Netgen-DMPlex and FEniCSx

P. E. Farrell*, U. Zerbinati*

* *Mathematical Institute*
University of Oxford

FEniCS 2023, Cagliari, 14th of June 2023

UNIVERSITY OF OXFORD
Mathematical Institute

Oxford Mathematics

NetGen-DMPlex

University of Oxford Mathematical Institute

January 2023 — First NetGen to DMPlex interface for Firedrake.

NetGen-DMPlex

Mathematical Institute

January 2023	First NetGen to DMPlex interface for Firedrake.
March 2023	PETSc DMPlex Transform wrapped in petsc4py.

NetGen-DMPlex

University of Oxford Mathematical Institute

- January 2023 — First NetGen to DMPlex interface for Firedrake.
- March 2023 — PETSc DMPlex Transform wrapped in petsc4py.
- April 2023 — DMPlex to NetGen interface for NGSolve.

NetGen-DMPlex

UNIVERSITY OF OXFORD
Mathematical Institute

- January 2023 — First NetGen to DMPlex interface for Firedrake.
- March 2023 — PETSc DMPlex Transform wrapped in petsc4py.
- April 2023 — DMPlex to NetGen interface for NGSolve.
- May 2023 — NGSolve-DMPlex used in Firedrake master.

NetGen-DMPlex

University of Oxford Mathematical Institute

- January 2023 — First NetGen to DMPlex interface for Firedrake.
- March 2023 — PETSc DMPlex Transform wrapped in petsc4py.
- April 2023 — DMPlex to NetGen interface for NGSolve.
- May 2023 — NGSolve-DMPlex used in Firedrake master.
- June 2023 — NGSolve-DMpPlex can be used in FEniCSx?

NetGen-DMPlex

University of Oxford Mathematical Institute

- January 2023 — First NetGen to DMPlex interface for Firedrake.
- March 2023 — PETSc DMPlex Transform wrapped in petsc4py.
- April 2023 — DMPlex to NetGen interface for NGSolve.
- May 2023 — NGSolve-DMPlex used in Firedrake master.
- June 2023 — NGSolve-DMpPlex can be used in FEniCSx?

Why NetGen?

NetGen is an advancing front 2D/3D-mesh generator, with many interesting features. Among the most important:

- Python wrapping (through pybind11),
- Multiple ways of describing the geometry to be meshed, i.e. its builtin **Constructive Solid Geometry** (**CSG**) and the **Open Cascade Technology** (**OCCT**) geometry kernel,
- Supports mesh refinement (also anisotropic mesh refinement).

Getting Started – Installing NetGen

UNIVERSITY OF OXFORD

Mathematical Institute

NetGen

```
pip install --pre ngsolve
```

NetGen/NGSolve can also be installed from source, just but be careful to link against the correct MPI.

PETSc

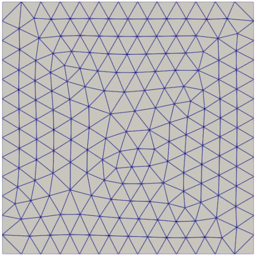
If you are using an external PETSc installation, it should be updated to include commit `654059db` and `text4aa747ac`.

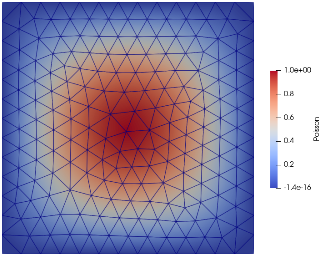
Getting Started – Unstructured Mesh

UNIVERSITY OF OXFORD
Mathematical Institute

```python
import netgen.gui
from netgen.geom2d import SplineGeometry
geo = SplineGeometry()
geo.AddRectangle((0,0),(np.pi,np.pi))
domain  = ngsio.model_to_mesh(geo, MPI.COMM_WORLD,
    hmax=0.25, gdim=2)
V = FunctionSpace(domain, ("CG", 3))
u = Function(V, dtype=default_scalar_type)
u.interpolate(lambda x: x[0]*x[1])
integrand = form(u*dx)
print(assemble_scalar(integrand))
with XDMFFile(domain.comm, "XDMF/triang.xdmf", "w") as
     xdmf:
    xdmf.write_mesh(domain)
```

