

Flervariabelanalys och Matlab

Kapitel 3

Thomas Wernstål
Carl-Henrik Fant
Matematiska Vetenskaper

26 september 2008

3 Multipelintegraler

3.1 Dubbelintegraler

Exempel. Vi skall beräkna dubbelintegralen $\iint_D (y \sin x + x \cos y) dx dy$ där D är rektangeln $D : \pi \leq x \leq 2\pi, 0 \leq y \leq \pi$.

Lösning. Ett enkelt sätt att beräkna närmevärden är att beräkna en Riemannsumma till integralen. Vi börjar med att definiera integranden som en anonym funktion

```
>> f=@ (x,y) y.*sin(x)+x.*cos(y);
```

Därefter gör vi en partition av D med 1000×1000 delrektanglar, alla med arean $\pi^2/1000000$:

```
>> n=1000;  
>> [x,y]=meshgrid(linspace(pi,2*pi,n+1),linspace(0,pi,n+1));
```

Hela rektangelarean är π^2 , delrektanglarnas area är $\pi^2/10^6$. Vi väljer sedan en punkt i varje delrektangel, t.ex. nedre vänstra hörnet. Då beräknas Riemannsumman med kommandot

```
>> dA=pi^2/n^2;  
>> I=sum(sum(f(x(1:n),y(1:n))))*dA
```

Jämför med integralens exakta värde, $-\pi^2$ så ser du att relativa felet, trots det mycket stora antalet delrektanglar, är $6 \cdot 10^{-3}$. Relativa felet är alltså ungefär samma som sidlängden i delrektanglarna. För att minska relativa felet med en faktor 10 till storleksordningen 10^{-4} , måste antalet delrektanglar, och antalet element i matriserna x och y , ökas med en faktor 100 till 10^8 . Matlab klarar inte mycket större matriser än de som använts här. Skall man beräkna Riemannsumma med finare indelning krävs det att kalkylerna organiseras på annat sätt. $\square$

Genom detta exempel kan man se att Riemannsumman inte ger en bra numerisk metod. Dessutom tar kalkylen lång tid vilket syns om man låter Matlab ta tiden:

```
>> tic,I=sum(sum(f(x(1:n-1),y(1:n-1))))*A,toc
```

För att beräkna approximativa värden på envariabelintegraler används ofta *Simpsons formel* som bygger på att man delar in integrationsområdet i m små intervall, och sedan approximerar integranden i vart och ett av intervallen med ett andragradspolynom som har samma värde som integranden i delintervallets ändpunkter och mittpunkt.

Summan av integralerna av andragradspolynomen ger Simpsons formel:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{m} \sum_{k=0}^{2m+1} w_k \cdot f(x_k).$$

Här är $x_k = a + \frac{k}{2m+1}(b-a)$ och w_k är i tur och ordning $\frac{1}{6}, \frac{4}{6}, \frac{2}{6}, \frac{4}{6}, \frac{2}{6}, \dots, \frac{4}{6}, \frac{2}{6}, \frac{4}{6}, \frac{1}{6}$. Med 500 delintervall och därmed 1001 punkter (inklusive mittpunkterna i delintervallen) ges följden w_k i Matlab av

```
>> m=500; n=2*m+1; w = [1 [3*ones(1,n-2)+(-1).^(2:n-1)] 1]/6;
```

Genom upprepad integration, där simpsons formel används först vid integralberäkning i ena variabeln och sedan i den andra kan man härleda en tvådimensionell Simpsons formel:

$$\iint_D f(x)dx \approx \frac{(b-a)(d-c)}{m^2} \sum_{j=0}^{2m+1} \sum_{k=0}^{2m+1} w_{j,k} \cdot f(x_j, y_k).$$

Här är $x_j = a + \frac{j}{2m+1}(b-a)$, $y_k = c + \frac{k}{2m+1}(d-c)$ och $w_{j,k} = w_j \cdot w_k$ med w_k enligt ovan.

