Luščenje modelskih parametrov in razdelčni modeli

6. domača naloga pri predmetu Modelska analiza I

Domen Vaupotič

V nalogi se bomo ukvarjali z luščenjem modelskih parametrov. Osnovni namen luščenja modelskih parametrov je določiti parametre (fizikalnega) modela tako, da model karseda dobro prilegamo k (eksperimentalnim) meritvam. Pri tem uporabljamo različne mere, s katerimi kvantitativno ovrednotimo, kako dobro se model ujema z meritvami. Ena izmed njih je statistični test χ^2, ki ga lahko uporabimo, kadar so merske negotovosti porazdeljene Gaussovsko. Tedaj je namreč minimizacija χ^2 ekvivalentna maksimizaciji funkcije verjetja.

$$\chi^2 = \sum_{i=1}^{N} \frac{(y_i - f(x_i; \vec{a}))^2}{\sigma_i^2}.$$

Izračunano vrednost primerjamo s porazdelitvijo χ^2 z ustreznim številom prostostnih stopenj $\nu = N - M$, pri čemer je N število izmerkov in M število parametrov modela.

Pri izbiri modela kot epistemološko podstat pogosto upoštevamo princip *Ockhamove britve*, s čimer težimo po izbiri *varčnih* modelov – modelov, ki imajo le toliko prostostnih stopenj, kot je nujno potrebno.

Najenostavnejši modeli so *linearni:*

$$f(x; \vec{a}) = \sum_{i=1}^{N} a_i \phi_i(x).$$

Tedaj lahko minimizacijo χ^2 pretvorimo v reševanje linearnega sistema enačb

$$A\vec{a} = \vec{b} \qquad A_{jk} = \sum_{i=1}^{N} \frac{\phi_j(x_i)\,\phi_k(x_i)}{\sigma_i^2}, \quad b_j = \sum_{i=1}^{N} \frac{y_i\,\phi_j(x_i)}{\sigma_i^2},$$

pri čemer so (x_i, y_i) meritve, ϕ_i bazne funkcije in σ_i merske negotovosti.

Kadar modela ne moremo zapisati v linearni obliki, se poslužimo nelinearnih modelov, pri katerih minimizacijo opravimo z eno izmed metod za numerično optimizacijo, npr. Levenberg-Marquardtovo iterativno metodo:

$$\vec{a}_{i+1} = \vec{a}_i - H^{-1}(\chi^2(\vec{a}_i)),$$

pri čemer H predstavlja Hessovo matriko drugih odvodov funkcije χ^2.

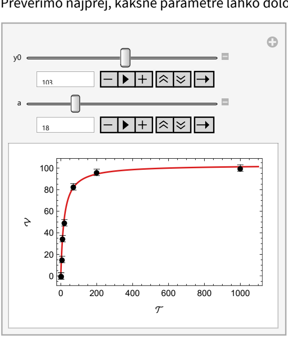
1. Farmakološki model odziva tkiva

V prvem delu obravnavamo enostavni farmakološki model vezave liganda na receptor X + Y ↔ Y*, pri katerem v ravnotežju velja

$$y = \frac{y_0 x}{x + a}.$$

Okometrična metoda

Preverimo najprej, kakšne parametre lahko določimo zgolj z opazovanjem krivulje: $y_0 = 103$, $a = 18$.



Reševanje linearnega sistema

Model lahko lineariziramo:

$$y = \frac{y_0 x}{x + a} \longrightarrow \frac{1}{y} = \frac{a}{y_0}\frac{1}{x} + \frac{1}{y_0} = a_1 + a_2 \frac{1}{x},$$

s čimer dobimo bazni funkciji $\phi_1 = 1$ in $\phi_2 = 1/x$ ter označimo pripadajoča koeficienta z $a_1 = 1/y_0$ in $a_2 = a/y_0$.

