

solve_tol() — Tolerance used by solvers and inverters

Description
Diagnostics

Syntax
Also see

Remarks and examples

Conformability

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

stata.com

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

$$eta = s * \frac{\text{trace}(\text{abs}(Z))}{n} \quad (1)$$

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 $s = \text{usertol}$.

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

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

1. If external real scalar `_solvetolerance` does not exist, as is usually the case, the value of *eta* is returned using $s = 1e-13$.
2. If external real scalar `_solvetolerance` does exist,
 - a. If `_solvetolerance` > 0, the value of *eta* is returned using $s = \text{solvetolerance}$.
 - b. If `_solvetolerance` ≤ 0, $-\text{_solvetolerance}$ is returned.

Conformability

`solve_tol(Z, usertol)`:

<i>Z</i> :	$r \times c$
<i>usertol</i> :	1×1
<i>result</i> :	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