

# GNLS: un Programma R per la Parametrizzazione dei Modelli agli Errori nelle Variabili \*

Simon L. Marshall  
Air Quality Services  
Department of Water and Environmental Regulation<sup>†</sup>  
Perth, WA 6000  
Australia

## Sunto

Il linguaggio di programmazione R, insieme ai suoi numerosi pacchetti per applicazioni specializzate, si è affermato come veicolo preferito per l'analisi esplorativa dei dati e la modellizzazione statistica. La funzione incorporata `nls()` disponibile per la stima dei parametri non lineari si basa su (i) la distinzione tra regressori e variabili di risposta; (ii) l'esistenza di una relazione funzionale esplicita tra di essi e (iii) l'assenza di errori nei regressori; poiché queste assunzioni non sono sempre giustificate, il presente articolo risponde alla necessità di un trattamento più generale che le eviti. Il problema dei minimi quadrati generalizzati viene formulato come una minimizzazione vincolata della somma ponderata dei residui al quadrato, da risolvere iterativamente per i regressori e i parametri. Gli input principali sono un sottoprogramma che valuta la funzione scalare o vettoriale da adattare, una matrice contenente le covarianze dei regressori in ogni punto di progetto e una stima iniziale dei parametri - tutte le derivate parziali richieste sono approssimate da differenze centrali. Il programma è adatto per l'applicazione a modelli definiti da una o più relazioni implicite

---

\*Da sottomettere al *Bollettino dell'Unione Matematica Italiana*.

<sup>†</sup>Indirizzo attuale: 24 Flinders Crescent, Bull Creek, Western Australia

non lineari tra parametri e regressori a ciascun dato, che si presentano in una vasta gamma di campi. È stata dimostrata l'applicazione a problemi di riduzione dei dati in chimica fisica e fisica chimica.

## 1 Introduzione

In un'ampia gamma di discipline, la definizione di relazioni quantitative tra insiemi di dati numerici implica il metodo dei minimi quadrati, mediante il quale è possibile determinare i valori ottimali dei parametri in una dipendenza funzionale postulata. L'esecuzione di tali calcoli non è mai stata così semplice. Per questo motivo, le funzioni di regressione lineare sono state integrate in molte calcolatrici tascabili economiche e le capacità di regressione lineare e non lineare più avanzate sono incorporate nel linguaggio R [R Core Team, 2017]. La sintassi naturale e le capacità grafiche di quest'ultimo rendono ora possibile l'analisi dei dati a un livello precedentemente accessibile solo agli esperti. Tuttavia, la quasi universalità dei metodi dei minimi quadrati non si riflette in una consapevolezza altrettanto diffusa dei loro presupposti impliciti. È quindi possibile eseguire l'analisi ai minimi quadrati senza tenere in debito conto questioni importanti come i pesi moltiplicativi applicati ai dati o persino se l'uso della somma dei quadrati come cifra di merito sia giustificato in primo luogo.

Si possono distinguere tre varietà principali di problemi di minimi quadrati. Nell'approccio più semplice, le variabili di risposta  $\{y_i\}$  sono correlate alle variabili regressori (indipendenti)  $\mathbf{x}_i$  e ai parametri  $\mathbf{z}$  mediante

$$y_i = f(\mathbf{x}_i, \mathbf{z}) + e \quad (1-1)$$

dove i regressori sono noti esattamente e il termine di errore  $e$  ha una varianza costante. I parametri vengono cercati per minimizzare la somma dei residui al quadrato,  $\sum_{i=1}^n [y_i - f(\mathbf{x}_i, \mathbf{z})]^2$ . Questo metodo può essere definito minimi quadrati non lineari non ponderati (o più correttamente, uniformemente ponderati) (UNLS, Uniformly-weighted Nonlinear Least Squares). In WNLS (Weighted Nonlinear Least Squares), si assume che i regressori siano noti esattamente, ma la varianza del termine di errore non è costante,

$$y_i = f(\mathbf{x}_i, \mathbf{z}) + e_i; \quad (1-2)$$

il peso è la varianza reciproca del termine di errore  $1/\sigma_i^2$  e la funzione obiettivo è  $\sum_{i=1}^n [y_i - f(\mathbf{x}_i, \mathbf{z})]^2 / \sigma_i^2$ . Nei problemi di minimi quadrati non lineari

generalizzati (GNLS, Generalized Nonlinear Least Squares) o di errori nelle variabili, si assume che tutte le variabili contengano errori. Si richiede che i parametri e i regressori minimizzino la somma ponderata dei residui al quadrato, soggetta a vincoli funzionali  $\mathbf{f}(\mathbf{x}_i, \mathbf{z}) = \mathbf{0}$ .

La letteratura sui problemi GNLS risale ai lavori pionieristici di Pearson [1901] e Deming [1943] tra gli altri (come rivisto da Macdonald e Thompson [1992]), precedendo di molti anni lo sviluppo di metodi pratici per la loro soluzione [Powell e Macdonald, 1972; Britt e Luecke, 1973; Anderson et al., 1978; Jefferys, 1980; Lybanon, 1984, 1985; Schwetlick e Tiller, 1985]. Considerando che anche il caso più semplice di adattamento di una funzione lineare a due variabili di precisione finita è non lineare, l'esecuzione manuale della soluzione iterativa richiesta sarebbe stata estremamente onerosa. La maggior parte della letteratura iniziale riguarda l'adattamento di modelli definiti da singole equazioni. I lavori successivi che hanno affrontato le sfide poste dalla parametrizzazione di modelli a più equazioni (come quelli che si presentano nell'adattamento delle equazioni di stato e di altri modelli termodinamici non lineari di interesse per l'ingegneria chimica) hanno comportato l'uso di programmi Fortran. L'applicazione potenziale di R a tali problemi non sembra essere stata studiata; lo scopo del presente articolo è dimostrare che R è, in effetti, idealmente adatto a questo scopo.

## 2 Parte Teorica

### 2.1 Programma quadratico non linearmente vincolato

Prima di procedere alla descrizione del codice R e delle routine associate, è forse utile presentare in questa sezione una derivazione abbreviata delle equazioni che definiscono l'approccio alla riduzione dei dati basato sugli errori nelle variabili. Ulteriori dettagli sono disponibili nei lavori citati nella sezione precedente. Il punto di partenza è l'identificazione del problema dei minimi quadrati come un programma quadratico non lineare: la minimizzazione di una funzione quadratica soggetta a vincoli funzionali non lineari. Più precisamente, dato un vettore di dati  $\mathbf{x}_0$ , si richiede di trovare un vettore di regressori  $\mathbf{x}$  e di parametri  $\mathbf{z}$  che soddisfino

$$\min \left\{ \frac{1}{2} \mathbf{r}^T \boldsymbol{\Sigma}^{-1} \mathbf{r} \right\} : \quad \mathbf{f}(\mathbf{x}, \mathbf{z}) = \mathbf{0}, \quad (2-1)$$

dove  $\mathbf{r} = \mathbf{x} - \mathbf{x}_0$  è un vettore di residui e  $\Sigma$  è la matrice di varianza-covarianza (solitamente diagonale a blocchi). La soluzione di questo programma non lineare può essere ottenuta con il metodo di Newton-Raphson, che prevede la linearizzazione su una soluzione di prova  $(\mathbf{x}_1, \mathbf{z}_1)$ , vale a dire,  $\mathbf{x} = \mathbf{x}_1 + \Delta\mathbf{x}$  e  $\mathbf{z} = \mathbf{z}_1 + \Delta\mathbf{z}$ , in termini di cui  $\mathbf{r} = \mathbf{r}_1 + \Delta\mathbf{x}$ . Il sistema di equazioni da risolvere che ne risulta può essere rappresentato in modo compatto in forma di matrice-vettore a blocchi come

$$\begin{bmatrix} \Sigma^{-1} & \mathbf{0} & (\nabla_x \mathbf{f})^T \\ \mathbf{0} & \mathbf{0} & (\nabla_z \mathbf{f})^T \\ \nabla_x \mathbf{f} & \nabla_z \mathbf{f} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{z} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\Sigma^{-1}\mathbf{r}_1 \\ \mathbf{0} \\ -\mathbf{f}(\mathbf{x}_1, \mathbf{z}_1) \end{bmatrix}; \quad (2-2)$$

si assume che tutte le derivate siano valutate in  $(\mathbf{x}_1, \mathbf{z}_1)$ . In questo caso,  $\boldsymbol{\lambda}$  è un vettore di moltiplicatori di Lagrange. L'insieme dei dati consiste di  $N$  punti, ciascuno con  $n_x$  regressori che soddisfano  $n_f$  equazioni di condizione che dipendono da  $n_z$  parametri nel vettore  $\mathbf{z}$ . Allora  $\Sigma$  avrà dimensioni  $Nn_x \times Nn_x$ , mentre  $\nabla_x \mathbf{f}$  e  $\nabla_z \mathbf{f}$  saranno rettangolari di dimensioni  $Nn_f \times Nn_x$  e  $Nn_f \times n_z$ , rispettivamente. Il primo passo per la soluzione di questo sistema è risolvere per l'incremento del regressore

$$\begin{aligned} \Sigma^{-1}\Delta\mathbf{x} + (\nabla_x \mathbf{f})^T \boldsymbol{\lambda} &= -\Sigma^{-1}\mathbf{r}_1 \\ \implies \Delta\mathbf{x} &= -[\mathbf{r}_1 + \Sigma(\nabla_x \mathbf{f})^T \boldsymbol{\lambda}] \end{aligned} \quad (2-3)$$

e sostituire questa soluzione nei vincoli linearizzati:

$$\begin{aligned} \nabla_x \mathbf{f} \Delta\mathbf{x} + \nabla_z \mathbf{f} \Delta\mathbf{z} &= -\mathbf{f} \\ \implies -\nabla_x \mathbf{f} [\mathbf{r}_1 + \Sigma(\nabla_x \mathbf{f})^T \boldsymbol{\lambda}] + \nabla_z \mathbf{f} \Delta\mathbf{z} &= -\mathbf{f}. \end{aligned} \quad (2-4)$$

Questo può essere riorganizzato in un'equazione dalla quale si possono determinare i moltiplicatori di Lagrange:

$$\begin{aligned} [\nabla_x \mathbf{f} \Sigma (\nabla_x \mathbf{f})^T] \boldsymbol{\lambda} &= \mathbf{f} - \nabla_x \mathbf{f} \mathbf{r}_1 + \nabla_z \mathbf{f} \Delta\mathbf{z} \\ \implies \boldsymbol{\lambda} &= [\nabla_x \mathbf{f} \Sigma (\nabla_x \mathbf{f})^T]^{-1} [\mathbf{f} - \nabla_x \mathbf{f} \mathbf{r}_1 + \nabla_z \mathbf{f} \Delta\mathbf{z}] \\ &\equiv \mathbf{W} [\mathbf{f} - \nabla_x \mathbf{f} \mathbf{r}_1 + \nabla_z \mathbf{f} \Delta\mathbf{z}] \end{aligned} \quad (2-5)$$

dove

$$\mathbf{W} \equiv [\nabla_x \mathbf{f} \Sigma (\nabla_x \mathbf{f})^T]^{-1} \quad (2-6)$$

si chiama la matrice dei pesi effettivi. L'incremento dei parametri si possono ottenere per la sostituzione di  $\boldsymbol{\lambda}$  nell'equazione media:

$$\begin{aligned} (\nabla_z \mathbf{f})^T \boldsymbol{\lambda} &= (\nabla_z \mathbf{f})^T \mathbf{W} [\mathbf{f} - \nabla_x \mathbf{f} \mathbf{r}_1 + \nabla_z \mathbf{f} \Delta \mathbf{z}] = \mathbf{0} \\ \implies (\nabla_z \mathbf{f})^T \mathbf{W} \nabla_z \mathbf{f} \Delta \mathbf{z} &= -(\nabla_z \mathbf{f})^T \mathbf{W} [\mathbf{f} - \nabla_x \mathbf{f} \mathbf{r}_1] \end{aligned} \quad (2-7)$$

quindi abbiamo

$$\Delta \mathbf{z} = -[(\nabla_z \mathbf{f})^T \mathbf{W} \nabla_z \mathbf{f}]^{-1} (\nabla_z \mathbf{f})^T \mathbf{W} [\mathbf{f} - \nabla_x \mathbf{f} \mathbf{r}_1]. \quad (2-8)$$

L'intero processo si ripete fino alla soddisfazione di un criterio di convergenza (che si basa di solito della deviazione relativa tra le stime successive dei parametri.) I comandi di manipolazione delle matrici integrati in R lo rendono particolarmente adatto alla realizzazione numerica di tali calcoli.

