Inleiding programmeren

1^e jaar wis-, natuur- en sterrenkunde

Universiteit van Amsterdam

september 2014

Opgaves bij college 4

numerieke integratie en fitten van data

1 Numeriek integreren: Riemannsom (theorie)

Evalueer een integraal door het te schrijven als som van kleine rechthoekjes: een zogenaamde Riemannsom.

Probleem: integraal
Gegeven $f(x)$ op $a \leq x \leq b$, bereken $\int_a^b f(x)\ dx$

Oplossing: schrijf de integraal als Riemannsom
Verdeel het interval (a, b) in N intervallen van gelijke lengte Δx en representeer $f(x)$ door de waarde van de functie $f(x_i)$ op de x-waarden x_i:

$$f_i = f(x_i), \text{ waarbij } x_i = a + i\Delta x \quad (i = 0, 1, 2, ..., N \text{ en } \Delta x = \frac{b-a}{N})$$

Schrijf de integraal dan als een Riemannsom:

$$\int_a^b f(x)\ dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x)\ dx \tag{1}$$

Implementatie: benader centrale waarde voor $f(x)$ in elke bin en doe sommatie
Met behulp van een lineaire benadering **(de trapeziumregel)** kunnen we de centrale waarde voor $f(x)$ benaderen in elke bin door het gemiddelde van de waarden van $f(x)$ op de linker en rechter rand van de bin. In deze lineaire benadering op het interval (x_i, x_{i+1}) is $f(x)$ dan te schrijven als:

$$f(x) = \frac{f_{i+1} + f_i}{2} \ + \ \mathcal{O}(\Delta x)$$

De sommatie voor de integraal uit vergelijking (1) is dan te schrijven als:

$$\begin{aligned}\int_a^b f(x)\ dx \ &\approx \ \frac{\Delta x}{2}(f_0 + 2f_1 + 2f_2 + ... + 2f_{N-1} + f_N) \ + \ \mathcal{O}((\Delta x)^2) \\ &\approx \ \Delta x(f_1 + f_2 + ... + f_{N-1}) + \frac{\Delta x}{2}(f_0 + f_N)\end{aligned}$$

Extra: hogere orde (meer precieze) benaderingen
Het is mogelijk de evaluatie van de integraal te verbeteren door niet te uit te gaan van de (te simpele) lineaire benadering. De **Simpsonregel** bijvoorbeeld is een parabolische benadering (let op, N=even) waarbij $f(x)$ op het interval (x_{i-1}, x_{i+1}) wordt benaderd door een parabool door de 3 punten (f_{i-1}, f_i, f_{i+1}). Zoek op of werk zelf uit als je deze wilt gebruiken.

2 Numeriek integreren: Monte Carlo (theorie)

Benader de integraal door gebruik te maken van random getallen. Gooi in een gebied rond de integratie regio random punten en kijk welke fractie binnen het integratiegebied valt.

Probleem: integraal
Gegeven $f(x)$ op $a \leq x \leq b$, bereken $\int_a^b f(x)\ dx$.

Oplossing: definieer een 'box' om de integraalregio heen:
Definieer een box om de integraalregio heen: definieer een $x_{min}, x_{max}, y_{min}$ en y_{max} zodanig dat: $x_{min} \leq a$ en $x_{max} \geq b$ en dat ook geldt:

$$\text{voor } a \leq x \leq b : \ y_{min} \leq f(x) \leq y_{max}$$

In de meeste toepassingen wordt gekozen voor $x_{min} = a$ en $x_{max} = b$.

Implementatie: Gooi random punten in de box:
Gooi een groot aantal random punten (x_i, y_i) in de box en bekijk voor elk punt of het binnen de functie ligt ('goed') of erbuiten ('fout'):

$$\begin{aligned} x_i : &\quad \text{random getal tussen } x_{min} \text{ en } x_{max} \\ y_i : &\quad \text{random getal tussen } y_{min} \text{ en } y_{max} \end{aligned}$$

Is het punt 'goed' (binnen de functie) of 'fout' (buiten de functie) ?

$$\text{'goed': } y_i < f(x_i) \text{ of 'fout': } y_i > f(x_i)$$

Let op: bij negatieve $f(x)$ draait definitie om. Visualiseer altijd de functie.

