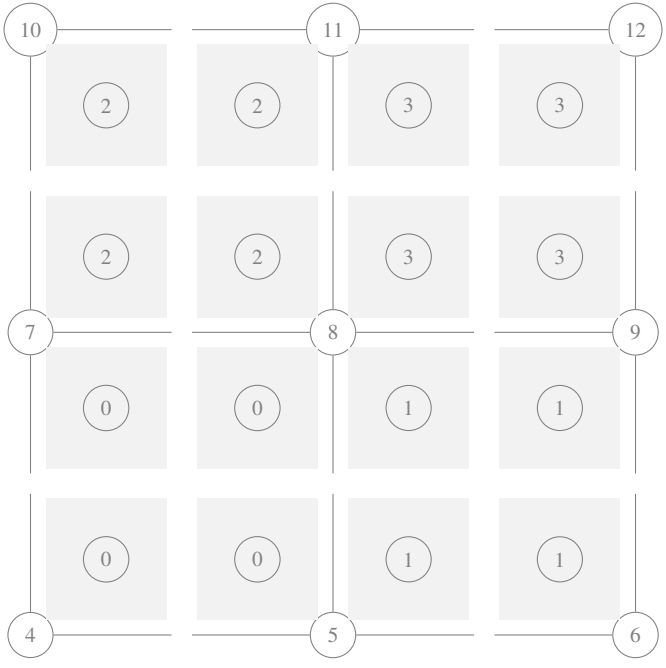

10
17
11
18
12
20
2
22
3
24
7
15
8
16
9
19
0
21
1
23
4
13
5
14
6

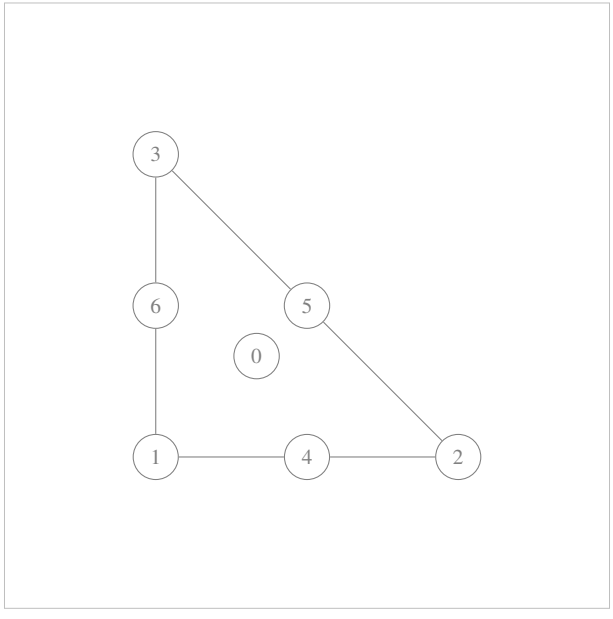

10
17
11
20
2
22
7
15
8

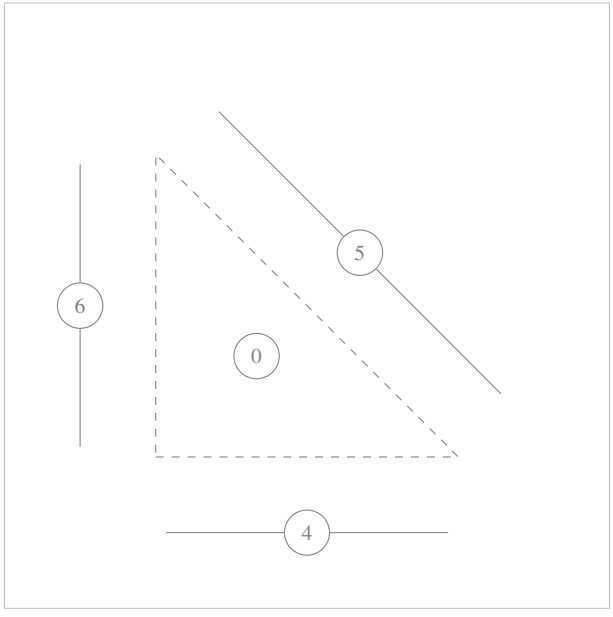

11
18
12
22
3
24
8
16
9

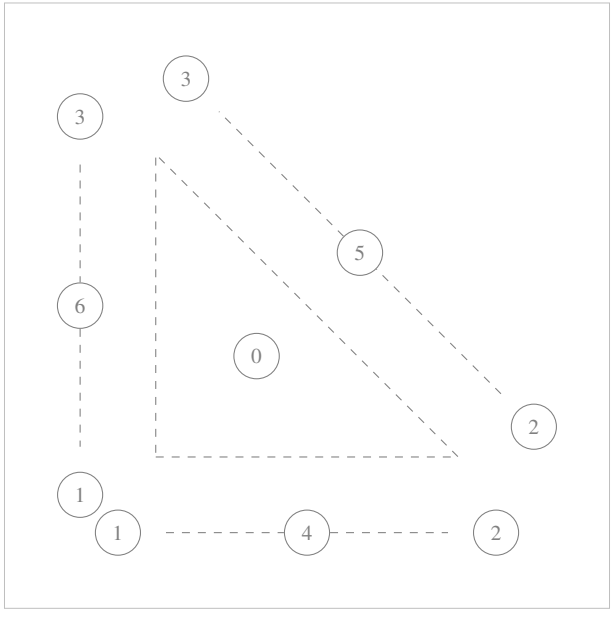

7
15
8
19
0
21
4
13
5

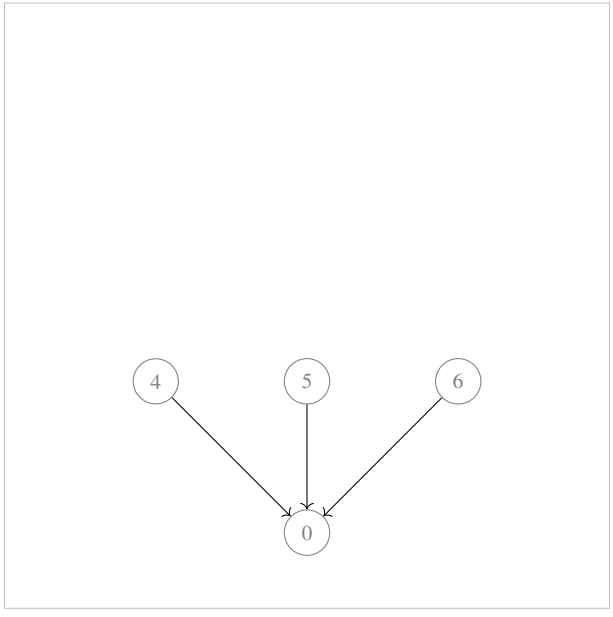

8
16
9
21
1
23
5
14
6

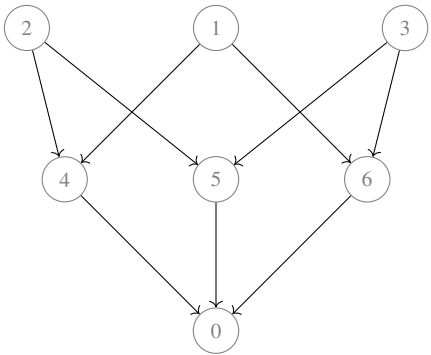

10
11
12
2
2
3
3
2
2
3
3
7
8
9
0
0
1
1
0
0
1
1
4
5
6

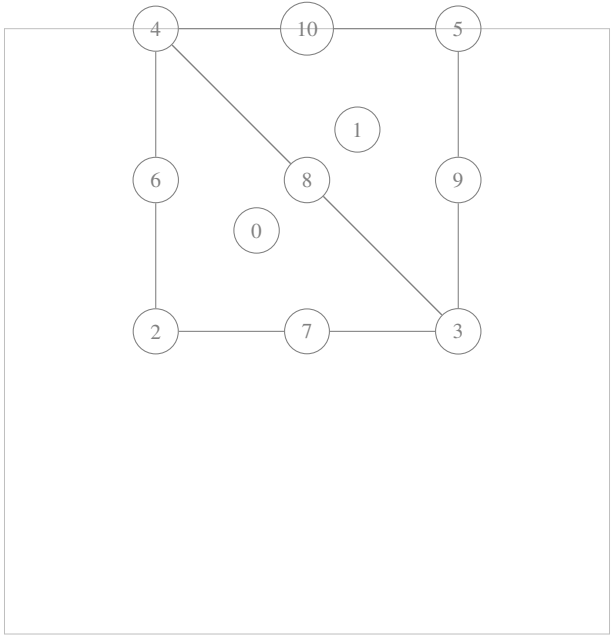

3
6
5
0
1
4
2

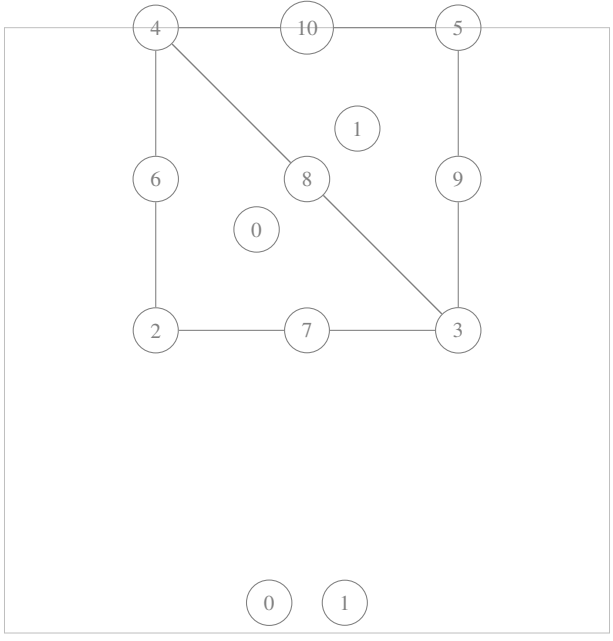

5
6
0
4

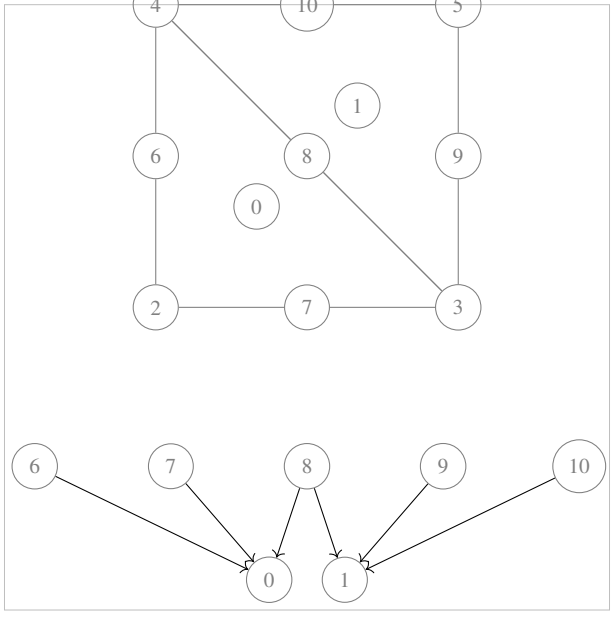

3
3
5
6
0
2
1
1
4
2

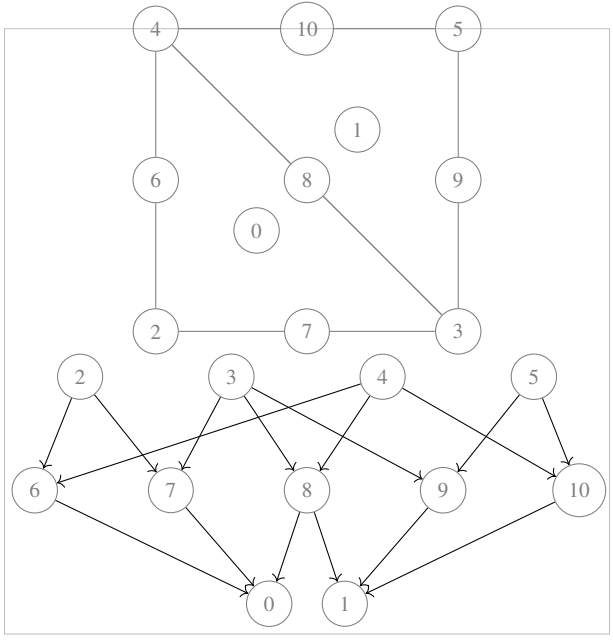