## 2.2 Relazione ai Minimi Quadrati Convenzionali

Per vedere come il trattamento precedente si collega con il metodo classico di minimi quadrati lineare e non lineare, supponiamo che ci sono due regressori per ciascuno di  $N$  punti, che soddisfanno un vettore  $N$ -dimensionale di vincoli funzionali  $\mathbf{f} = \mathbf{0}$ , e che il vettore dei dati e la sua matrice varianza-covarianza sono

$$\mathbf{x}_0 = [x_{0,11}, x_{0,12}, \dots, x_{0,N1}, x_{0,N2}]^T \quad \boldsymbol{\Sigma} = \text{diag}[\sigma_{11}^2, \sigma_{12}^2, \dots, \sigma_{N1}^2, \sigma_{N2}^2]. \quad (2-9)$$

Se i componenti del vettore di vincoli funzionali siano della forma

$$f_i \equiv g(x_{i1}, \mathbf{z}) - x_{i2} = 0, \quad i = 1, \dots, N, \quad (2-10)$$

dove  $x_{i1}$  e  $x_{i2}$  svolgono i ruoli di variabili indipendenti e dipendenti, rispettivamente, il gradiente dei vincoli rispetto ai regressori è

$$\nabla_x \mathbf{f} = \begin{bmatrix} \frac{\partial g}{\partial x_1}(x_{11}, \mathbf{z}) & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \frac{\partial g}{\partial x_1}(x_{21}, \mathbf{z}) & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \frac{\partial g}{\partial x_1}(x_{N1}, \mathbf{z}) & -1 \end{bmatrix}. \quad (2-11)$$

La matrice diagonale delle varianze effettive  $\nabla_x \mathbf{f} \Sigma (\nabla_x \mathbf{f})^T \equiv \mathbf{W}^{-1}$  ha  $i$ -esimo elemento

$$\begin{bmatrix} \frac{\partial g}{\partial x_1}(x_{i1}, \mathbf{z}) & -1 \end{bmatrix} \begin{bmatrix} \sigma_{i1}^2 & 0 \\ 0 & \sigma_{i1}^2 \end{bmatrix} \begin{bmatrix} \frac{\partial g}{\partial x_1}(x_{i1}, \mathbf{z}) \\ -1 \end{bmatrix} = \sigma_{i1}^2 \left( \frac{\partial g}{\partial x_1}(x_{i1}, \mathbf{z}) \right)^2 + \sigma_{i2}^2. \quad (2-12)$$

Quando le variabili indipendenti sono note esattamente,  $\sigma_{i1}^2 = 0$ , e  $\mathbf{W}$  contiene le varianze reciproche delle variabili dipendente. In questo caso, i componenti dispari del vettore dei residui svaniscono,

$$\mathbf{r}_1 \equiv \mathbf{x} - \mathbf{x}_0 = [0, x_{12} - x_{0,12}, 0, x_{22} - x_{0,22}, \dots, 0, x_{N2} - x_{0,N2}]^T. \quad (2-13)$$

La combinazione di questo con l'espressione precedente per  $\nabla_x \mathbf{f}$  dà

$$\mathbf{f} - (\nabla_x \mathbf{f}) \mathbf{r}_1 = \begin{bmatrix} g(x_{11}, \mathbf{z}) - x_{0,12} \\ \vdots \\ g(x_{N1}, \mathbf{z}) - x_{0,N2} \end{bmatrix}. \quad (2-14)$$

La matrice dei coefficienti nelle equazioni  $\mathbf{A} \Delta \mathbf{z} = -\mathbf{b}$  per  $\Delta \mathbf{z}$  è

$$\mathbf{A} = \begin{bmatrix} \partial g(x_{11}, \mathbf{z}) / \partial z_1 & \dots & \partial g(x_{N1}, \mathbf{z}) / \partial z_1 \\ \vdots & \dots & \vdots \\ \partial g(x_{11}, \mathbf{z}) / \partial z_P & \dots & \partial g(x_{N1}, \mathbf{z}) / \partial z_P \end{bmatrix} \begin{bmatrix} 1/\sigma_{12}^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1/\sigma_{N2}^2 \end{bmatrix} \quad (2-15)$$

$$\times \begin{bmatrix} \partial g(x_{11}, \mathbf{z}) / \partial z_1 & \dots & \partial g(x_{11}, \mathbf{z}) / \partial z_P \\ \vdots & \dots & \vdots \\ \partial g(x_{N1}, \mathbf{z}) / \partial z_1 & \dots & \partial g(x_{N1}, \mathbf{z}) / \partial z_P \end{bmatrix},$$

e il vettore della destra è

$$\mathbf{b} = \begin{bmatrix} \partial g(x_{11}, \mathbf{z}) / \partial z_1 & \dots & \partial g(x_{N1}, \mathbf{z}) / \partial z_1 \\ \vdots & \dots & \vdots \\ \partial g(x_{11}, \mathbf{z}) / \partial z_P & \dots & \partial g(x_{N1}, \mathbf{z}) / \partial z_P \end{bmatrix} \begin{bmatrix} 1/\sigma_{12}^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1/\sigma_{N2}^2 \end{bmatrix} \times$$

$$\begin{bmatrix} g(x_{11}, \mathbf{z}) - x_{0,12} \\ \vdots \\ g(x_{N1}, \mathbf{z}) - x_{0,N2} \end{bmatrix}. \quad (2-16)$$

Nel caso particolare di un modello lineare, gli elementi delle matrici di derivate parziali sono indipendenti di tutti i parametri. Questi sono le equazione che definiscono il metodo convenzionale di minimi quadrati, sviluppato nella maggior parte di libri sulla statistica e la riduzione di dati (vedere, ad esempio Press et al. [1992, pagine 675-683].)

## 3 Realizzazione in R

La realizzazione dello schema precedente dipende ovviamente della valutazione efficace e generale della relazione funzionale postulata e le sue derivate parziali rispetto ai regressori e ai parametri da adattare. Questi sono discussi nei paragrafi seguenti.

### 3.1 Valutazione della Funzione

La funzione da adattare deve essere valutata non solo come funzione dei regressori  $\mathbf{x}$  ma anche dei parametri regolabili  $\mathbf{z}$ . Più in generale, risulta desiderabile consentire un vettore di parametri fissi addizionali  $\mathbf{y}$ , che possono o meno essere richiesti; questi tre vettori sono forniti nell'elenco degli argomenti del sottoprogramma della funzione R. Ad esempio, una retta di intercetta  $z_1$  e pendenza  $z_2$  può essere rappresentata come una funzione implicita delle variabili  $x_1$  e  $x_2$

$$f = z_1 + z_2x_1 - x_2, \quad (3-1)$$

la valutazione della quale si può realizzare dal sottoprogramma

```
fline <- function(X,Y,Z){
  # Linear relation between the components of x
  #
  nf <- 1
  fvalue <- vector(length=nf)
  x <- X[1]
  y <- X[2]
  a <- Z[1]
  b <- Z[2]
  fvalue[1] <- a+b*x-y
  return(fvalue)
}
```

Questo calcolo è, naturalmente, triviale; il codice precedente viene fornito qui per illustrare la forma del sottoprogramma con cui l'ottimizzatore GNLS è progettato per funzionare. In questo caso, non c'è bisogno del vettore  $\mathbf{Y}$ , cui componenti si possono impostare a 0 nella routine chiamante.

## 3.2 Derivazione Numerica

La maggior parte degli ottimizzatori minimi quadrati coinvolgono le derivate parziali della funzione obiettivo, che dipendono dalle derivate della funzione da adattare. L'utilizzazione di derivate analitiche è da preferire per funzioni abbastanza semplici, in particolare quelle che sono lineari rispetto ai loro parametri. Ma derivate analitiche sono non sempre disponibili o convenienti, come si può trovare per funzioni molto complicate o relazioni implicite, e di solito non si possono trasferire ad altri problemi. Per trattare di tutte queste possibilità, le derivate richieste da GNLS sono approssimate da quozienti di differenze, in particolare il quoziente di differenze centrali

$$\frac{\partial f}{\partial x} \approx \frac{f(x+h) - f(x-h)}{2h}. \quad (3-2)$$

Nonostante che la derivazione numerica mediante questa formula o le sue controparti basate su differenze in avanti o all'indietro sia riconosciuta come processo intrinsecamente impreciso, risultati soddisfacenti si possono ottenere se la quantità  $h$  è scelta correttamente. La procedura utilizzata nel presente lavoro si basa sui suggerimenti forniti da Press et al. [1992, page 182], secondo cui  $h$  è posto uguale a  $x$  volte la radice cubica del relativo errore macchina, oppure la radice cubica dell'errore macchina relativo se  $x = 0$ . Questo dipende dal processore; per il computer Macintosh su cui sono stati eseguiti tutti i calcoli qui descritti, questo è approssimativamente  $4.8 \times 10^{-6}$ . L'errore macchina relativo, definito come il più piccolo numero positivo  $\varepsilon$  tale che  $1 + \varepsilon$  sia sensibilmente diverso da 1, è convenientemente accessibile in R come parametro `.Machine$double.eps`.

Nel metodo dei minimi quadrati non lineari generalizzati, le derivate parziali del vettore vincolo funzionale  $\mathbf{f}$  devono essere prese rispetto sia ai parametri del modello  $\mathbf{z}$ , sia ai regressori  $\mathbf{x}$ . Se le rispettive dimensioni di questi vettori sono  $n_f$ ,  $n_z$  e  $n_x$ , le derivate risultanti sono restituite come matrici Jacobiane con  $n_f$  righe e  $n_z$  o  $n_x$  colonne. Poiché le funzioni da derivare dipendono non solo da  $\mathbf{z}$  e  $\mathbf{x}$ , ma anche da eventuali parametri fissi in  $\mathbf{y}$ , tutti e tre i vettori devono essere forniti come argomenti ai sottoprogrammi di derivazione, insieme al nome della funzione da derivare, e la dimensione di  $\mathbf{f}$ . Per la derivazione rispetto ai regressori, una possibile realizzazione R è:

```
xjacob <- function(funk,n,x,y,z){
# This subprogram calculates the Jacobian matrix of a vector
# valued function funk of m variables with n components,
```

```

# with respect to x, for given vectors of fixed and
# adjustable parameters y and z, respectively. The required
# partial derivatives are approximated as central difference
# quotients.
#
scale <- (.Machine$double.eps)^(1/3)
m <- length(x)
xplus <- vector(length=m)
xminus <- vector(length=m)
fplus <- vector(length=n)
fminus <- vector(length=n)
aj <- matrix(nrow=n,ncol=m)
for(j in 1:m){
  xplus <- x
  xminus <- x
  if(x[j]==0){
    h=scale
  } else {
    h=scale*abs(x[j])
  }
  xplus[j] <- xplus[j]+h
  xminus[j] <- xminus[j]-h
  fplus <- funk(xplus,y,z)
  fminus <- funk(xminus,y,z)
  for(i in 1:n){
    aj[i,j] <- (fplus[i]-fminus[i])/(2*h)
  }
}
return(aj)
}

```

Derivazione rispetto ai parametri da adattare si realizza da:

```

zjacob <- function(funk,n,x,y,z){
  # This subprogram calculates the Jacobian matrix of a vector
  # valued function funk of m variables with n components,
  # with respect to adjustable parameters z, for given vectors
  # of fixed parameters y and regressors x. The required
  # partial derivatives are approximated as central difference
  # quotients.
  #
  scale <- (.Machine$double.eps)^(1/3)
  m <- length(z)
  zplus <- vector(length=m)
  zminus <- vector(length=m)

```

```

fplus <- vector(length=n)
fminus <- vector(length=n)
aj <- matrix(nrow=n,ncol=m)
for(j in 1:m){
  zplus <- z
  zminus <- z
  if(z[j]==0){
    h=scale
  } else {
    h=scale*abs(z[j])
  }
  zplus[j] <- zplus[j]+h
  zminus[j] <- zminus[j]-h
  fplus <- funk(x,y,zplus)
  fminus <- funk(x,y,zminus)
  for(i in 1:n){
    aj[i,j] <- (fplus[i]-fminus[i])/(2*h)
  }
}
return(aj)
}

```

### 3.3 Il Programma GNLS

Gli input da fornire all'ottimizzatore GNLS sono `ndata` punti di dati come un vettore di dimensioni `X0[1:ndata*nx]`; una matrice `Sigmax[1:ndata*nx, 1:nx]` che contiene i `ndata` blocchi di  $\Sigma$ ; i parametri fissi `y`; le stime iniziali dei parametri `z[1:nz]`; e il sottoprogramma `funk` che valuta le `nf` equazioni di condizioni che si applicano in ciascuno punto.

```

gnls <- function(funk,nf,X0,Sigmax,ndata,y,z){
  # This subprogram performs a Generalized Nonlinear Least Squares
  # fit of the nf-dimensional vector-valued function funk to a vector
  # X0 of regressors containing ndata data with nx variables at each
  # point, for which the block diagonal variance - covariance matrix is
  # Sigmax, supplied as a rectangular matrix with ndata*nx rows and
  # nx columns. Also supplied is a vector z[1:nz] containing the
  # initial estimates of the adjustable parameters, and a vector y that
  # contains any fixed parameters.
  #
  # Iteration parameters and key dimensions
  itmax <- 20
  tol <- 1.e-10
  nz <- length(z)