Dobimo sistem dveh enačb z dvema neznankama.

$$2.65422 \times 10^7\, a_1 + 181798.\, a_2 == 290347.$$
$$181798.\, a_1 + 4660.2\, a_2 == 2679.95$$

Sistem rešimo in s tem določimo najboljše parametre. Iz matrike A lahko določimo še negotovost parametrov. Ujemanje modela za vsako posamezno točko prikažemo z residuali:

$$\text{residual}(x_i) = \frac{y_i - f(x_i)}{\sigma_i}.$$

$$A^{-1} = \begin{pmatrix} 5.14 \times 10^{-8} & -2.01 \times 10^{-6} \\ -2.01 \times 10^{-6} & 2.93 \times 10^{-4} \end{pmatrix}$$

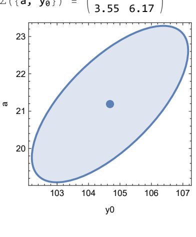
	vrednost	σ
a_1	9.55×10^{-3}	2.27×10^{-4}
a_2	2.02×10^{-1}	1.71×10^{-2}

	vrednost	σ
a	21.2	2.3
y_0	104.7	2.5

Za izračun kovariančne matrike moramo opraviti ustrezno transformacijo.

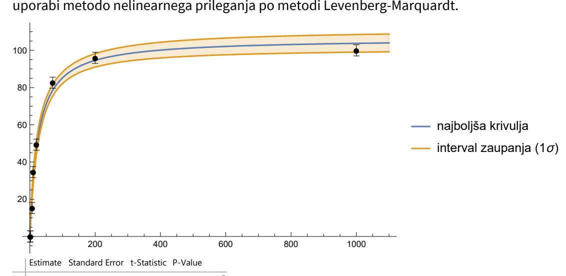
$$\vec{a} = (a, y_0) = (a_2/a_1, 1/a_1)$$
$$\vec{b} = (a_1, a_2) = (1/y_0, a/y_0)$$
$$\Sigma(\vec{a}) \approx D\,\Sigma(\vec{b})\,D^T, \quad D_{ki} = \left(\frac{\partial a_i}{\partial b_i}\right)_{\vec{b}=\vec{\mu}}$$
$$D = \begin{pmatrix} -a_2/a_1^2 & 1/a_1 \\ -1/a_1^2 & 0 \end{pmatrix}$$

$$\Sigma(\{a, y_0\}) = \begin{pmatrix} 4.39 & 3.55 \\ 3.55 & 6.17 \end{pmatrix}$$



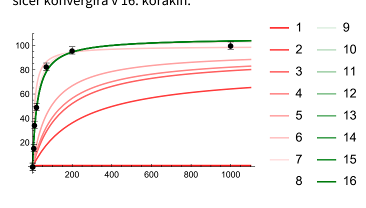
Vgrajene metode

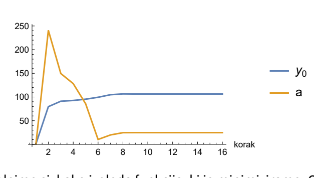
Za prileganje lahko uporabimo tudi vgrajeno metodo *NonlinearModelFit*, ki modela ne linearizira, temveč uporabi metodo nelinearnega prileganja po metodi Levenberg-Marquardt.



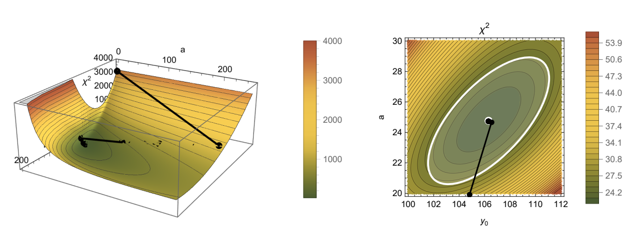
	Estimate	Standard Error	t-Statistic	P-Value
y_0	106.317	4.75897	22.3404	5.26184×10^{-7}
a	24.7625	4.16392	5.94691	0.00101053

Oglejmo si še zaporedne iteracije iskanja najboljših parametrov, kot jih opravi metoda *NonlinearModelFit*, ki sicer konvergira v 16. korakih.