Vikterna $w_{j,k}$ ges av matrisen:

```
>> W = w' * w;
```

För att beräkna integralen ovan med denna tvådimensionella Simpsons formel, och samtidigt ta beräkningstiden skriver man:

```
>> m=500; n=2*m+1;
>> [x,y]=meshgrid(linspace(pi,2*pi,n),linspace(0,pi,n));
>> w = [1 [3*ones(1,n-2)+(-1).^(2:n-1)] 1]/6;
>> W = w' * w;
>> dA=pi^2/m^2;
>> tic, I=sum(sum(W.*f(x,y)))*dA, toc
```

Tidsåtgången blir ungefär samma som för Riemannsumman, men relativa felet blir nu av storleksordningen $1/m^4$.

Matlabs integralberäkningsprogram `quad` använder Simpsons formel men med en adaptiv metod där intervallindelningen styrs av beräkningsresultaten. Där funktionen varierar mycket görs finare indelning, där variationen är liten görs grov indelning. Det finns också ett par andra beräkningsmetoder att tillgå, `quadl`, och `quadgk`. Du kan läsa mer om detta i Matlabs *Product help* under rubriken *Numerical Integration (Quadrature)*

Dubbelintegraler över axelparallella rektanglar beräknas med hjälp av `dblquad`. I princip används upprepad integration genom att Matlab först gör en indelning av den andra variabelns intervall, beräknar enkelintegralen i alla dessa delningspunkter med avseende på den första variabeln med hjälp av `quad`. Simpsons formel tillämpas sedan med dessa beräknade integralvärden. Indelningen av andra variabelns intervall förfinas, integraler beräknas i de nya punkterna, Simpsons formel tillämpas igen. Proceduren upprepas till förändringen är liten nog. Även i detta fall görs indelningen finare där det behövs. Man kan välja annan metod för beräkning av enkelintegralerna. Trippelintegralsberäkning, som vi kommer till i avsnitt 3.2 bygger på samma idé.

Exempel. Vi skall beräkna samma dubbelintegral som ovan $\iint_D (y \sin x + x \cos y) \, dx dy$ där D är rektangeln $D : \pi \leq x \leq 2\pi, 0 \leq y \leq \pi$.

Lösning. Eftersom vi redan definierat integranden som en anonym funktion kan vi beräkna dubbelintegralen, med relativt fel 10^{-6} vilket är grundinställningen, genom kommandot

```
>> I = dblquad(f, pi, 2*pi, 0, pi)
```

Jämför med integralens exakta värde, $-\pi^2$ så ser du att relativa felet i själva verket är ungefär 10^{-9} .

För att minska beräkningstiderna kan vi ibland acceptera större relativt fel. Kommandot

```
>> I = dblquad(f, pi, 2*pi, 0, pi, 0.006)
```

ger samma relativa fel som riemannsumman ovan, men hundra gånger snabbare. $\square$

Vid beräkning av dubbelintegraler över ett allmännare område D , som ligger i en axelparallell rektangel R i planet, använder vi att

$$\iint_D f(x, y) \, dx dy = \iint_R f(x, y) \chi_D(x, y) \, dx dy$$

där χ_D är karakteristiska funktionen för området D dvs.

$$\chi_D(x, y) = \begin{cases} 1 & \text{om } (x, y) \in D \\ 0 & \text{annars} \end{cases}.$$

Exempel.

Vi skall beräkna $\iint_D (y \sin x + x \cos y) \, dx dy$ där

$$D = \{(x, y) : 0 \leq x \leq 2, 0 \leq y \leq x^2\}.$$

Lösning. Vi definierar anonyma funktioner som motsvarar karakteristiska funktionen, integranden och, för enkelhets skull, produkten av dessa två:

```
>> karD = @(x,y) (0 <= x).* (x <= 2).*(0 <= y).* (y <= x.^2);
>> f = @(x,y) y.*sin(x)+x.*cos(y);
>> fD = @(x,y) karD(x,y).*f(x,y);
```

Eftersom $D \subset \{(x, y): 0 \leq x \leq 2, 0 \leq y \leq 4\}$, så beräknar vi sedan integralen med kommandot

```
>> I = dblquad(fD, 0, 2, 0, 4)
```