4
5
6
0


2
1
3
4
5
6
0

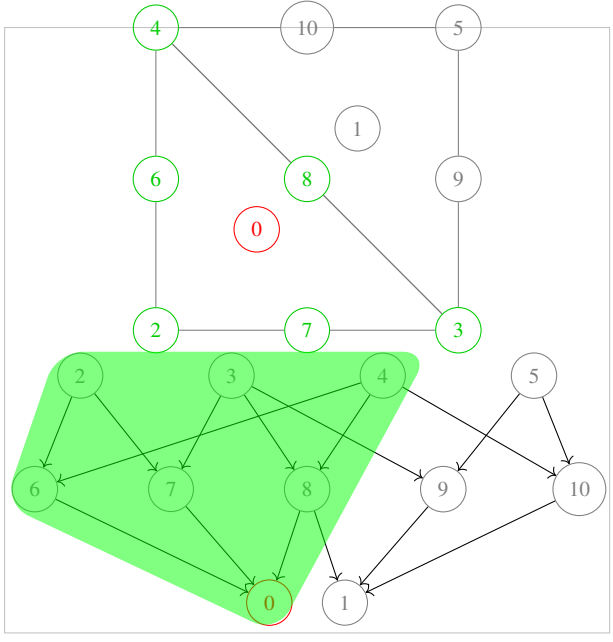

4
10
5
1
6
8
9
0
2
7
3

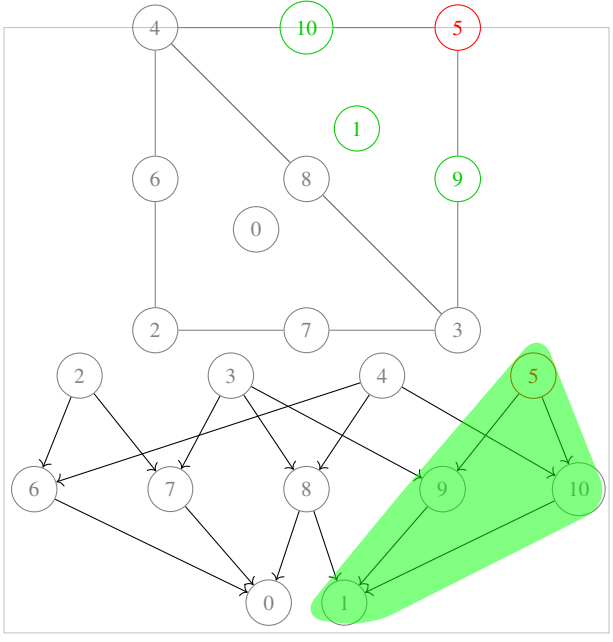

4
10
5
1
6
8
9
0
2
7
3
0
1

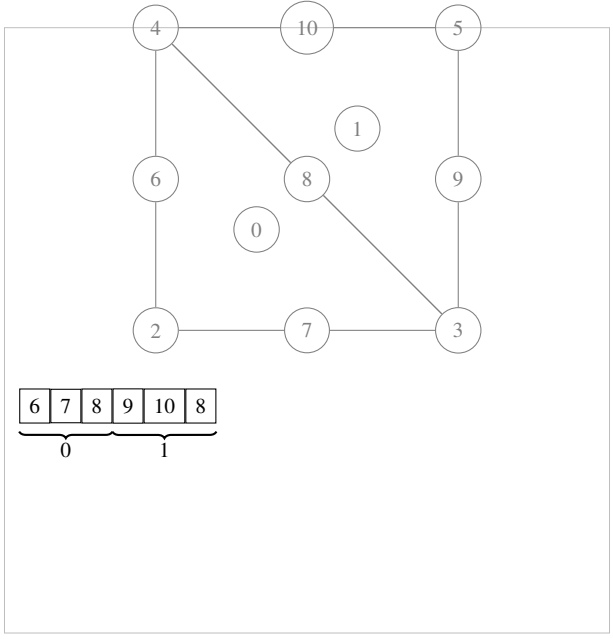

4
10
5
1
6
8
9
0
2
7
3
6
7
8
9
10
0
1

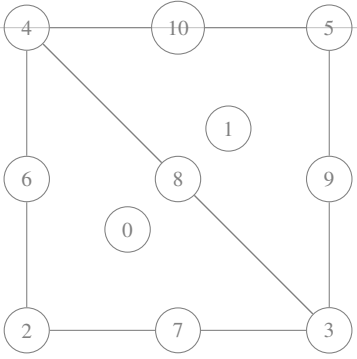

4
10
5
1
6
8
9
0
2
7
3
2
3
4
5
6
7
8
9
10
0
1

Outline


4
10
5
1
6
8
9
0
2
7
3
2
3
4
5
6
7
8
9
10
0
1

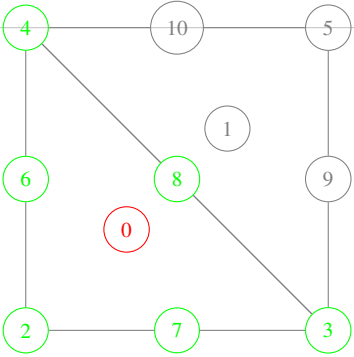

4
10
5
1
6
8
9
0
2
7
3
2
3
4
5
6
7
8
9
10
0
1


4
10
5
1
6
8
9
0
2
7
3
6
7
8
9
10
8
0
1



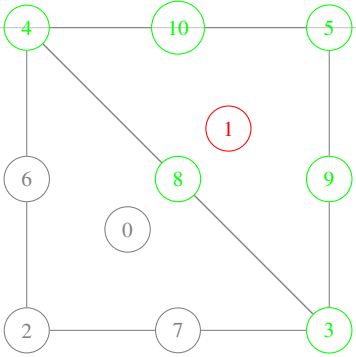
6	7	8	9	10	8	4	2	2	3	3	4	3	5	5	4

0 1 6 7 8 9 10



6	7	8	9	10	8	4	2	2	3	3	4	3	5	5	4
0				1			6	7		8		9		10	

