

FIT per HP 48 G/GX
Versione 1.0
(C) 1994-2001 Gian Piero Luciani

I file del programma "FIT" sono proprietà di Gian Piero Luciani.

Questo software è libero e non è fornita alcuna garanzia di alcun tipo, l'autore non si assume responsabilità derivate dall'uso del programma; eventuali danni derivati dall'uso di questo software, sono a carico dell'utilizzatore.

Questa è la documentazione in italiano, viene riportata insieme alla documentazione in inglese in quanto il mio inglese è piuttosto scadente e la corrispondente documentazione potrebbe essere scarsamente comprensibile e piena di errori.

Ringrazio anticipatamente chiunque vorrà aiutarmi a tradurre in modo corretto questo documento.

Chi trovasse il programma utile o interessante o semplicemente ne avesse voglia, può mandarmi una cartolina con un bel francobollo!

Cosa fa il programma FIT.

Il programma nasce dalla esigenza di calcolare, su una serie di dati raccolti in esperimenti scientifici (chimica, cinetica, etc.), la migliore interpolazione con i minimi quadrati con funzioni che di solito non sono facilmente calcolabili con questo sistema.

Il programma contiene gli algoritmi per calcolare i minimi quadrati delle seguenti funzioni:

1) Polinomi di grado n.: $Y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$

2) Gaussiane: $Y = a \cdot e^{-b(x-x_0)^2}$

3) Biesponenziali: $Y = a \cdot e^{-bx} + c \cdot e^{-dx}$

4) Van Deemter: $Y = a + \frac{b}{x} + cx$

5) Morse: $Y = a(1 - e^{-b(x-c)})^2$

6) funzione: $Y = a(1 - e^{-bx})$

7) funzione: $Y = a - b \cdot e^{-cx}$

8) funzione: $Y = a \cdot x^n \cdot e^{-bx}$

Installazione.

Il programma è scritto quasi totalmente in USER RPL si presenta sotto forma di una libreria, installabile e funzionante su qualsiasi porta, o come directory a seconda dei gusti personali.

Libreria:	File	fit.lib
	Libreria numero:	1700
	Bytes:	7619
Directory:	CRC:	# 13098d
	File	fit.dir
	Bytes:	7593.5
	CRC:	# 9785d

Il programma può essere scaricato in uno dei soliti modi (vedi manuale capitoli 27 e 28) e funziona correttamente in qualsiasi porta.

Come si usa

Per utilizzare il programma basta dare il comando FIT.

Viene mostrato il menu delle diverse funzioni dal quale si può scegliere con la freccia il tipo di curva e selezionare con OK quella preferita.

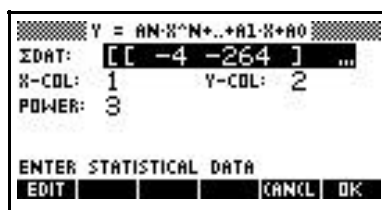


A questo punto viene presentata una seconda finestra nella quale si introducono i dati necessari a completare il calcolo.

Si descrivono di seguito le diverse opzioni:

Polinomi di grado n

Programma di interpolazione polinomiale, il programma esegue l'interpolazione con il polinomio di grado n qualsiasi (purchè minore al numero dei punti) utilizzando il sistema dei minimi quadrati.



Le informazioni necessarie sono: la matrice dei dati dei punti da interpolare che è costituita dalla solita matrice statistica di HP, l'indicazione della colonna che contiene la variabile indipendente x, l'indicazione della colonna che contiene la variabile dipendente y, la potenza del polinomio da calcolare. Inseriti i dati e dato OK il programma esegue il calcolo e trasferisce il risultato, sotto forma di espressione algebrica, nella catasta.

E' abbastanza facile vedere il risultato dell'interpolazione con il programma PLOT di HP, per prima cosa si entra nel programma con RS 8 e si seleziona il tipo di grafico SCATTER, si cancellano eventuali grafici precedenti e si plottano i punti impostando la scalatura automatica. Poi si torna al menu PLOT e si seleziona il tipo di grafico FUNCTION, si prende la funzione calcolata dalla catasta e si riplotta sopra i punti appena visualizzati.

