

# polysurf

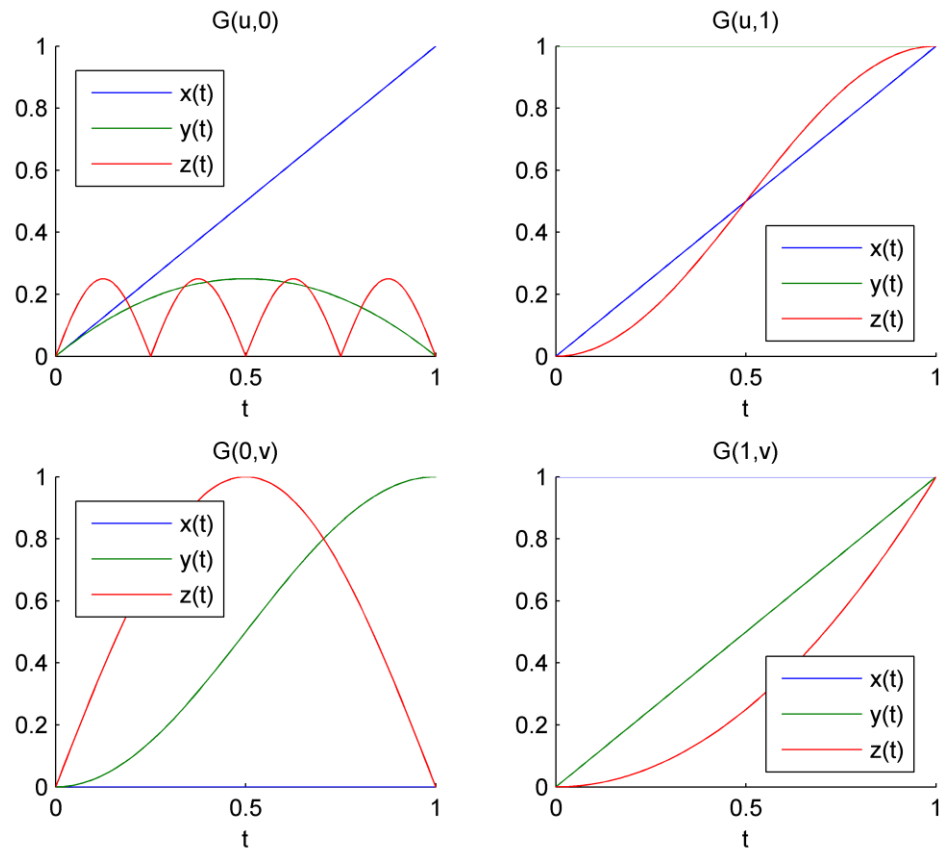
-a surface based on bilinear interpolation between 4 polylines implemented in MATLAB®

---

Date: 10/2016

Author: Jari Repo, University West, [jari.repo@hv.se](mailto:jari.repo@hv.se)

