Degree-one interpolants on a concave quadrilateral

Singularity conditions are addressed using Gröbner Bases

July 8, 2007

code

```
BeginPackage["concaveQuadrilateralDegreeOneInterpolants`", "Global`"]

concaveQuadrilateralDegreeOneInterpolants`

concaveQuadrilateralDegreeOneInterpolants::usage =
 "concaveQuadrilateralDegreeOneInterpolants[nodes,{x,y}], second node to be the origin:
nodes: {{x1,y1},{0,0},{x3,y3},{x4,y4}}
generates degree one interpolant for all  nodes
   in x-y coordinates for a concave quadrilateral described by nodes."

s2::usage = "s2[nodes,{x,y}], second node to be the origin:
nodes: {{x1,y1},{0,0},{x3,y3},{x4,y4}}
generates degree one interpolant only for the second
   node in x-y coordinates for a concave region described by nodes."

equationOfAnEdge::usage = "
   equationOfAnEdge[nodes,{x,y},{i,j}]  yields
   equation in x,y coordinates of the line joining nodes number i and
   j in the intercept form such that the value at the origin is 1."

alongAnEdge::usage =
 "alongAnEdge[f,{x, y},nodes,{i,j},τ] yields a function of 0< τ < 1, of a
   function f in x,y coordinates between i and j nodes.

alongAnEdge[f,{x, y,z},nodes,{2,j},τ] also substitutes for z."

equationOfAnEdge[nodes_, {x_, y_}, {i_, j_}] :=
 Module[{s}, s = Det[{{x, y, 1}, Append[nodes[[i]], 1], Append[nodes[[j]], 1]}];
  (* check the sign *)
  s / (s /. {x → 0, y → 0}) // Expand]

alongAnEdge[f_, {x_, y_}, nodes_, {i_, j_}, τ_: τ] := Module[{τRule},
  τRule = Thread[{x, y} -> nodes[[i]] + τ (nodes[[j]] - nodes[[i]])
   ];
  f /. τRule // Expand // Together // Chop]
```

```
alongAnEdge[f_, {x_, y_, z_}, nodes_, {2, j_}, τ_: τ] := Module[{s},
  s = alongAnEdge[f, {x, y}, nodes, {2, j}, τ];

  (s /. z → τ * (Sqrt[(nodes[[2]] - nodes[[j]]).(nodes[[2]] - nodes[[j]])])) // Expand //
    Together // Chop
 ]

Begin["`Private`"]

concaveQuadrilateralDegreeOneInterpolants`Private`

s2[nodes_, {x_, y_}] := Module[{z, num2, deno, α, β, γ, x4, y4, eq1, s },

  (* numerator as the product of side34 and side 41 *)
  num2 = Times @@ (equationOfAnEdge[nodes, {x, y}, #] & /@ {{3, 4}, {4, 1}});
  deno = 1 + α x + β y + γ z;

  (* denominator should pass through the
   downward projected point of node4 *)

  {x4, y4} = nodes[[4]];
  eq1 = 0 == deno /. {x → x4, y → y4, z → -Sqrt[x4^2 + y4^2]};
  deno = deno /. (Solve[eq1, γ] // Flatten) [[1]];
  s = (num2 / deno);


  (* along side21 and side23 from 2 the profile should be (1-τ)
  *)

  along21 = Cancel[alongAnEdge[s, {x, y, z}, nodes, {2, 1}, τ] / (1 - τ)];

  along23 = Cancel[alongAnEdge[s, {x, y, z}, nodes, {2, 3}, τ] / (1 - τ)];

  (* unknown α and β can now be solved *)

  (s /. Flatten[
      Solve[Thread[Cancel[Rationalize[{along21, along23}]] == 1], {α, β}]
     ]) /. z → Sqrt[x^2 + y^2]

 ]

concaveQuadrilateralDegreeOneInterpolants[nodes_, {x, y}] :=
 Module[{s, s1, s3, s4, eqHomogeneity, eqXLinearity, eqYLinearity, solS134 },
  s = s2[nodes, {x, y}];
  eqHomogeneity = s + s1 + s3 + s4 == 1;
  eqXLinearity = {s1, s, s3, s4}.First[Transpose[nodes]] == x;
  eqYLinearity = {s1, s, s3, s4}.Last[Transpose[nodes]] == y;
  solS134 =
   Solve[{eqHomogeneity, eqXLinearity, eqYLinearity}, {s1, s3, s4}] // Together // Flatten;
  {s1, s, s3, s4} = {s1, s, s3, s4} /. solS134

 ]

End[]

concaveQuadrilateralDegreeOneInterpolants`Private`

EndPackage[]
```

Example

■ case-closed form expression

```
nodes = {{-x1, 0}, {0, 0}, {-x3, -y3}, {x4, y4}}
```

```
{{-x1, 0}, {0, 0}, {-x3, -y3}, {x4, y4}}
```

```
s = s2[nodes, {x, y}] // Simplify;
```

A numerical example:

```
(*nodes={{-1,0},{0,0},{-.5,-.5},{1,.5}} *)
```

```
ExpandAll[s /. N[{x1 → 1, x3 → 1 / 2, y3 → 1 / 2, x4 → 1, y4 → 1 / 2}]] // Together
```

$$\frac{1. - 3.\,x - 4.\,x^2 + 2.\,y + 22.\,x\,y - 24.\,y^2}{1. + 24.0669\,x + 12.6257\,y + 28.0669\,\sqrt{x^2 + y^2}}$$

■ case-1: verification

```
nodes = {{-1, 0}, {0, 0}, {-.5, -.5}, {1, .5}}
```

```
{{-1, 0}, {0, 0}, {-0.5, -0.5}, {1, 0.5}}
```

```
s = s2[nodes, {x, y}]
```

$$\frac{(1. + 1.\,x - 4.\,y)\,(1. - 4.\,x + 6.\,y)}{1 + 24.0669\,x + 12.6257\,y + 28.0669\,\sqrt{x^2 + y^2}}$$