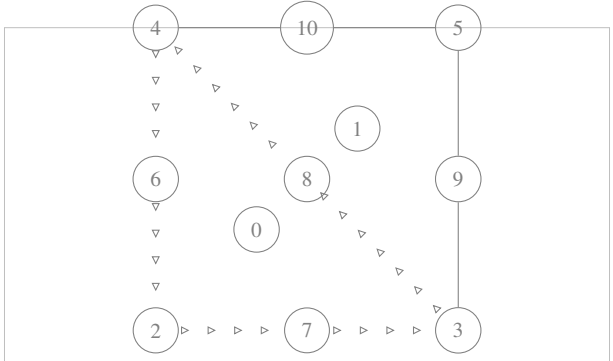
$\vee\vee_0$ = 6 7 8 4 2 2 3 3 4



6	7	8	9	10	8	4	2	2	3	3	4	3	5	5	4
0				1			6	7		8		9		10	

$\vee\!\!\vee_0$ = 6 7 8 4 2 2 3 3 4

$\vee\!\!\vee_1$ = 9 10 8 3 5 5 4 3 4



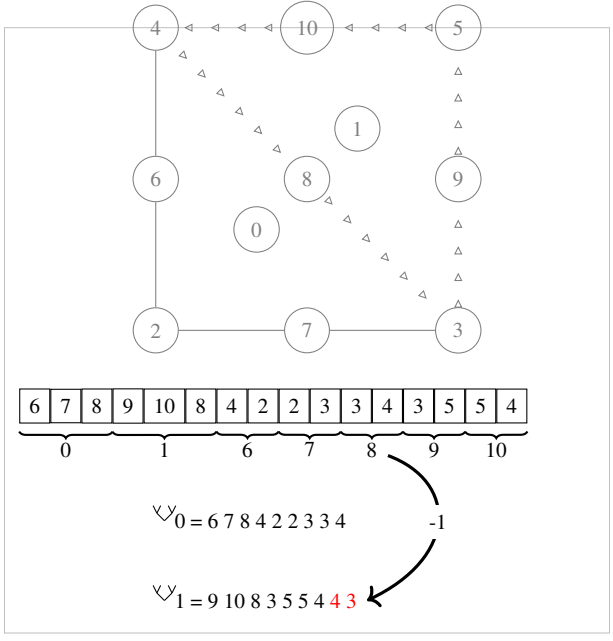
6	7	8	9	10	8	4	2	2	3	3	4	3	5	5	4

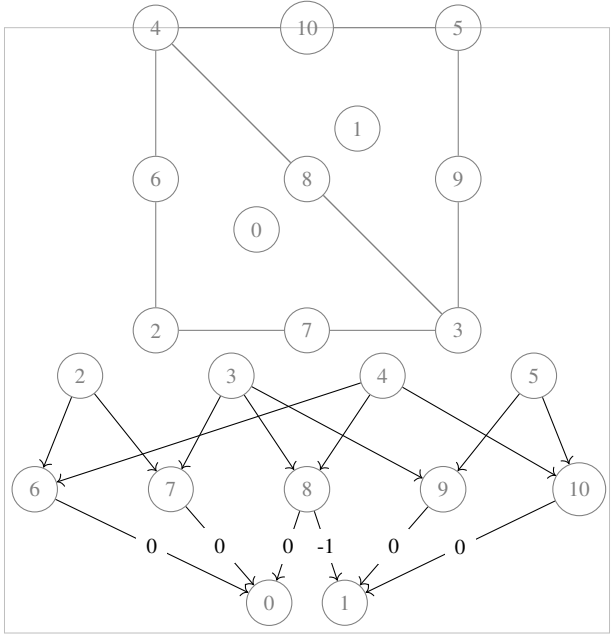
0 1 6 7 8 9 10

0

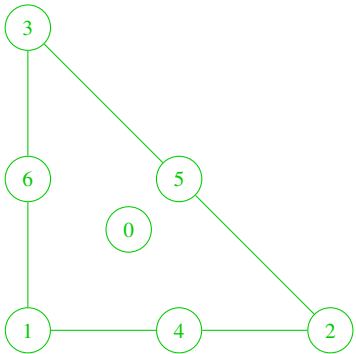
$\vee\!\!\vee_0$ = 6 7 8 4 2 2 3 3 4

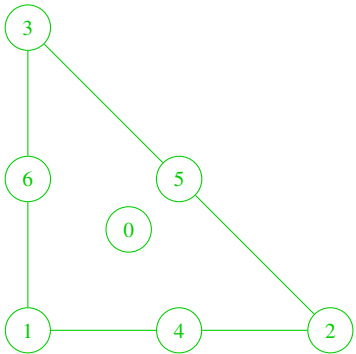
$\vee\!\!\vee_1$ = 9 10 8 3 5 5 4 4 3

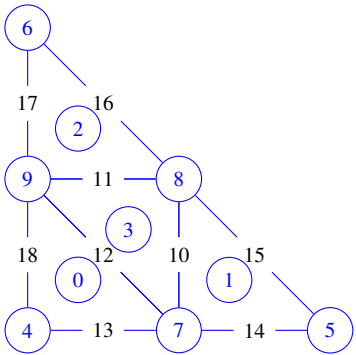

4
10
5
1
6
8
9
0
2
7
3
6 7 8 9 10 8 4 2 2 3 3 4 3 5 5 4
0 1 6 7 8 9 10
-1
$\vec{y}_0$ = 6 7 8 4 2 2 3 3 4
$\vec{y}_1$ = 9 10 8 3 5 5 4 4 3