```

```

nx <- ncol(Sigmax)
#
# Arrays for the whole data set
A <- matrix(nrow=nz,ncol=nz)      #Normal eqn. coefficient matrix
B <- vector(length=nz)           #Normal eqn. right hand side
Deltax <- vector(length=ndata*nx) #Regressor increments
Deltaz <- vector(length=nz)      #Parameter increments
F <- vector(length=(nf*ndata))   #Constraints
Fx <- matrix(nrow=nf*ndata,ncol=nx) #x-Jacobian
Fz <- matrix(nrow=nf*ndata,ncol=nz) #z-Jacobian
G <- vector(length=nf*ndata)     #f-grad_xfr+grad_zfDelta z
R <- vector(length=(nx*ndata))   #Residuals
W <- matrix(nrow=nf*ndata,ncol=nf) #Weighting matrix
X <- vector(length=(nx*ndata))   #Regressors
#
# Arrays for individual points
a <- matrix(nrow=nz,ncol=nz)     #Normal eqn. coefficient matrix
b <- vector(length=nz)           #Normal eqn. right hand side
deltax <- vector(length=nx)      #Regressor increments
f <- vector(length=nf)           #Constraints
fx <- matrix(nrow=nf,ncol=nx)    #x-Jacobian
fz <- matrix(nrow=nf,ncol=nz)    #z-Jacobian
g <- vector(length=nf)           #f-grad_xfr+grad_zfDelta z
lambda <- vector(length=nf)      #Lagrange multipliers
r <- vector(length=nx)           #Residuals
sigmax <- matrix(nrow=nx,ncol=nx) #Variance-covariance matrix
w <- matrix(nrow=nf,ncol=nf)     #Weighting matrix
x <- vector(length=nx)           #Regressors
#
# Initial regressor and parameter vectors
X <- X0
R[1:(nx*ndata)] <- 0
#
# Main iteration loop:
cat("Iterating...","\n")
for(iter in 1:itmax){
  # Initialization
  for(k in 1:nz){
    A[k,] <- 0
    B[k] <- 0
  }
  F <- NULL
  Fx <- NULL
  Fz <- NULL
  G <- NULL
}

```

```

W <- NULL
#
# Weighting matrix, normal equations
for(i in 1:ndata){
  #
  # Calculation of single-point arrays:
  r <- R[(nx*(i-1)+1):(nx*i)]
  x <- X[(nx*(i-1)+1):(nx*i)]
  sigmax <- Sigmax[(nx*(i-1)+1):(nx*i),]
  sigmax <- sigmax + 1.e-15 * diag(nx)
  f <- funk(x,y,z)
  fx <- xjacob(funk,nf,x,y,z)
  fz <- zjacob(funk,nf,x,y,z)
  g <- f - fx %%% r
  w <- solve(fx %%% (tcrossprod(sigmax,fx)))
  a <- crossprod(fz, (w %%% fz))
  b <- crossprod(fz, (w %%% g))
  #
  # Accumulation of whole-set arrays:
  A <- A+a
  B <- B+b
  F <- rbind(F,f)
  Fx <- rbind(Fx,fx)
  Fz <- rbind(Fz,fz)
  G <- rbind(G,g)
  W <- rbind(W,w)
}
# Solution for parameter increment vector
A <- solve(A)
Deltaz <- -A %%% B
G <- G + Fz %%% Deltaz
Deltax <- NULL
#
# Lagrange multipliers and regressor increment vector
for(i in 1:ndata){
  for(j in 1:nx){
    r[j] <- R[nx*(i-1)+j]
    sigmax[j,] <- Sigmax[nx*(i-1)+j,]
  }
  for(k in 1:nf){
    fx[k,] <- Fx[nf*(i-1)+k,]
    g[k] <- G[nf*(i-1)+k]
    w[k,] <- W[nf*(i-1)+k,]
  }
  lambda <- w %%% g
}

```

```

    deltax <- -(r + sigmax %*% crossprod(fx,lambda))
    Deltax <- rbind(Deltax,deltax)
  }
  #
  # Convergence assessed on relative error of parameter estimates
  relerr <- sqrt(crossprod(Deltaz)/crossprod(z))
  cat(iter,relerr,z,"\n")
  # Converged - exit loop and return
  if(relerr < tol){break}
  # Not yet converged - update z and R and try again
  z <- z+Deltaz
  R <- R+Deltax
  X <- R+X0
}
#
# Weighted sum of squared residuals etc.
nu <- -nz
chisq <- 0
for(i in 1:ndata){
  for(j in 1:nx){
    r[j] <- R[nx*(i-1)+j]
    sigmax[j,] <- Sigmax[nx*(i-1)+j,]
  }
  for(k in 1:nx){
    test <- ceiling(sigmax[k,k])
    if(test >= 1){nu <- nu+1} #Fixed regressors excluded from d.o.f.
  }
  sigmax <- sigmax + 1.e-15 * diag(nx)
  chisq <- chisq + crossprod(r,solve(sigmax)) %*% r
}
cat("Degrees of freedom = ",nu,"\n")
prob <- pchisq(chisq,df=nu,lower.tail=FALSE)
#
# Standard errors in parameter estimates
sigmaz <- vector(length=nz)
for(k in 1:nz){
  sigmaz[k] <- sqrt(A[k,k]*chisq/nu)
}
# Assembly of results in a list:
results <- list(z,sigmaz,X,chisq,prob)
return(results)
}

```

Tabella 1: Dati di prova di Pearson [1901] per l'adattamento di una retta, con i pesi moltiplicativi suggeriti da York [1966].

$i$	$x_{i1}$	$x_{i2}$	$W_{ix1} = 1/\sigma_{ix1}^2$	$W_{ix2} = 1/\sigma_{ix2}^2$
1	0.0	5.9	1000.0	1.0
2	0.9	5.4	1000.0	1.8
3	1.8	4.4	500.0	4.0
4	2.6	4.6	800.0	8.0
5	3.3	3.5	200.0	20.0
6	4.4	3.7	80.0	20.0
7	5.2	2.8	60.0	70.0
8	6.1	2.8	20.0	70.0
9	6.5	2.4	1.8	100.0
10	7.4	1.5	1.0	500.0

## 4 Modelli con una Equazione

L'applicazione più comune dell'analisi dei minimi quadrati è alla parametrizzazione di una sola relazione funzionale tra due variabili misurate. La distinzione convenzionale tra le variabili dipendenti e indipendenti è utile solo nel caso limitante dove i valori di una variabile sono noti esattamente. Quando si suppone che entrambi siano di precisione finita, la scelta di uno come dipendente è dettata da ciò che può essere risolto più facilmente in termini dell'altro; il trattamento degli errori è anche legato alla formulazione della funzione da adattare. Quando i vincoli sono definiti implicitamente, la distinzione è arbitraria. In questa sezione, questi argomenti sono considerati con particolare riferimento all'adattamento di funzioni lineari o polinomiali, che si presentano frequentemente in contesti pratici.

### 4.1 Adattamento di una retta

Gli effetti dei pesi moltiplicativi nell'adattamento di una retta si può illustrare per un insieme di date usato originalmente da Pearson [1901], con pesi proposti da York [1966]. Questi dati, che sono diventati un punto di riferimento per verificare l'operazione di programmi dei minimi quadrati non lineari, si presentano nella Tabella 1 e si mostrano nella Figura 1, dove le lunghezze delle barre di errore sono uguali al doppio della deviazione standard di ciascuno dato. Identificando i regressori  $(x_1, x_2)$  con i valori della

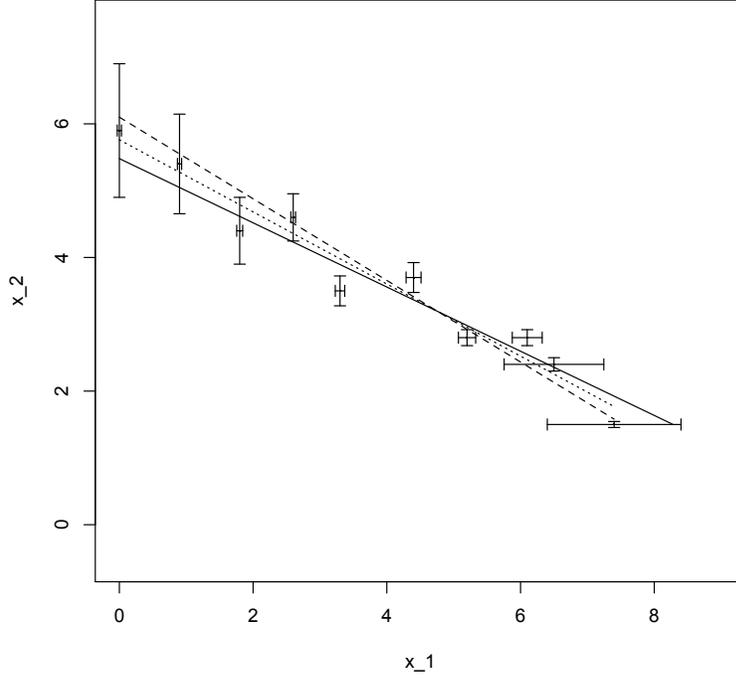


Figura 1: Adattamenti di una retta ai dati di Pearson [1901], illustrando gli effetti di diversi pesi moltiplicativi. Retta continua: adattamento GNLS, pesi applicati ai valori  $x_1$  e  $x_2$ ; retta tratteggiata: valori  $x_1$  fissi, pesi applicati ai valori  $x_2$ ; retta punteggiata: regressione lineare con pesi uniformi.

Tabella 1, l' $i$ -esimo componente del vettore dei vincoli funzionali è

$$f_i = z_1 + z_2 x_{i1} - x_{i2}, \quad (4-1)$$

come valutata a ciascuno punto dal sottoprogramma `fline` della sezione precedente.

La retta continua della Figura 1 è l'adattamento con gli errori nelle variabili (GNLS), dove l' $i$ -esimo blocco della matrice dei pesi  $\Sigma$  è

$$(\Sigma)_i = \begin{bmatrix} W_{ix1}^{-1} & 0 \\ 0 & W_{ix2}^{-1} \end{bmatrix} \quad (4-2)$$

e  $W_{ix1}$  e  $W_{ix2}$  sono rispettivamente le seconde e quarte colonne del Tabello 1; questo produce le stime dei parametri

$$z_1 = 5.47991 \pm 0.2394977, \quad z_2 = -0.4805334 \pm 0.04708018. \quad (4-3)$$

La retta tratteggiata è dei minimi quadrati pesati con regressori fissi (WNLS), con blocchi di varianza-covarianza

$$(\Sigma)_i = \begin{bmatrix} 0 & 0 \\ 0 & W_{ix2}^{-1} \end{bmatrix}, \quad (4-4)$$

che produce

$$z_1 = 6.100109 \pm 0.4240595, \quad z_2 = -0.610813 \pm 0.06234095. \quad (4-5)$$

Per i minimi quadrati uniformemente ponderati (UNLS), la varianza costante si annulla nelle equazioni normali e può essere impostata uguale a 1, quindi il blocco varianza-covarianza è

$$(\Sigma)_i = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad (4-6)$$

che produce

$$z_1 = 5.761185 \pm 0.1894852, \quad z_2 = -0.5395773 \pm 0.04212655. \quad (4-7)$$

Anche se le differenze tra queste stime dei parametri mostrano che l'attribuzione di varianze ai regressori esercita un effetto significativo, la vicinanza della prima (tutte le variabili ponderate) alla terza (regressori fissi, pesi trascurati) è da notare. Probabilmente questo è una conseguenza dell'eteroschedasticità introdotta dai pesi di York. Essendo i più alti per il primo valore  $x$  e l'ultimo valore  $y$ , questi pesi tendono a forzare la linea attraverso questi due punti.

## 4.2 Adattamento dei Polinomi

Considerando la linearità dell'equazione 4-1 rispetto sia alle variabili che ai parametri, e la comparabilità dei pesi assegnati alle variabili agli estremi opposti degli intervalli di valori, la parametrizzazione di questa funzione per regressione di  $x_2$  su  $x_1$  o di  $x_1$  on  $x_2$  sono essenzialmente equivalenti. Una grande parte di questa equivalenza svanisce con la postulazione di una relazione non

lineare, come una dipendenza polinomiale che è lineare rispetto ai parametri, ma non lineare rispetto le variabili. Tali funzioni trovano applicazioni al problema della costruzioni di equazioni di stato dei gas che descrivono la relazione tra la densità, la pressione, e la temperatura. A parte della sua ovvia importanza pratica, questo problema è interessante per ammettere delle diverse formulazioni, che sono analiticamente e computazionalmente inequivalenti.