Getting Started – Unstructured Mesh

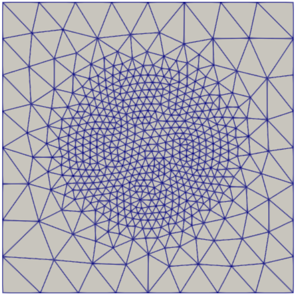




Getting Started – CSG 2D


```python
from netgen.geom2d import SplineGeometry
geo = SplineGeometry()
geo.AddRectangle(p1=(-1,-1),p2=(1,1),bc="rectangle",
     leftdomain=1,rightdomain=0)
geo.AddCircle(c=(0,0),r=0.5,bc="circle",
     leftdomain=2,rightdomain=1)
geo.SetMaterial (1, "outer")
geo.SetMaterial (2, "inner")
geo.SetDomainMaxH(2, 0.05)
domain  = ngsio.model_to_mesh(geo, MPI.COMM_WORLD, hmax=0.25, gdim=2)
with XDMFFile(domain.comm, "XDMF/csg2D.xdmf", "w") as xdmf:
     xdmf.write_mesh(domain)
```

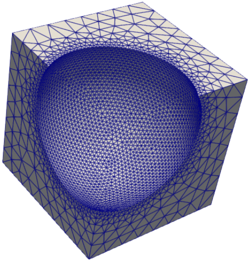
Getting Started – CSG 2D



Getting Started – CSG 3D

Mathematical Institute

```python
from netgen.csg import CSGeometry, Sphere, OrthoBrick, Pnt
geo = CSGeometry()
brick = OrthoBrick(Pnt(-1,-1,-1),Pnt(1,1,1))
sphere = Sphere(Pnt(0.5,0.5,0.5),1)
sphere.maxh(0.05)
geo.Add(brick-sphere)
domain  = ngsio.model_to_mesh(geo, MPI.COMM_WORLD, hmax=0.25, gdim=3)
with XDMFFile(domain.comm, "XDMF/csg3D.xdmf", "w") as xdmf:
    xdmf.write_mesh(domain)
```



The Open Cascade Technology Kernel

- Basic OCCT objects can be used in NetGen such as: `Box`, `Cylinder`, `Point`, `Segment` and `ArcOfCircle`.
- The `fuse`, `cut` and `common` operations between OCCT objects have been wrapped in NetGen.
- Transformation operations such as `Move` and `Rotate` have also been wrapped into NetGen.

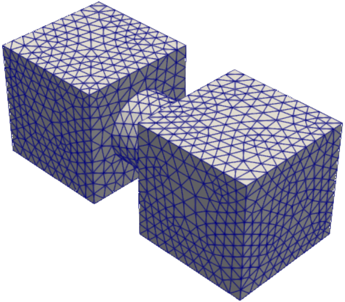
The Open Cascade Technology Kernel

UNIVERSITY OF OXFORD
Mathematical Institute

```python
from netgen.csg import CSGeometry, Sphere, OrthoBrick,
    Pnt
box = Box(Pnt(0,0,0), Pnt(1,1,1))
cyl = Cylinder(Pnt(1,0.5,0.5), X, r=0.3, h=0.5)
solid1 = box + cyl
solid2 = solid1.Rotate(Axis((0,0,0),Y),180).Move
    ((2.5,0.,1.))
solid = solid2 + solid1
geo = OCCGeometry(solid)
domain  = ngsio.model_to_mesh(geo, MPI.COMM_WORLD,
    hmax=0.25, gdim=3)
with XDMFFile(domain.comm, "XDMF/csg3D.xdmf", "w") as
    xdmf:
    xdmf.write_mesh(domain)
```

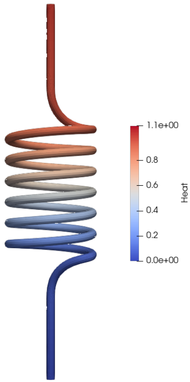
The Open Cascade Technology Kernel

UNIVERSITY OF OXFORD
Mathematical Institute



The OCCT Kernel – Poisson Problem

```
cyl = Cylinder((0,0,0), Z, r=0.01, h=0.03).faces[0]
heli = Edge(Segment((0,0), (12*pi, 0.03)), cyl)
ps,vs,pe,ve = heli.start, heli.start_tangent, heli.end, heli.end_tangent
e1 = Segment((0,0,-0.03), (0,0,-0.01))
c1 = BezierCurve( [(0,0,-0.01), (0,0,0), ps-vs, ps])
e2 = Segment((0,0,0.04), (0,0,0.06))
c2 = BezierCurve( [pe, pe+ve, (0,0,0.03), (0,0,0.04)])
spiral = Wire([e1, c1, heli, c2, e2])
circ, coil = Face(Wire([Circle((0,0,-0.03), Z, 0.001)])), Pipe(spiral, circ)
geo = OCCGeometry(coil)
```