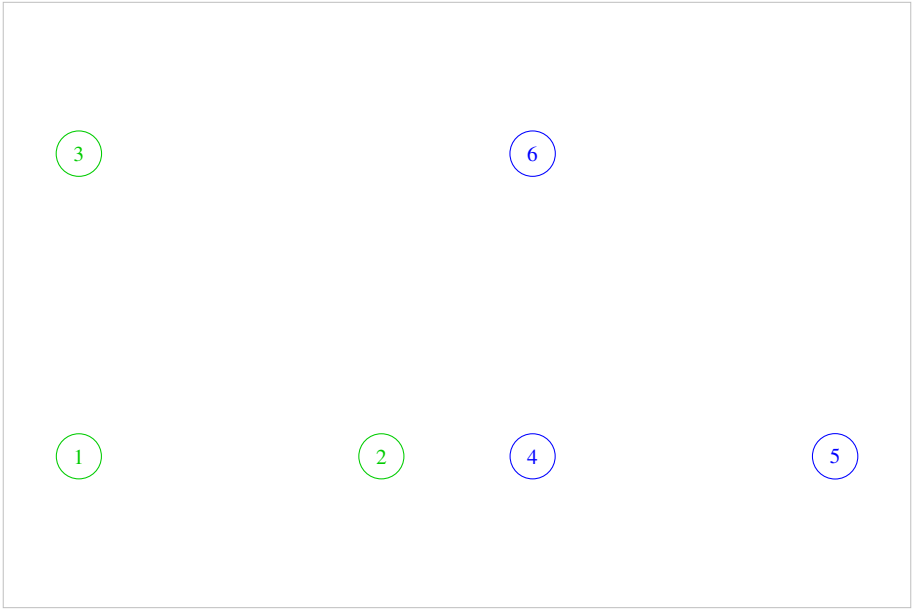

4
10
5
1
6
8
9
0
2
7
3
2
3
4
5
6
7
8
9
10
0
0
0
-1
0
0
0
1

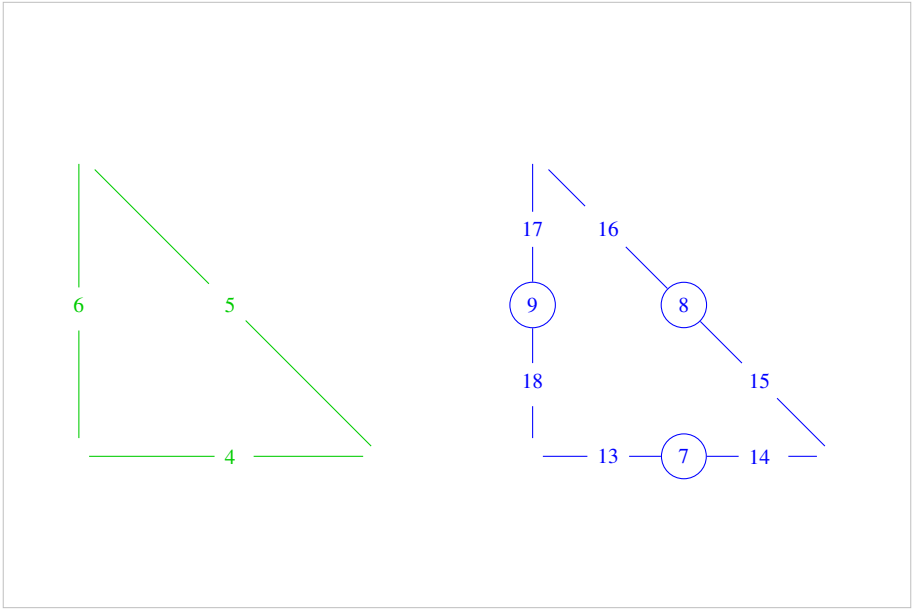
Outline


3
6
5
0
1
4
2


3
6
5
0
1
4
2


6
17
16
2
9
11
8
3
18
12
10
15
0
1
4
13
7
14
5


3
6
1
2
4
5


6
5
4
17
16
9
8
18
15
13
7
14


1
4
2
4
13
7
14
5

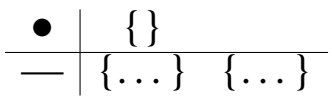

1
4
2
4
13
7
14
5


1
4
2
4
13
7
14
5


1
4
2
4
13
7
14
5


1
4
2


4
13
7
14
5


{}
{...}
{...}

$$— \mid \{(1, \ldots), \ldots\} \quad \{\ldots\}$$

$$— \mid \{(1, 0, \ldots), \ldots\} \quad \{\ldots\}$$

$$— \mid \{(1, 0, 0), \dots\} \quad \{\dots\}$$

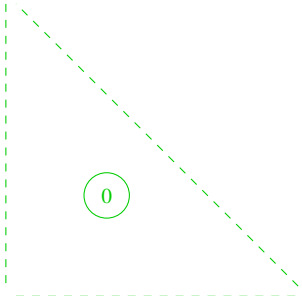
$$— \mid \{(1, 0, 0), (0, \ldots)\} \quad \{\ldots\}$$

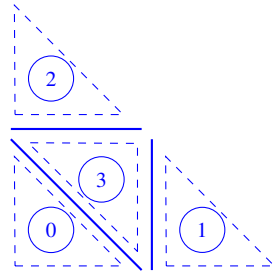
$$— \mid \{(1, 0, 0), (0, 0)\} \quad \{\dots\}$$

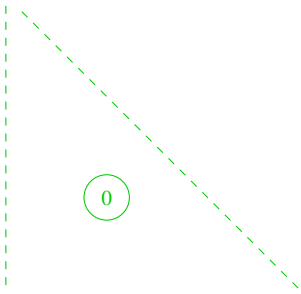
$$— \mid \{(1, 0, 0), (0, 0)\} \quad \{(0, 0), (1, \ldots)\}$$

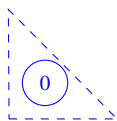
$$— \mid \{(1, 0, 0), (0, 0)\} \quad \{(0, 0), (1, 1, \ldots)\}$$

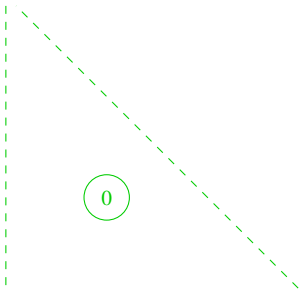
— | {(1, 0, 0), (0, 0)} {(0, 0), (1, 1, 0)}

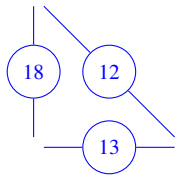

0

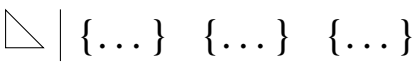

2
3
0
1

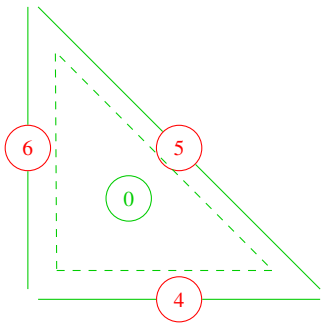

0

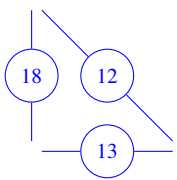

0

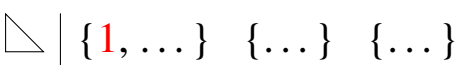

0

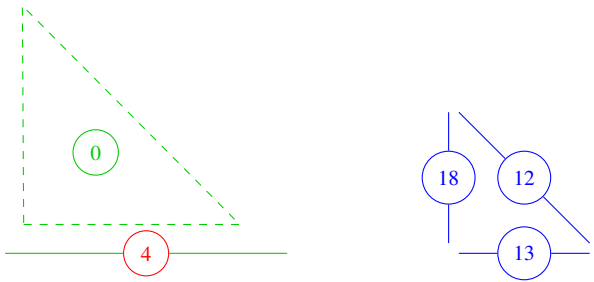

18
12
13


{...}
{...}
{...}

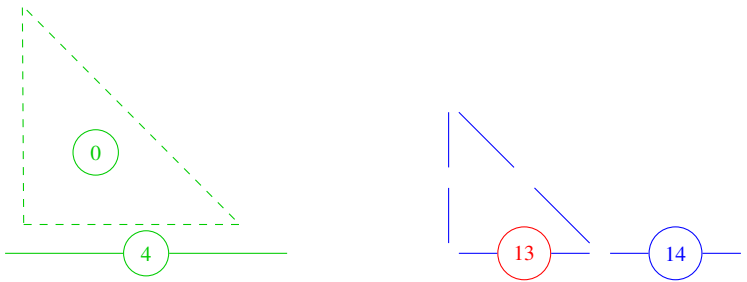

6
5
0
4

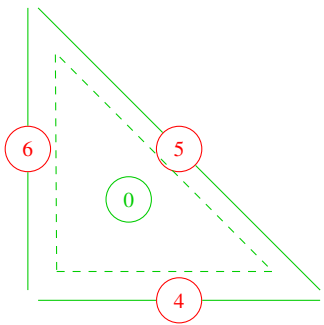

18
12
13

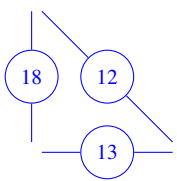

{1, . . . } {. . . } {. . . }

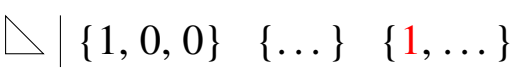


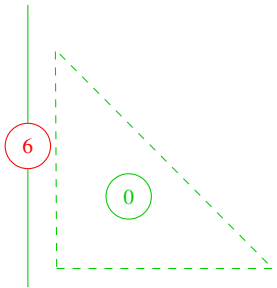
$\{1, 0, \ldots\}$ $\{\ldots\}$ $\{\ldots\}$

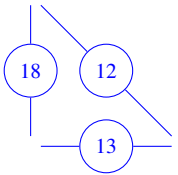

0
4
13
14
{1, 0, 0} {...} {...}


6
5
0
4

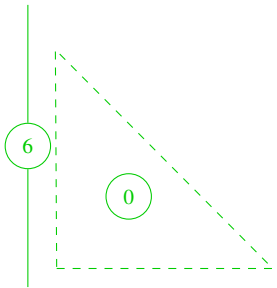

18
12
13

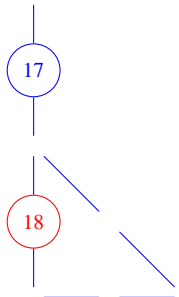

{1, 0, 0} {. . . } {1, . . . }



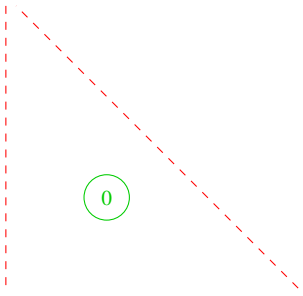


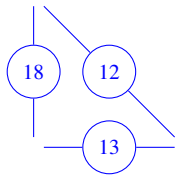
$\{1, 0, 0\} \quad \{\dots\} \quad \{1, 2, \dots\}$



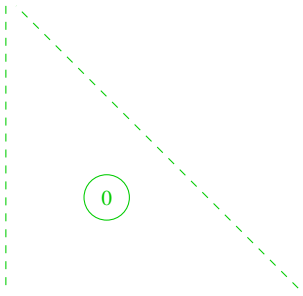


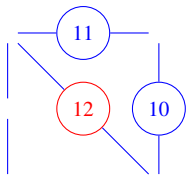
$\{1, 0, 0\} \quad \{\dots\} \quad \{1, 2, 1\}$



