Lukas Bundrock, Alexandre Girouard, Denis S. Grebenkov, Michael Levitin, and Iosif Polterovich

Exterior Steklov problem for the kite and the disks

Script accompanying the paper *The exterior Steklov problem for Euclidean domains*

The script processes the outputs of FreeFEM scripts ExteriorSteklovKite.edp and ExteriorSteklovDisks.edp

Auxiliary

In[493]:=
```
SetDirectory[NotebookDirectory[]];
<< MaTeX`
SetOptions[MaTeX,
  "BasePreamble" → {"\\usepackage{amsmath}", "\\usepackage{fourier}",
"\\usepackage[lining]{ebgaramond}",
     "\usepackage[scr=boondox]{mathalpha}",
    "\\usepackage{xcolor}\\definecolor{darkgreen}{rgb}{0.00, 0.67, 0.00}"},
  FontSize → 12, Magnification → 1];
clr = ColorData[97, "ColorList"]
```

Out[496]=
```
{■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■}
```

- rasterizeBackground from https://mathematica.stackexchange.com/questions/3190/saner-alternative-to-contourplot-fill

In[497]:=
```
rasterizeBackground[g_, res_ : 450] :=
 Show[Rasterize[Show[g, PlotRangePadding → 0, ImagePadding → 0,
     ImageMargins → 0, LabelStyle → Opacity[0], FrameTicksStyle → Opacity[0],
     FrameStyle → Opacity[0], AxesStyle → Opacity[0], TicksStyle → Opacity[0],
     PlotRangeClipping → False], "Graphics", ImageResolution → res] /.
   Raster[data_, rect_, rest__] :→ Raster[data,
     Transpose@OptionValue[AbsoluteOptions[g, PlotRange], PlotRange], rest],
  Sequence @@ Options[g], Sequence @@ Options[g, PlotRange]]
```

Load FreeFEM data

In[498]:=
```
LoadConformal[filestr_] := Module[{strConf, nev, evs, ncol, np, Pts, Ufs, j},
   strConf = OpenRead[filestr];
   Read[strConf, Record];
   nev = Read[strConf, Number];
   evs = Read[strConf, Table[Number, {nev}]];
   Read[strConf, Record];
   {np, ncol} = Read[strConf, {Number, Number}];
   Pts = Read[strConf, Table[Table[Number, {ncol}], {np}]];
   Close[strConf];
   Ufs = Table[Interpolation[
      Pts[[All, {1, 2, j + 2}]], InterpolationOrder → 1], {j, 1, nev - 1}];
   {nev, evs, Ufs}];
```

Kite

Definitions

In[499]:=
```
xK[t_] := 3 / 2 Cos[t] + 7 / 10 Cos[2 t] - 4 / 10;
yK[t_] := 3 / 2 Sin[t] - 3 / 10 Cos[t];
boundaryK = Table[{xK[t], yK[t]}, {t, 0, 2 Pi, 2 Pi / 400}];
PboundaryK = Polygon[boundaryK];
```

In[503]:=
```
kt = MaTeX["\\mathcal{K}"];
frt = {{-3, MaTeX["-3"]}, {0, MaTeX["0"]}, {3, MaTeX["3"]}};
```

In[505]:=
```
PlotKiteUf[nn_, cc_] := Show[
   rasterizeBackground[DensityPlot[cc UfsKite[[nn - 1]][x, y],
     {x, -3, 3}, {y, -3, 3}, PerformanceGoal → "Quality",
     ColorFunction → (*"Rainbow"*)"ThermometerColors",
     MeshFunctions → {#3 &, #3 &}, Mesh → 5, MeshStyle → Gray,
     PlotRange → All, PlotPoints → 25, FrameTicks → {{frt, None}, {frt, None}}]],
   Graphics[{FaceForm[White], EdgeForm[{Thick, Black}], PboundaryK,
     Inset[kt, {0, 0}]}], PlotLabel → MaTeX["\\sigma_{" <> ToString[nn] <>
      "}\\left(\\mathcal{K}^\\mathrm{ext}\\right)\\approx" <>
      ToString[Round[evsKite[[nn]] 1000] / 1000.]]
  ];
```

Plots

In[506]:=
```
{nevKite, evsKite, UfsKite} = LoadConformal["external-kite-data-web.txt"];
```

```
figUfsKiteAll = Legended[Grid[
    Table[PlotKiteUf[3 i + j - 2, 1], {i, 1, 3}, {j, 1, 3}]
   ], Placed[BarLegend[
     {"ThermometerColors", {-0.5, 0.5}}, LegendLayout → "Row"], Below] ];
```

In[508]:=
```
Export["figUfsKiteAll.jpeg", figUfsKiteAll, ImageSize → 600];
```

In[ ]:=
```
figUfsKiteSelection =
  GraphicsRow[{PlotKiteUf[3, 1], PlotKiteUf[7, 1]}, ImageSize → 580];
```

In[ ]:=
```
Export["figUfsKiteSelection.pdf", figUfsKiteSelection]
```

Three disks

Definitions

In[509]:=
```
regdisk = RegionUnion[Disk[], Disk[{-2, 0}, 2 / 3], Disk[{2, -2}, 3 / 2]];
```

In[510]:=
```
td = MaTeX["\\mathcal{T}"];
frt = {{-5, MaTeX["-5"]}, {0, MaTeX["0"]}, {5, MaTeX["5"]}};
```

In[512]:=
```
PlotDisksUf[nn_, cc_] :=
  Show[rasterizeBackground[
    DensityPlot[
     cc UfsDisks⟦nn - 1⟧[x, y], {x, -5, 5}, {y, -5, 5},
     PerformanceGoal → "Quality",
     ColorFunction → (*"Rainbow"*)"ThermometerColors",
     MeshFunctions → {#3 &, #3 &}, Mesh → 5, MeshStyle → Gray,
     PlotRange → All, PlotPoints → 25, FrameTicks → {{frt, None}, {frt, None}}]
   ], Graphics[{FaceForm[White], EdgeForm[{Thick, Black}], Disk[],
     Disk[{-2, 0}, 2 / 3], Disk[{2, -2}, 3 / 2], Inset[td, {0, 0}]}],
   AspectRatio → 1, PlotLabel → MaTeX["\\sigma_{" <> ToString[nn] <>
      "}\\left(\\mathcal{T}^\\mathrm{ext}\\right)\\approx" <>
      ToString[Round[evsDisks⟦nn⟧ 1000] / 1000.]]
  ];
```

Plots

In[513]:=
```
{nevDisks, evsDisks, UfsDisks} = LoadConformal["external-disks-data-web.txt"];
```

```
figUfsDisksAll = Legended[Grid[
    Table[PlotDisksUf[3 i + j - 2, 1], {i, 1, 3}, {j, 1, 3}]
   ], Placed[BarLegend[
     {"ThermometerColors", {-0.5, 0.5}}, LegendLayout → "Row"], Below] ];
```

In[515]:=
```
Export["figUfsDisksAll.jpeg", figUfsDisksAll, ImageSize → 600];
```

```
figUfsDisksSelection =
  GraphicsRow[{PlotDisksUf[4, 1], PlotDisksUf[8, 1]}, ImageSize → 580];
```

In[•]:=
```
Export["figUfsDisksSelection.pdf", figUfsDisksSelection]
```

Out[•]=

figUfsDisksSelection.pdf