Risoluzione di equazioni in MATLAB

Eugenio G. Omodeo

OLTRE L'ALGEBRICO! *Dal* [DAVE HESLOP]:

Figure 3.2: *How can we find the true count rate of a Geiger counter which suffers from dead-time.*

Trieste, 17 ottobre 2022

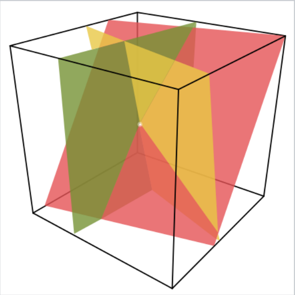
In mathematics, a **system of linear equations** (or **linear system**) is a collection of one or more linear equations involving the same set of variables.[1][2][3][4][5] For example,

$$\begin{aligned} 3x + 2y - z &= 1 \\ 2x - 2y + 4z &= -2 \\ -x + \tfrac{1}{2}y - z &= 0 \end{aligned}$$

is a system of three equations in the three variables x, y, z. A **solution** to a linear system is an assignment of values to the variables such that all the equations are simultaneously satisfied. A solution to the system above is given by

$$\begin{aligned} x &= 1 \\ y &= -2 \\ z &= -2 \end{aligned}$$

since it makes all three equations valid. The word "system" indicates that the equations are to be considered collectively, rather than individually.



A linear system in three variables determines a collection of planes. The intersection point is the solution.

Il sistema visto sopra, ad es., può essere pensato cosí:

$$\begin{bmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & \frac{1}{2} & -1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix};$$

o anche, passando alle trasposte:

$$\begin{bmatrix} x & y & z \end{bmatrix} * \begin{bmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & \frac{1}{2} & -1 \end{bmatrix}' = \begin{bmatrix} 1 & -2 & 0 \end{bmatrix}.$$

Anyway, the best way to think about all matrix division is in terms of solving linear systems. MATLAB interprets

```
>> x = A/B
```

as **"solve the linear system x*B = A (for x)"**. And, similarly,

```
>> x = A\B
```

is **"solve the linear system A*x = B (for x)"**. MATLAB will solve the system if at all possible (ie if the dimensions are consistent), giving, in general, the least-squares solution (ie minimizing the 2-norm of the residual). This means it will "solve" over/under/determined systems, in the most natural way possible -- the actual solution if there is one, or the least-squares solution otherwise.

Command Window

New to MATLAB? See resources for Getting Started.

```
>> X = [ 3   2   -1
         2  -2    4
        -1 0.5   -1 ] \ [ 1 ; -2 ; 0 ]

X =

    1.0000
   -2.0000
   -2.0000

>> X = [ 1  -2  0 ] / [ 3   2   -1
                        2  -2  0.5
                       -1   4   -1 ]

X =

    1.0000   -2.0000   -2.0000

>>
```

Live Editor – EqSecondoGrado.mlx *

+7 | crackSafe.m | digitClose.m | crackWeakSafe.m | classifyingSediments.m | nsceltek.m | daOttimizzare.m | bidimDaOttimizzare.m

Risolvo equazioni di secondo grado, di cui una a coefficienti letterali

```
syms x a b c

solve( 3*x^2 - 4*x + 1 == 0, x )

solve( 3*x^2 - 2*x + 1, x )

eqn = a*x^2 + b*x + c == 0

solve( eqn, x )
```

$$\begin{pmatrix} \frac{1}{3} \\ 1 \end{pmatrix}$$

ans =

$$\begin{pmatrix} \frac{1}{3} - \frac{\sqrt{2}\,\mathrm{i}}{3} \\ \frac{1}{3} + \frac{\sqrt{2}\,\mathrm{i}}{3} \end{pmatrix}$$

eqn = $a\,x^2 + b\,x + c = 0$

ans =

$$\begin{pmatrix} -\frac{b + \sqrt{b^2 - 4\,a\,c}}{2\,a} \\ -\frac{b - \sqrt{b^2 - 4\,a\,c}}{2\,a} \end{pmatrix}$$