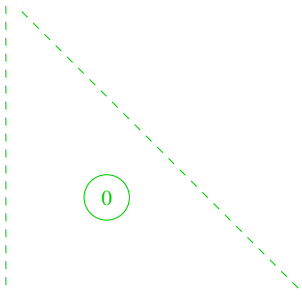


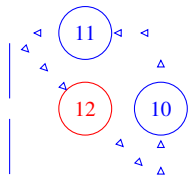
$\{1, 0, 0\}$ $\{0, \ldots\}$ $\{1, 2, 1\}$





$\{1, 0, 0\} \quad \{0, 2, \ldots\} \quad \{1, 2, 1\}$

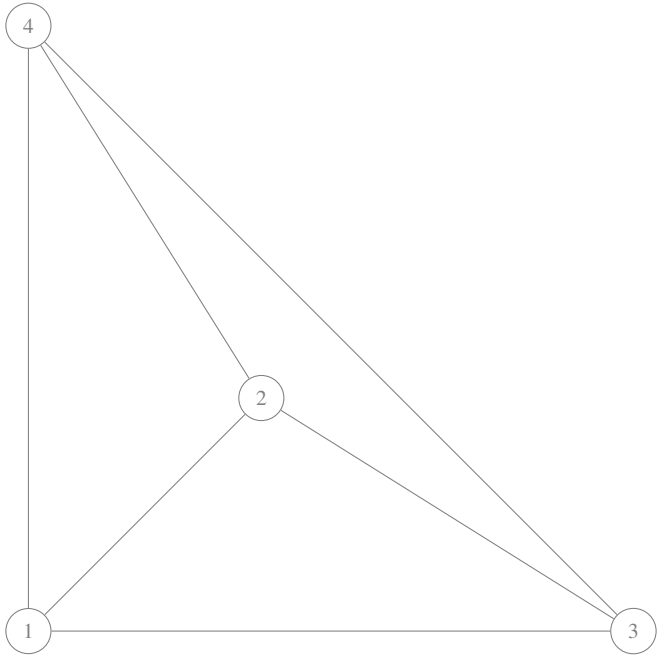


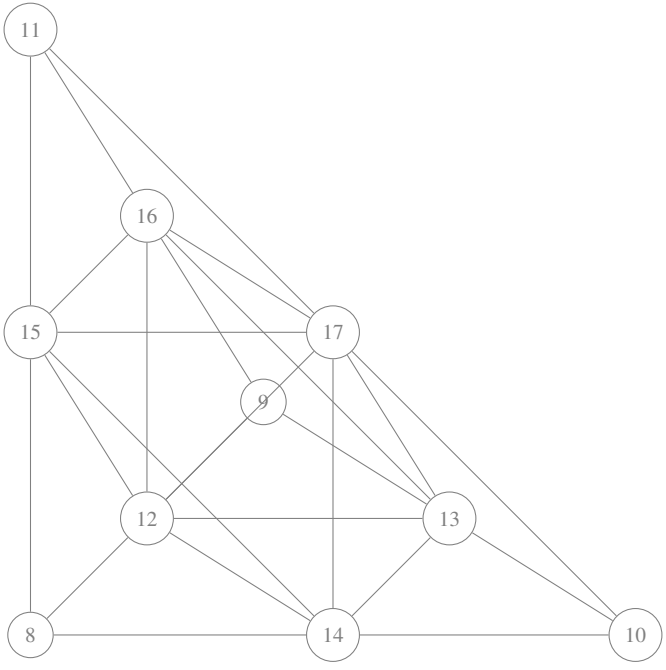


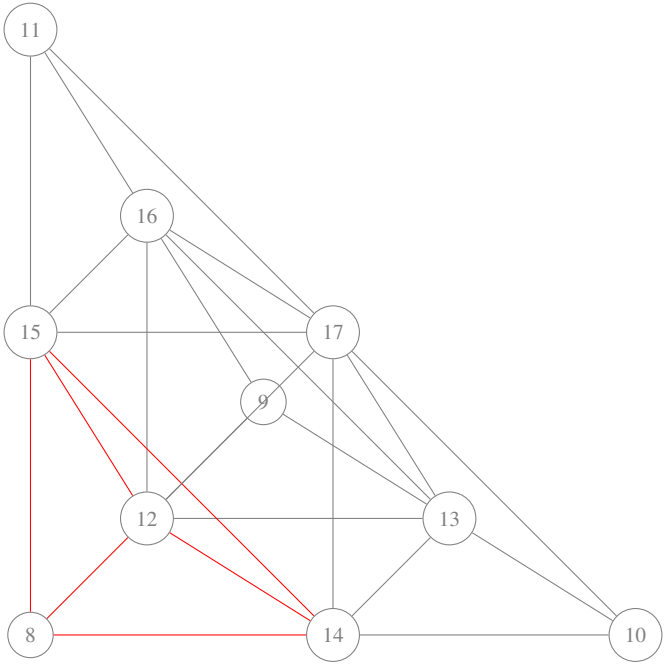
{1, 0, 0} {0, 2 (-1) } {1, 2, 1}

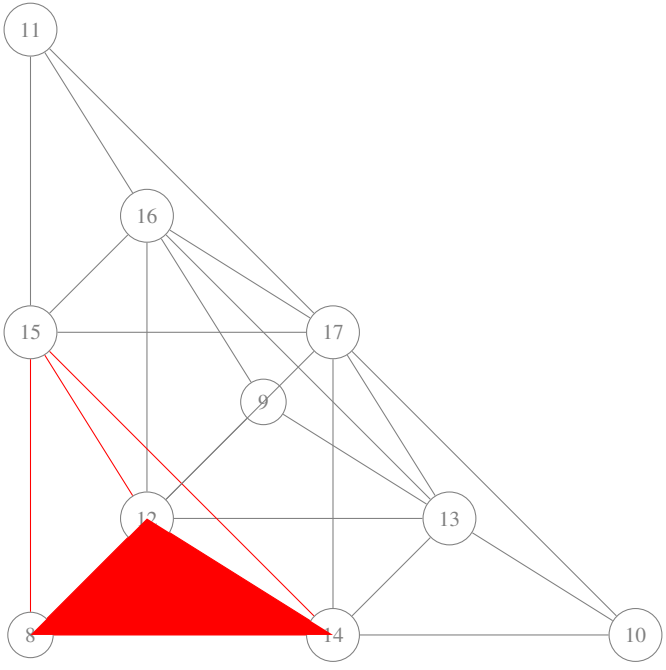
•	{}					
—	•	(1, 0, 0)	•	(0, 0)		
—	•	(0, 0)	•	(1, 1, 0)		

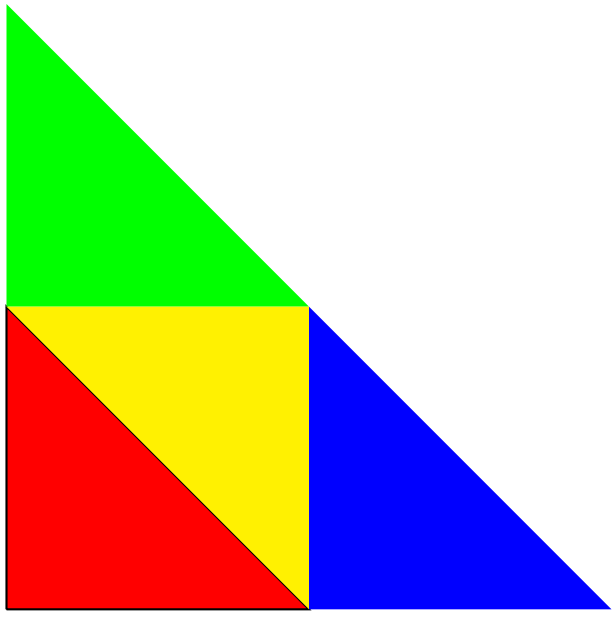
—	•	(1, 0, 0)	•	(1, 1, 0)		
—	•	(1, 1, 0)	•	(1, 2, 0)		
—	•	(1, 2, 0)	•	(1, 0, 0)		
◺	—	(1, 0, 0)	—	(0, 2 (-1))	—	(1, 2, 1)
◺	—	(1, 0, 1)	—	(1, 1, 0)	—	(0, 0 (-1))
◺	—	(0, 1 (-1))	—	(1, 1, 1)	—	(1, 2, 0)
◺	—	(0, 0)	—	(0, 1)	—	(0, 2)