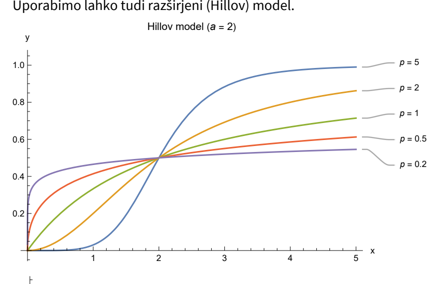


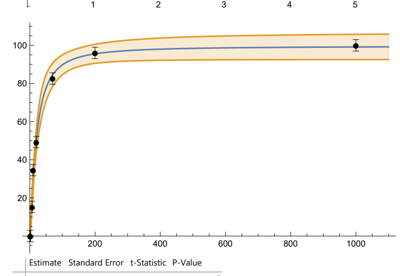
Oglejmo si, kako izgleda funkcija, ki jo minimiziramo. Opazimo, da lahko v bližini minimuma funkcijo zares lepo aproksimiramo s kvadratno formo. Zelo lepo se v tem pogledu vidi in razume tudi pomen kovariančne elipse (označene z belo).



Razširjeni model

Uporabimo lahko tudi razširjeni (Hillov) model.





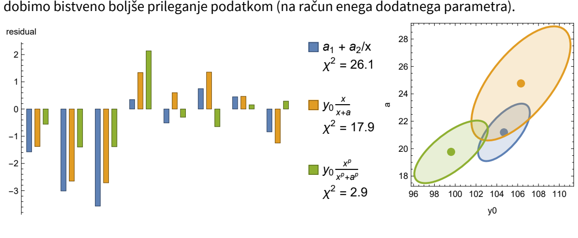
	Estimate	Standard Error	t-Statistic	P-Value
y_0	99.6056	3.53158	28.2042	1.0493×10^{-6}
a	19.7674	2.28365	8.65607	0.000340032
p	1.3641	0.176849	7.71339	0.000584633

$\chi^2 = 2.9$

$\chi^2_\nu = 1.87$

$$\Sigma(\{a, y_0\}) = \begin{pmatrix} 22.6478 & 13.221 \\ 13.221 & 17.3383 \end{pmatrix}$$

Primerjajmo vse tri modele. Opazimo, da se kovariančne elipse delno prekrivajo in da z razširjenim modelom dobimo bistveno boljše prileganje podatkom (na račun enega dodatnega parametra).



2. Čistilnost ledvic

V drugem delu naloge se ukvarjamo z meritvami čistilnosti ledvic, ki jih obravnavamo z razdelčnim modelom. V najosnovnejšem modelu ledvice predstavimo z enim razdelkom, čemur ustreza "enoparametrični" eksponentni model (ena razpolovna konstanta). Model lahko razširimo v dvorazdelčnega, ki vsebuje dve razpolovni konstanti. Nadalje lahko k meritvam prilegamu tudi drugačne modele.

Nelinearno prileganje

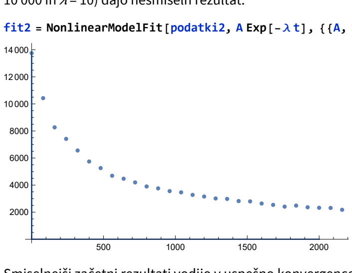
Modela tokrat ne moremo več linearizirati, zato je edina možna metoda nelinearno prileganje. Hitro opazimo, da samodejno prileganje ni uspešno, saj pride do prekoračitve števil.

```
fit2 = NonlinearModelFit[podatki2, A Exp[-λ t], {A, λ}, t];
```

General: Overflow occurred in computation.

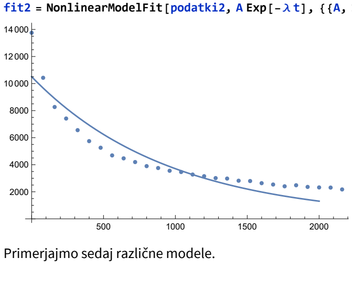
Metodi moramo torej podati ustrezne začetne približke (sicer vse postavi na 1). Slabi začetni približki (npr. A = 10 000 in λ = 10) dajo nesmiseln rezultat.

```
fit2 = NonlinearModelFit[podatki2, A Exp[-λ t], {{A, 10000}, {λ, 10}}, t];
```