Equazioni di stato si possono classificare generalmente come esplicite di pressione o esplicita di densità, secondo che la pressione  $p$  sia espressa come funzione della densità  $\rho$  ( $\text{kg}\cdot\text{m}^{-3}$ ) e la temperatura  $T$  (K), o la densità sia espressa come funzione della pressione e la temperatura, rispettivamente. Tra le equazioni esplicite di pressione, la più semplice è la forma polinomiale

$$p = \frac{\rho RT}{M} [1 + B(T)\rho + C(T)\rho^2 + D(T)\rho^3 + \dots] \quad (4-8)$$

dove  $R$  è la costante dei gas e  $M$  è la massa molare ( $\text{kg}\cdot\text{mol}^{-1}$ .) Le funzioni  $B$ ,  $C$ ,  $D$ , ... si chiamano i secondi, terzi, quarti ... coefficienti viriali. Secondo la teoria meccanica-statistica, i coefficienti misurano le contribuzioni di interazioni tra due, tre, quattro ... moleculi alle deviazioni dall'equazione di stato dei gas perfetti

$$p = \frac{\rho RT}{M}; \quad (4-9)$$

in alternativa, possono essere considerate come costanti empiriche da determinare dalle isoterme pressione-densità misurate. I coefficienti viriali sono fortemente dipendenti dalla temperatura, e generalmente diminuiscono in valore assoluto all'aumentare della temperatura. L'analisi dei dati non isotermici implica la determinazione delle costanti nelle funzioni assunte per descrivere questa dipendenza dalla temperatura, la cui forma dipende dall'intervallo di temperature rappresentato nelle misurazioni. Le osservazioni pertinenti che si possono fare su questa equazione sono le seguenti:

1. Con la migliore strumentazione disponibile, la precisione realizzabile nelle misure di pressione è notevolmente superiore a quella delle misure di densità.
2. A basse densità, la maggior parte della variazione di pressione è spiegata dal primo termine (equazione dei gas ideali), che tenderà a mascherare gli altri contributi.

3. I termini viriali superiori sono tipicamente piccoli; l'intervallo delle densità misurate è solitamente insufficiente per consentire una stima statisticamente significativa per  $D$ .

Per l'adattamento di un'equazione esplicita di pressione, la densità gioca il ruolo di variabile indipendente. L'Osservazione 1, che gli errori nella densità sono maggiori degli errori nella pressione, contraddice quindi un presupposto chiave fatto nei minimi quadrati convenzionali e indica la necessità di un trattamento GNLS. La predominanza del termine di gas ideale a cui si allude nell'Osservazione 2 può essere mitigata scrivendo l'equazione del viriale nella forma

$$Z \equiv \frac{pM}{\rho RT} = 1 + B(T)\rho + C(T)\rho^2 + D(T)\rho^3 + \dots \quad (4-10)$$

dove la quantità  $Z$  è il fattore di comprimibilità; in questi termini, l'equazione di stato dei gas perfetti è semplicemente  $Z = 1$ , che è il termine costante del polinomio viriale. I contributi dei termini viriali superiori si possono determinare più precisamente. Trascurando i termini quarti e superiori, e introducendo la densità ridotta  $\rho_r = \rho/\rho_c$ , dove  $\rho_c$  è la densità critica, l'equazione viriale troncata a  $C$  è

$$Z = 1 + B_r(T)\rho_r + C_r(T)\rho_r^2, \quad (4-11)$$

con i coefficienti viriali scalati

$$B_r(T) = B(T)\rho_c, \quad C_r(T) = C(T)\rho_c^2.$$

Ci sono due maniere a cui gli errori requisiti per un adattamento GNLS di quest'equazione si possono determinare. Secondo la prima, le due variabili dell'adattamento sono identificate come le quantità derivate  $\rho_r$  e  $Z$ , per le quale le varianze requisiti vengono calcolati dagli errori delle misure secondo la formula di Gauss per la propagazione di errori,

$$\sigma_Z^2 = \left(\frac{\partial Z}{\partial \rho}\right)^2 \sigma_\rho^2 + \left(\frac{\partial Z}{\partial p}\right)^2 \sigma_p^2 + \left(\frac{\partial Z}{\partial T}\right)^2 \sigma_T^2, \quad \sigma_{\rho_r}^2 = \frac{\sigma_\rho^2}{\rho_c^2}. \quad (4-12)$$

Questo approccio ha il vantaggio di utilizzare due variabili invece di tre. Però prendendo in considerazione la formulazione implicita dei vincoli funzionali, l'approccio alternativo più diretto di rappresentare questi come

$$\frac{pM}{\rho RT} - 1 - B_r(T)\frac{\rho}{\rho_c} - C_r(T)\left(\frac{\rho}{\rho_c}\right)^2 = 0 \quad (4-13)$$

e usare direttamente le misure e i loro errori (cioè, senza calcolari prima  $Z$  and  $\rho_r$ ) è da preferire.

Queste idee si possono fare più definite per l'adattamento dell'equazione viriale a tre parametri

$$Z = 1 + \left( \frac{M_1}{T_r^{1/4}} + \frac{M_2}{T_r^{5/4}} \right) \rho_r + \frac{M_3}{T_r} \rho_r^2 : \quad \rho_r = \frac{\rho}{\rho_c}, T_r = \frac{T}{T_c} \quad (4-14)$$

a 168  $\rho p T$  misure su metana supercritica da Kleinrahm et al. [1988] nelle intervalli  $273.15 \leq T/\text{K} \leq 323.15$ ,  $0.1 \leq p/\text{MPa} \leq 8$ . I parametri fissi  $R=8.31451 \text{ J}\cdot\text{K}^{-1}\cdot\text{mol}^{-1}$ , and  $M = 16.0428 \times 10^{-3} \text{ kg}\cdot\text{mol}^{-1}$  requisiti per calcolare  $Z$ , e le costanti criticali  $\rho_c/\text{kg}\cdot\text{m}^{-3} = 162.660$ ,  $p_c/\text{MPa} = 4.5992$ ,  $T_c/\text{K} = 190.551$  requisiti per calcolare la temperatura e la densità ridotte sono forniti nel vettore  $\mathbf{y} = \mathbf{y}[1:5]$ . Con le identificazioni  $\mathbf{x}_i \equiv (\rho, p, T)$ ,  $\mathbf{y} = [R, M, \rho_c, p_c, T_c]^T$ ,  $\mathbf{z} \equiv [M_1, M_2, M_3]^T$ , l' $i$ -esimo componente del vettore dei vincoli

$$f_i = \frac{x_{i2}y_2}{x_{i1}x_{i3}y_2} - 1 - \left[ \frac{z_1}{(x_{i3}/y_5)^{1/4}} + \frac{z_2}{(x_{i3}/y_5)^{5/4}} \right] \frac{x_{i1}}{y_3} - \frac{z_3}{(x_{i3}/y_5)} \left( \frac{x_{i1}}{y_3} \right)^2. \quad (4-15)$$

Utilizzando le stime fornite da Kleinrahm et al. [1988] per la precisione delle loro misure, *viz.*,

$$\begin{bmatrix} \delta x_{0,i1} \\ \delta x_{0,i2} \\ \delta x_{0,i3} \end{bmatrix} = \begin{bmatrix} \max(0.0002, 0.0002x_{0,i1}) \\ \max(0.00003, 0.00007x_{0,i2}) \\ 0.003 \end{bmatrix}, \quad (4-16)$$

i blocchi diagonale della matrice varianza-covarianza possono essere approssimati da

$$\Sigma_i = \text{diag} [(\delta x_{0,i1})^2, (\delta x_{0,i2})^2, (\delta x_{0,i3})^2]. \quad (4-17)$$

Quando iniziato al punto  $\mathbf{z}_1 = [0.6695033, -1.808498, 0.3917327]^T$ , generato dal adattamento UNLS di  $Z - 1$  in funzione di  $\rho_r$ , l'iterazione GNLS converge dopo tre iterazioni al

$$\mathbf{z} = \begin{bmatrix} 0.6694699 \pm 0.0002190418 \\ -1.808442 \pm 0.0003494479 \\ 0.3917198 \pm 0.0001985114 \end{bmatrix}. \quad (4-18)$$

I dati e l'adattamento sono mostrati nella Figura 2. L'alta qualità di questa rappresentazione risulta non solo dall'alta precisione dei dati originali, ma anche dall'ottimizzazione evolutiva utilizzato da Kleinrahm et al. [1988] per

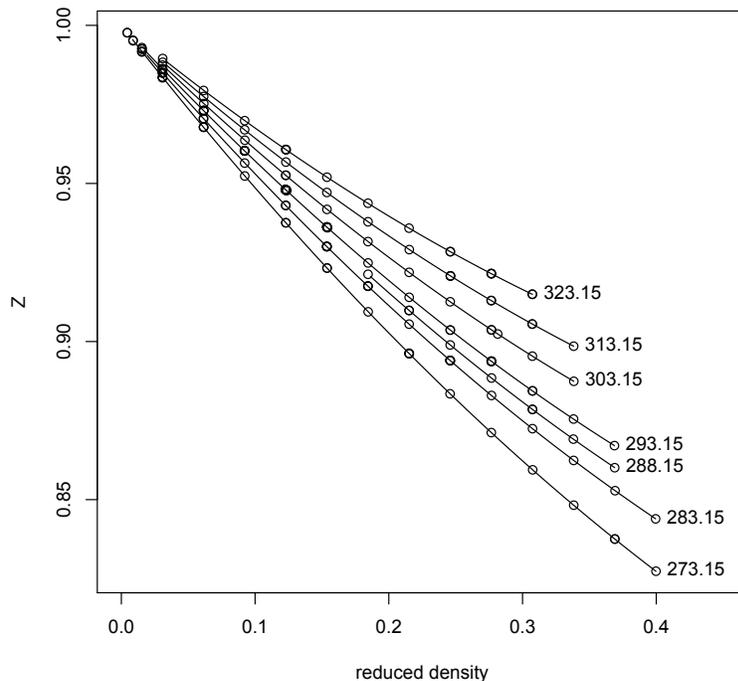


Figura 2: Adattamento dell'equazione viriali troncata a  $C$  4-14 a 168 misure volumetriche di metana da Kleinrahm et al. [1988]. La matrice diagonale delle varianze è data dall'equazione 4-17, e le curve sono calcolate con le stime GNLS dei parametri nell'equazione 4-18.

trovare la migliore forma funzionale per la dipendenza dei coefficienti viriali della temperatura. Senza descrivere i dettagli di quello processo, è da notare che poiché la figura di merito usata per distinguere le funzioni possibile si basa di somme di quadrati, l'uso di GNLS invece di UNLS potrebbe risultare in scelte diverse, poiché ci sono più gradi di libertà nel primo trattamento che nel secondo.