Varje annan rektangel som omfattar D duger här. □

3.1.1 Övningar

3.1.1 Beräkna följande integraler m.h.a. `dblquad`.

- a) $\int \int_D y \sin(y + xy) dx dy$
där $D = \{(x, y): 0 \leq x \leq 1, -\pi/2 \leq y \leq \pi/2\}$.
- b) $\int \int_D x^2 + e^{x^2+y^2} dx dy$, där $D = \{(x, y): x^2 + y^2 \leq 1\}$
- c) $\int \int_D e^{x+ey} dx dy$, där $D = \{(x, y): 1 \leq x \leq 2 - y, |xy| \leq 1\}$.
(Skissa först för hand på området D . Gå tillbaka till avsnittet om nivåkurvor om du vill få hjälp av Matlab.)

3.1.2 Låt D vara det område i planet som begränsas av nivåkurvorna $x \sin y + y \cos x = xy$ och $(x + 3)^2 + (y - 2)^2 = 1.5$ och som innehåller punkten $(-3, 1)$. Beräkna tyngdpunkten för området D .

Några avsnitt/uppgifter ur kursboken som anknyter till detta avsnitt: 14.2, 14.4

3.2 Trippelintegraler

Det är inte någon väsentlig skillnad på beräkning av trippelintegral eller dubbelintegral med Matlab.

Exempel. Låt oss se hur man använder `triplequad` för att beräkna integralen $\iiint_D xy^2z^3 dx dy dz$, där $D: 0 < x < 1, 0 < y < 2, 1 < z < 2$. Vi börjar som tidigare med att definiera integranden som en anonym funktion

```
>> f=@(x,y,z) x.*y.^2.*z.^3;
```

För att beräkna integralen ger vi sedan kommandot

```
>> I=triplequad(f,0,1,0,2,1,2)
```

□

Om integrationsområdet inte är en parallelepiped definierar vi en karakteristisk funktion för området, multiplicerar med den och integrerar, precis som i tvådimensionella fallet.

3.2.1 Övningar

3.2.1 Använd kommandot `triplequad` för att beräkna trippelintegralen

$$\iiint_D \frac{dx dy dz}{(x+y+z+1)^3} \text{ där } D : x > 0, y > 0, z > 0, x+y+z < 2.$$

3.2.2 Beräkna volymen av en sfär med radie 1 genom att beräkna integralen

$$\iiint_D dx dy dz \text{ där } D : x^2 + y^2 + z^2 \leq 1, \text{ dels med hjälp av sfärisk substitution och } \text{triplequad}, \text{ dels med tekniken med karakteristisk funktion för } D \text{ och } \text{triplequad}.$$

Några avsnitt/uppgifter ur kursboken som anknyter till detta avsnitt:

14.5.1-12, 14.6.15-20

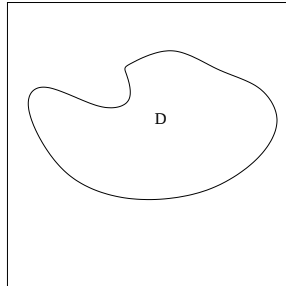
3.3 Monte Carlo metoden

I detta avsnitt skall vi se hur man kan beräkna flerdimensionella integraler approximativt med den så kallade *Monte Carlo-metoden*. Vi kommer inte att göra någon matematisk analys av hur effektiv och noggrann metoden är, eftersom en sådan analys förutsätter kunskaper i matematisk statistik.

De metoder som använts i det föregående utnyttjar, som redan nämnts, *Simpsons formel*. I Matlabprogrammen *quad*, *dblquad* och *triplequad* används en adaptiv metod där intervallindelningen styrs av beräkningsresultaten. Trots detta krävs många beräkningssteg för att ge önskad noggrannhet.

Matlabprogrammen *quad*, *dblquad* och *triplequad* har tre nackdelar. För det första har de problem med diskontinuiteter, som naturligt kommer in om integrationsområdet inte är en rektangel eller ett rätblock. För det andra ökar beräkningstiden väsentligt med ökande dimension. För det tredje, och viktigast, finns inget färdigt matlabprogram för n -dimensionella integraler med $n > 3$.

