

TANA17 Matematiska beräkningar med Matlab

Projekt 2. Beräkning av Tyngdpunkt

1 Introduktion

Ofta då man designar produkter i ett CAD verktyg vill man utföra beräkningar för att verifiera produktens hållfasthet, vikt eller andra mekaniska egenskaper. De verktyg som utför dessa beräkningarna behöver en beskrivning av produktens geometri. För $2D$ beräkningar används oftast trianglar och för $3D$ används pyramider. De flesta CAD system kan exportera en beskrivning av produktens geometri som är enkel att beskriva med hjälp av trianglar eller pyramider.

I detta projekt skall vi undersöka hur en sådan beskrivning av geometrin kan användas för att beräkna tyngdpunkt för ett objekt i $2D$.

1.1 Geometri beskrivning med trianglar

En triangel kan beskrivas genom att man anger x - och y -koordinater för dess hörnpunkter. Exempelvis beskrivs enhetstriangeln genom att vi anger de tre punkterna

$$P_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \text{och} \quad P_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Eftersom många trianglar kan dela hörnpunkter är det praktiskt att samla alla hörnpunkter i en matris. Vi låter `Nodes` vara en $2 \times m$ matris där varje kolumn `Nodes(:,i)` innehåller x - och y -koordinater för en av hörnpunkterna.

Triablarna brukar ofta kallas för *Element*. Varje element har tre hörnpunkter. För att hålla reda på vilka hörnpunkter som skall användas för att bygga upp varje triangel skapar vi en $3 \times n$ matris `Elements`. Varje kolumn i `Elements` matrisen talar om vilka hörnpunkter som ingår i just den triangeln. Exempelvis om

```
>> Elements( : , 9 )'  
ans =  
    43  21 192
```

så betyder det att de tre hörnpunkterna som bygger upp triangel nummer 9 är

```
>> P1 = Nodes(:,43); P2 = Nodes(:,21); P3=Nodes(:,192);
```

I projektet skall två olika geometrier användas. Filen `Mesh1.m` innehåller noder och element som representerar en ellips. Filen `Mesh2.m` representerar en platta med två kylflänsar placerade på ovansidan. Variabeln `Mesh` skapas när filerna körs i Matlab. Denna variabel är en struktur med två fält `Nodes` och `Elements`. Vill du komma åt matrisen `Nodes` som innehåller koordinaterna för triangelarnas hörn skriver du

```
>> Nodes = Mesh.Nodes;
```

på motsvarande sätt kommer du åt `Elements` matrisen.

1.2 Tyngdpunksberäkning

För en $2D$ yta som beskrivs av ett område V , med konstant densitet gäller att tyngdpunkten ges av

$$c_m = \iint_V \vec{r} dx dy, \quad \vec{r} = \begin{pmatrix} x \\ y \end{pmatrix}.$$

I vårt fall består området V av ett antal trianglar T_i . Det gäller alltså att

$$V = \bigcup_{i=1}^n T_i.$$

Vi kan utföra tyngdpunksberäkningen för varje triangel T_i separat. Vi får att

$$c_m = \sum_{i=1}^n A_i \vec{r}_i$$

där A_i och $\vec{r}_i$ är arean respektive mittpunkten för triangeln nummer i .

Arean för en triangel T_i med hörnpunkter P_1 , P_2 och P_3 i $\mathbb{R}^2$ fås genom att vi bildar vektorerna

$$\vec{v}_1 = P_2 - P_1, \text{ och } \vec{v}_2 = P_3 - P_1.$$

Arean av triangeln fås sedan genom att beräkna *determinanten*

$$A_i = \frac{1}{2} |(\vec{v}_1, \vec{v}_2)|.$$

Mitt punkten för triangeln fås genom att vi beräknar medelvärdet av dess hörnpunkter,

$$\vec{r}_i = \frac{1}{3}(P_1 + P_2 + P_3).$$

Det är alltså lätt att hitta tyngdpunkt och area (eller vikten) för varje enskild triangel. Gör en enkel skiss som tydligt förklarar de olika storheterna. Genom att summera bidragen från samtliga trianglar hittar vi tyngdpunkt och area för hela kroppen.