Poiché il componente di  $\mathbf{f}$  fornito dal sottoprogramma che valuta le equazioni dei vincoli per ogni punto dei dati non è adatto per la costruzione di grafici, è una buona pratica di fornire un sottoprogramma che valuta la funzione adatta. Per l'equazione viriale troncata a  $C$  questi codici sono

```

Cvirial.eos <- function(x,y,z){
  # This evaluates the z-explicit form of the C-truncated virial eos.
  #
  nf <- 1
  f <- vector(length=nf)
  # Input variables....
  rhokm3 <- x[1]
  pPa <- x[2]*1.e+06
  TK <- x[3]
  # Fixed parameters...
  Rgas <- y[1]
  mw <- y[2]*1.e-03
  rhoc <- y[3]
  pc <- y[4]*1.e+06
  Tc <- y[5]
  # Adjustable parameters...
  N1 <- z[1]
  N2 <- z[2]
  N3 <- z[3]
  # EOS variables...
  Tr <- TK/Tc
  rhor <- rhokm3/rhoc
  Z <- (pPa*mw)/(rhokm3*Rgas*TK)
  # Function value
  f[1] <- Z - 1 - rhor*(N1/Tr^(1/4)+N2/Tr^(5/4)) - rhor*rhor*N3/Tr
  return(f)
}

Cvirial.eval <- function(rhor,Tr,M1,M2,M3){
  # This subprogram returns the compressibility factor according to the
  # C-truncated virial eos fitted to the density measurements of Kleinrahm
  # et al. (1988)
  #
  B <- M1/Tr^0.25+M2/Tr^1.25
  C <- M3/Tr
  Z <- (C*rhor+B)*rhor+1
  return(Z)
}

```

### 4.3 Vincoli Impliciti vs. Espliciti

La forma implicita della funzione postulata è adottata qui non solo come conseguenza naturale della formulazione implicita del problema di ottimizzazione, ma anche perché offre più di flessibilità delle formulazioni esplicite. Una formula esplicita per la variabile risposta in funzione dei regressori e dei

parametri - se esista (come si suppone dalla funzione `nls()`) - è non necessariamente utile per adattamento minimi-quadrati. Questo si può illustrare per un cerchio di raggio  $r$  centrato al punto  $(x_{01}, x_{02})$ , quando rappresentato esplicitamente secondo

$$g(x_1, r, x_{01}, x_{02}) = x_{02} \pm \sqrt{r^2 - (x_1 - x_{01})^2}. \quad (4-19)$$

A parte dell'ambiguità del segno, le derivate parziali di  $g$ , viz.,

$$\frac{\partial g}{\partial r} = \frac{r}{\sqrt{r^2 - (x_1 - x_{01})^2}}, \quad \frac{\partial g}{\partial x_{01}} = \frac{x_1 - x_{01}}{\sqrt{r^2 - (x_1 - x_{01})^2}}, \quad \frac{\partial g}{\partial x_{02}} = 1, \quad (4-20)$$

sono molto grandi quando  $x_1$  avvicina le estremità orizzontali del cerchio. Questi grandi valori sono molto male approssimati dai quozienti di differenza, il che potrebbe causare problemi numerici nella soluzione delle equazioni normali (es. mal-condizionamento). Però con la formulazione implicita

$$f(x_1, x_2 | r, x_{01}, x_{02}) = (x_1 - x_{01})^2 + (x_2 - x_{02})^2 - r^2 \quad (4-21)$$

l'ambiguità del segno svanisce, e le derivate parziali

$$\frac{\partial f}{\partial r} = -2r, \quad \frac{\partial f}{\partial x_{01}} = -2(x_1 - x_{01}), \quad \frac{\partial f}{\partial x_{02}} = -2(x_2 - x_{02}), \quad (4-22)$$

sono matematicamente rispettabili per tutti i valori di  $x_1$  e  $x_2$ . La formulazione implicita può essere illustrata per l'adattamento di un cerchio  $f(x_1, x_2 | z_1, z_2, z_3) = 0$  ai dati sintetici rumorosi generati secondo le equazioni

$$\begin{aligned} x_{i1}^{(0)} &= x_{01} + r \cos \frac{i\pi}{15} + \varepsilon_{x1} \text{rnd}[\mathcal{N}(0, 1)] \\ x_{i2}^{(0)} &= x_{02} + r \sin \frac{i\pi}{15} + \varepsilon_{x2} \text{rnd}[\mathcal{N}(0, 1)] \end{aligned} \quad (4-23)$$

dove  $i = 1, \dots, 30$ ,  $x_{01} = x_{02} = 0$ ,  $\text{rnd}[\mathcal{N}(0, 1)]$  è un numero aleatorio dalla distribuzione normale con valore atteso 0 e varianza 1, e le ampiezze degli errori sono  $\varepsilon_{x1}=0.1$  e  $\varepsilon_{x2}=0.2$ . Per l'adattamento del cerchio a questi dati, le stime iniziali

$$z_1^{(0)} \equiv \bar{x}_{i1} = \frac{1}{n} \sum_{i=1}^n x_{i1}^{(0)}, \quad z_2^{(0)} \equiv \bar{x}_{i2} = \frac{1}{n} \sum_{i=1}^n x_{i2}^{(0)},$$

e

$$z_3^{(0)} \equiv r = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_1 - \bar{x}_1)^2 + (x_2 - \bar{x}_2)^2} \quad (4-24)$$

si possono usare. Per una realizzazione, i valori di queste quantità sono

$$\bar{x}_{i1} = -0.0082881, \quad \bar{x}_{i2} = -0.03694573, \quad r = 10.05141;$$

il sottoprogramma GNLS produce, dopo 8 iterazioni, le stime di parametri

$$z_1 = x_0 = 0.008619789 \pm 0.02536196, \quad z_2 = y_0 = -0.03285755 \pm 0.03579911, \\ z_3 = r = 10.04151 \pm 0.0206947. \quad (4-25)$$

Il cerchio adattato è mostrato nella Figura 3.

## 5 Modelli a Equazioni Multiple

Le sezioni precedenti descrivono la soluzione di problemi di errori nelle variabili in cui una singola equazione di condizione deve essere soddisfatta dai regressori calcolati in ogni punto dati. L'argomento di questa sezione è l'adattamento di modelli in cui si deve soddisfare un sistema di equazioni. Per dati circolari come quelli appena considerati, le posizioni dei dati possono essere specificate parametricamente in termini di coordinate polari ( $x = r \cos \theta, y = r \sin \theta$ ), ma questa rappresentazione è evidentemente scomoda senza specificazione del punto di origine e una direzione di riferimento (rispetto alla quale viene misurata la coordinata angolare). Un esempio interessante di applicazione di GNLS a problemi che ammettono formulazioni equivalenti a equazione singola o doppia parametrica è l'adattamento di modelli teorici di rilassamento dielettrico ai dati di permittività complessa, costituiti da parti reali e immaginarie misurate in funzione della frequenza. Per questo scopo, è possibile utilizzare l'ampia raccolta di misure accurate sui fluidi dielettrici di riferimento, pubblicata da Gregory e Clarke [2012]. Nel presente contesto, la caratteristica più importante di questi dati è l'indicazione delle incertezze sperimentali per tutte le misure, che consente un'assegnazione ben informata degli elementi della matrice di varianza-covarianza per i regressori. In base alla teoria di Debye, la permittività complessa  $\epsilon' + i\epsilon''$  di un fluido con un singolo tempo di rilassazione  $\tau$  ha componenti reali e immaginarie che dipendono della frequenza angolare  $\omega$  secondo

$$\epsilon' = \epsilon_\infty + \frac{\epsilon_0 - \epsilon_\infty}{1 + (\omega\tau)^2}, \quad \epsilon'' = \frac{\omega\tau(\epsilon_0 - \epsilon_\infty)}{1 + (\omega\tau)^2}, \quad (5-1)$$

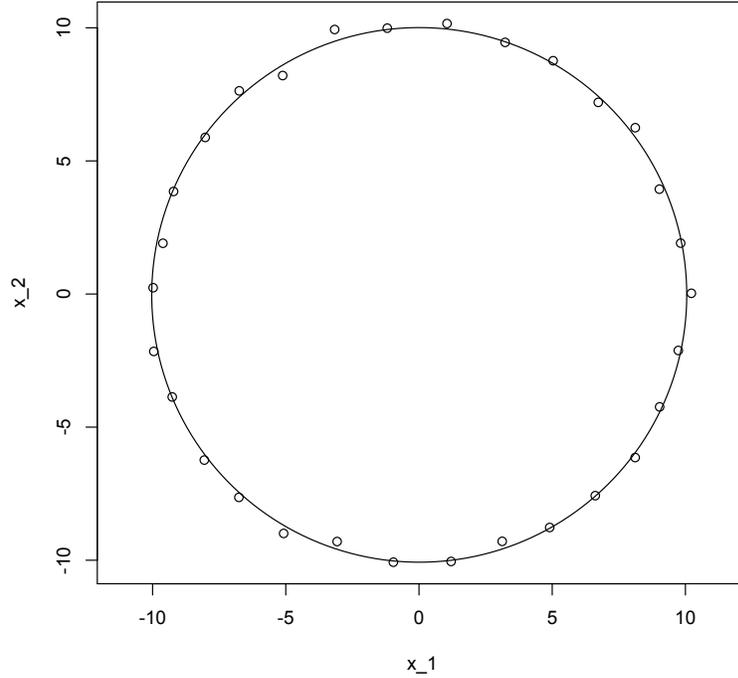


Figura 3: Cerchio adatto ai dati sintetici rumorosi generati dall'equazioni 4-23 con  $\epsilon_x=0.1$  e  $\epsilon=0.2$ .

dove  $\epsilon_0$  e  $\epsilon_\infty$  sono rispettivamente le permittività statica e di frequenza infinita che, oltre a  $\tau$ , sono da determinare. Estrapolazione della permittività ai limiti di frequenza zero e infinita si può realizzare mediante il grafico Cole-Cole, che si basa sull'osservazione che un grafico del componente immaginario  $\epsilon''$  contro il componente reale  $\epsilon'$  dovrebbe essere una semicerchio, centrato al punto  $((\epsilon_0 + \epsilon_\infty)/2, 0)$ . Che sia davvero così si può verificare per la formazione dell'equazione di un tale cerchio

$$\left[ \epsilon' - \frac{\epsilon_0 + \epsilon_\infty}{2} \right]^2 + (\epsilon'')^2 = \left[ \frac{\epsilon_0 - \epsilon_\infty}{2} \right]^2, \quad (5-2)$$

nella quale sono sostituite le espressioni per  $\epsilon'$ ,  $\epsilon''$ . Gli errori nei componenti della permittività possono spostare il centro del cerchio dall'asse; conoscenza

di questo spostamento è necessario per un'estrapolazione corretta. Quindi, se il semicerchio Cole-Cole si supponga rappresentare per l'equazione

$$(\epsilon' - z_1)^2 + (\epsilon'' - z_2)^2 - z_3^2 = 0, \quad (5-3)$$

e i parametri  $z_1$ ,  $z_2$ , e  $z_3$  sono stimati da GNLS come nella sezione precedente, le permittività limitanti si possono ottenere da

$$\epsilon_0 = z_1 + \sqrt{z_3^2 - z_2^2}, \quad \epsilon_\infty = z_1 - \sqrt{z_3^2 - z_2^2}. \quad (5-4)$$

Anche se solo due dei parametri Debye sono presenti nell'equazione del semicerchio Cole-Cole, il terzo ( $\tau$ ) può essere stimato dall'osservazione che i componenti della permittività si possono combinare secondo l'equazione

$$(\epsilon' - \epsilon_\infty)\omega\tau = \epsilon'' \quad (5-5)$$

che può essere risolta per  $\tau$  per minimi quadrati lineari. Secondo la formulazione pseudo-inversa dei minimi quadrati lineari, il vettore dei parametri  $\mathbf{z}$  di un modello definito per il sistema lineare sovradeterminato

$$\mathbf{A}\mathbf{z} = \mathbf{y} \quad (5-6)$$

è dato esplicitamente per

$$\mathbf{z} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}. \quad (5-7)$$

La forma di questa che si applica all'estimazione del parametro singolo  $\tau$  di questo modello è

$$\tau = \frac{\sum_{i=1}^n (\epsilon''_i \omega_i) (\epsilon'_i - \epsilon_\infty)}{\sum_{i=1}^n [(\epsilon'_i - \epsilon_\infty) \omega_i]^2}. \quad (5-8)$$

I parametri  $\epsilon_0$ ,  $\epsilon_\infty$  e  $\tau$  così determinati dovrebbero fornire buone stime iniziali per l'adattamento GNLS delle equazioni di Debye ai dati di permittività. Per illustrare ciò, si consideri la permittività complessa del metanolo a 20°C, determinata a 32 frequenze tra 0,1 e 5 GHz. Partendo dal centroide (26.9175, 9.334687) e dal raggio corrispondente 6.57293, l'adattamento del semicerchio di Cole-Cole converge in 7 iterazioni alle stime ai minimi quadrati del centro ( $z_1, z_2$ )

$$z_1 = 19.5213 \pm 0.008971141, \quad z_2 = -0.08013724 \pm 0.01349626, \quad (5-9)$$

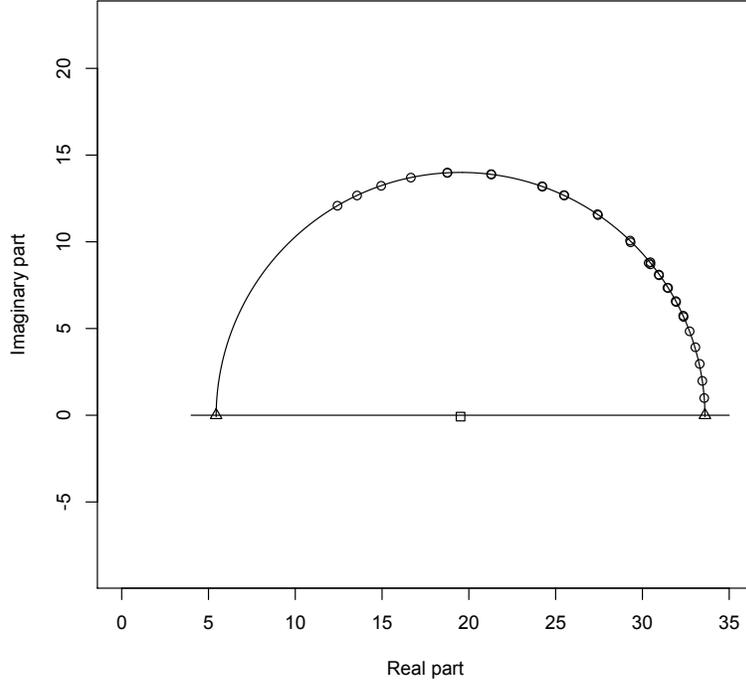


Figura 4: Grafico Cole-Cole dei complessi componenti di permittività per il metanolo a 20°C. Cerchi: misure di Gregory e Clarke [2012]. Triangoli: permittività estrapolate  $\epsilon_\infty$  (sinistra),  $\epsilon_0$  (destra). Quadrato: centro.

e raggio

$$z_3 = 14.08024 \pm 0.01360774. \quad (5-10)$$

La rapidità di convergenza è notevole, se si considera che le equazioni 4-24 forniscono una scarsa approssimazione. In termini di parametri del cerchio,

$$\epsilon_\infty = z_1 - \sqrt{z_3^2 - z_2^2} = 5.441297, \quad (5-11)$$

$$\epsilon_0 = z_1 + \sqrt{z_3^2 - z_2^2} = 33.60131, \quad (5-12)$$

e la stima lineare ai minimi quadrati di  $\tau$  è 0.05574936. I dati e il cerchio adattato sono mostrati nella Figura 4, da cui si evince l'eccellente accordo.