Variabili utilizzate:

MSG	:	lista dei messaggi dell'interfaccia grafica
ΣDAT	:	variabile statistica dell'HP 48
x	:	indice della colonna della variabile indipendente
y	:	indice della colonna della variabile dipendente
N	:	grado del polinomio

Algoritmo:

I coefficienti del polinomio da interpolare si ottengono risolvendo la seguente equazione:

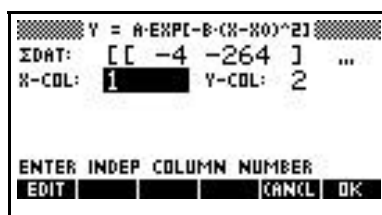
$$\begin{pmatrix} N & \sum_{i=1}^n x_i & \dots & \sum_{i=1}^n x_i^n \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \sum_{i=1}^n x_i^n & \dots & \dots & \sum_{i=1}^n x_i^{2n} \end{pmatrix}^{-1} \times \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n y_i x_i \\ \dots \\ \sum_{i=1}^n y_i x_i^n \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_n \end{pmatrix}$$

Esempio:

Xi	Yi	grado del polinomio N=3 (y=x ³ -7x+6)	
0	6	esatto	trovato
1	0	a0	6
2	0	a1	-7.00000000001
3	12	a2	3.75E-12
4	42	a3	1
5	96		

Gaussiana

Programma di interpolazione della funzione gaussiana, data una serie di punti il programma è in grado di calcolare la migliore gaussiana in grado di approssimarli.



Le informazioni necessarie sono: la matrice dei dati dei punti da interpolare, l'indicazione della colonna che contiene la variabile indipendente x, l'indicazione della colonna che contiene la variabile dipendente y.

Per il resto il programma ha lo stesso comportamento del programma appena descritto.

Variabili esterne utilizzate:

MSG	:	lista dei messaggi dell'interfaccia grafica
ΣDAT	:	variabile statistica dell'HP 48
x	:	indice della colonna della variabile indipendente
y	:	indice della colonna della variabile dipendente

Algoritmo:

L'equazione che si vuole interpolare è la gaussiana che viene espressa dall'equazione:

$$y = A \times e^{-B \cdot (x - x_0)^2}$$

Questa equazione si può trasformare in un polinomio di 2° grado, che può essere risolto con lo stesso algoritmo utilizzato nel caso precedente.

Per prima cosa si assume:

$$\ln(y) = \ln(A) - B \cdot (x - x_0)^2$$

$$\ln(y) = \ln(A) - Bx^2 + 2BX_0 - BX_0^2$$

$$\ln(y) = (-B)x^2 + (2BX_0)X + (\ln(A) - BX_0^2)$$

E quindi si calcola il polinomio di interpolazione di secondo grado dal quale si ottengono i coefficienti:

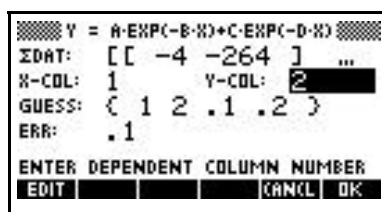
$$\begin{aligned} a_0 &= \ln(A) - BX_0^2 & A &= e^{(a_0 - \frac{a_1^2}{4a_2})} \\ a_1 &= 2BX_0 & B &= -a_2 \\ a_2 &= -B & X_0 &= -\frac{a_1}{2a_2} \end{aligned} \quad \text{ovvero}$$

Esempio:

Xi	Yi	$y = 3 \cdot \exp(-0.2 \cdot (X - 2.3)^2)$		
0	1.04145			
1	2.13958			
2	2.94648	A	3	3.00000051775
3	2.71995	B	0.2	0.200000299828
4	1.68306	X0	2.3	2.29999886033
5	0.69810			

Biesponenziali

Programma di interpolazione biesponenziale, il programma esegue l'interpolazione con la funzione biesponenziale, ovvero $y = A \cdot \exp(-B \cdot x) + C \cdot \exp(-D \cdot x)$



In questo caso oltre alle informazioni richieste nei programmi precedenti è necessario introdurre una lista di quattro valori (guess) e l'errore.