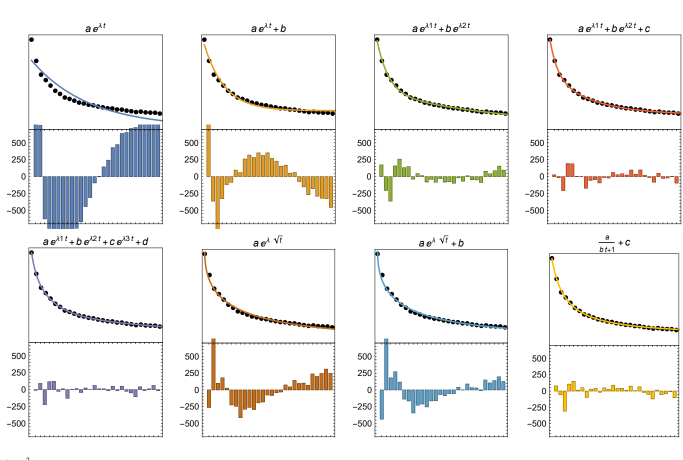


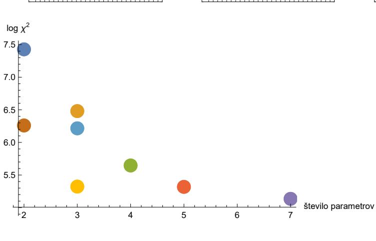
Smiselnejši začetni rezultati vodijo v uspešno konvergenco metode, vendar takoj opazimo, da model slabo opiše dane meritve.

```
fit2 = NonlinearModelFit[podatki2, A Exp[-λ t], {{A, 14000}, {λ, .0001}}, t];
```



Primerjajmo sedaj različne modele.





Opazimo, da se model s korensko-eksponentno odvisnostjo (●) prilega bolje kot navadni eksponentni model (●), čeprav imata isto število parametrov. Zanimivo opazimo, da se recipročni model z zgolj tremi parametri(●), ki ga sicer nismo teoretično izpeljali iz razdelčnih modelov, prilega bolje kot ostali modeli.

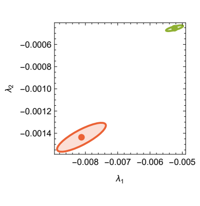
Oglejmo si, kako vpliva dodajanje "ozadja" (konstanta) k dvorazdelčnemu eksponentnemu modelu. Opazimo, da se s tem nekoliko opveča negotovost obeh parametrov λ, prav tako se njuna vrednost zelo spremeni, tako da se kovariančni elipsi ne prekrivata.

$\left(a\, e^{\lambda_1 t} + b\, e^{\lambda_2 t}\right)$ ●

	Estimate	Standard Error	t-Statistic	P-Value
a	8.03×10^3	2.12×10^2	3.79×10^1	6.14×10^{-23}
b	5.55×10^3	$2. \times 10^2$	2.77×10^1	1.01×10^{-19}
λ1	-5.24×10^{-3}	2.64×10^{-4}	-1.98×10^1	2.14×10^{-16}
λ2	-4.54×10^{-4}	2.62×10^{-5}	-1.73×10^1	4.47×10^{-15}

$\left(a\, e^{\lambda_1 t} + b\, e^{\lambda_2 t} + c\right)$ ●

	Estimate	Standard Error	t-Statistic	P-Value
a	5.45×10^3	4.32×10^2	1.26×10^1	8.34×10^{-12}
b	6.29×10^3	3.59×10^2	1.76×10^1	8.11×10^{-15}
λ1	-8.12×10^{-3}	7.55×10^{-4}	-1.08×10^1	1.9×10^{-10}
λ2	-1.44×10^{-3}	1.3×10^{-4}	-1.1×10^1	1.17×10^{-10}
c	1.99×10^3	1.03×10^2	1.92×10^1	1.15×10^{-15}



3. Magnetni spektrometer

V tretjem delu naloge se ukvarjamo s kalibracijskimi podatki magnetnega spektrometra. Najti želimo preslikavo $(x_{fp}, \theta_{fp}) \to \theta_{tg}$.