Bepaal de integraal:
De integraal is de fractie punten die binnen de grafiek vallen keer de oppervlakte van de totale box. In ons geval (rechthoek als box) geldt dan:

$$\int_a^b f(x)\ dx = \frac{N_{\text{goed}}}{N_{\text{goed}} + N_{\text{fout}}} (x_{max} - x_{min})(y_{max} - y_{min})$$

Extra:
In 'echte' toepassingen wordt voor efficiëntie maximalisatie de box zo gekozen dat hij de integraal zo nauw mogelijk omsluit: grootste fractie 'goede' worpen.

computing hints:

- Een random getal tussen 0 en 1 krijg je door: `xi = random()`

opgavenset week 4 (deel 1)

opgave [1]: numeriek integreren
Bereken, gebruikmakend van de Riemannsom **en** de Monte Carlo-techniek de volgende integralen. Maak ook een grafiek met behulp van Matplotlib.

a) $\int_0^1 x^x \, dx$
Hint: test je functie door te testen of je $\int_0^1 x^2 \, dx$ goed voorspelt.

b) $\int_{0.1}^{2.0} \sin(x) \, dx$
Hint: test je methode door de integraal $\int_0^\pi sin(x) \, dx$ te bepalen.

c) $\int_0^\pi \sin(x^2) \, dx$

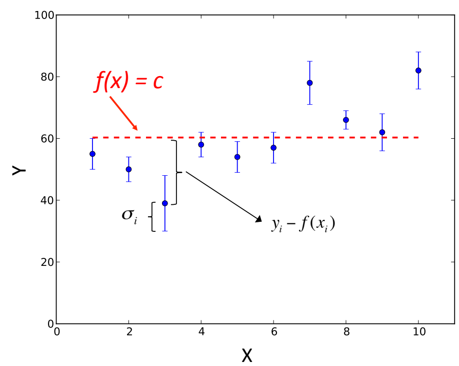
3 Fitten van data en foutenbepaling (theorie)

Om de onderliggende fenomenen van (natuurkundige) verschijnselen te achterhalen wordt data verzameld om afhankelijkheden te onderzoeken. Dat kan de massa van het Higgs boson zijn, de vervaltijd van uranium, maar ook het aantal kinderen in een gezin als functie van de gemiddelde lengte van de ouders. Je kan dan zoeken naar een (causaal) verband: lineair, exponentieel,etc. en daarbij ook de bijbehorende parameters bepalen met hun onzekerheid. Als je een goede beschrijving hebt gevonden kan je daarmee vervolgens ook voorspellingen doen.

Om de 'beste' waarde te vinden hebben we een maat nodig om de *'goedheid'* van de fit quantificeert. We doen dat hier met de χ^2-maat: de som van de gemiddelde afwijking van de meetpunten tot het model gewogen met hun fout: 'hoeveel standaardafwijkingen ligt dit punt weg van mijn functie'.

$$\chi^2 = \sum_{i \text{ (datapunten)}} \left(\frac{y_i - f(x_i|\vec{\alpha})}{\sigma_i} \right)^2,$$

met $\vec{\alpha}$ de vector functie-parameters.



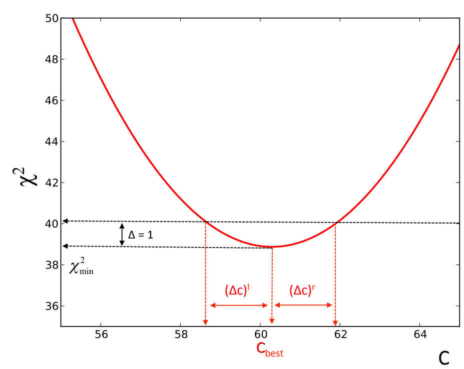
Voorbeeld: 1 dimensie/parameter (zie plot) $f(x|\vec{\alpha}) = c$.

a) de beste waarde van c: c_{best}

De waarde van c waarbij de χ^2 minimaal is.

b) de onzekerheid op $c_{\text{best}}(\Delta_c)$

De fout in de positieve richting $(\Delta_c)^r$ en negatieve richting $(\Delta c)^l$ zijn die waardes van c waarbij de χ^2 1 hoger is dan χ^2_{min}. In de meeste gevallen geldt $(\Delta c)^l = (\Delta c)^r = \Delta c$



Het eindresultaat van je meting is dan: $c = c_{\text{best}}\ {}^{+(\Delta_c)^r}_{-(\Delta_c)^l}$

opgavenset week 4 (deel 2)

opgave [2]: fitten van een model aan de data

De onderstaande data-set geeft voor een specifieke voetballer het percentage goede passes (y) weer als functie van het aantal gespeelde wedstrijden in oranje (x). De onzekerheid op het aantal goede passes is weergegeven als σ_y.

x	1	2	3	4	5	6	7	8	9	10
y	55	50	39	58	54	57	78	66	62	82
σ_y	5	4	9	4	5	5	7	3	6	6

a) maak een plot van deze data met fouten
Computing tip: gebruik de functie `plt.errorbar(x,y, yerr=yerror)`

b) bereken de beste waarde van c als $f(x)$ =c en de bijbehorende onzekerheid Δc.

c) Wat gebeurt er met Δc als de fout in elk meetpunt 2x kleiner wordt ?

4 hacker opgaves

Als je met de verplichte opgaves van deze week klaar bent kan je gaan werken aan de hacker opgaves. Je kan die vinden als apart tabblad op de website van het vak. Een van de hacker opgaves (hacker opgave 3) is een uitbreiding van de fitting-opgave 3 hierboven.