Per risolvere questo problema (che è il più complesso dell'intero gruppo ed è quello per il quale sono stati scritti questi programmi) è necessario ricorrere ad un metodo iterativo di Newton Raphson che per funzionare bene richiede un valore approssimato dei parametri che deve cercare; tanto migliori sono questi valori tanto migliore e più veloce è il calcolo.

Inizialmente è bene porre Err ad un valore abbastanza grande (per esempio = 0.1) per poi ridurlo in un secondo tempo. Infatti ripetendo il calcolo come guess sono presi i valori del risultato del calcolo precedente.

Variabili esterne utilizzate:

- MSG : lista dei messaggi dell'interfaccia grafica
- ΣDAT : variabile statistica dell'HP 48
- x : indice della colonna della variabile indipendente
- y : indice della colonna della variabile dipendente
- X0 : lista dei valori iniziali dei coefficienti dell'equazione cercata
- Er : errore; è la somma delle differenze tra i coefficienti dell'equazione cercata calcolati in due iterazioni consecutive.

Algoritmo

Si utilizza l'algoritmo dei minimi quadrati, ma poichè la funzione in oggetto non può essere ricondotta ad un polinomio, si utilizza un metodo iterativo (Newton-Raphson) che sfrutta il seguente algoritmo:

Sia la funzione da interpolare:

$$Y = a \cdot e^{-bx} + c \cdot e^{-dx}$$

Allora la funzione chi quadro è uguale (salvo alcune costanti che non influenzano il calcolo) a:

$$\chi^2 \propto (y_i - a \cdot e^{-bx_i} + c \cdot e^{-dx_i})^2$$

Se si vogliono trovare i migliori valori dei parametri (a,b,c,d) bisogna minimizzare questa funzione, ovvero trovare i valori di (a,b,c,d) che rendono nulla la derivata prima di chi quadro.

Per fare questo si utilizza l'algoritmo iterativo di Newton-Raphson che riporto di sotto:

$$\begin{bmatrix} a_n \\ b_n \\ c_n \\ d_n \end{bmatrix} = \begin{bmatrix} a_{n-1} \\ b_{n-1} \\ c_{n-1} \\ d_{n-1} \end{bmatrix} - \begin{bmatrix} \frac{\partial^2 \chi^2}{\partial a \partial a} & \frac{\partial^2 \chi^2}{\partial a \partial b} & \frac{\partial^2 \chi^2}{\partial a \partial c} & \frac{\partial^2 \chi^2}{\partial a \partial d} \\ \frac{\partial^2 \chi^2}{\partial b \partial a} & \frac{\partial^2 \chi^2}{\partial b \partial b} & \frac{\partial^2 \chi^2}{\partial b \partial c} & \frac{\partial^2 \chi^2}{\partial b \partial d} \\ \frac{\partial^2 \chi^2}{\partial c \partial a} & \frac{\partial^2 \chi^2}{\partial c \partial b} & \frac{\partial^2 \chi^2}{\partial c \partial c} & \frac{\partial^2 \chi^2}{\partial c \partial d} \\ \frac{\partial^2 \chi^2}{\partial d \partial a} & \frac{\partial^2 \chi^2}{\partial d \partial b} & \frac{\partial^2 \chi^2}{\partial d \partial c} & \frac{\partial^2 \chi^2}{\partial d \partial d} \end{bmatrix}^{-1} \times \begin{bmatrix} \frac{\partial \chi}{\partial a} \\ \frac{\partial \chi}{\partial b} \\ \frac{\partial \chi}{\partial c} \\ \frac{\partial \chi}{\partial d} \end{bmatrix}$$

In realtà questo algoritmo è piuttosto lento per cui il calcolo viene eseguito in due fasi, nella prima si esegue una forma ridotta ed approssimata e poi si applica l'algoritmo vero e proprio.