En enkel metod som kan vara en lösning på dessa problem är Monte Carlo-metoden. Vi skall först se hur den kan användas till att åskådliggöra områden i planet och beräkna arean. Området D i figuren nedan är inritat i en ruta som är 1×1 .



Antag att området har arean A .

Om man väljer två koordinater $0 \leq x \leq 1$ och $0 \leq y \leq 1$ slumpvis, så är sannolikheten att punkten (x, y) ligger i området D precis A (man måste ställa en del krav på hur området ser ut, för att detta påstående skall vara sant). Om man nu väljer N punkter, (x_i, y_i) , $1 \leq i \leq N$, så bör antalet punkter, M , som ligger inom det markerade området i genomsnitt vara $A \cdot N$. Med andra ord bör vi få $M \approx A \cdot N$ och $A \approx M/N$.

Om området ligger inom en axelparallell rektangel R med area W så tar man punkterna slumpvis i rektangeln och multiplicerar kvoten med W : $A \approx W \cdot M/N$.

Detta är idén bakom Monte Carlo-metoden:

1. Välj N punkter (x_i, y_i) slumpvis i rektangeln R .
2. Undersök hur många av de N punkterna som ligger i område D . Kalla detta antal för M .
3. Tag $A \approx W \cdot M/N$ som en approximation av områdets area.

Man kan visa att om området uppfyller viss villkor, som uppfylls av de flesta områden man kan tänkas komma på, så är beräkningsfelet med denna metod av storleksordning $1/\sqrt{N}$. I det tvådimensionella fallet är detta jämförbart med Riemannsumman men i högre dimension är det väsentligt bättre.

Exempel. Vi ritar området D som ges av $1 \leq xy \leq 4$, $x \leq y \leq 4x$, $x > 0$, $y > 0$ samt beräknar områdets area.

Lösning. Vi observerar först att om $\chi_D(\mathbf{x})$ är områdets karakteristiska funktion så ges antalet träffar i D av

$$M = \sum_{i=1}^N \chi_D(x_i, y_i).$$

Alltså har vi att

$$A \approx \frac{W}{N} \sum_{i=1}^N \chi_D(x_i, y_i).$$

I detta exempel har vi

```
>> chiD = @(x,y) (x.*y>=1) .* (x.*y<=4) .* (y>=x) .* (y<=4*x);
```

De två hyperblerna skär linjerna i punkterna $(\frac{1}{2}, 2)$, $(1, 1)$, $(1, 4)$ och $(2, 2)$. Området ligger i rektangeln $\frac{1}{2} \leq x \leq 2$, $1 \leq y \leq 4$

Vi genererar sedan punkter slumpvis i kvadraten med sida 1, multiplicerar med intervallens längder och translaterar så att vänster ändpunkt är rätt:

```
>> N=100; X=rand(N,2);  
>> x = 1/2 + 3/2*X(:,1); y = 1 + 3*X(:,2);
```

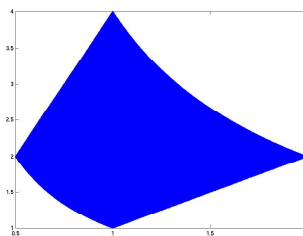
För att rita området låter vi Matlab bestämma alla punkter där karakteristiska funktionen är 1.

```
>> M=find(chiD(x,y)==1); plot(x(M),y(M),'.')
```

Eftersom rektangelarean är $\frac{3}{2} \cdot 3 = \frac{9}{2}$ får vi sedan arean till:

```
>> A = 9/2*sum(chiD(x,y))/N;
```

När vi ser att det hela fungerar ökar vi antalet punkter till en miljon, får en fin graf och arean till ungefär 2.08. Den exakta arean är $3 \ln(2) \approx 2.076$. $\square$



En väsentlig fördel med metoden är att varken beräkningarna eller felet påverkas av dimensionen. Om man har en n -dimensionell kropp som ligger i ett n -dimensionellt

rätblock av volym W och slumpmässigt tar N punkter i detta rätblock, så är kroppens (n-dimensionella) volym $V \approx W \cdot \frac{M}{N}$, där M är antalet punkter som ligger i kroppen.