2 Implementering av Algoritmen

Vi skall skriva Matlab funktioner som visualiserar kroppen och beräknar dess tyngdpunkt. Bägge funktionerna får väldigt lika struktur.

Om vi börjar med att visualisera område och trianglar så behöver vi en hjälpfunktion

```
>> DrawTriangle( Ti , Nodes )
```

Vi låter **Ti** vara en kolumnvektor med index för de tre hörnpunkterna som definierar triangeln T_i . Som tidigare är **Nodes** den matris som innehåller koordinater för samtliga hörnpunkter. Funktionen skall hitta de koordinaterna för de tre hörnpunkterna P_1 , P_2 och P_3 i matrisen **Nodes** och sedan plotta linjer från

$$P_1 \longrightarrow P_2 \longrightarrow P_3 \longrightarrow P_1,$$

och vi har ritat upp en av trianglarna. Denna funktion används sedan av funktionen

```
>> DrawMesh( Mesh )
```

som har hela **Mesh** strukturen som inparameter. Eftersom varje kolumn i **Elements** matrisen svarar mot en triangel så behöver vi en **for**-loop där vi använder hjälpfunktionen ovan för att plotta en triangel i taget. Tänk på att använda **axis equal** i slutet så att trianglarna får rimliga proportioner.

På liknande sätt kan vi beräkna *tyngdpunkten* och *vikten* för hela området V . Målet är att skriva en funktion

```
>>[ MassCenter , Area ] = CalculateMassCenter( Mesh )
```

För att underlätta beräkningen behöver vi en hjälpfunktion

```
>>[ Ri , Ai ] = CalculateTriangle( Ti , Nodes )
```

som beräknar mass centrum $\vec{r}_i$ och arean A_i för triangeln T_i . Vi behöver hitta hörnpunkterna P_1 , P_2 och P_3 för att utföra beräkningen. I huvudfunktionen **CalculateMassCenter** så skapar vi två variabler **MassCenter** och **Area** som bägge sätts till noll i början av funktionen. Vi beräknar sedan bidragen från en triangel i taget och summerar tills vi hittat arean och mass centrum för hela kroppen.

3 Resultat

Då funktionerna är färdiga skall ni använda dem för att beräkna masscentrum för de båda områden som ges av filerna **Mesh1** respektive **Mesh2**.

Verkligt masscentrum för ellipsen skall vara $(0,0)$. Ellipsen har axlar $a = 0.6$ och $b = 1.0$ vilket ger arean $A = ab\pi = 1.8850$. Detta kan du använda för att göra troligt att din funktion räknar rätt.

4 Redovisning och Rapporten

Rapporten skall ha följande kapitel och innehåll

- *Introduktion* där problemet, dvs tyngdpunktsberäkning givet en beskrivning av geometrin introduceras kortfattat. Ni skall även ge en kort beskrivning av formlerna för tyngdpunkts- och areaberäkning. Metoden att beskriva ett komplicerat område genom att dela upp det i trianglar skall också beskrivas.
- *Matlab program* där ni beskriva alla Matlab funktioner och program som ni skrivit. Ni skall inte klippa in programkoden utan för varje funktion beskriv kort funktionens syfte, vilka inparametrar den behöver och vad den gör.
- *Resultat* där det beskrivs hur lösningsmetoden fungerade. Viktigt är att jämföra resultatet för ellipsen med de verkliga resultaten. Redovisa också resultatet för den lite mer komplicerade geometrin.
- *Diskussion* där resultatet sammanfattas. Fungerade metoden? Blev resultatet bra?

De grafer som minst måste ingå i rapporten är

- Bägge geometrierna skall illustreras genom att ni ritar upp alla trianglarna med **DrawMesh** funktionen.

Se dessutom generella krav på rapport och matlab program från hemsidan.