Partendo da queste stime dei parametri, l'adattamento concomitante di  $\epsilon'$  e  $\epsilon''$  in funzione di  $\omega$  converge in 5 iterazioni alle stime ai minimi quadrati

$$z_1 = \epsilon_0 = 33.56849 \pm 0.00869708,$$

$$z_2 = \epsilon_\infty = 5.561308 \pm 0.02203392,$$

e

$$z_3 = \tau = 0.05624048 \pm 6.026653 \times 10^{-5},$$

che confermano che il semicerchio di Cole-Cole fornisce un'eccellente approssimazione. Le parti reali e immaginarie sono mostrate in funzione della frequenza angolare nella Figura 5. I sottoprogrammi di funzione utilizzati per la stima dei parametri e per valutare la soluzione sono i seguenti:

```
debye.relax <- function(x,y,z){
  # This subroutine evaluates the real and imaginary parts of the
  # complex permittivity according to the Debye relaxation theory
  #
  nf <- 2
  f <- vector(length=2)
  omega <- x[1]
  epsreal <- x[2]
  epsimag <- x[3]
  eps0 <- z[1]
  epsinf <- z[2]
  tau <- z[3]
  f[1] <- epsinf+(eps0-epsinf)/(1+(omega*tau)^2)-x[2]
  f[2] <- (eps0-epsinf)*(omega*tau)/(1+(omega*tau)^2)-x[3]
  return(f)
}

debye.eval <- function(omega,eps0,epsinf,tau){
  # This subprogram returns the real and imaginary parts of the complex
  # permittivity as a function of angular frequency omega, according to
  # the Debye equations for a single relaxation time
  #
  f <- vector(length=2)
  eps1 <- epsinf + (eps0-epsinf)/(1+(omega*tau)^2)
  eps11 <- omega*tau*(eps1-epsinf)
  f <- c(eps1,eps11)
  return(f)
}
```

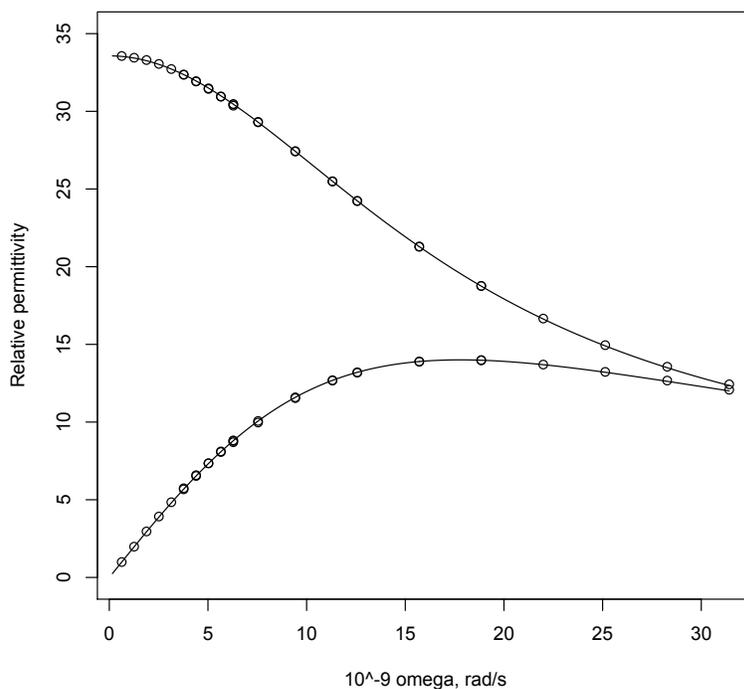


Figura 5: Adattamento parametrico dell'equazioni 5-1. Punti: misure della permittività del metanolo di Gregory e Clarke [2012] a 20° C. Curva superiore: parte reale; curva inferiore: parte immaginaria.

I risultati qui presentati mostrano che il programma GNLS produce eccellenti adattamenti delle misure di rilassamento dielettrico, ma dovrebbe funzionare altrettanto bene nell'analisi della risposta transitoria di qualsiasi altro sistema che obbedisce a un'equazione costitutiva lineare soggetta a forzature periodiche (come la polarizzazione sinusoidale di piccola ampiezza delle reazioni elettrochimiche). Si può anche osservare che l'analisi di tali dati è un problema specializzato e che il GNLS non è l'unico e non necessariamente il migliore approccio. Sebbene a questo scopo siano stati sviluppati metodi ai minimi quadrati complessi non lineari [Macdonald e Garber, 1977; Macdonald et al., 1982; Macdonald, 1992, 1995], l'approccio GNLS ha il vantaggio di essere applicabile a qualsiasi modello a più equazioni.

## 6 Conclusioni

Il programma R descritto nel presente articolo esegue la stima dei parametri ai minimi quadrati come un'ottimizzazione non linearmente vincolata, in cui i vincoli vengono linearizzati su una soluzione di prova, le equazioni risultanti per i vettori di incremento vengono risolte e il processo iterato fino alla convergenza. Questa formulazione più generale consente di incorporare la precisione finita di tutti i dati e di eliminare la distinzione tra variabili dipendenti e indipendenti e la necessità di una relazione funzionale esplicita tra di esse. Tutte le derivate parziali richieste sono approssimate da quozienti della differenza centrale secondo principi ben consolidati. L'applicazione alla riduzione dei dati è illustrata dall'adattamento di dati volumetrici all'equazione viriale, e di dati della permittività complessa al diagramma di Cole-Cole e alle equazioni di Debye.

## Bibliografia

- T.F. Anderson, D.S. Abrams, and E.A. Grens II. Evaluation of parameters for nonlinear thermodynamic models. *AIChE Journal*, 24:20–29, 1978.
- H.I. Britt and R.H. Luecke. The estimation of parameters in nonlinear, implicit models. *Technometrics*, 15:233–247, 1973.
- W.E. Deming. *Statistical Adjustment of Data*. Wiley, New York, 1943.
- A.P. Gregory and R.N. Clarke. Tables of the complex permittivity of dielectric reference liquids at frequencies to 5 Ghz. NPL Technical Report MAT 23, Materials Division, National Physical Laboratory, Teddington, Middlesex, January 2012.
- W.H. Jefferys. On the method of least squares. *Astrophysical Journal*, 85: 177–181, 1980.
- R. Kleinrahm, W. Duschek, W. Wagner, and M. Jaeschke. Measurement and correlation of the (pressure, density, temperature) relation of methane in the temperature range from 273.15 K to 323.15 K at pressures to 8 MPa. *Journal of Chemical Thermodynamics*, 20:621–631, 1988.
- M. Lybanon. A better least-squares method when both variables have uncertainties. *American Journal of Physics*, 52:22–26, 1984.

- M. Lybanon. A simple generalized least-squares algorithm. *Computers and Geosciences*, 11:501–508, 1985.
- J.R. Macdonald. Comparison and application of two methods for the least squares analysis of immittance data. *Solid State Ionics*, 58:97–107, 1992.
- J.R. Macdonald. Exact and approximate nonlinear least-squares inversion of dielectric relaxation spectra. *Journal of Chemical Physics*, 102:6241–6250, 1995.
- J.R. Macdonald and J.A. Garber. Analysis of impedance and admittance data for solids and liquids. *Journal of the Electrochemical Society*, 124: 1022–1030, 1977.
- J.R. Macdonald, J. Schoonman, and A.P. Lehen. The applicability and power of complex nonlinear least squares for the analysis of impedance and admittance data. *Journal of Electroanalytical Chemistry and Interfacial Electrochemistry*, 131:77–95, 1982.
- J.R. Macdonald and W.J. Thompson. Least-squares fitting when both variables contain errors: pitfalls and possibilities. *American Journal of Physics*, 60:66–73, 1992.
- Simon L. Marshall. Generalized least-squares parameter estimation from multi-equation implicit models. *AIChE Journal*, 49(10):2577–2594, 2003.
- Simon L. Marshall. A new maximum-likelihood method for reduction of distribution-coefficient data. *AIChE Journal*, 49(10):2595–2608, 2003b.
- Simon L. Marshall and James G. Blencoe. Generalized least-squares fit of multi-equation models. *American Journal of Physics*, 73(1):69–82, 2005.
- K. Pearson. On lines and planes of closest fit to points in space. *Philosophical Magazine*, 2:559–572, 1901.
- D.R. Powell and J.R. Macdonald. A rapidly convergent iterative method for the solution of the generalized nonlinear least squares problem. *Computer Journal*, 15:148–155, 1972.
- William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in FORTRAN 77: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 2 edition, 1992.

- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL <https://www.R-project.org/>.
- H. Schwetlick and V. Tiller. Numerical methods for estimating parameters in nonlinear models with errors in the variables. *Technometrics*, 27:17–24, 1985.
- D. York. Least-squares fitting of a straight line. *Canadian Journal of Physics*, 44:1079–1086, 1966.

## A Codici Informatici (non destinati alla pubblicazione)

In questa sezione sono forniti i codici fonti dei programmi R usati per generare le figure nell'articolo presente. Poiché i codici dei sottoprogrammi che hanno eseguito i calcoli sono inclusi nel test, i paragraffi seguenti contengono principalmente i scritti che chiamano questi sottoprogrammi.

### A.1 Figura 1

Il programma seguente `GNLSpaperFigure1.R` adatta una retta ai dati di Pearson [1901] con pesi moltiplicativi proposti da York [1966].

```
# Fit of a straight line to Pearson's Data with York's Weights
#
WXY <- read.csv("PDYW.csv")
ndata <- nrow(WXY)
nx <- 2
nz <- 2
nf <- 1
X0 <- vector(length=ndata*nx)
X <- vector(length=ndata*nx)
Sigmax <- matrix(nrow=ndata*nx,ncol=nx)
z <- vector(length=nz)
#
# Errors-in-variables fit
for(i in 1:ndata){
  X0[nx*(i-1)+1] <- WXY[[1]][i]
  X0[nx*(i-1)+2] <- WXY[[2]][i]
  Sigmax[nx*(i-1)+1,] <- c(1/WXY[[3]][i],0)
  Sigmax[nx*(i-1)+2,] <- c(0,1/WXY[[4]][i])
}
y <- 0
z <- c(6,-0.5)
fit <- gnls(fline,nf,X0,Sigmax,ndata,y,z)
z <- fit[[1]]
sigmaz <- fit[[2]]
for(k in 1:nz){
  cat(z[k],"+/-",sigmaz[k],"\n")
}
#
# Plot of data, error bars, and fitted line
#
```

```

X <- fit[[3]]
xx0 <- vector(length=ndata)
yy0 <- vector(length=ndata)
x <- vector(length=ndata)
y <- vector(length=ndata)
sigmax <- vector(length=ndata)
sigmay <- vector(length=ndata)
for(i in 1:ndata){
  xx0[i] <- X0[nx*(i-1)+1]
  yy0[i] <- X0[nx*(i-1)+2]
  x[i] <- X[nx*(i-1)+1]
  y[i] <- X[nx*(i-1)+2]
  sigmax[i] <- sqrt(Sigmax[nx*(i-1)+1,1])
  sigmay[i] <- sqrt(Sigmax[nx*(i-1)+2,2])
}
plot(xx0,yy0,xlim=c(0,9),ylim=c(0,7),xlab="x_1",ylab="x_2",pch=".",asp=1)
for(i in 1:ndata){
  arrows(x0=xx0[i],y0=yy0[i]-sigmay[i],x1=xx0[i],y1=yy0[i]+sigmay[i],
  code=3,angle=90,length=0.05)
  arrows(x0=xx0[i]-sigmax[i],y0=yy0[i],x1=xx0[i]+sigmax[i],y1=yy0[i],
  code=3,angle=90,length=0.05)
}
lines(x,y)
#
# Fixed regressor fit, nonuniform weights
for(i in 1:ndata){
  Sigmax[nx*(i-1)+1,] <- c(0,0)
  Sigmax[nx*(i-1)+2,] <- c(0,1/WXY[[4]][i])
}
z <- c(6,-0.5)
fit <- gnls(fline,nf,X0,Sigmax,ndata,y,z)
z <- fit[[1]]
sigmaz <- fit[[2]]
for(k in 1:nz){
  cat(z[k],"+/-",sigmaz[k],"\n")
}
X <- fit[[3]]
for(i in 1:ndata){
  x[i] <- X[nx*(i-1)+1]
  y[i] <- X[nx*(i-1)+2]
}
lines(x,y,lty=2)
#
# Fixed regressor fit, uniform weights
for(i in 1:ndata){

