backslash (\): left matrix division

Calling sequence: `X = A\B`[1]

Description:

Backslash is the left matrix division: `X = A\B` is a solution to `A*X = B`.

(1) L'equazione $AX = B$ potrebbe avere *più di una* soluzione.

(2) Come evidente dall'esecuzione della prima parte degli esempi, l'assegnamento fornisce *un'approssimazione* di una soluzione dell'equazione.

$\diamondsuit$

If `A` is square and non-singular, `X = A\B` is equivalent to `X = inv(A)*B` in exact arithmetic, but the computations are more accurate and cheaper in floating point arithmetic.

(3) I due assegnamenti, *quando si opera in aritmetica esatta*, sono equivalenti (posto che, in tal caso, `inv(A)` $= A^{-1}$) *ma*, quando si opera in aritmetica floating point, *non lo sono* quasi mai.

(4) Il primo assegnamento fornisce *un'approssimazione della soluzione* del sistema $AX = B$ (quasi sempre) più accurata ed ha *costo aritmetico* (quasi sempre) inferiore rispetto al secondo.

Hence, to compute the solution of the linear system of equations `A*X = B`, the backslash operator should be used, and the `inv` function should be avoided.

(5) Si ricordi che l'assegnamento fornisce *un'approssimazione* della soluzione. Ometteremo di ripetere nel seguito questa puntualizzazione.

$\diamondsuit$

In the case where `A` is square, the solution `X` can be computed either from LU factorization or from a linear least squares solver. If the condition number of `A` is smaller than `1/(10 * %eps)` (i.e. if `A` is well conditioned), the LU factorization with row pivoting is used. If not (i.e. if `A` is poorly conditioned), then `X` is the minimum-norm solution which minimizes $\|$`A*X - B`$\|$ using a complete orthogonal factorization of `A` (i.e. `X` is the solution of a linear least squares problem).[2]

(6) La costante predefinita `%eps` vale $2 \cdot \text{precisione di macchina} \approx 2.22 \cdot 10^{-16}$. Dunque: Se $c($`A`$) < 4.5 \cdot 10^{14}$ allora `backslash` calcola un'approssimazione della soluzione con il procedimento che usa EGPP poi SA poi SI. Se, invece, $c($`A`$) \geqslant 4.5 \cdot 10^{14}$ allora `backslash` calcola un'approssimazione di una *soluzione nel senso dei minimi quadrati* utilizzando un procedimento che calcola una *fattorizzazione QR* di `A`. Se `A` è invertibile esiste *una sola* soluzione nel senso dei minimi quadrati, ed è questa che Scilab approssima. Se, invece, `A` non è invertibile, esistono *infinite* soluzioni nel senso dei minimi quadrati e Scilab ne approssima una ma, contrariamente a quanto affermato nel testo, *non necessariamente quella di norma minima*. Ad esempio:

```
-->A = [1,1;0,0]
 A  =

    1.    1.
    0.    0.
```

[1] In queste pagine si fa riferimento alla versione 5.5.0 di Scilab. Notare la posizione della descrizione del comando nella struttura dell'*help* di Scilab (la riga immediatamente sopra il nome del comando): `Scilab Help >> Scilab > Scilab keywords > backslash`.

[2] Per i dettagli, vedere [3] pag. 142–143, [2] sezioni "Driver Routines, Linear Least Squares (LLS) Problems", "Computational Routines, Orthogonal Factorizations and Linear Least Squares Problems" e [4] pag. 144–148.

```
-->b = [1,0]'
 b  =

    1.
    0.

-->x = A\b
Warning :
matrix is close to singular or badly scaled. rcond =   0.0000D+00
computing least squares solution. (see lsq).

 x  =

    1.
    0.
```

In questo esempio `A` non è invertibile ed il sistema $\mathtt{A}x = \mathtt{b}$ ha infinite soluzioni e quindi infinite soluzioni nel senso dei minimi quadrati. Scilab segnala, correttamente, che `A` non è invertibile e che fornirà un'approssimazione di una soluzione nel senso dei minimi quadrati (Scilab riporta anche, come valore di `rcond`, *l'inverso del numero di condizionamento* – Scilab estende la nozione di numero di condizionamento alle matrici non invertibili: se M è una matrice non invertibile, $c(M) = \infty$ e quindi `rcond` $= 0$). Il vettore `x` calcolato è una soluzione del sistema, ma anche $y = (\frac{1}{2}, \frac{1}{2})^{\mathsf{T}}$ lo è e $\| \mathrm{x} \| = 1 > \frac{1}{\sqrt{2}} = \| y \|$: il valore fornito da `backslash` *non* è un'approssimazione della soluzione nel senso dei minimi quadrati di norma minima.

♢

If `A` is not square, `X` is a least square solution, i.e. norm(`A*X - B`) is minimal (Euclidean norm). If `A` is full column rank, the least square solution, `X = A\B`, is uniquely defined (there is a unique `X` which minimizes norm(`A*X - B`)). If `A` is not full column rank, then the least square solution is not unique, and `X = A\B`, in general, is not the solution with minimum norm (the minimum norm solution is `X = pinv(A)*B`).

