1 The `@gmshSurf` class definition

`@gmshSurf` creates a file for the open source meshing program `GMSH`, the following assumes a knowledge of `GMSH`.

Definition

The `@gmshSurf` class creates an object with the following fields:

```
gms.dpath=dpath; % full path and filename /home/user/test.geo
gms.pnts=[];
gms.pntsC={};
gms.pN=0;
gms.pNC=0;
gms.lines=[];
gms.lineloops=[];
gms.surfaces=[];
gms.surfaceloops=[];
gms.volume=[];
gms.physicalline={};
gms.physicalsurface={};
gms.physicalvolume={};
gms.comments=[];
gms.pointINsurfaces=[];
gms.periodicpoints=[];
```

With associated functions: `addlineloops.m, addpntsC.m, addsurfaceloops.m, addsurfC.m, addvolume.m, display.m, gmshSurf.m, rungmshSurf.m, translateSurf.m, writeComp.m, addlines.m, addpnts.m, addsurfaces.m, addsurf.m, createsurface.m, get.m, readmsh.m, set.m, writeC.m` and `write.m`.

It is simplist to demonstart the usage through example.

Simple Example

Creating the object:

```
$dpath=[pwd ’’basicobject.geo’]; %linux path stucture
$A=gmshSurf(dpath);
$mshv=0.5; % Set the general mesh resolution for all points
$mshvd=0.25; % Set the mesh resolution for point d
$sqz=2;
$a=[-sqz -sqz 0 mshv];
$b=[-sqz sqz 0 mshv];
$c=[ sqz sqz 0 mshv];
$d=[ sqz -sqz 0 mshvd];
$A=addpnts(A, [a]); % adds 1 point
$A=addpnts(A, [b; c; d]); % add multiple points
$write(A);% write out the object to the .geo file
```

This writes out the following to the `basicobject.geo` file:

```
// Variables
Point(1)={-2,-2,0,0.5};
Point(2)={-2,2,0,0.5};
Point(3)={2,2,0,0.5};
Point(4)={2,-2,0,0.25};
```

Finishing off creating the square:

```
% We now can add some lines
% Available linetypes:  Line, Spline, BSpline, Circle and Ellipse
$ltype='Line';
$[A lna]=addlines(A,[1 2],ltype); % line 1
$[A lnb]=addlines(A,[2 3],ltype); % line 2
$[A lnc]=addlines(A,[3 4],ltype); % line 3
$[A lnd]=addlines(A,[4 1],ltype); % line 4
% Now we can create a lineloop out of the lines
$[A lnloopa]=addlineloops(A,[lna lnb lnc lnd]); % lineloop 1
% and a surface from the lineloop
% Available surfacetypes:  Plane, Ruled, Compound and Physical
$stype='Plane';
$[A surfa]=addsurfaces(A,lnloopa,stype);
$write(A);
```

The additional code in the `basicobject.geo` file is:

```
// Lines, Splines, BSplines, Circles and Ellipses
Line(1)={1,2};
Line(2)={2,3};
Line(3)={3,4};
Line(4)={4,1};

// Line Loops
Line Loop(1)={1  ,2  ,3  ,4};

// Surfaces
Plane Surface(1)={1};
```

If desired `GMSH` can be run directly, creating the `.msh` file and returning a `meshout` object with fields: `meshout.v` *(Vertices)* `meshout.ed` *(Edges)* and `meshout.tr` *(Triangles)*, as described in the `GMSH` documentation.

```
$meshout=rungmshSurf(A);
```

The final outputed mesh is given in fig. 1.

Curved Surface

This allows complex shapes to be easily created:

```
$A=gmshSurf(dpath);
% Let's create a curved surface y=y0*sin2̂(x/x0) x:  0 -> x0*pi/2
```

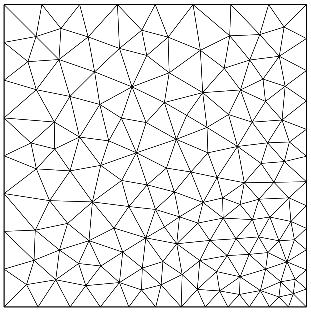

Figure 1: Basic meshed example using `@gmshSurf` and `GMSH`.

```
$mshv=0.5; % Set the general mesh resolution for all points
$N=20;
$y0=4;
$x0=5;
$x=linspace(0,x0*pi/2,N);
$y=y0*sin(x/x0).2̂;
$pnts=[x' y' zeros(length(x),1) mshv*ones(length(x),1)];
$A=addpnts(A, pnts);
$endpnt=[x(end) y(1) 0 mshv];
$[A pN]=addpnts(A, endpnt);
$ltype='lines';
$lnum=1;
$for n=2:pN, A=set(A,ltype,lnum,[n-1 n],'l');lnum=lnum+1; end;
$A=set(A,'lines',lnum,[n 1],'l');lnum=lnum+1; % put in the last point
$A=set(A,'lineloops',1:(lnum-1));
$snum=1;
$A=set(A,'surfaces',0,1,'p');snum=snum+1;
$write(A);
$meshout=rungmshSurf(A);
```