```

```

    Sigmax[nx*(i-1)+1,] <- c(0,0)
    Sigmax[nx*(i-1)+2,] <- c(0,1)
  }
z <- c(6,-0.5)
fit <- gnls(fline,nf,X0,Sigmax,ndata,y,z)
z <- fit[[1]]
sigmaz <- fit[[2]]
for(k in 1:nz){
  cat(z[k],"+/-",sigmaz[k],"\n")
}
X <- fit[[3]]
for(i in 1:ndata){
  x[i] <- X[nx*(i-1)+1]
  y[i] <- X[nx*(i-1)+2]
}
lines(x,y,lty=3)

```

Il file dal quale i dati sono estratti è `PDYW.csv`, cui contenuti sono come segue:

```

x,y,wx,wy
0.0,5.9,1000.0,1.0
0.9,5.4,1000.0,1.8
1.8,4.4,500.0,4.0
2.6,4.6,800.0,8.0
3.3,3.5,200.0,20.0
4.4,3.7,80.0,20.0
5.2,2.8,60.0,70.0
6.1,2.8,20.0,70.0
6.5,2.4,1.8,100.0
7.4,1.5,1.0,500.0

```

## A.2 Figura 2

L'adattamento GNLS dell'equazione viriale troncata a  $C$  si realizza dal programma

```

# Program to fit volumetric data to the virial equation of state
#
XYZ <- read.csv("methanedensity.csv",header=FALSE)
rhopT <- as.matrix(XYZ)
ndata <- nrow(rhopT)
nx <- ncol(rhopT)
nz <- 3
#

```

```

# Array declarations
y <- vector(length=5)
dx0 <- vector(length=nx)
x0 <- vector(length=nx) # 1=rho/(kg/m3), 2=p/MPa), 3=T/K
X0 <- vector(length=(nx*ndata))
rhor <- vector(length=ndata)
pr <- vector(length=ndata)
Tr <- vector(length=ndata)
Z <- vector(length=ndata)
A <- matrix(nrow=ndata,ncol=3)
b <- vector(length=ndata)
#
# Critical parameters etc
y[1] <- 8.31451 #Gas constant
y[2] <- 16.0428 #Molar mass, g/mol
y[3] <- 162.660 #Critical density, kg m-3
y[4] <- 4.5992 #Critical pressure, MPa
y[5] <- 190.551 #Critical temperature, K
#
# Parameters delineating isothermal data sets
ifirst <- vector(length=10)
temp <- vector(length=9)
ntemp <- 0
tzero <- 0
ifirst[10] <- ndata+1
#
# Formation of data matrices
Sigmax <- NULL
for(i in 1:ndata){
  x0 <- rhopT[i,]
  dx0[1] <- 0.0002*x0[1]
  if(dx0[1] < 0.0002){dx0[1] <- 0.0002}
  dx0[2] <- 0.00007*x0[2]
  if(dx0[2] < 0.00003){dx0[2] <- 0.00003}
  dx0[3] <- 0.003
  sigmax <- diag(dx0) %*% diag(dx0)
  Sigmax <- rbind(Sigmax,sigmax)
  for(j in 1:nx){
    X0[nx*(i-1)+j] <- x0[j]
  }
  # Reduced state variables
  rhor[i] <- x0[1]/y[3]
  pr[i] <- x0[2]/y[4]
  Tr[i] <- x0[3]/y[5]
  Z[i] <- (1e+06*x0[2])*(1e-03*y[2])/(x0[1]*y[1]*x0[3])
}

```

```

if(x0[3] != tzero){
  ntemp <- ntemp+1
  ifirst[ntemp] <- i
  temp[ntemp] <- x0[3]
}
tzero <- x0[3]
# cat(rhor[i],Z[i],Tr[i],"\n")
# Normal equation coefficient matrix and right-hand side vector
A[i,1] <- rhor[i]/Tr[i]^0.25
A[i,2] <- rhor[i]/Tr[i]^1.25
A[i,3] <- rhor[i]*rhor[i]/Tr[i]
b[i] <- Z[i]-1
}
#
# Initial parameter estimates by unweighted linear least squares
ATA <- crossprod(A)
ATb <- crossprod(A,b)
z <- solve(ATA) %%% ATb
cat(z,"\n")
nf <- 1
f <- vector(length=nf)
results <- gnls(Cvirial.eos,nf,X0,Sigmax,ndata,y,z)
z <- results[[1]]
sigmaz <- results[[2]]
for(k in 1:nz){
  cat(z[k]," +/- ",sigmaz[k],"\n")
}
#
# Plot of isothermal data sets
xcalc <- vector(length=100)
ycalc <- vector(length=100)
# for(i in 1:(ntemp-1)){
for(i in 1:9){
  itest <- (i-2)*(i-4)
  if(itest != 0){
    j1 <- ifirst[i]
    j2 <- ifirst[i+1]-1
    cat("T/K = ",temp[i],j1,j2,"\n")
    Tri <- temp[i]/y[5]
    xdata <- vector(length=(j2-j1+1))
    ydata <- vector(length=(j2-j1+1))
    for(j in j1:j2){
      # cat(rhor[j],Z[j],"\n")
      jplot <- j-j1+1
      xdata[jplot] <- rhor[j]
    }
  }
}

```

```

        ydata[jplot] <- Z[j]
    }
    dxcalc <- (rhor[j2]-rhor[j1])/100
    for(k in 1:100){
        xcalc[k] <- rhor[j1]+k*dxcalc
        ycalc[k] <- Cvirial.eval(xcalc[k],Tri,z[1],z[2],z[3])
    }
    if(i == 1){plot(xdata,ydata,xlab="reduced density",ylab="Z",
        xlim=c(0,0.45))}
    if(i > 1){points(xdata,ydata)}
    lines(xcalc,ycalc)
    tlabel <- as.character(temp[i])
    text(xdata[j2-j1+1],ydata[j2-j1+1],tlabel,pos=4)
}
}

```

I dati volumetrici densità ( $\text{kg}\cdot\text{m}^{-3}$ ) - pressione (MPa) - temperatura (K) di citet1988kle/dus sono contenuti nel `methanedensity.csv`:

```

0.7155,0.10106,273.15
0.7174,0.10132,273.15
1.4247,0.20074,273.15
1.4251,0.20076,273.15
2.4955,0.35030,273.15
2.4977,0.35066,273.15
2.5015,0.35121,273.15
4.9973,0.69580,273.15
5.0012,0.69626,273.15
5.0030,0.69667,273.15
9.9979,1.36972,273.15
10.0001,1.37008,273.15
10.0096,1.37143,273.15
15.0083,2.02345,273.15
20.0078,2.65561,273.15
20.0124,2.65616,273.15
25.0150,3.26943,273.15
25.0287,3.27124,273.15
30.0204,3.86480,273.15
34.9853,4.43913,273.15
35.0019,4.44056,273.15
35.0305,4.44400,273.15
39.9826,5.00058,273.15
44.9996,5.54987,273.15
50.0283,6.08669,273.15
55.0227,6.60726,273.15

```

60.0217,7.11648,273.15  
60.0225,7.11677,273.15  
65.0305,7.61675,273.15  
39.9927,5.19778,281.15  
39.9978,5.19851,281.15  
39.9990,5.19857,281.15  
40.0001,5.19871,281.15  
45.0088,5.77588,281.15  
50.0013,6.33838,281.15  
54.9899,6.88858,281.15  
2.4976,0.36376,283.15  
2.5037,0.36468,283.15  
4.9975,0.72249,283.15  
5.0060,0.72356,283.15  
5.0151,0.72485,283.15  
9.9812,1.42166,283.15  
10.0027,1.42447,283.15  
14.9992,2.10524,283.15  
19.9877,2.76611,283.15  
19.9990,2.76768,283.15  
24.9848,3.41013,283.15  
24.9862,3.41035,283.15  
24.9872,3.41042,283.15  
25.0044,3.41249,283.15  
25.0053,3.41264,283.15  
29.9898,4.03808,283.15  
29.9907,4.03826,283.15  
30.0003,4.03916,283.15  
35.0026,4.65116,283.15  
39.9746,5.24472,283.15  
39.9838,5.24577,283.15  
39.9875,5.24596,283.15  
40.0326,5.25148,283.15  
45.0244,5.83384,283.15  
50.0091,6.40263,283.15  
55.0126,6.96214,283.15  
60.0523,7.51560,283.15  
64.9857,8.04806,283.15  
49.9882,6.46354,285.15  
30.0035,4.12797,288.15  
35.0011,4.75561,288.15  
35.0020,4.75599,288.15  
40.0000,5.36976,288.15  
45.0227,5.97377,288.15  
50.0039,6.56060,288.15

50.0067,6.56085,288.15  
54.9803,7.13624,288.15  
60.0071,7.70771,288.15  
2.4967,0.37666,293.15  
4.9969,0.74881,293.15  
5.0110,0.75080,293.15  
5.0196,0.75197,293.15  
10.0003,1.47850,293.15  
10.0019,1.47866,293.15  
10.0115,1.48001,293.15  
10.0118,1.48004,293.15  
14.9786,2.18553,293.15  
14.9822,2.18588,293.15  
14.9832,2.18611,293.15  
14.9838,2.18617,293.15  
14.9936,2.18767,293.15  
14.9942,2.18759,293.15  
15.0015,2.18878,293.15  
19.9414,2.87268,293.15  
19.9853,2.87830,293.15  
19.9875,2.87871,293.15  
19.9911,2.87917,293.15  
19.9951,2.87963,293.15  
19.9954,2.87975,293.15  
20.0042,2.88091,293.15  
20.0049,2.88130,293.15  
20.0150,2.88275,293.15  
20.0155,2.88242,293.15  
20.0202,2.88307,293.15  
20.0308,2.88484,293.15  
20.0778,2.89136,293.15  
20.1206,2.89720,293.15  
24.9894,3.55459,293.15  
25.0015,3.55626,293.15  
25.0070,3.55692,293.15  
25.0090,3.55721,293.15  
25.0100,3.55746,293.15  
25.0457,3.56195,293.15  
25.0808,3.56663,293.15  
30.0086,4.21666,293.15  
35.0104,4.86167,293.15  
40.0014,5.49194,293.15  
40.0163,5.49366,293.15  
44.9913,6.10922,293.15  
45.0030,6.11116,293.15

45.0049,6.11147,293.15  
45.0075,6.11131,293.15  
45.0204,6.11297,293.15  
50.0121,6.71991,293.15  
50.0134,6.72021,293.15  
55.0052,7.31643,293.15  
59.9777,7.90126,293.15  
4.9984,0.77544,303.15  
10.0262,1.53627,303.15  
15.0227,2.27452,303.15  
19.9933,2.99221,303.15  
20.0086,2.99463,303.15  
25.0078,3.70039,303.15  
29.9955,4.39042,303.15  
35.0011,5.06958,303.15  
39.9991,5.73500,303.15  
45.0070,6.39015,303.15  
45.0213,6.39221,303.15  
45.7732,6.48960,303.15  
49.9933,7.03284,303.15  
55.0366,7.67355,303.15  
5.0300,0.80682,313.15  
10.0123,1.58840,313.15  
14.9972,2.35363,313.15  
20.0135,3.10766,313.15  
24.9981,3.84268,313.15  
30.0034,4.56697,313.15  
35.0232,5.28100,313.15  
40.0203,5.98061,313.15  
40.0294,5.98176,313.15  
40.0323,5.98193,313.15  
44.9828,6.66525,313.15  
45.0007,6.66760,313.15  
49.9763,7.34485,313.15  
49.9823,7.34553,313.15  
55.0181,8.02320,313.15  
5.0266,0.83303,323.15  
10.0035,1.64088,323.15  
14.9993,2.43634,323.15  
19.9958,3.21730,323.15  
20.0008,3.21825,323.15  
20.0593,3.22728,323.15  
25.0068,3.98700,323.15  
30.0009,4.74188,323.15  
34.9945,5.48493,323.15

```

40.0101,6.22151,323.15
40.0152,6.22184,323.15
44.9853,6.94284,323.15
45.0056,6.94571,323.15
45.0185,6.94721,323.15
49.9777,7.65886,323.15
50.0059,7.66223,323.15