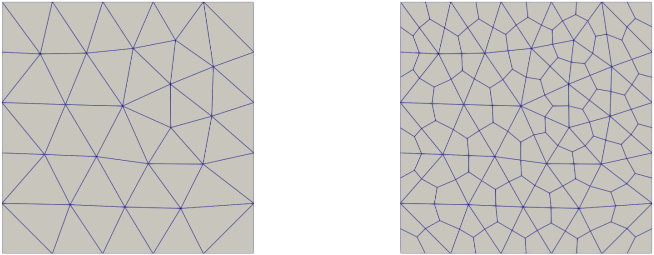


PETSc Transform – Quad Mesh


```python
import netgen.gui
from netgen.geom2d import SplineGeometry
geo = SplineGeometry()
geo.AddRectangle((0,0),(1,1))
tr = PETSc.DMPlexTransform().create(comm=PETSc.COMM_WORLD)
tr.setType(PETSc.DMPlexTransformType.REFINETOBOX)
domain  = ngsio.model_to_mesh(geo, MPI.COMM_WORLD, hmax=0.2, gdim=2, transform=tr)
with XDMFFile(domain.comm, "XDMF/quad.xdmf", "w") as xdmf:
     xdmf.write_mesh(domain)
```

PETSc Transform – Quad Mesh

UNIVERSITY OF OXFORD
Mathematical Institute

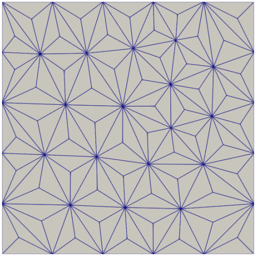


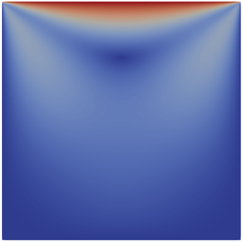
PETSc Transform – Barycentric Mesh

University of Oxford
Mathematical Institute

```
import netgen.gui
from netgen.geom2d import SplineGeometry
geo = SplineGeometry()
geo.AddRectangle((0,0),(1,1))
tr = PETSc.DMPlexTransform().create(comm=PETSc.
    COMM_WORLD)
tr.setType(PETSc.DMPlexTransformType.REFINEALFELD)
msh = ngsio.model_to_mesh(geo, MPI.COMM_WORLD, hmax
    =0.01, gdim=2, transform=tr)
```

PETSc Transform – Barycentric Refinement



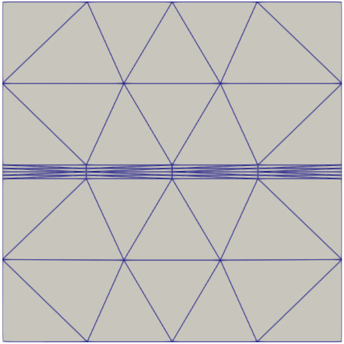


Anisotropic Refinement

Mathematical Institute

```
geo = CSGeometry()
box = OrthoBrick(Pnt(0,0,0),Pnt(1,1,1))
top = Plane(Pnt(0,0,0.52),Vec(0,0,1))
bot = Plane(Pnt(0,0,0.48),Vec(0,0,-1))
plate = box * top * bot
geo.Add((box-top).mat("air"))
geo.Add(plate.mat("plate"))
geo.Add((box-bot).mat("air"))
geo.CloseSurfaces(bot,top)
nmesh = geo.GenerateMesh(maxh=0.25)
def ngs_routine(msh,geo):
    ZRefinement(msh,geo); msh.Split2Tets();
    return msh, geo
domain  = ngsio.model_to_mesh(geo, MPI.COMM_WORLD,
    hmax=0.25, gdim=3,transform=None,routine=
    ngs_routine)
```

Anisotropic Refinement



Conclusion

UNIVERSITY OF OXFORD
Mathematical Institute

- It is now possible to describe the geometry using NetGen, this allows for **anisotropic mesh** refinement.
- It is possible to use **Open Cascade** to describe a geometry, through NetGen.
- It is now possible to use DMPlex transformation in two and three dimensions. Such as **Alfeld, Powell-Sabin** refinements.