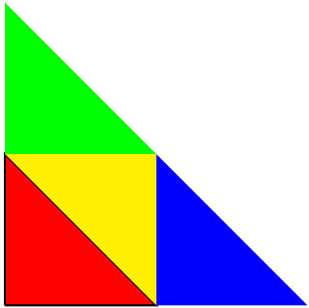
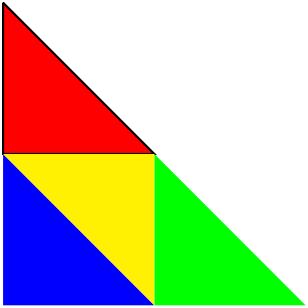
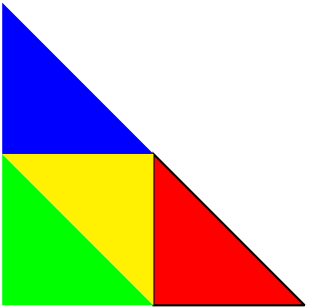
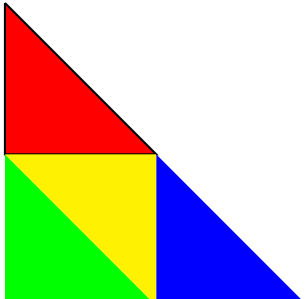
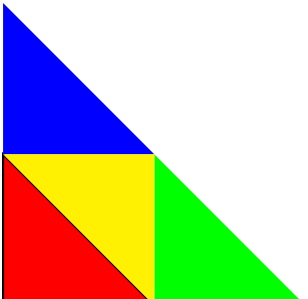
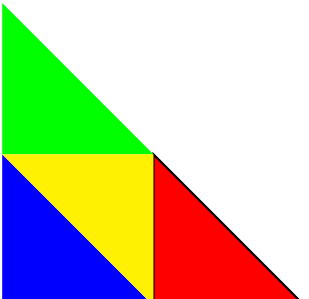

4
2
1
3

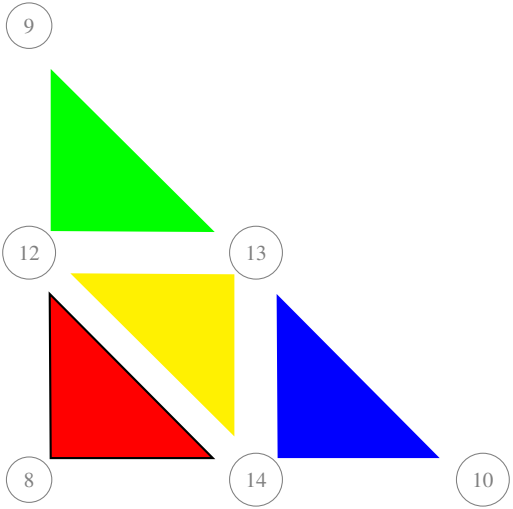

11
16
15
17
9
12
13
8
14
10


11
16
15
17
9
12
13
8
14
10

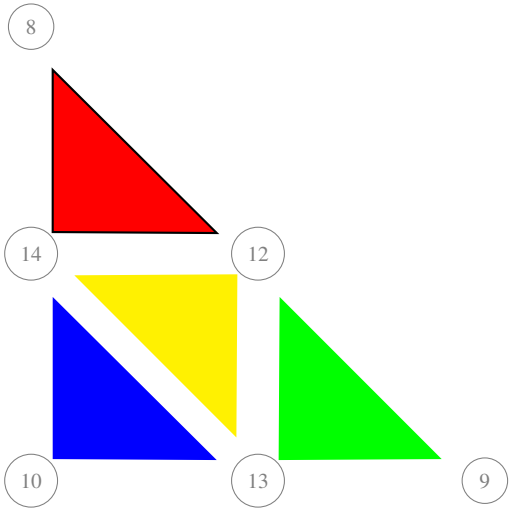

11
16
15
17
9
12
13
8
14
10



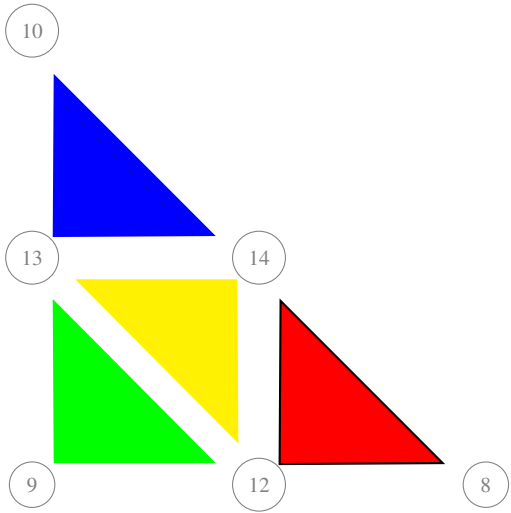









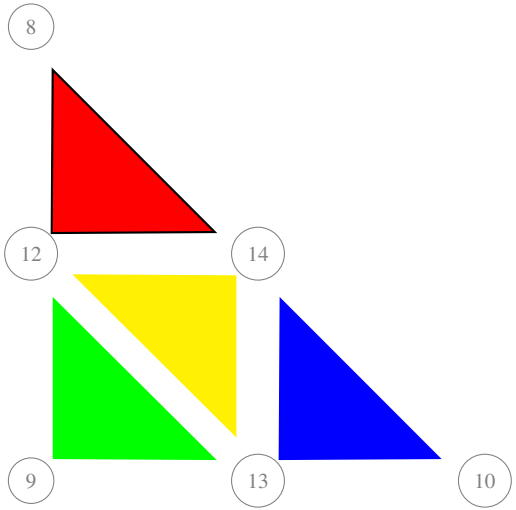
$$(0, 0) \longrightarrow (0, 0)$$



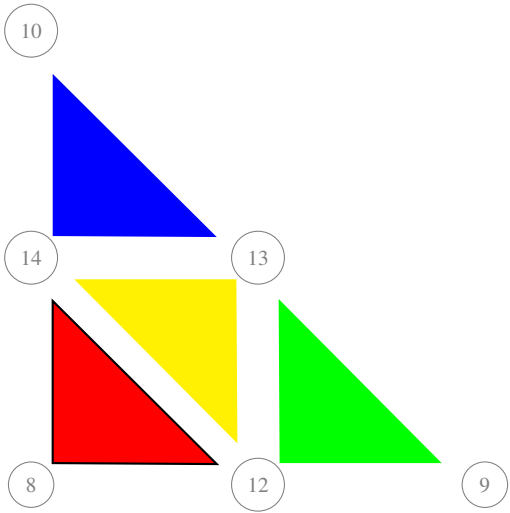
$(0, 0) \longrightarrow (1, 1)$



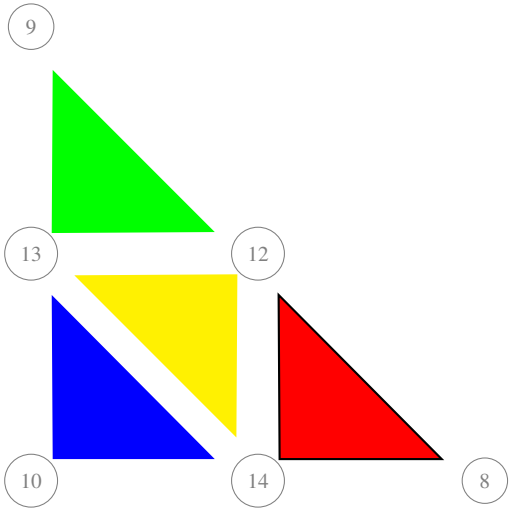
$(0, 0) \longrightarrow (2, 2)$



$(0, 0) \longrightarrow (2, -1)$



$(0, 0) \longrightarrow (0, -2)$

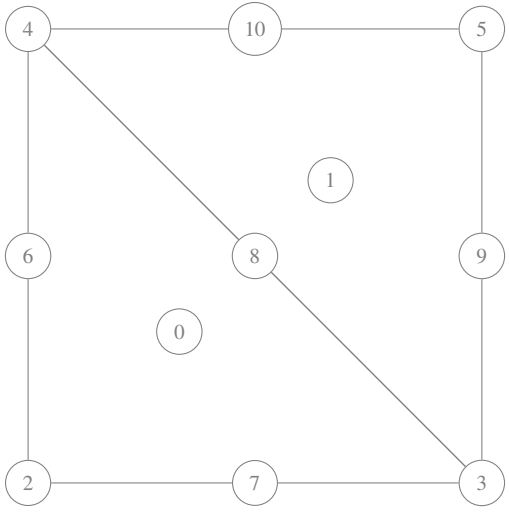


$(0, 0) \longrightarrow (1, -3)$

Outline

Doublet

Triangle



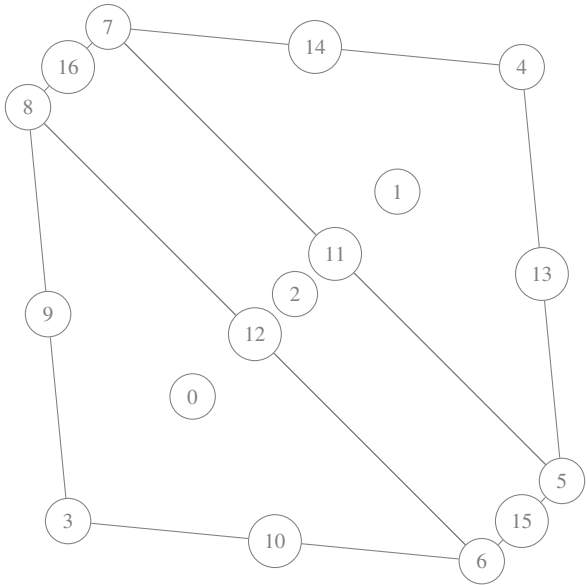
Doublet

Triangle

```
./src/dm/impls/plex/tests/ex69
  -dm_plex_box_faces 1,1
    -dm_plex_cohesive_label_fault 8
  -dm_refine 1
    -dm_plex_transform_type cohesive_extrude
    -dm_plex_transform_cohesive_width 0.25
    -dm_plex_transform_active fault
  -coarse_dm_view :tri.tex:ascii_latex
  -dm_view :trif.tex:ascii_latex
  -dm_plex_view_tikzscale 3.
```