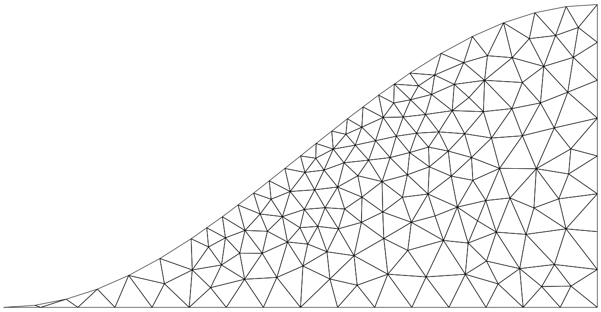

Figure 2: Basic meshed example using `@gmshSurf` and `GMSH`.

2 The @gmsh class definition

Definition

The @gmsh class allows us combine several @gmshSurf objects, the class contains the following fields:

```
Gms.name = name;
Gms.gmsurfs = [];
```

Simple Example

A simple example combining two gmsh surfaces:

```
$G=gmsh(dpath);
$A=gmshSurf;
$mshv=0.5; % Set the general mesh resolution for all points
$mshvd=0.25; % Set the mesh resolution for point d
$sqz=2;
$a=[-sqz -sqz 0 mshv];
$b=[-sqz sqz 0 mshv];
$c=[ sqz sqz 0 mshv];
$d=[ sqz -sqz 0 mshvd];
$A=addpnts(A, [a; b; c; d]); % add multiple points
$ltype='Line';
$[A lna]=addlines(A,[1 2],ltype); % line 1
$[A lnb]=addlines(A,[2 3],ltype); % line 2
$[A lnc]=addlines(A,[3 4],ltype); % line 3
$[A lnd]=addlines(A,[4 1],ltype); % line 4
$[A lnloopa]=addlineloops(A,[lna lnb lnc lnd]); % lineloop 1
$stype='Plane';
$[A surfa]=addsurfaces(A,lnloopa,stype);
% Add surfaces A to the larger object
$G=addsurface(G,A);
$B=gmshSurf;
$a=a+[2*sqz 0 0 -mshvd];
$b=b+[2*sqz 0 0 0];
$c=c+[2*sqz 0 0 0];
$d=d+[2*sqz 0 0 mshvd];
$B=addpnts(B, [a; b; c; d]); % add multiple points
$ltype='Line';
$[B lna]=addlines(B,[1 2],ltype); % line 1
$[B lnb]=addlines(B,[2 3],ltype); % line 2
$[B lnc]=addlines(B,[3 4],ltype); % line 3
$[B lnd]=addlines(B,[4 1],ltype); % line 4
$[B lnloopa]=addlineloops(B,[lna lnb lnc lnd]); % lineloop 1
$stype='Plane';
```

```
$[B surfa]=addsurfaces(B,lnloopa,stype);
% Add surfaces B to the larger object
$G=addsurface(G,B);
% We can now write out the compisite object
$write(G);
% We can now directly run GMSH
$meshout=rungmsh(G);
```

The combined surfaces are shown in fig. 3:

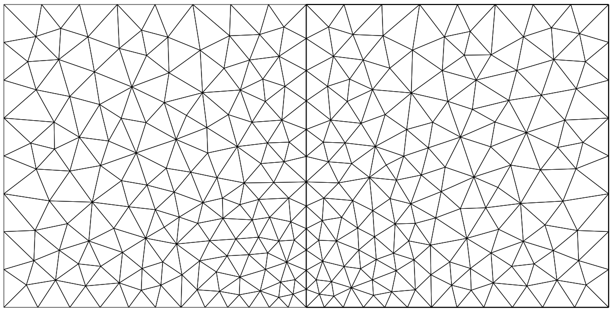

Figure 3: Basic meshed example using `@gmsh` and `GMSH`.