(7) Per approfondire la teoria riguardante i sistemi di equazioni lineari con matrice *non quadrata* vedere [1]. Lo stesso riferimento contiene nozioni relative alla *pseudoinversa* di una matrice. Il comando `pinv(A)` fornisce un'approssimazione della matrice pseudoinversa di `A`.

♢

`A.\B` is the matrix with `(i,j)` entry `A(i,j)\B(i,j)`. If `A` (or `B`) is a scalar, `A.\B` is equivalent to `A*ones(B).\B` (or `A.\(B*ones(A))`).

`A\.B` is an operator with no predefined meaning. It may be used to define a new operator (see *overloading*) with the same precedence as `*` or `/`.

Examples:

```
-->A = [  9.   -36.    30.
        -36.   192.  -180.
         30.  -180.   180. ];

-->b = [  3.
        -24.
         30. ];

-->x = A\b
```

```
x  =

    1.
    1.
    1.

-->A*x - b  // close to zero

 ans  =

 10^(-14) *

  - 0.7105427
    0.
    0.
```

(8) Il vettore $(1, 1, 1)^{\mathsf{T}}$ è soluzione (*l'unica*: `A` è invertibile) del sistema $\mathtt{A}x = \mathtt{b}$;

(9) Come si può constatare eseguendo il comando `x == [1;1;1]`, *nessuna* delle componenti di `x` vale 1. Quindi il vettore fornito è *un'approssimazione* della soluzione, come rilevato anche dall'ultima istruzione che mostra il *residuo* associato al vettore `x`.

$\diamondsuit$

```
-->A = rand(3,2);

-->b = [1;1;1];

-->x = A\b;

-->y = pinv(A)*b;

-->x-y

 ans  =

 10^(-14) *

    0.1304512
  - 0.1110223
```

(10) Si considera un sistema di tre equazioni in due incognite con matrice dei coefficienti `A` generata in modo "casuale." Salvo casi improbabili, le due colonne di `A` sono *linearmente indipendenti* (la matrice `A` *ha* "full column rank"), dunque il sistema $\mathtt{A}x = \mathtt{b}$ ha una sola soluzione nel senso dei minimi quadrati, e `x` ed `y` sono due approssimazioni di tale soluzione.

(11) Come si constata dall'essere `x-y` $\neq 0$, le due approssimazioni sono diverse e quindi devono essere state calcolate utilizzando *procedure diverse*.[3] Si osservi che, a causa del funzionamento dell'interfaccia utente/Scilab, le istruzioni `x` (ovvero: "mostra il valore di `x`") e `y` (ovvero: "mostra il valore di `y`") possono generare risultati *identici*.

```
-->A = rand(2,3); b = [1;1];
```

[3] Il valore effettivo della differenza può non coincidere con quello mostrato nel testo perché la matrice `A` ha elementi "casuali."

```
-->x = A\b;

-->y = pinv(A)*b;

-->x-y

 ans  =

    0.8855461
  - 1.5367794
    0.8354957

-->A*x - b

 ans  =

 10^(-15) *

    0.2220446
    0.2220446

-->A*y - b

 ans  =

 10^(-15) *

  - 0.5551115
    0.
```

(12) Si considera un sistema di due equazioni in tre incognite con matrice dei coefficienti `A` generata in modo "casuale." Le tre colonne di `A` sono *linearmente dipendenti* (la matrice `A` *non ha* "full column rank"), dunque il sistema $\mathtt{A}x = \mathtt{b}$ ha infinite soluzioni nel senso dei minimi quadrati. Infatti: salvo casi improbabili, la matrice `A` ha rango due, quindi l'insieme delle soluzioni del sistema $\mathtt{A}x = \mathtt{b}$ *ha infiniti elementi* e coincide con l'insieme delle soluzioni nel senso dei minimi quadrati.

(13) Come si constata dal risultato generato dall'istruzione `x-y`, i vettori `x` ed `y` sono approssimazioni di due soluzioni nel senso dei minimi quadrati *distinte*, come esplicitato nel testo. Inoltre, si può constatare che `norm(x)` $>$ `norm(y)`, risultato anche questo suggerito nel testo.

(14) Come già osservato, l'insieme delle soluzioni del sistema $\mathtt{A}x = \mathtt{b}$ *non è vuoto* e coincide con l'insieme delle soluzioni nel senso dei minimi quadrati. Ne segue che i vettori `x` ed `y` sono approssimazioni di due *soluzioni* (distinte) del sistema. È quindi ragionavole il risultato generato dalle istruzioni `A*x - b` e `A*y - b` che mostra come i vettori siano *poco* diversi da zero (si ricordi che *se* un vettore v è soluzione del sistema $\mathtt{A}x = \mathtt{b}$, *allora* $\mathtt{A}v - \mathtt{b} = 0$).