# Step 1 – Define surface boundary functions



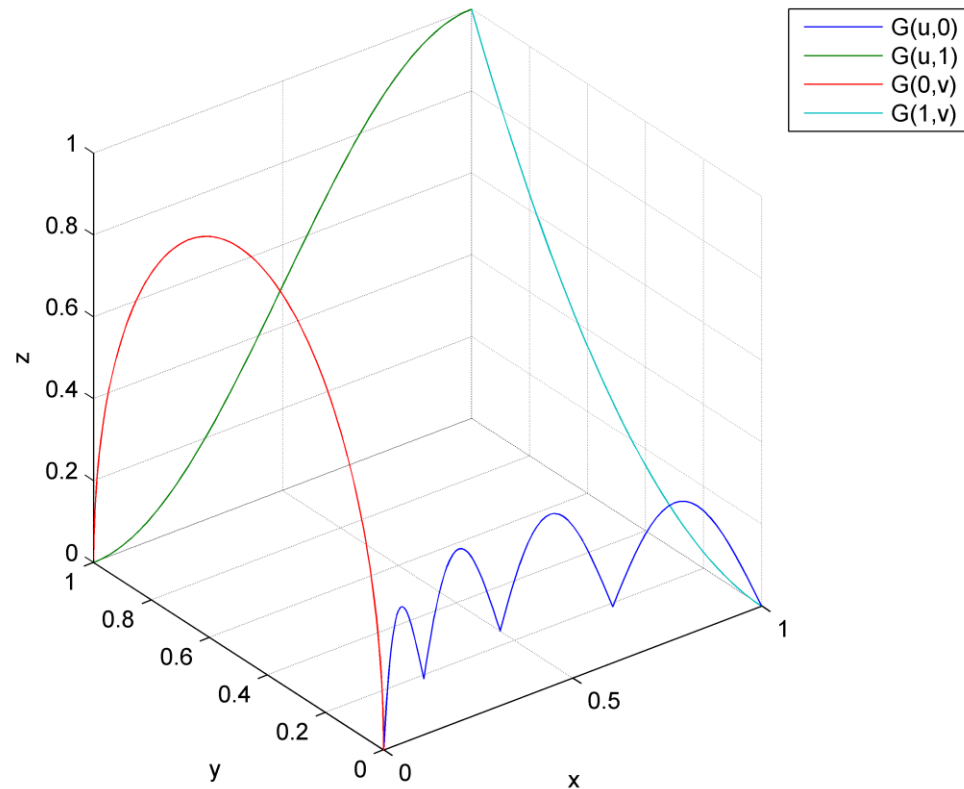
```
% G(u,0) -- parabolically twisted abs. sine
Gu0 = { @(t) t; ...
        @(t) -4*.25*t.*(t-1); ...
        @(t) abs(.25*sin(4*pi*t)) };

% G(u,1) -- squared quarter sine
Gu1 = { @(t) t; ...
        @(t) ones(size(t)); ...
        @(t) sin(pi/2*t).^2 };

% G(0,v) -- circular arc
G0v = { @(t) zeros(size(t)); ...
        @(t) (1/2)*(1-cos(pi*t)); ...
        @(t) sin(pi*t) };

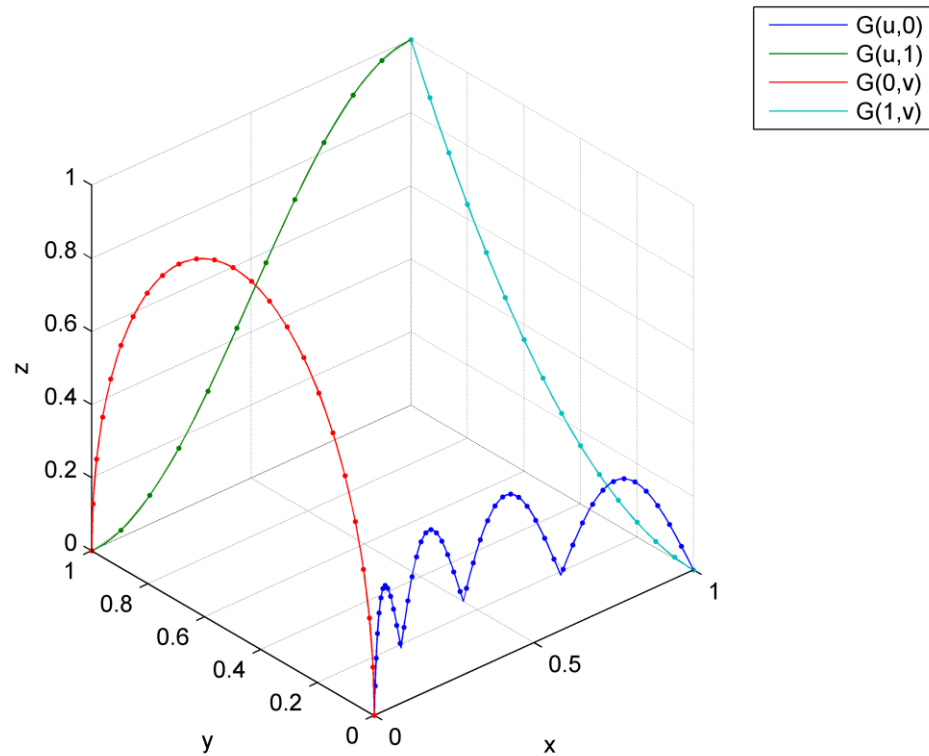
% G(1,v) -- parabolic
G1v = { @(t) ones(size(t)); ...
        @(t) t; ...
        @(t) t.^2 };
```