```

### A.3 Figura 3

L'adattamento di un cerchio a dati rumorosi è realizzato dal programma

```

# This program fits a circle to a set of points X
#
XX <- read.table("circle_data")
# Assembly of input data arrays
#
npoint <- nrow(XX)
nx <- ncol(XX)
epsx <- 0.1
epsy <- 0.2
X0 <- vector(length=(nx*npoint))
Sigmax <- matrix(nrow=(npoint*nx),ncol=nx)
x0 <- vector(length=npoint)
y0 <- vector(length=npoint)
x1 <- vector(length=npoint)
y1 <- vector(length=npoint)
xbar <- 0
ybar <- 0
for(i in 1:npoint){
  x0[i] <- XX[[1]][i]
  y0[i] <- XX[[2]][i]
  xbar <- xbar+x0[i]
  ybar <- ybar+y0[i]
  X0[nx*(i-1)+1] <- x0[i]
  X0[nx*(i-1)+2] <- y0[i]
  Sigmax[nx*(i-1)+1,] <- c(epsx*epsx,0)
  Sigmax[nx*(i-1)+2,] <- c(0,epsy*epsy)
}
xbar <- xbar/npoint
ybar <- ybar/npoint
rbar <- 0
for(i in 1:npoint){
  rbar <- rbar+sqrt((x0[i]-xbar)^2+(y0[i]-ybar)^2)
}

```

```

rbar <- rbar/npoint
cat(xbar,ybar,rbar,"\n")
#
# Initial parameter values etc.
nz <- 3
nf <- 1
f <- vector(length=nf)
z <- vector(length=nz)
z[1] <- xbar
z[2] <- ybar
z[3] <- rbar
results <- gnls(fcircle,nf,X0,Sigmax,npoint,y,z)
#
# Retrieval of results from "results" list...
z <- results[[1]]
sigmaz <- results[[2]]
for(k in 1:nz){
  cat(z[k],"+/-",sigmaz[k],"\n")
}
X1 <- results[[3]]
for(i in 1:npoint){
  x1[i] <- X1[nx*(i-1)+1]
  y1[i] <- X1[nx*(i-1)+2]
  cat(x0[i],x1[i],y0[i],y1[i],"\n")
}
plot(x0,y0,asp=1,xlab="x_1",ylab="x_2")
theta <- seq(0,2*pi,length=200)
lines(x=z[1]+z[3]*cos(theta),y=z[2]+z[3]*sin(theta))

```

che opera sul sottoprogramma fcircle

```

fcircle <- function(x,y,z){
# This subprogram evaluates the implicit form of the equaation for a
# circle
#
  X <- x[1]
  Y <- x[2]
  X0 <- z[1]
  Y0 <- z[2]
  R <- z[3]
  f[1] <- (X-X0)*(X-X0)+(Y-Y0)*(Y-Y0)-R*R
  return(f)
}

```

e dati nel file circle\_data

```

9.820596 1.911918
9.022108 3.938883
8.117322 6.247215
6.729026 7.199807
5.034405 8.763135
3.233283 9.452695
1.051394 10.1603
-1.191579 9.987549
-3.166612 9.936446
-5.117229 8.206854
-6.7478 7.632368
-8.025259 5.881579
-9.213448 3.852297
-9.612323 1.906814
-9.979829 0.2364966
-9.95436 -2.155752
-9.263088 -3.866426
-8.058455 -6.23765
-6.760183 -7.639452
-5.08093 -8.993774
-3.073138 -9.297716
-0.962514 -10.07442
1.204268 -10.04661
3.12235 -9.290957
4.905204 -8.769935
6.615954 -7.574589
8.113818 -6.143212
9.03238 -4.240055
9.733186 -2.12013
10.22281 0.02794955

```

## A.4 Figure 4 e 5

L'adattamento del semicerchio Cole-Cole alle misure di permittività complessa del metanolo [Gregory e Clarke, 2012] si usa per generare le stime iniziali dei parametri per l'adattamento GNLS delle due equazioni di Debye. Il programma seguente GNLSpaperFigure4+5.R

```

# This program fits a circle to a set of points X
#
XX <- read.csv("methanol20.csv")
# Assembly of input data arrays
#
npoint <- nrow(XX)

```

```

nx <- 2
X0 <- vector(length=(nx*npoint))
Sigmax <- matrix(nrow=(npoint*nx),ncol=nx)
x0 <- vector(length=npoint)
y0 <- vector(length=npoint)
xbar <- 0
ybar <- 0
for(i in 1:npoint){
  x0[i] <- XX[[4]][i]
  epsx <- XX[[5]][i]
  y0[i] <- XX[[6]][i]
  epsy <- XX[[7]][i]
  X0[nx*(i-1)+1] <- x0[i]
  X0[nx*(i-1)+2] <- y0[i]
  Sigmax[nx*(i-1)+1,] <- c(epsx*epsx/4,0)
  Sigmax[nx*(i-1)+2,] <- c(0,epsy*epsy/4)
  xbar <- xbar+x0[i]
  ybar <- ybar+y0[i]
}
xbar <- xbar/npoint
ybar <- ybar/npoint
rbar <- 0
for(i in 1:npoint){
  rbar <- rbar+sqrt((x0[i]-xbar)^2+(y0[i]-ybar)^2)
}
rbar <- rbar/npoint
cat(xbar,ybar,rbar,"\n")
#
# Initial parameter values etc.
nz <- 3
nf <- 1
f <- vector(length=nf)
z <- vector(length=nz)
y <- 0
z[1] <- xbar
z[2] <- ybar
z[3] <- rbar
results <- gnls(fcircle,nf,X0,Sigmax,npoint,y,z)
#
# Retrieval of results from "results" list...
z <- results[[1]]
sigmaz <- results[[2]]
for(k in 1:nz){
  cat(z[k],"+/-",sigmaz[k],"\n")
}

```

```

#
# Estimate of the Debye parameters
eps0 <- z[1]+sqrt(z[3]*z[3]-z[2]*z[2])
epsinf <- z[1]-sqrt(z[3]*z[3]-z[2]*z[2])
ata <- 0
aty <- 0
for(i in 1:npoint){
  omega <- 2*pi*XX[[3]][i]
  eps1 <- XX[[4]][i]
  eps11 <- XX[[6]][i]
  aty <- aty+(eps1-epsinf)*(omega*eps11)
  ata <- ata+(omega*(eps1-epsinf))*(omega*(eps1-epsinf))
  # cat(term,"\n")
}
tau <- aty/ata
cat(eps0,epsinf,tau,"\n")
#
# Plot of Cole-Cole semicircle
theta <- seq(0,pi,length=200)
plot(x=z[1]+z[3]*cos(theta),y=z[2]+z[3]*sin(theta),asp=1,type="l",
xlim=c(0,35),ylab="Imaginary part",xlab="Real part")
horizx <- c(4,35)
horizy <- c(0,0)
lines(horizx,horizy)
points(z[1],z[2],pch=0)
points(x0,y0)
points(eps0,0,pch=2)
points(epsinf,0,pch=2)
#
# Fit of real and imaginary permittivities to the Debye equations
nx <- 3
X0 <- vector(length=(nx*npoint))
Sigmax <- matrix(nrow=(npoint*nx),ncol=nx)
for(i in 1:npoint){
  X0[nx*(i-1)+1] <- XX[[3]][i]*2*pi
  X0[nx*(i-1)+2] <- XX[[4]][i]
  epsx <- XX[[5]][i]
  X0[nx*(i-1)+3] <- XX[[6]][i]
  epsy <- XX[[7]][i]
  Sigmax[nx*(i-1)+1,] <- c(0,0,0)
  Sigmax[nx*(i-1)+2,] <- c(0,epsx*epsx/4,0)
  Sigmax[nx*(i-1)+3,] <- c(0,0,epsy*epsy/4)
}
# Initial parameter values etc.
nz <- 3

```

```

nf <- 2
f <- vector(length=nf)
z <- vector(length=nz)
z[1] <- eps0
z[2] <- epsinf
z[3] <- tau
results <- gnls(debye.relax,nf,X0,Sigmax,npoint,y,z)
#
# Retrieval of results from "results" list...
z <- results[[1]]
sigmaz <- results[[2]]
for(k in 1:nz){
  cat(z[k],"+/-",sigmaz[k],"\n")
}
#
# Retrieval of data for plotting
angular.freq <- vector(length=npoint)
epsreal <- vector(length=npoint)
epsimag <- vector(length=npoint)
for(k in 1:npoint){
  angular.freq[k] <- X0[nx*(k-1)+1]
  epsreal[k] <- X0[nx*(k-1)+2]
  epsimag[k] <- X0[nx*(k-1)+3]
}
# Calculation of fitted values for plotting
eps0 <- z[1]
epsinf <- z[2]
tau <- z[3]
fplot <- vector(length=2)
omega.calc <- vector(length=200)
eps1.calc <- vector(length=200)
eps11.calc <- vector(length=200)
for(i in 1:200){
  omega <- i*pi/20
  omega.calc[i] <- omega
  fplot <- debye.eval(omega,eps0,epsinf,tau)
  eps1.calc[i] <- fplot[1]
  eps11.calc[i] <- fplot[2]
}
# Plots
plot(angular.freq,epsreal,ylab="Relative permittivity",
xlab="10^-9 omega, rad/s",ylim=c(0,35))
points(angular.freq,epsimag)
lines(omega.calc,eps1.calc)
lines(omega.calc,eps11.calc)

```

esegue entrambi questi calcoli, utilizzando il sottoprogramma `fcircle` della sezione precedente, oltre ai sottoprogrammi `debye.relax` e `debye.eval` nel testo dell'articolo. Il file dei dati `methanol20.csv` è come segue:

```
TempC,uncertainty,fGHz,epsreal,uncertainty,epsimag,uncertainty
```

```
20,0.1,0.1,33.56,0.10,0.99,0.05
```

```
20,0.1,0.2,33.45,0.10,1.98,0.04
```

```
20,0.1,0.3,33.30,0.11,2.96,0.05
```

```
20,0.1,0.4,33.05,0.11,3.92,0.07
```

```
20,0.1,0.5,32.72,0.12,4.84,0.08
```

```
20,0.1,0.6,32.35,0.14,5.74,0.10
```

```
20,0.1,0.7,31.92,0.15,6.58,0.12
```

```
20,0.1,0.8,31.44,0.16,7.35,0.14
```

```
20,0.1,0.9,30.94,0.18,8.11,0.16
```

```
20,0.1,1.0,30.36,0.20,8.78,0.18
```

```
20,0.1,1.0,30.46,0.21,8.70,0.14
```

```
20,0.1,1.2,29.32,0.29,9.97,0.12
```

```
20,0.1,1.5,27.44,0.31,11.54,0.06
```

```
20,0.1,1.8,25.50,0.26,12.67,0.09
```

```
20,0.1,2.0,24.24,0.24,13.18,0.10
```

```
20,0.1,2.5,21.30,0.21,13.88,0.11
```

```
20,0.1,3.0,18.76,0.18,13.97,0.12
```

```
20,0.1,3.5,16.66,0.16,13.70,0.12
```

```
20,0.1,4.0,14.95,0.15,13.23,0.11
```

```
20,0.1,4.5,13.56,0.14,12.67,0.11
```

```
20,0.1,5.0,12.43,0.13,12.08,0.11
```

```
20,0.1,0.6,32.37,0.14,5.67,0.07
```

```
20,0.1,0.7,31.94,0.11,6.53,0.04
```

```
20,0.1,0.8,31.48,0.11,7.34,0.05
```

```
20,0.1,0.9,30.96,0.10,8.07,0.05
```

```
20,0.1,1.0,30.46,0.11,8.82,0.04
```

```
20,0.1,1.2,29.29,0.10,10.07,0.04
```

```
20,0.1,1.5,27.41,0.10,11.59,0.04
```

```
20,0.1,1.8,25.48,0.09,12.69,0.05
```

```
20,0.1,2.0,24.22,0.09,13.20,0.06
```

```
20,0.1,2.5,21.28,0.11,13.90,0.09
```

```
20,0.1,3.0,18.76,0.15,13.99,0.15
```