Pri tem bomo uporabili potenčne bazne funkcije obeh spremenljivk do p-tega reda v x_{fp} in do q-tega reda v θ_{fp}.

```
baza[p_, q_] := {∑_{j=0}^{q} (∑_{i=0}^{p} x^i) θ^j // Expand} //. Plus[a_, b_] :> {a, b} // Flatten
```

```
baza[0, 0]
baza[3, 0]
baza[2, 3]
```

$\{1\}$

$\{1, x, x^2, x^3\}$

$\{1, x, x^2, \theta, x\theta, x^2\theta, \theta^2, x\theta^2, x^2\theta^2, \theta^3, x\theta^3, x^2\theta^3\}$

Zaradi velikega števila izmerkov in velikega števila baznih funkcij se moramo poslužiti bolj učinkovite metode za minimizacijo χ^2. Uporabimo razcep na singularne vrednosti (SVD).

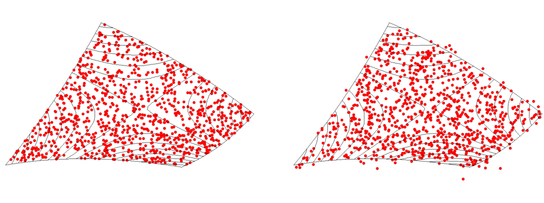
```
SVDFit[baza_, podatki_, tol_ : 10^-15] := Module[{A, b, U, W, V, w, wRec, thresh, a, σ, χ2},
  A = (baza /. {x → #[[1]], θ → #[[2]]}) & /@ podatki;
  b = podatki[[All, 3]];
  {U, W, V} = SingularValueDecomposition[A, Tolerance → 10^-50];

  (* Prileganje s SVD dodatno izboljšamo tako, da odstranimo vektorje, ki pripadajo zelo
   majhnim singularnim vrednostim (mejo izberemo glede na največjo singularno vrednost).
    To storimo tako, da postavimo ustrezno singularno vrednost na 0,
  hkrati pa tudi njeno recipročno vrednost na 0. *)

  thresh = tol * Max@Diagonal@W;
  {w, wRec} = (If[# ≥ thresh, {#, 1 / #}, {0, 0}] & /@ Diagonal[W]) // Transpose;

  a = ∑_{i=1}^{Length@baza} (U[[All, i]].b) V[[All, i]] wRec[[i]];

  σ = Sqrt[∑_{i=1}^{Length@baza} V[[i, All]]^2 * wRec[[i]]^2];

  χ2 = Norm[A.a - b]^2;

  {a, σ, χ2, w}
 ]
```

Test delovanja algoritma

Preverimo, ali algoritem deluje, tako da ga uporabimo na znani porazdelitvi podatkov:

$$(x, \theta) \to 3 + 2x - x^2 + \theta^2 - 4x\theta + \epsilon,$$

pri čemer je ϵ naključno porazdeljena merska napaka.

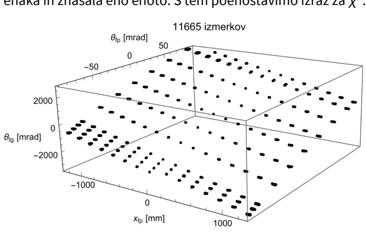


		$\vec{a}$	$\sigma(\vec{a})$	χ^2
brez napake	1	3.	7.2×10^{-2}	5.4×10^{-24}
	x	2.	1.2×10^{-2}	
	x^2	-1.	1.2×10^{-2}	
	θ	3.7×10^{-16}	5.4×10^{-3}	
	$x\theta$	-4.	5.1×10^{-3}	
	$x^2\theta$	1.6×10^{-17}	1.6×10^{-2}	
	θ^2	1.	1.5×10^{-3}	
	$x\theta^2$	-3.9×10^{-16}	3.8×10^{-3}	
	$x^2\theta^2$	-3.9×10^{-16}	1.1×10^{-3}	
z napako	1	5.	$7. \times 10^{-2}$	3.5×10^5
	x	1.7	9.2×10^{-3}	
	x^2	-1.1	1.5×10^{-2}	
	θ	-8.5×10^{-2}	6.7×10^{-3}	
	$x\theta$	-4.1	1.6×10^{-2}	
	$x^2\theta$	-8.6×10^{-3}	5.8×10^{-3}	
	θ^2	9.9×10^{-1}	2.8×10^{-3}	
	$x\theta^2$	2.4×10^{-2}	$3. \times 10^{-3}$	
	$x^2\theta^2$	2.1×10^{-3}	1.2×10^{-3}	