Se tutto va bene, ad ogni iterazione i valori di (a,b,c,d) sono più vicini ai valori ricercati e sul display viene mostrato un valore di X2 sempre più piccolo.

Quando X2 è sufficientemente piccolo il programma esegue l'algoritmo completo (viene mostrato Y2) sino al raggiungimento dell'errore voluto.

L'errore è calcolato sulla base dello scostamento della somma dei singoli parametri in due successive iterazioni.

Ai curiosi consiglio di guardare il listato del programma, posso solo aggiungere che funziona.

Vista la complessità dei calcoli, non è detto che si ottenga sempre un risultato accettabile, in questi casi si deve provare a cambiare i valori di guess di partenza.

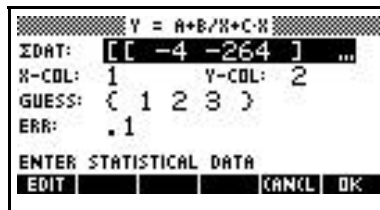
Inoltre i calcoli richiedono un pò di tempo per cui è bene non avere troppa fretta.

Esempio:

Xi	Yi	y=10*exp(-0.5*x)+5*exp(-0.05*x)			
0	15				
1	10.82		guess		
2	8.2	a	9		Er = 0.1
3	6.53	b	1		
4	5.45	c	3		
5	4.71	d	0.1		
6	4.2				
7	3.82		esatto		trovato
8	3.53	a	10		9.999329
9	3.3	b	0.5		0.500251
10	3.1	c	5		5.000664
11	2.92	d	0.05		0.050101

Van Deemter

Programma di interpolazione della funzione di Van Deemter



Tutti i prossimi programmi utilizzano gli stessi algoritmi visti per il programma di interpolazione biesponenziale e necessitano delle stesse informazioni di base.

In questo caso il guess è composto da una lista di 3 parametri.

Per i curiosi si può dire che la curva di Van Deemter è utilizzata in cromatografia per mettere in relazione il flusso dell'eluente con l'efficienza della colonna.

In pratica i costruttori delle colonne dichiarano qual è il flusso ottimale in base all'eluente utilizzato e siccome questo è molto simile a parità di colonna questo programma ha una utilità abbastanza ridotta.

Variabili esterne utilizzate:

- MSG : lista dei messaggi dell'interfaccia grafica
- ΣDAT : variabile statistica dell'HP 48
- x : indice della colonna della variabile indipendente
- y : indice della colonna della variabile dipendente
- Xa : lista dei valori iniziali dei coefficienti dell'equazione cercata
- Er : errore; è la somma delle differenze tra i coefficienti dell'equazione cercata calcolati in due iterazioni consecutive.

Algoritmo

Si utilizza lo stesso algoritmo spiegato prima, nella forma originale senza approssimazioni.

In questo caso ad ogni ciclo viene mostrato Er che è la somma delle differenze dei parametri cercati tra una iterazione e l'altra.

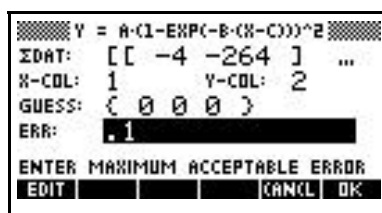
Come si vedrà facendo delle prove, questa funzione è molto meno sensibile alla scelta dei valori iniziali.

Esempio:

Xi	Yi	y=A+B/X+CX		
1	29			
2	26			
3	25.7	A	10	Er = 0.1
4	26	B	10	
5	26.6	C	10	
6	27.3			
7	28.1			
8	29	A	20	trovato 20.003197
9	29.9	B	8	8.002840
10	30.8	C	1	0.998690

Morse:

Programma di interpolazione con la funzione di Morse



Il programma utilizza gli stessi algoritmi visti per il programma di interpolazione biesponenziale e necessitano delle stesse informazioni di base, in questo caso tuttavia vi è una funzione di previsione del valore di guess che entra in funzione quando $\text{guess} = \{ 0 \ 0 \ 0 \}$.

