solve_tol( ) — Tolerance used by solvers and inverters

Description Syntax Remarks and examples Conformability Diagnostics Also see

Description

`solve_tol(`Z`,` *usertol*`)` returns the tolerance used by many Mata solvers to solve $AX = B$ and by many Mata inverters to obtain A^{-1}. *usertol* is the tolerance specified by the user or is missing value if the user did not specify a tolerance.

Syntax

real scalar `solve_tol(`*numeric matrix Z*`,` *real scalar usertol*`)`

Remarks and examples

The tolerance used by many Mata solvers to solve $AX = B$ and by many Mata inverters to obtain A^{-1} is

$$\begin{aligned} eta &= s * \frac{\texttt{trace(abs(}Z\texttt{))}}{n} && \text{when } s > 0 \\ eta &= -s && \text{when } s \leq 0 \end{aligned} \tag{1}$$

where $s =$ 1e–13 or a value specified by the user, n is the `min(rows(`Z`), cols(`Z`))`, and Z is a matrix related to A, usually by some form of decomposition, but could be A itself (for instance, if A were triangular). See, for instance, [M-5] **solvelower( )** and [M-5] **cholsolve( )**.

When *usertol* > 0 and *usertol* $<$ `.` is specified, `solvetol()` returns *eta* calculated with $s =$ *usertol*.

When *usertol* ≤ 0 is specified, `solvetol()` returns $-$*usertol*.

When *usertol* $\geq$ `.` is specified, `solvetol()` returns a default result, calculated as

1. If the `matasolvetol` setting is set to `.` (missing), the value of *eta* is computed using $s =$ 1e–13.
2. If the `matasolvetol` setting is set to positive, the value of *eta* is computed using $s =$ `st_numscalar("c(matasolvetol)")`.
3. If the `matasolvetol` setting is set to 0 or negative, the value of *eta* is $-$`st_numscalar("c(matasolvetol)")`.

Conformability

`solve_tol(`*Z*`,` *usertol*`)`:

Z:	$r \times c$
usertol:	1×1
result:	1×1

Diagnostics

`solve_tol(`*Z*`,` *usertol*`)` skips over missing values in *Z* in calculating (1); *n* is defined as the number of nonmissing elements on the diagonal.

Also see

[M-4] **Utility** — Matrix utility functions