Esercizio: RISOLVERE TRAMITE MATLAB LE SEGUENTI EQUAZIONI DI SECONDO E TERZO GRADO NELL'INCOGNITA x:

$$\begin{array}{rcrcrcrcl} & & -2 \cdot x^2 & + & 10 \cdot x & - & 12 & = & 0\,, \\ -2 \cdot x^3 & + & 18 \cdot x^2 & - & 52 \cdot x & + & 48 & = & 0\,, \\ x^3 & - & 0.731 \cdot x^2 & - & 3.64 \cdot x & - & 125.92 & = & 0\,, \\ a \cdot x^3 & + & b \cdot x^2 & + & c \cdot x & + & d & = & 0\,. \end{array}$$

Esercizio: PRODOTTO DI POLINOMI MONOVARIATI

Calcolate la lista dei coefficienti di ciascuno dei polinomi:

$$\begin{array}{r} -2 \cdot (x-2)\,, \\ -2 \cdot (x-2) \cdot (x-3)\,, \\ -2 \cdot (x-2) \cdot (x-3) \cdot (x-4)\,, \\ -2 \cdot (x-2) \cdot (x-3) \cdot (x-4) \cdot (x+1)\,, \end{array}$$

e dell'ultimo polinomio ottenuto trovate con MATLAB le radici.

Oltre l'algebrico! *Dal* [Dave Heslop]:

Often a Geiger counter will be used to measure the radioactivity of rocks. One problem with simple Geiger counters is that at high activity (high count rates) they will underestimate the true activity. This is because after a γ-ray enters the detector the system is "dead" for a short period of time during which it cannot measure. If another γ-ray enters the system during this dead-time it will not be counted and thus the observed count rate will be less than the true count rate. This situation is made worse because not only are the γ-rays which enter during the dead-time not counted, they do extend the dead-time.

Figure 3.2: *How can we find the true count rate of a Geiger counter which suffers from dead-time.*

The relationship between the observed count rate (N_{obs}) and the true count rate (N_{true}) is an exponential law equation:

Figure 3.2: *How can we find the true count rate of a Geiger counter which suffers from dead-time.*

The relationship between the observed count rate (N_{obs}) and the true count rate (N_{true}) is an exponential law equation:

$$N_{obs} = N_{true} e^{-N_{true}\tau}$$

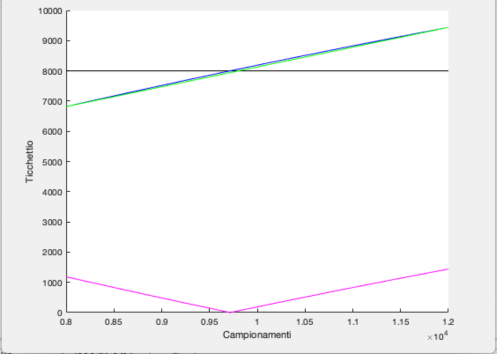
where τ is the dead-time per pulse (20 x 10^{-6} seconds in old instruments). Note this is a transcendental equation which means that it cannot be rewritten in the form $N_{true} = \ldots$. In this exercise you will write an M-function to determine the value of N_{true} for a given input value of N_{obs} and τ.

Cosa otterremo lanciando geigerTest

```
Ntrue =

        9716

>>
```



Editor – /Users/eugenioomodeo/Documents/MATLAB/geigerTest.m

geiger.m | geigerTest.m | digitCompare.m | crackSafe.m | crackWeakSafe.m | classifyingSediments.m | nsceltek.m | daOttimizzare

```
clc,clear,close all,
Nobs=8000;      % numero di click al secondo osservati
tau=20e-6;      % tempo morto per impulso, espresso in secondi
Ntest=[8e3:12e3];  % da 8000 a 12000
Ntrue=geiger(Nobs,tau,Ntest) % la stima del valore reale più calzante
```

. . . . . . A SEGUIRE. . . . . .

Grazie per la vostra attenzione!