**Syntax**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Remarks and examples</th>
<th>Conformability</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric matrix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_luinv(</td>
<td></td>
<td></td>
</tr>
<tr>
<td>numeric matrix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>void</td>
<td></td>
<td></td>
</tr>
<tr>
<td>void</td>
<td></td>
<td></td>
</tr>
<tr>
<td>real scalar</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Description**

`luinv(A)` and `luinv(A, tol)` return the inverse of real or complex, square matrix `A`.

`_luinv(A)` and `_luinv(A, tol)` do the same thing except that, rather than returning the inverse matrix, they overwrite the original matrix `A` with the inverse.

In all cases, optional argument `tol` specifies the tolerance for determining singularity; see Remarks and examples below.

`_luinv_la(A, b)` is the interface to the [M-1] LAPACK routines that do the work. The output `b` is a real scalar, which is 1 if the LAPACK routine used a blocked algorithm and 0 otherwise.

**Remarks and examples**

These routines calculate the inverse of `A`. The inverse matrix `A^{-1}` of `A` satisfies the conditions:

\[
AA^{-1} = I \\
A^{-1}A = I
\]


`luinv(A)` is logically equivalent to `lusolve(A, I(rows(A)))`; see [M-5] `lusolve()` for details and for use of the optional `tol` argument.
Conformability

\texttt{luinv(\textit{A}, \textit{tol})}: 
\begin{align*}
\textit{A}: & \quad n \times n \\
\textit{tol}: & \quad 1 \times 1 \quad \text{(optional)} \\
\textit{result}: & \quad n \times n 
\end{align*}

\texttt{luinv\_la(\textit{A}, \textit{b})}: 
\begin{align*}
\textit{input}: \\
\textit{A}: & \quad n \times n \\
\textit{tol}: & \quad 1 \times 1 \quad \text{(optional)} \\
\textit{output}: \\
\textit{A}: & \quad n \times n \\
\textit{b}: & \quad 1 \times 1 \\
\textit{result}: & \quad 1 \times 1
\end{align*}

Diagnostics

The inverse returned by these functions is real if \textit{A} is real and is complex if \textit{A} is complex. If you use these functions with a singular matrix, returned will be a matrix of missing values. The determination of singularity is made relative to \textit{tol}. See \textit{Tolerance} under Remarks and examples in [M-5] \texttt{lusolve()} for details.

\texttt{luinv(\textit{A})} and \texttt{luinv\_la(\textit{A})} return a matrix containing missing if \textit{A} contains missing values.

\texttt{luinv(\textit{A})} aborts with error if \textit{A} is a view.

\texttt{luinv\_la(\textit{A}, \textit{b})} should not be used directly; use \texttt{luinv()}.

See [M-5] \texttt{lusolve()} and [M-1] \textbf{tolerance} for information on the optional \textit{tol} argument.

Also see

[M-5] \texttt{invsym()} — Symmetric real matrix inversion

[M-5] \texttt{cholinv()} — Symmetric, positive-definite matrix inversion

[M-5] \texttt{qrinv()} — Generalized inverse of matrix via QR decomposition

[M-5] \texttt{pinv()} — Moore–Penrose pseudoinverse

[M-5] \texttt{lusolve()} — Solve AX=B for X using LU decomposition

[M-5] \texttt{lud()} — LU decomposition

[M-4] \textbf{matrix} — Matrix functions

[M-4] \textbf{solvers} — Functions to solve AX=B and to obtain A inverse