Doublet

Triangle



Algorithm Overview

1. Label interior faces to extrude
2. Orient labeled submanifold
3. Add impinging points to the label
4. Give a refine type to each mesh point
5. Create transformation table for each type
6. Execute transformation

Mesh Orientation

The orientation algorithm

- works in parallel,
- works on submeshes
- can be constrained by a label

Refine Types

rt	Action
$\text{ct} * 2 + 0$	Identity
$\text{ct} * 2 + 1$	Unsplit point
$(\text{ct} * 2 + 0) * 100 + F$	Impinging point, split some faces
$(\text{ct} * 2 + 1) * 100 + F$	Split point, unsplit some faces

F has bit f set if face f participates

Refine Types

Vertex

Type	Eqn	rt
unsplit vertex	$ct * 2 + 1$	1
split vertex	$(ct * 2 + 1) * 100 + 0$	100

Refine Types

Segment

Type	Eqn	rt
unsplit segment	$\mathrm{ct} * 2 + 1$	3
split both endpoints	$(\mathrm{ct} * 2 + 1) * 100 + 0$	300
unsplit endpoint 0	$(\mathrm{ct} * 2 + 1) * 100 + 1$	301
unsplit endpoint 1	$(\mathrm{ct} * 2 + 1) * 100 + 2$	302
impinging endpoint 0	$(\mathrm{ct} * 2 + 0) * 100 + 1$	201
impinging endpoint 1	$(\mathrm{ct} * 2 + 0) * 100 + 2$	202

Refine Types

Triangle

Type	Eqn	rt
split all faces	$(\text{ct} * 2 + 1) * 100 + 0$	700
unsplit edge 0	$(\text{ct} * 2 + 1) * 100 + 1$	701
unsplit edge 1	$(\text{ct} * 2 + 1) * 100 + 2$	702
unsplit edge 0 1	$(\text{ct} * 2 + 1) * 100 + 3$	703
unsplit edge 2	$(\text{ct} * 2 + 1) * 100 + 4$	704
unsplit edge 0 2	$(\text{ct} * 2 + 1) * 100 + 5$	705
unsplit edge 1 2	$(\text{ct} * 2 + 1) * 100 + 6$	706
impinging edge 0	$(\text{ct} * 2 + 0) * 100 + 1$	601
impinging edge 1	$(\text{ct} * 2 + 0) * 100 + 2$	602
impinging edge 0 1	$(\text{ct} * 2 + 0) * 100 + 3$	603
impinging edge 2	$(\text{ct} * 2 + 0) * 100 + 4$	604
impinging edge 0 2	$(\text{ct} * 2 + 0) * 100 + 5$	605
impinging edge 1 2	$(\text{ct} * 2 + 0) * 100 + 6$	606

Refine Types

Quadrilateral

Type	Eqn	rt
split all faces	$(\text{ct} * 2 + 1) * 100 + 0$	900
unsplit edge 0	$(\text{ct} * 2 + 1) * 100 + 1$	901
unsplit edge 1	$(\text{ct} * 2 + 1) * 100 + 2$	902
⋮	⋮	⋮
unsplit edge 2 3	$(\text{ct} * 2 + 1) * 100 + 12$	912
unsplit edge 0 2 3	$(\text{ct} * 2 + 1) * 100 + 13$	913
unsplit edge 1 2 3	$(\text{ct} * 2 + 1) * 100 + 14$	914
impinging edge 0	$(\text{ct} * 2 + 0) * 100 + 1$	801
impinging edge 1	$(\text{ct} * 2 + 0) * 100 + 2$	802
⋮	⋮	⋮
impinging edge 2 3	$(\text{ct} * 2 + 0) * 100 + 12$	812
impinging edge 0 2 3	$(\text{ct} * 2 + 0) * 100 + 13$	813
impinging edge 1 2 3	$(\text{ct} * 2 + 0) * 100 + 14$	814

Refine Types

Tetrahedron

Type	Eqn	rt
impinging face 0	$(\mathrm{ct} * 2 + 0) * 100 + 1$	1201
impinging face 1	$(\mathrm{ct} * 2 + 0) * 100 + 2$	1202
impinging face 0 1	$(\mathrm{ct} * 2 + 0) * 100 + 3$	1203
impinging face 2	$(\mathrm{ct} * 2 + 0) * 100 + 4$	1204
impinging face 0 2	$(\mathrm{ct} * 2 + 0) * 100 + 5$	1205
impinging face 1 2	$(\mathrm{ct} * 2 + 0) * 100 + 6$	1206
impinging face 0 1 2	$(\mathrm{ct} * 2 + 0) * 100 + 7$	1207
impinging face 3	$(\mathrm{ct} * 2 + 0) * 100 + 8$	1208
impinging face 0 3	$(\mathrm{ct} * 2 + 0) * 100 + 9$	1209
impinging face 1 3	$(\mathrm{ct} * 2 + 0) * 100 + 10$	1210
impinging face 0 1 3	$(\mathrm{ct} * 2 + 0) * 100 + 11$	1211
impinging face 2 3	$(\mathrm{ct} * 2 + 0) * 100 + 12$	1212
impinging face 0 2 3	$(\mathrm{ct} * 2 + 0) * 100 + 13$	1213
impinging face 1 2 3	$(\mathrm{ct} * 2 + 0) * 100 + 14$	1214

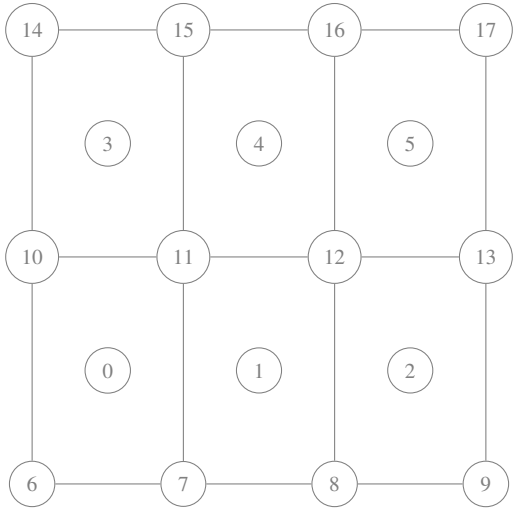
Refine Types

Hexahedron

There are 62 cases

Embedded Fault

Quadrilateral



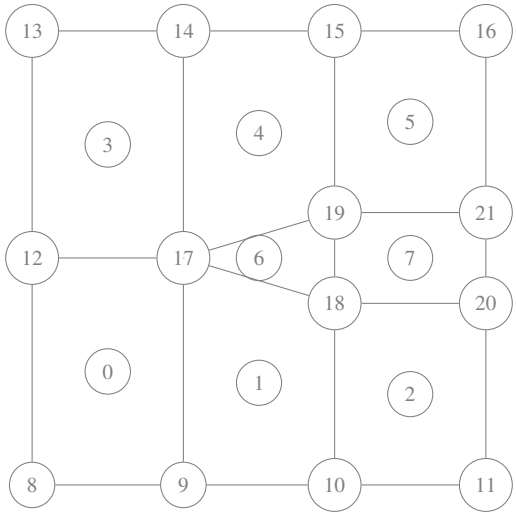
Embedded Fault

Quadrilateral

```
./src/dm/impls/plex/tests/ex69
  -dm_plex_simplex 0
    -dm_plex_box_faces 3,2
    -dm_plex_cohesive_label_fault 22,23
  -dm_refine 1
    -dm_plex_transform_type cohesive_extrude
    -dm_plex_transform_active fault
    -dm_plex_transform_cohesive_width 0.2
  -coarse_dm_view :quad.tex:ascii_latex
  -dm_view :quadf.tex:ascii_latex
  -dm_plex_view_tikzscale 3.
  -dm_plex_view_numbers_depth 1,0,1
```

