Stången: Introduktion till MyPoisson och Dugga 2

[1]

Inledning

En stång med cirkulärt tvärsnitt är fast inspänd i vänstra ändpunkten och utsatt för en dragkraft i högra ändpunkten. Bestäm tvärsnittsradien så att förlängningen får ett givet värde. (En uppgift i Dugga 2 handlar om detta.)

Stångens ekvationer är

$$\begin{cases} -\mathrm{D}\big(EA\,\mathrm{D}u\big) = K_x A & \text{för } x \in I = (0,L), \\ EA\,\mathrm{D_N}u + ku = P & \text{för } x = 0, L. \end{cases}$$

Vi har $K_x = 0$, $k_0 = \infty$, $k_L = 0$, så att ekvationerna blir

$$\begin{cases} -\mathrm{D}\big(EA\,\mathrm{D}u\big) = 0 & \text{för } x \in I = (0,L), \\ u(0) = 0, & (k_0 = \infty) \\ EA\,u'(L) = P & (k_L = 0). \end{cases}$$

Ekvationerna löses lätt genom att integrera två gånger,

$$u(x) = C_1 x + C_2,$$

och använda randvillkoren

$$0 = u(0) = C_2, \quad P = EAu'(L) = EAC_1.$$

Vi får

$$u(x) = \frac{P}{EA}x.$$

Förlängningen är, med cirkulär tvärsnittsarea $A = \pi r^2$,

$$\Delta L = u(L) = \frac{P}{EA}L = \frac{P}{E\pi r^2}L.$$

Nu kan man enkelt bestämma r så att förlängningen får ett visst värde:

$$r = \sqrt{\frac{PL}{E\pi\Delta L}}.$$

(I duggauppgiften går det inte att genomföra dessa räkningar för hand.)

Matlab-beräkningar

Vi ska göra detta med Matlab. Ladda ned filerna `MyPoisson.m`, `EqData1.m`, `BdryData1.m` och prova dem:

[1] 2 maj 2013, Stig Larsson, Matematiska vetenskaper, Chalmers tekniska högskola

```
L=1;    % the length of the rod
n=101;  % the number of points, n-1 is the number of intervals
p=linspace(0,L,n);
t=[1:n-1; 2:n; ones(1,n-1)];
e=[1 n; 1 2];

EqData  =@EqData1;      % function handles
BdryData=@BdryData1;
[U,A,b] =MyPoissonSolver(p,t,e,EqData,BdryData);
plot(p,U)
```

Obs att vi sparar funktionshandtagen i variablerna `EqData, BdryData`. Detta gör det bekvämt att byta ut dem senare. Nu fyller vi i data som svarar mot stången. Vi väljer $E = 1.0 \times 10^7$ `[N/m`^{2}`]`, $P = 10.0$ `[N]`, $r = 1.0$ `[cm]`, $L = 1.0$ `[m]`. Det önskade värdet är $\Delta L = 1.0$ `[mm]`.

Filen `EqDataStang11.m`:

```
function [a, d, c, f] = EqDataStang11(x, tag)

E=1e7;     % N/m^2 Young's modulus
r=0.01;    % m     radius
A=pi*r^2;  % m^2   area
a=E*A;
d=0; c=0; f=0;
```

Filen `BdryDataStang11.m`:

```
function [k, uA, g] = BdryDataStang11(x, tag)

P=10;  % N the force
if tag==1  k=1e8; uA=0;   g=0; end
if tag==2  k=0;   uA=0;   g=P; end
```

Vi provar med

```
EqData  =@EqDataStang11;      % function handles
BdryData=@BdryDataStang11;
[U,A,b] =MyPoissonSolver(p,t,e,EqData,BdryData);
plot(p,U)
dL=U(end)  % the extension of the rod
```

Nu har vi skrivit in värden på alla parametrar E, r, P i datafilerna. Om vi vill variera några av dem, till exempel r och P, så bör vi betrakta dem som invariabler till funktionerna. Filen `EqDataStang12.m`:

```
function [a, d, c, f] = EqDataStang12(x, tag, r)

E=1e7;     % N/m^2
A=pi*r^2;  % m^2
a=E*A;
d=0; c=0; f=0;
```

Filen `BdryDataStang12.m`:

```
function [k, uA, g] = BdryDataStang12(x, tag, P)

if tag==1  k=1e8; uA=0;   g=0; end
if tag==2  k=0;   uA=0;   g=P; end
```

Vi provar

```
P=10;   % N force
r=0.01; % m radius
EqData  =@(x,tag) EqDataStang12(x,tag,r);
BdryData=@(x,tag) BdryDataStang12(x,tag,P);
[U,A,b] =MyPoissonSolver(p,t,e,EqData,BdryData);
dL=U(end)  % the extension of the rod
plot(p,U)
```

Obs att vi nu gjort om funktionshandtagen till *anonyma funktioner* av `(x,tag)` därför att funktionerna `EqDataStang12` och `BdryDataStang12` har för många invariabler nämligen `(x,tag,r)` respektive `(x,tag,P)`.

Nu kan vi lätt göra en parameterstudie där vi varierar radien och beräknar förlängningen:

```
P=10;  % N
BdryData=@(x,tag)BdryDataStang12(x,tag,P);
rr=[]; dL=[];
for r=0.01:.001:0.03
  EqData=@(x,tag)EqDataStang12(x,tag,r);
  [U,A,b]=MyPoissonSolver(p,t,e,EqData,BdryData);
  dL=[dL;U(end)]; % save the extension in a column matrix
  rr=[rr;r];      % save the radius in a column matrix
end
[rr,dL]           % display a table
```

Nu kan vi läsa av i tabellen när ΔL blir 1 mm. En bättre metod är att lösa ekvationen

$$\Delta L(r) - 0.001 = 0$$

med Newtons metod. Vi bygger in funktionen $f(r) = \Delta L(r) - 0.001$ i MATLAB-funktionen `stangfunk`:

```
function y = stangfunk(r)
L=1;    % the length of the rod
n=101;  % the number of points, n-1 is the number of intervals
p=linspace(0,L,n);
t=[1:n-1; 2:n; ones(1,n-1)];
e=[1 n; 1 2];
P=10;
EqData  =@(x,tag)EqDataStang12(x,tag,r);
BdryData=@(x,tag)BdryDataStang12(x,tag,P);
[U,A,b] =MyPoissonSolver(p,t,e,EqData,BdryData);
dL=U(end);
y=dL-0.001;
end
```

Sedan beräknar vi den optimala radien:

```
R=newton(@stangfunk,.01,1e-6)
```