Questa funzione descrive la curva di potenziale di un legame chimico in funzione della distanza tra gli atomi che compongono il legame.

Variabili esterne utilizzate:

- MSG : lista dei messaggi dell'interfaccia grafica
- ΣDAT : variabile statistica dell'HP 48
- x : indice della colonna della variabile indipendente
- y : indice della colonna della variabile dipendente
- Xa : lista dei valori iniziali dei coefficienti dell'equazione cercata
- Er : errore; è la somma delle differenze tra i coefficienti dell'equazione cercata calcolati in due iterazioni consecutive.

Algoritmo

Si utilizza lo stesso algoritmo spiegato per l'interpolazione biesponenziale nella forma originale senza approssimazioni, quando $\text{guess} = \{ 0 \ 0 \ 0 \}$ un secondo algoritmo cerca di indovinare i migliori parametri da utilizzare nella routine iterativa.

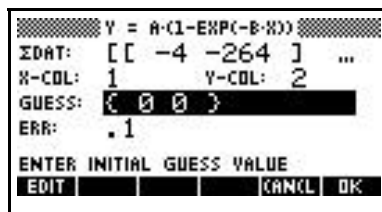
In questo caso ad ogni ciclo viene mostrato Er che è la somma delle differenze dei parametri cercati tra una iterazione e l'altra.

Esempio:

Xi	Yi	$y=A(1-\exp(-B(X-C)))^2$		
0	124.77			
1	32.30		guess	
2	2.62	A	0	Er = 0.01
3	1.94	B	0	
4	13.13	C	0	
5	27.84			
6	42.26		esatto	trovato
7	54.87	A	100	119.16
8	65.28	B	0.3	0.255
9	73.57	C	2.5	2.735
10	80.03			

$$Y=A(1-\exp(-BX))$$

Questo programma esegue l'interpolazione con la funzione: $y=A(1-\exp(-BX))$.



Il programma utilizza gli stessi algoritmi visti per il programma di interpolazione biesponenziale e necessita delle stesse informazioni di base; come nel caso della funzione di Morse è presente una funzione di previsione del valore di guess che entra in funzione quando $\text{guess} = \{ 0 \ 0 \}$

Variabili esterne utilizzate:

MSG	:	lista dei messaggi dell'interfaccia grafica
ΣDAT	:	variabile statistica dell'HP 48
x	:	indice della colonna della variabile indipendente
y	:	indice della colonna della variabile dipendente
Xb	:	lista dei valori iniziali dei coefficienti dell'equazione cercata
Er	:	errore; è la somma delle differenze tra i coefficienti dell'equazione cercata calcolati in due iterazioni consecutive.

Algoritmo

Si utilizza lo stesso algoritmo utilizzato per l'interpolazione biesponenziale nella forma originale senza approssimazioni, quando $\text{guess} = \{ 0 \ 0 \}$ un secondo algoritmo cerca di indovinare per via semiempirica i migliori parametri da utilizzare nella routine iterativa.

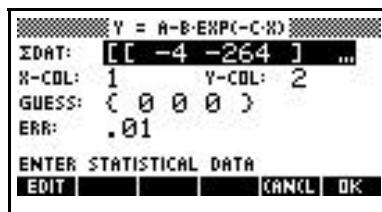
In questo caso ad ogni ciclo viene mostrato Er che è la somma delle differenze dei parametri cercati tra una iterazione e l'altra.