Om man istället vill beräkna en multipelintegral,

$$I = \int \cdots \int_{\Omega} f(\mathbf{x}) d\mathbf{x},$$

så gör man precis på samma sätt, fast nu blir

$$I \approx \frac{W}{N} \sum_{i=1}^N \chi_{\Omega}(\mathbf{x}_i) f(\mathbf{x}_i).$$

Det finns förbättringar av metoden. Vill du veta mer kan du läsa på:

http://www.wikidoc.org/index.php/Monte_Carlo_integration

Exempel. Vi illustrerar hur man kan beräkna integralen

$$\iiint_D 1/\sqrt{x^2 + y^2 + z^2} dx dy dz$$

då $D = \{(x, y, z) : x^2 + y^2 + z^2 \leq 1, x > 0, y > 0, z > 0\}$.

Lösning. Integrationsområdet ligger i kuben $0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1$.

Som vanligt definierar vi områdets karakteristiska funktion och integranden som anonyma funktioner:

```
>>chiD=@(x,y,z) (x.^2+y.^2+z.^2 < 1)
>>f=@(x,y,z) 1./sqrt(x.^2+y.^2+z.^2)
```

och produkten:

```
>>fD=@(x,y,z) chiD(x,y,z).*f(x,y,z)
```

Vi låter Matlab bestämma en miljon punkter i kuben slumpvis med kommandosekvensen

```
>>N=10^6; X=rand(N,3);x=X(:,1); y= X(:,2); z=X(:,3);
```

Integralberäkningen ges nu av:

```
int = sum(fD(x,y,z))/N
```

Det exakta värdet är $\pi/4$ som är ungefär 0.7854, vilket även triplequad ger. Medelvärde för 10 beräkningar med denna metod blev 0.7858. Styrkan ligger i möjligheten att öka antalet dimensioner. □

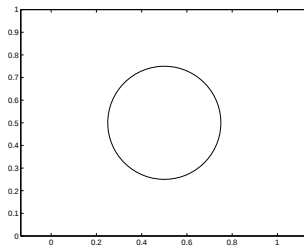
3.3.1 Övningar

I figuren nedan ges området Λ av

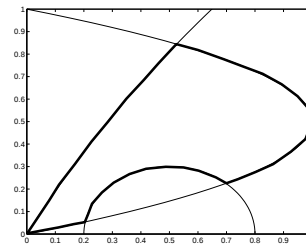
$$\Lambda = \{(x, y); (x - 0.5)^2 + (y - 0.5)^2 < 1/16\},$$

och Γ består av det område som definieras av olikheterna

$$\begin{aligned} y - 2 \ln(1 + x) &< 0, \\ x + 4(y - 1/2)^2 &< 1, \text{ och} \\ (x - 1/2)^2 + y^2 &> 0.09. \end{aligned}$$



Området Λ .



Området Γ .

3.3.1 Beräkna först det exakta värdet av arean av området Λ och integralen,

$$\iint_{\Lambda} (x^2 + y^2) dx dy.$$

Utför samma beräkningar med Monte Carlometoden. För att se hur det beräknade värdet varierar för fixt N och hur det förbättras för ökande N kan du utföra beräkningarna med $N = 10, 100, 1000, 10000$ punkter, och upprepa varje sådan beräkning 10-15 gånger. Studera hur relativa felet varierar från gång till gång då N hålls fixt och hur relativa felen förändras då N ökas.

3.3.2 Gör motsvarande för området Γ respektive integralen

$$\iint_{\Gamma} (x^2 + y^2) dx dy.$$

I detta fall kan du jämföra med vad `dblquad` ger i stället för det exakta värdet.

3.3.3 Beräkna volymen av en fyrdimensionell sfär $x^2 + y^2 + z^2 + u^2 \leq 1$ med hjälp av Monte-Carlo metoden.