svsolve() — Solve AX=B for X using singular value decomposition

Description

svsolve(A, B, ...), uses singular value decomposition to solve \( AX = B \) and return \( X \). When \( A \) is singular, \svsolve() computes the minimum-norm least-squares generalized solution. When \textit{rank} is specified, in it is placed the rank of \( A \).

\_svsolve(A, B, ...) does the same thing, except that it destroys the contents of \( A \) and it overwrites \( B \) with the solution. Returned is the rank of \( A \).

In both cases, \textit{tol} specifies the tolerance for determining whether \( A \) is of full rank. \textit{tol} is interpreted in the standard way—as a multiplier for the default if \textit{tol} > 0 is specified and as an absolute quantity to use in place of the default if \textit{tol} \leq 0 is specified.

Syntax

\[
\text{numeric matrix } \hspace{0.5cm} \svsolve(A, B) \\
\text{numeric matrix } \hspace{0.5cm} \svsolve(A, B, \text{rank}) \\
\text{numeric matrix } \hspace{0.5cm} \svsolve(A, B, \text{rank}, \text{tol}) \\
\text{real scalar } \hspace{0.5cm} \_\svsolve(A, B) \\
\text{real scalar } \hspace{0.5cm} \_\svsolve(A, B, \text{tol})
\]

where

\( A \): \hspace{1cm} \text{numeric matrix} \\
\( B \): \hspace{1cm} \text{numeric matrix} \\
\text{rank}: \hspace{1cm} \text{irrelevant; real scalar returned} \\
\text{tol}: \hspace{1cm} \text{real scalar}

Remarks and examples

\svsolve(A, B, ...) is suitable for use with square or nonsquare, full-rank or rank-deficient matrix \( A \). When \( A \) is of full rank, \svsolve() returns the same solution as \lusolve() (see \textit{[M-5] lusolve()}), ignoring roundoff error. When \( A \) is singular, \svsolve() returns the minimum-norm least-squares generalized solution. \qrsvolve() (see \textit{[M-5] qrsvolve()}), an alternative, returns a generalized least-squares solution that amounts to dropping rows of \( A \).

Remarks are presented under the following headings:

- Derivation
- Relationship to inversion
- Tolerance
Derivation

We wish to solve for $X$

$$AX = B \quad (1)$$

Perform singular value decomposition on $A$ so that we have $A = USV'$. Then (1) can be rewritten as

$$USV'X = B$$

Premultiplying by $U'$ and remembering that $U'U = I$, we have

$$SV'X = U'B$$

Matrix $S$ is diagonal and thus its inverse is easily calculated, and we have

$$V'X = S^{-1}U'B$$

When we premultiply by $V$, remembering that $VV' = I$, the solution is

$$X = VS^{-1}U'B \quad (2)$$

See [M-5] svd() for more information on the SVD.

Relationship to inversion

For a general discussion, see Relationship to inversion in [M-5] lusolve().

For an inverse based on the SVD, see [M-5] pinv(). pinv($A$) amounts to svsolve($A$, I(rows($A$))), although pinv() has separate code that uses less memory.

Tolerance

In (2) above, we are required to calculate the inverse of diagonal matrix $S$. The generalized solution is obtained by substituting zero for the $i$th diagonal element of $S^{-1}$, where the $i$th diagonal element of $S$ is less than or equal to $eta$ in absolute value. The default value of $eta$ is

$$eta = \text{epsilon}(1) \times \text{rows}(A) \times \text{max}(S)$$

If you specify $tol > 0$, the value you specify is used to multiply $eta$. You may instead specify $tol \leq 0$ and then the negative of the value you specify is used in place of $eta$; see [M-1] Tolerance.
Conformability

\texttt{svsolve}(A, B, rank, tol):
\begin{align*}
\text{input:} & \\
A & : \quad m \times n \\
B & : \quad m \times k \\
tol & : \quad 1 \times 1 \quad \text{(optional)}
\end{align*}
\begin{align*}
\text{output:} & \\
rank & : \quad 1 \times 1 \quad \text{(optional)} \\
result & : \quad n \times k
\end{align*}
\texttt{svsolve}(A, B, tol):
\begin{align*}
\text{input:} & \\
A & : \quad m \times n \\
B & : \quad m \times k \\
tol & : \quad 1 \times 1 \quad \text{(optional)}
\end{align*}
\begin{align*}
\text{output:} & \\
A & : \quad 0 \times 0 \\
B & : \quad m \times k \\
result & : \quad 1 \times 1
\end{align*}

Diagnostics

\texttt{svsolve}(A, B, \ldots) \text{ and } \texttt{_svsolve}(A, B, \ldots) \text{ return missing results if } A \text{ or } B \text{ contain missing.}

\texttt{_svsolve}(A, B, \ldots) \text{ aborts with error if } A \text{ (but not } B) \text{ is a view.}

Also see

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

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

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

[M-5] \texttt{solvelower()} — Solve AX=B for X, A triangular

[M-4] \texttt{Matrix} — Matrix functions

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