Esempio:

Xi	Yi	$y=A(1-\exp(-BX))$		
0	0.00			
1	25.92		guess	
2	45.12	A	0	
3	59.34	B	0	Er = 0.01
4	69.88			
5	77.69			
6	83.47		esatto	trovato
7	87.75	A	100	99.99917
8	90.93	B	0.3	0.300005
9	93.28			
10	95.02			

Y=A-Bexp(-CX)

Questo programma esegue l'interpolazione con la funzione: $y=A-B\exp(-CX)$



Il programma utilizza gli stessi algoritmi visti per il programma di interpolazione biesponenziale e necessita delle stesse informazioni di base; come nel caso della funzione di Morse è presente una funzione di previsione del valore di guess che entra in funzione quando $\text{guess} = \{ 0 \ 0 \ 0 \}$

Variabili esterne utilizzate:

MSG	:	lista dei messaggi dell'interfaccia grafica
ΣDAT	:	variabile statistica dell'HP 48
x	:	indice della colonna della variabile indipendente
y	:	indice della colonna della variabile dipendente
Xa	:	lista dei valori iniziali dei coefficienti dell'equazione cercata
Er	:	errore; è la somma delle differenze tra i coefficienti dell'equazione cercata calcolati in due iterazioni consecutive.

Algoritmo

Si utilizza lo stesso algoritmo utilizzato per l'interpolazione biesponenziale nella forma originale senza approssimazioni, quando $\text{guess} = \{ 0 \ 0 \ 0 \}$ un secondo algoritmo cerca di indovinare per via semiempirica i migliori parametri da utilizzare nella routine iterativa.

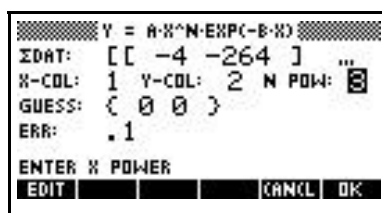
In questo caso ad ogni ciclo viene mostrato Er che è la somma delle differenze dei parametri cercati tra una iterazione e l'altra.

Esempio:

Xi	Yi	y=A-Bexp(-CX)		
0	5.00			
1	5.91		guess	
2	6.65	A	0	
3	7.26	B	0	Er = 0.01
4	7.75	C	0	
5	8.16			
6	8.49		esatto	trovato
7	8.77	A	10	9.993044
8	8.99	B	5	4.991586
9	9.17	C	0.2	0.200418
10	9.32			

$Y=A \cdot X^n \cdot \exp(-BX)$

Questo programma esegue l'interpolazione con la funzione: $Y=A \cdot X^n \cdot \exp(-BX)$.



Il programma utilizza gli stessi algoritmi visti per il programma di interpolazione biesponenziale e necessita delle stesse informazioni di base; come nel caso della funzione di Morse è presente una funzione di previsione del valore di guess che entra in funzione quando $\text{guess} = \{ 0 \ 0 \}$. In più bisogna indicare la potenza di n desiderata.

Variabili esterne utilizzate:

MSG : lista dei messaggi dell'interfaccia grafica
ΣDAT : variabile statistica dell'HP 48
x : indice della colonna della variabile indipendente
y : indice della colonna della variabile dipendente
Xb : lista dei valori iniziali dei coefficienti dell'equazione cercata
Er : errore; è la somma delle differenze tra i coefficienti dell'equazione cercata calcolati in due iterazioni consecutive.

Algoritmo

Si utilizza lo stesso algoritmo utilizzato per l'interpolazione biesponenziale nella forma originale senza approssimazioni, quando $\text{guess} = \{ 0 \ 0 \}$ un secondo algoritmo cerca di indovinare per via semiempirica i migliori parametri da utilizzare nella routine iterativa.

In questo caso ad ogni ciclo viene mostrato Er che è la somma delle differenze dei parametri cercati tra una iterazione e l'altra.

Esempio $y=A \cdot X^n \cdot \exp(-BX)$:

Xi	Yi	y=A*X^n*exp(-BX)			
0	0.00				
1	9.51	guess			
2	72.39	A	15	Er = 0.01	
3	232.39	B	0.01		
4	523.99	n	3		
5	973.50				
6	1600.17	esatto			
7	2417.08	A	10	trovato 10.000124	
8	3432.04	B	0.2	5.000114	
9	4648.31				
10	6065.31				

Gian Piero Luciani
Via Piave, 7
44022 Comacchio (FE) - ITALIA

Email: gluciani@area.ra.it
lucianigp@hotmail.com