A(1,1)*x(1)+A(1,2)*x(2) = b(1)
A(2,1)*x(1)+A(2,2)*x(2) = b(2)
where A(i,j) a square matrices, x(i), b(i) are vectors

How to get Schur complement:
1) A(1,1)=L*U
2) A(1,2)=(L*U)^-1*A(1,2)
3) b(1)=(L*U)^-1*b(1)
4) A(2,2)=A(2,2)-A(2,1)*A(1,2)
5) b(2)=b(2)-A(2,1)*b(1)
then, the Schur complement is stored at A(2,2) and b(2)

1)
A(1,1)=L*U

LAPACK DGETRF for double precision
ZGETRF for complex
SUBROUTINE ZGETRF(M, N, A, LDA, IPIV, INFO)
INTEGER
INFO, LDA, M, N
INTEGER
IPIV(*)
DOUBLE
PRECISION A(LDA, *)
M (input) INTEGER The number of rows of the matrix A. M >= 0.
N (input) INTEGER The number of columns of the matrix A. N >= 0.
A (input/output) DOUBLE PRECISION array, dimension (LDA,N) On entry, the M-by-N matrix to be factored.
On exit, the factors L and U from the factorization A = P*L*U; the unit diagonal elements of L are not stored.
LDA (input) INTEGER The leading dimension of the array A. LDA >= max(1,M).
IPIV (output) INTEGER array, dimension (min(M,N)) The pivot indices; for 1 <= i <= min(M,N), row i of the matrix was interchanged with row **IPIV**(i).
INFO (output) INTEGER = 0: successful exit
< 0: if INFO = -i, the i-th argument had an illegal value
> 0: if INFO = i, U(i,i) is exactly zero. The factorization has been completed, but the factor U is exactly singular, and division by zero will occur if it is used to solve a system of equations.
2)
A(1,2)=(L*U)^-1*A(1,2)
in other words (where A=A(1,2))
R = U^-1*L^-1*A
L^-1*A = B
U^-1*B=R
Algorithm:
a) Given A and L, solve A=L*B
b) Given B and U, solve B=L*R for R
DTRSM - ZDTRM(SIDE, UPLO, TRANS, DIAG, M, N, ALPHA, A, LDA, B, LDB)
B<=L^-1*A
A is of size mxn
L is of size mxm
SIDE = 'L'
UPLO = 'L' lower triangular matrix
TRANS = 'N' no transpoze of L
DIAG = indicates if the diagonal of *L* is to be taken to equal the identity matrix (DIAG = "Unit") or the values in the matrix (DIAG = "Non unit").

M=m, N=n
ALPHA=1
A <- L
B <- A
The leading dimensions of the matrices are given in LDA(for L) and LDB (for A).
R<=U^-1*B
B is of size nxn
U is of size nxn
SIDE = 'L'
UPLO = 'U' upper triangular matrix
TRANS = 'N' no transpoze of U
DIAG = indicates if the diagonal of *U* is to be taken to equal the identity matrix (DIAG = "Unit") or the values in the matrix (DIAG = "Non unit").
M=n, N=n
ALPHA=1
A <- U
B <- B
The leading dimensions of the matrices are given in LDA(for U) and LDB (for B).

3)
b(1)=(L*U)^-1*b(1)

in other words (where b=b(1))
d = U^-1*L^-1*b
L^-1*b = e
b=L*e
d=U^-1*e
e=U*d
Algorithm:
a) Given b and L, solve b=L*e for e
a) Given e and U, solve e=U*d for d
DTRSM - ZDTRM(SIDE, UPLO, TRANS, DIAG, M, N, ALPHA, A, LDA, B, LDB)
L^-1*b = e
b is of size mx1
L is of size mxm
SIDE = 'L'
UPLO = 'L' lower triangular matrix
TRANS = 'N' no transpoze of L
DIAG = indicates if the diagonal of *L* is to be taken to equal the identity matrix (DIAG = "Unit") or the values in the matrix (DIAG = "Non unit").
M=m, N=1
ALPHA=1
A <- L
B <- b
The leading dimensions of the matrices are given in LDA(for L) and LDB (for b).
d<-U^-1*e
e is of size mx1
U is of size mxm
SIDE = 'L'
UPLO = 'U' upper triangular matrix
TRANS = 'N' no transpoze of U
DIAG = indicates if the diagonal of *U* is to be taken to equal the identity matrix (DIAG = "Unit") or the values in the matrix (DIAG = "Non unit").
M=m, N=1
ALPHA=1
A <- U
B <- e

The leading dimensions of the matrices are given in LDA(for U) and LDB (for e).

4) A(2,2)=A(2,2)-A(2,1)*A(1,2)

DGEMM/ZGEMM (double / complex)
ZGEMM (transa, transb, l, n, m, alpha, a, lda, b, ldb, beta, c, ldc)
C = alpha A *B + beta* C
here alpha = -1, beta = 1, A(2,1)=mult, B=A(1,2), C=A(2,2)
transa = 'N' ***A*** is used in the computation.
transb = 'N', ***B*** is used in the computation.
l is the number of rows in matrix ***C***.
n is the number of columns in matrix ***C***.
m is the number of columns in matrix ***A***.
alpha is the scalar alpha.
a is the matrix ***A***, where: ***A*** has l rows and m columns.
If transa equal to 'N', its size must be lda by (at least) m.
lda is the leading dimension of the array specified for a.
b is the matrix ***B***, where: ***B*** has m rows and n columns.
ldb is the leading dimension of the array specified for b.
beta is the scalar beta.
c is the l by n matrix ***C***.
ldc is the leading dimension of the array specified for c.
On Return c is the l by n matrix ***C***, containing the results of the computation.

5) b(2)=b(2)-A(2,1)*b(1)

ZGEMV
(TRANS, M, N, ALPHA, A, LDA, X, INCX, BETA, Y, INCY)
TRANS = 'N' for y := alpha*A*x + beta*y,
M = number of rows in A
N = number of columns in A
ALPHA=-1
A <= A(2,1)
LDA On entry, LDA specifies the first dimension of A as declared in the calling (sub) progra
X<=b(1)
INCX=1
BETA=1
Y<=b(1)
INCY=1