$\diamondsuit$

```
// if rank is deficient

-->A = rand(3,1)*rand(1,2);

-->b = [1;1;1];

-->x = A\b;
Rango insufficiente. rango = 1

-->y = pinv(A)*b;
```

```
-->A*x - b

 ans  =

    0.2519766
  - 0.0752298
  - 0.5640432

-->A*y - b

 ans  =

    0.2519766
  - 0.0752298
  - 0.5640432
```

(15) Si considera un sistema di tre equazioni in due incognite con matrice dei coefficienti `A` generata moltiplicando una colonna con tre componenti "casuali" ed una riga con due componenti "casuali." Le due colonne di `A` sono *linearmente dipendenti* (la matrice `A` *non ha* "full column rank"), dunque il sistema $\mathtt{A}x = \mathtt{b}$ ha infinite soluzioni nel senso dei minimi quadrati.

(16) Come si può constatare eseguendo l'istruzione `x-y`, i vettori `x` ed `y` sono approssimazioni di due soluzioni nel senso dei minimi quadrati *distinte*, come esplicitato nel testo. Inoltre, si può constatare che `norm(x)` $>$ `norm(y)`, risultato anche questo suggerito nel testo.

(17) Salvo casi improbabili, la matrice `A` ha rango uno (come correttamente rilevato dall'istruzione `A\b`) e l'insieme delle soluzioni del sistema $\mathtt{A}x = \mathtt{b}$ è *vuoto*. Infatti, le istruzioni `A*x - b` e `A*y - b` mostrano come il residuo sia, in entrambi i casi, *decisamente* diverso da zero. Si osservi, però, che se v e w sono soluzioni del sistema $\mathtt{A}x = \mathtt{b}$ nel senso dei minimi quadrati allora: $\mathtt{A}v = \mathtt{A}w =$ *la* proiezione ortogonale di `b` sullo spazio generato dalle colonne di `A` e quindi i vettori $\mathtt{A}v - \mathtt{b}$ e $\mathtt{A}w - \mathtt{b}$ sono *uguali*. È quindi ragionevole che l'interfaccia utente/Scilab mostri i vettori `A*x - b` e `A*y - b` identici. In effetti i due vettori *non* sono uguali, come si può constatare eseguendo l'istruzione `(A*x - b) - (A*y - b)`, perché, si ricordi, `x` ed `y` sono *approssimazioni* di due soluzioni nel senso dei minimi quadrati.

```
-->A = rand(2,1)*rand(1,3);

-->b = [1;1];

-->x = A\b;
Rango insufficiente. rango = 1

-->y = pinv(A)*b;

-->A*x - b

 ans  =

  - 0.1684007
    0.1245334

-->A*y - b

 ans  =

  - 0.1684007
    0.1245334
```

(18) Si considera un sistema di due equazioni in tre incognite con matrice dei coefficienti `A` generata moltiplicando una colonna con due componenti "casuali" ed una riga con tre componenti "casuali." Le tre colonne della matrice sono *linearmente dipendenti* (la matrice `A` *non ha* "full column rank"), dunque il sistema $\mathtt{A}x = \mathtt{b}$ ha infinite soluzioni nel senso dei minimi quadrati.

(19) Come si può constatare eseguendo l'istruzione `x-y`, i vettori `x` ed `y` sono approssimazioni di due soluzioni nel senso dei minimi quadrati *distinte*, come esplicitato nel testo. Inoltre, si può constatare che `norm(x)` $>$ `norm(y)`, risultato anche questo suggerito nel testo.

(20) Salvo casi improbabili, la matrice `A` ha rango uno (come correttamente rilevato dall'istruzione `A\b`) e l'insieme delle soluzioni del sistema $\mathtt{A}x = \mathtt{b}$ *è vuoto*. Infatti le istruzioni `A*x - b` e `A*y - b` mostrano come il residuo sia, in entrambi i casi, *decisamente* diverso da zero. Anche in questo caso l'interfaccia utente/Scilab mostra i vettori `A*x - b` e `A*y - b` identici anche se in effetti *non lo sono*, come si può constatare eseguendo l'istruzione `(A*x - b) - (A*y - b)`.

Riferimenti

[1] Aceto L. e Ciampa M, *Decomposizione ai valori singolari*, 2009.[4]

[2] Anderson E. ecc, *LAPACK Users' Guide*, Third Edition, 1999.[5]

[3] Baudin M, *Programming in Scilab*, 2011.[6]

[4] Lambers J, *MAT 610: Numerical Linear Algebra*, 2014.[7]

[4] http://pagine.dm.unipi.it/~a008363/Didattica_10-11/x-AlgLinFondGeo_Materiale.php
[5] http://www.netlib.org/lapack/lug/lapack_lug.html
[6] http://forge.scilab.org/index.php/p/docprogscilab/downloads/
[7] http://www.math.usm.edu/lambers/mat610/book610.pdf