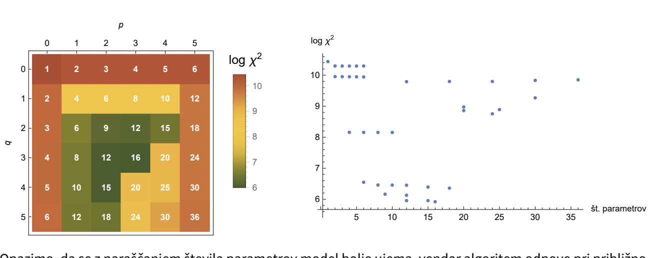
Opazimo, da nam (do numerične natančnosti algoritma) uspe v primeru brez šuma povsem rekonstruirati parametre porazdelitve. Tudi v primeru šuma se približamo pravim vrednostim.

Eksperimentalni podatki

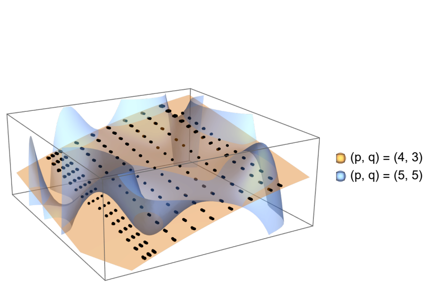
Meritve so podane kot trojica $(x_{fp}, \theta_{fp}, \theta_{tg})$, pri čemer so natančnosti meritev kotov velikostnega reda miliradianov, položajev pa okrog milimetra. Najprej reskalirajmo podatke tako, da bo absolutna napaka obojih enaka in znašala eno enoto. S tem poenostavimo izraz za χ^2. Torej pretvorimo vse kote v miliradiane.



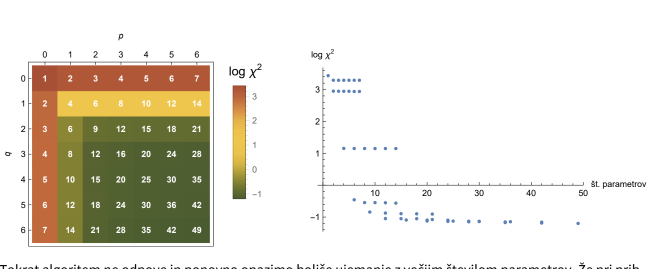
Sedaj prilegajmo vse možne kombinacije polinomskih nastavkov do $p \le 6$ in $q \le 6$.



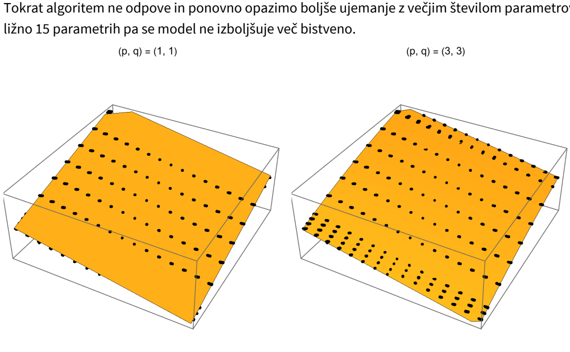
Opazimo, da se z naraščanjem števila parametrov model bolje ujema, vendar algoritem odpove pri približno 15 parametrih.



Vzrok morda leži v numerični nestabilnosti napisanega algoritma, ki je posledica zelo različnih razponov vrednosti. Reskalirajmo zato podatke tako, da bodo vse tri količne v območju [−1, 1] in ponovimo prileganje.



Tokrat algoritem ne odpove in ponovno opazimo boljše ujemanje z večjim številom parametrov. Že pri približno 15 parametrih pa se model ne izboljšuje več bistveno.



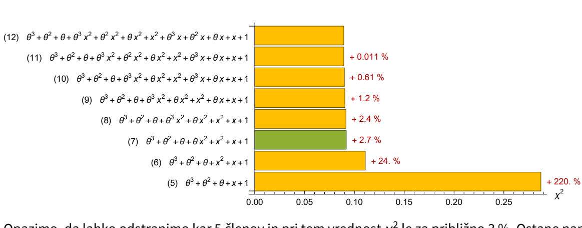
Najprimernejši model

Vzemimo kot izhodiščni najprimernejši model $(p, q) = (2, 3)$ z dvanajstimi parametri, pri katerem je kakovost ujemanja zelo dobro, število parametrov pa še ne preveliko.