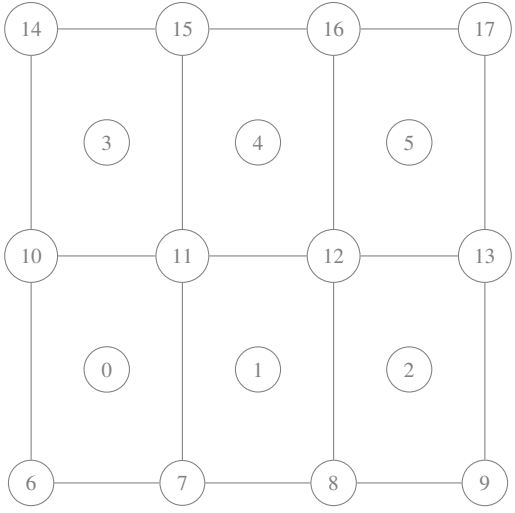
Embedded Fault

Quadrilateral



T-junction

Quadrilateral



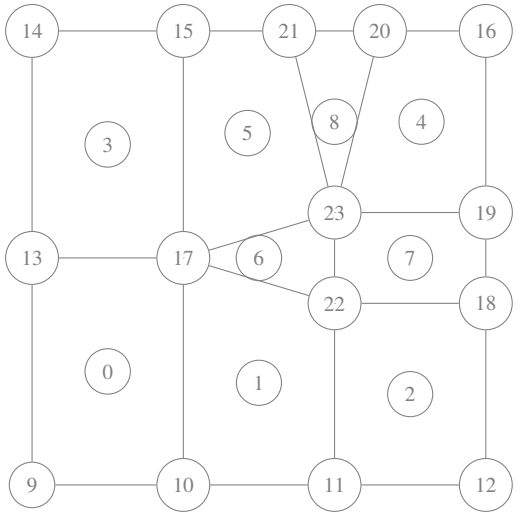
T-junction

Quadrilateral

```
./src/dm/impls/plex/tests/ex69
  -dm_plex_simplex 0
    -dm_plex_box_faces 3,2
    -dm_plex_cohesive_label_fault0 22,23
    -dm_plex_cohesive_label_fault1 32
  -f0_dm_refine 1
    -f0_dm_plex_transform_type cohesive_extrude
    -f0_dm_plex_transform_active fault0
    -f0_dm_plex_transform_cohesive_width 0.2
  -f1_dm_refine 1
    -f1_dm_plex_transform_type cohesive_extrude
    -f1_dm_plex_transform_active fault1
    -f1_dm_plex_transform_cohesive_width 0.2
  -f0_coarse_dm_view :quad.tex:ascii_latex
  -dm_view :quadf.tex:ascii_latex
  -dm_plex_view_tikzscale 3. -dm_plex_view_numbers_depth 1,0,1
```

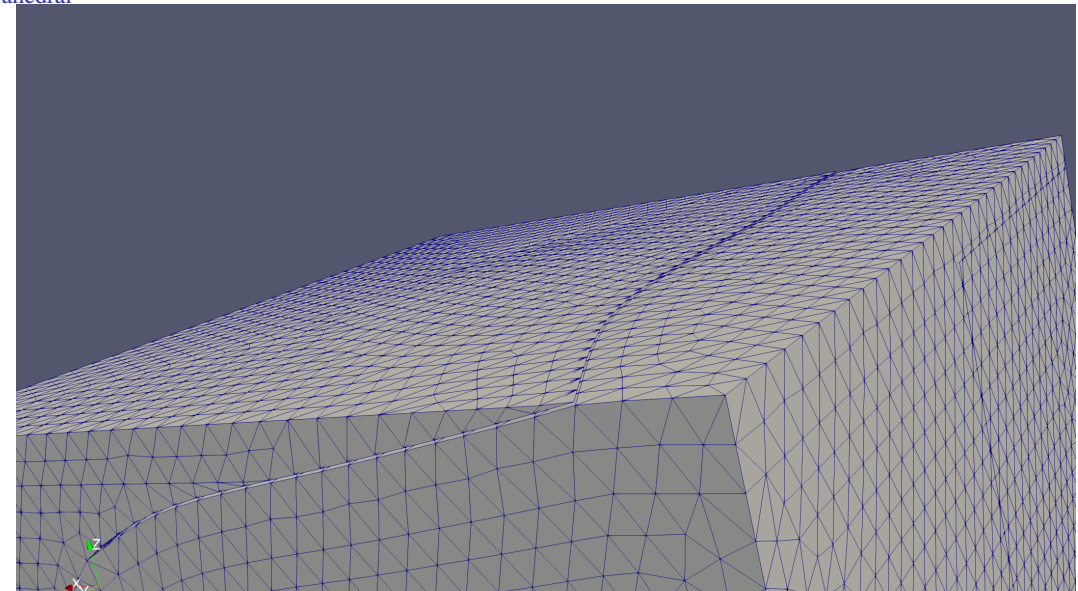
T-junction

Quadrilateral



PyLith Subduction

Tetrahedral



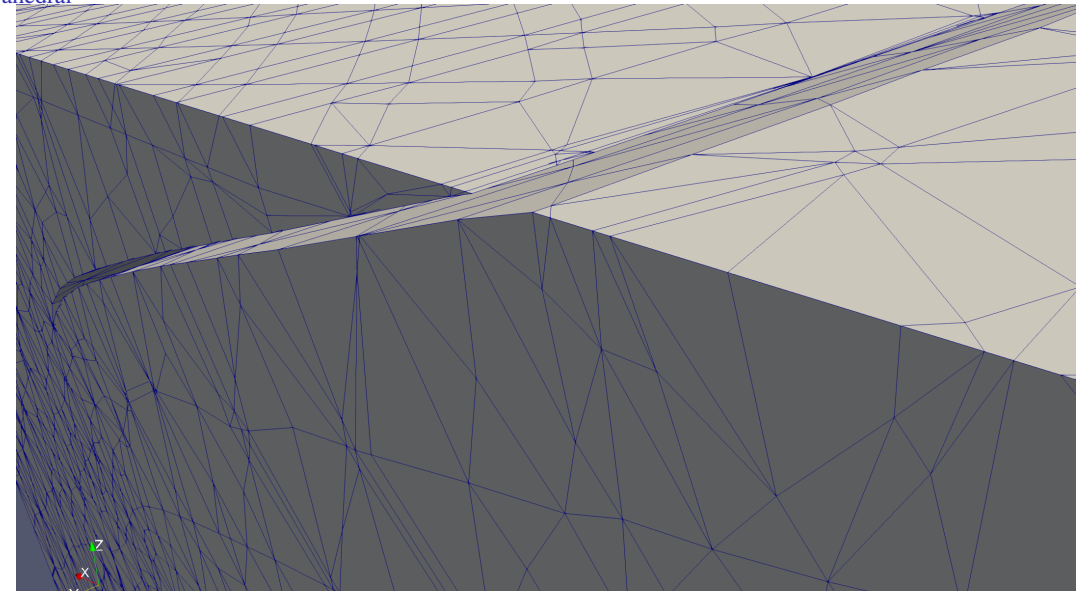
PyLith Subduction

Tetrahedral

```
./src/dm/impls/plex/tests/ex69
  -dm_plex_filename
    ${PYLITH_DIR}/examples/subduction-3d/input/mesh_tet.exo
  -f0_dm_refine 1
    -f0_dm_plex_transform_type cohesive_extrude
    -dm_plex_cohesive_label_Vertex\ Sets 10
    -f0_dm_plex_transform_active "Vertex␣Sets10"
    -f0_dm_plex_transform_cohesive_width 2000
    -f0_dm_plex_transform_extrude_use_tensor 0
  -dm_view hdf5:mesh.h5
```

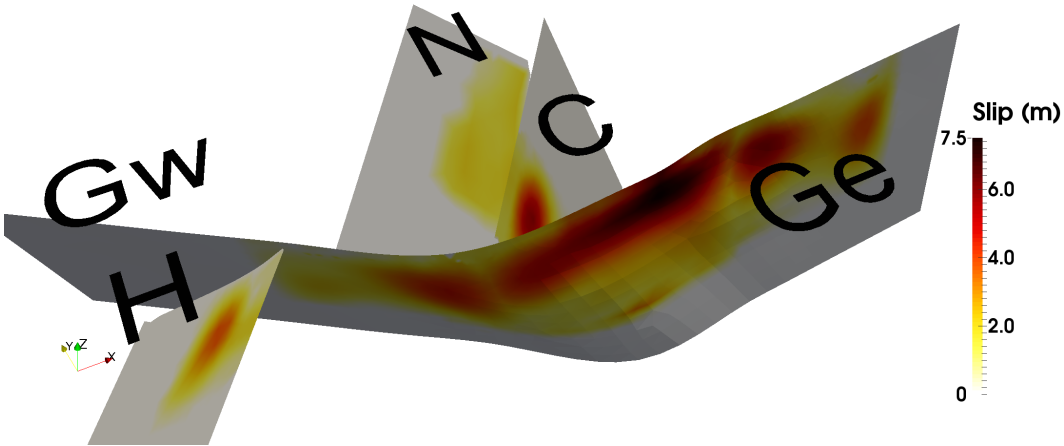
PyLith Subduction

Tetrahedral



PyLith Subduction

Tetrahedral



We can dynamically select:

- FEM discretization
- Particle layout
- Background mesh
- Poisson solver
- Projection solver
- Time integrators

References I