# Step 1 – View surface boundary functions



```
fh3d = figure('name','3-d plot');
set(gcf,'renderer','opengl','doublebuffer','on')
set(gca,'fontsize',9)
cameratoolbar
X = [Gu0{1}(t); Gu1{1}(t); G0v{1}(t); G1v{1}(t)]';
Y = [Gu0{2}(t); Gu1{2}(t); G0v{2}(t); G1v{2}(t)]';
Z = [Gu0{3}(t); Gu1{3}(t); G0v{3}(t); G1v{3}(t)]';
ph = plot3(X,Y,Z);
view(3)
grid on
xlabel('x'),ylabel('y'),zlabel('z')
title('Surface boundary curves')
legend('G(u,0)', 'G(u,1)', 'G(0,v)', 'G(1,v)', ...
       'location','eastoutside')
```

# Step 2 – Generate polylines by sampling



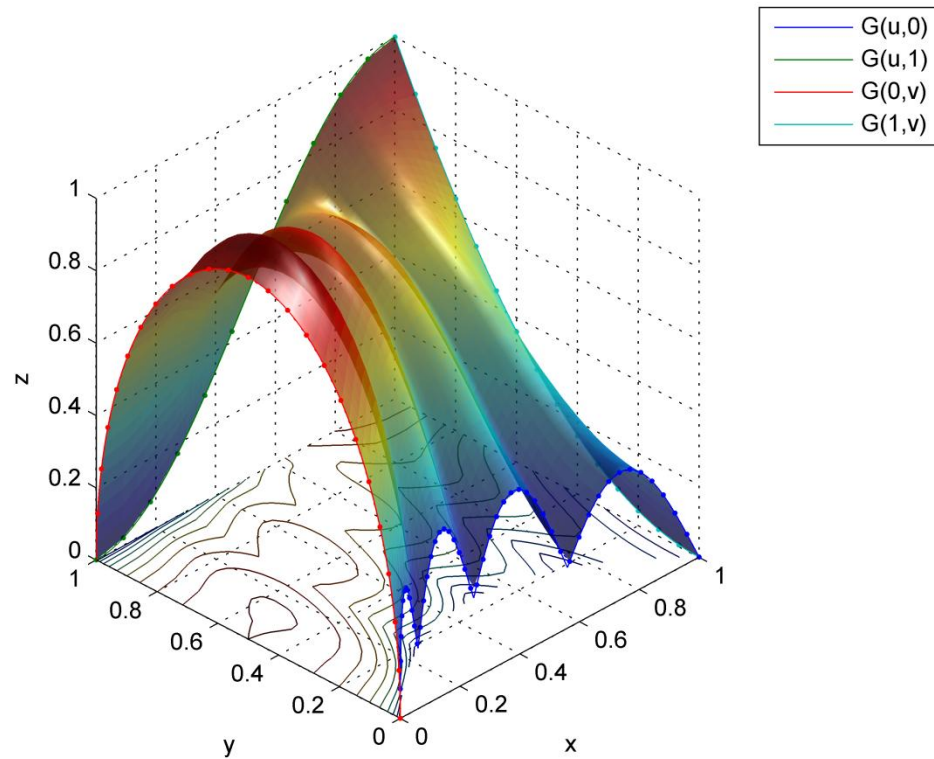
```
t1 = linspace(0,1, 50);
t2 = linspace(0,1, 12);
t3 = linspace(0,1, 25);
t4 = linspace(0,1, 16);

% Store sampled boundary functions into a 4-cell array
P = cell(1,4);
P{1} = [Gu0{1}(t1); Gu0{2}(t1); Gu0{3}(t1)]; % G(u,0)
P{2} = [Gu1{1}(t2); Gu1{2}(t2); Gu1{3}(t2)]; % G(u,1)
P{3} = [G0v{1}(t3); G0v{2}(t3); G0v{3}(t3)]; % G(0,v)
P{4} = [G1v{1}(t4); G1v{2}(t4); G1v{3}(t4)]; % G(1,v)

% To view the content in P
% P{1},P{2},P{3},P{4}

% Output the individual polylines in the 3-d view
hold on
for k=1:4
    plot3(P{k}(1,:),P{k}(2,:),P{k}(3,:),'.: ',...
          'Color',get(ph(k),'Color'))
end
hold off
```

# Step 3 – Generate bilinear surface



```
surfu = 50;  
surfv = 50;  
  
disp('Generating surface ...')  
[Gx,Gy,Gz] = polysurf(P,surfu,surfv);  
  
hold on  
% plot3(Gx,Gy,Gz,'.')  
sh = surf(Gx,Gy,Gz);  
set(sh,'linestyle','none','facealpha',.75)  
hold off  
axis equal  
  
% add some shading  
lighting phong  
light
```

# Step 3 – Bilinear surface (randomized edges)

see "polysurfdemo\_random.m"