člen	koeficient	negotovost	relativna negotovost
1	1.21×10^{-1}	2.9×10^{-2}	2.4×10^{-1}
x	-4.65×10^{-1}	7.5×10^{-2}	1.6×10^{-1}
x^2	-1.35×10^{-2}	3.3×10^{-2}	2.5
θ	5.71×10^{-1}	7.6×10^{-2}	1.3×10^{-1}
$x\theta$	2.51×10^{-3}	$4. \times 10^{-2}$	1.6×10^1
$x^2\theta$	-3.53×10^{-3}	4.8×10^{-2}	1.4×10^1
θ^2	-1.01×10^{-1}	1.3×10^{-1}	1.3
$x\theta^2$	1.47×10^{-4}	4.7×10^{-2}	3.2×10^2
$x^2\theta^2$	1.94×10^{-3}	$1. \times 10^{-1}$	5.4×10^1
θ^3	1.47×10^{-2}	$6. \times 10^{-2}$	4.1
$x\theta^3$	-2.44×10^{-3}	$1. \times 10^{-1}$	4.2×10^1
$x^2\theta^3$	-2.84×10^{-3}	1.8×10^{-1}	6.5×10^1

$\chi^2 = 0.089$

Sedaj poskusimo narediti ta model še varčnejši, tako da postopoma odstranjujemo člene z majhnimi koeficienti in vsakič znova izvedemo prileganje z danimi baznimi funkcijami.



Opazimo, da lahko odstranimo kar 5 členov in pri tem vrednost χ^2 le za približno 3 %. Ostane nam model s 7 parametri.

člen	koeficient	negotovost	relativna negotovost
1	1.2×10^{-1}	2.3×10^{-2}	1.9×10^{-1}
x	-4.65×10^{-1}	$5. \times 10^{-2}$	1.1×10^{-1}
x^2	-1.26×10^{-2}	1.1×10^{-2}	8.5×10^{-1}
$x^2\theta$	-5.71×10^{-3}	3.1×10^{-2}	5.5
θ	5.72×10^{-1}	2.7×10^{-2}	4.7×10^{-2}
θ^2	-1.01×10^{-1}	3.3×10^{-2}	3.3×10^{-1}
θ^3	1.36×10^{-2}	6.1×10^{-2}	4.5

$\chi^2 = 0.092$

Pri večini zgornjih metod s SVD lahko opazimo zelo velike negotovosti. Primerjajmo rezultate z vgrajeno metodo za linearno napoved.

	Estimate	Standard Error	Confidence Interval
1	0.120473	0.019472	{0.0823043, 0.158641}
x	−0.464784	0.0156874	{−0.495533, −0.434034}
x^2	−0.0125759	0.0289152	{−0.0692544, 0.0441027}
$x^2\theta$	−0.00571108	0.0413216	{−0.0867084, 0.0752862}
θ	0.571608	0.0480997	{0.477325, 0.665892}
θ^2	−0.100631	0.0268562	{−0.153273, −0.0479881}
θ^3	0.0136175	0.0586341	{−0.101315, 0.12855}

Opazimo, da so dobljeni parametri identični, primerljive pa so tudi napake. Uporabimo sedaj ta najboljši model na prvotnih podatkih, reskaliranih glede na merske napake.

člen	koeficient	negotovost	relativna negotovost
1	3.48×10^1	1.2×10^{-4}	3.4×10^{-6}
x	-1.18	3.6×10^{-4}	3.1×10^{-4}
x^2	-2.52×10^{-5}	1.9×10^{-2}	7.7×10^2
$x^2\theta$	-1.26×10^{-7}	5.2×10^{-4}	4.2×10^3
θ	$2. \times 10^1$	4.4×10^{-6}	2.2×10^{-7}
θ^2	-3.84×10^{-2}	1.3×10^{-5}	3.3×10^{-4}
θ^3	5.63×10^{-5}	6.2×10^{-8}	1.1×10^{-3}

$\chi^2 = 9.26 \times 10^5$