





---



## **Part 2—Using CXML Subprograms**

### **4 Using the Level 1 BLAS Subprograms and Extensions**

4.1



9.2.2













## **VLIB Routines**

VCOS



**Tables**

1      **General Conventions Used in this Documentation . . . . .**





---

## Preface





- A manpage for each subprogram that provides details of its use and the operations it implements

Refer to the section of this Preface titled "Using Manpages" if you need















---



## 4 How CXML Achieves High Performance

CXML relies on the following design techniques to achieve high performance:









---

**Preparing and Storing Program Data**

# Preparing and Storing Program Data

## 1.1 Data and Data Types





## Preparing and Storing Program Data

### 1.3 Storing Data





## Preparing and Storing Program Data

### 1.3 Storing Data



## Preparing and Storing Program Data

**Preparing and Storing Program Data**  
**1.3 Storing Data**

For



## Preparing and Storing Program Data







## Preparing and Storing Program Data

### 1.3 Storing Data

For example, element



## Preparing and Storing Program Data































































## Using the Level 1 BLAS Subprograms and Extensions







## Using the Level 1 BLAS Subprograms and Extensions









## Using the Sparse Level 1 BLAS Subprograms















































---

## Using the Level 3 BLAS Subprograms

The Level 3 BLAS subprograms perform matrix-matrix operations commonly













## Using the Level 3 BLAS Subprograms



## Using the Level 3 BLAS Subprograms









## Using LAPACK Subprograms





## Using LAPACK Subprograms

### 8.2 Naming Conventions



















## Using LAPACK Subprograms

















## Using the Signal Processing Subprograms

### 9.1 Fourier Transform

6.







## Using the Signal Processing Subprograms

## Using the Signal Processing Subprograms

### 9.1 Fourier Transform





## Using the Signal Processing Subprograms

### 9.1 Fourier Transform

- The `_APPLY` subroutine uses the internal data structures to compute the FFT.
-











## Using the Signal Processing Subprograms





## Using the Signal Processing Subprograms











**Using the Signal Processing Subprograms**  
**9.3 Convolution and Correlation**

**Table 9–15 (Cont.) Summary of Correlation Subroutines**

---







## Using the Signal Processing Subprograms

### 9.4 Digital Filtering

Figure 9–3 Lowpass Nonrecursive Filter for Varying Wiggles

## Using the Signal Processing Subprograms

### 9.4 Digital Filtering

**Table 9–17 Naming Conventions: Digital Filter Subroutines**

<b>Character Group</b>	<b>Mnemonic</b>	<b>Meaning</b>
First group	S	Single-precision real data

## Using the Signal Processing Subprograms

### 9.5 Error Handling









---

## Using the Iterative Solvers for Sparse Linear Systems



















# Using the Iterative Solvers for Sparse Linear Systems

## 10.2 Interface to the Iterative Solver









**Using the Iterative Solvers for Sparse Linear Systems**  
**10.2 Interface to the Iterative Solver**

**Table 10–7 (Cont.) Parameters for the SOLVER Subroutine**

---

**Argument**

































## Using the Iterative Solvers for Sparse Linear Systems

### 10.9 A Look at Some Iterative Solvers







## Using the Iterative Solvers for Sparse Linear Systems











































## Using the Iterative Solvers for Sparse Linear Systems





# Using the Direct Solvers for Sparse Linear Systems

## 11.2 Methods for Solutions



## Using the Direct Solvers for Sparse Linear Systems





## Using the Direct Solvers for Sparse Linear Systems

## Using the Direct Solvers for Sparse Linear Systems

















**Using the Direct Solvers for Sparse Linear Systems**  
**11.8 Summary of Skyline Solver Subprograms**





## Using the Direct Solvers for Sparse Linear Systems



## Using the Direct Solvers for Sparse Linear Systems



## Using the Direct Solvers for Sparse Linear Systems

## **Using the Direct Solvers for Sparse Linear Systems**

### **11.11 A Look at Some Skyline Solvers**











## Using the Direct Solvers for Sparse Linear Systems

## Using the Direct Solvers for Sparse Linear Systems







## Using the Direct Solvers for Sparse Linear Systems

## Using the Direct Solvers for Sparse Linear Systems







## Using the Direct Solvers for Sparse Linear Systems

### 11.11 A Look at Some Skyline Solvers



















## Using the Direct Solvers for Sparse Linear Systems













**Using the VLIB Routines**  
**12.2 Vector Storage**



## 12.7 Error Handling















## Using Sort Subprograms

---

## **Part 3—CXML Subprogram Reference**























**BLAS Level 1 Reference**

**SDOT DDOT DSDOT CDOTC ZDOTC CDOTU ZDOTU**

**Description**

















**BLAS Level 1 Reference**  
**SROTM DROTM**







**BLAS Level 1 Reference  
SROTMG DROTMG**





**BLAS Level 1 Reference**  
**SSWAP DSWAP CSWAP ZSWAP**

y





**BLAS Level 1 Extensions Reference  
ISAMIN IDAMIN ICAMIN IZAMIN**

---





**Example**



**BLAS Level 1 Extensions Reference  
SAMIN DAMIN SCAMIN DZAMIN**

**If *incx***





**BLAS Level 1 Extensions Reference**  
**SMIN DMIN**

---

**SMIN DMIN**



## BLAS Level 1 Extensions Reference







**Example**

INTEGER\*4

**BLAS Level 1 Extensions Reference**  
**SVCAL DVCAL CVCAL ZVCAL CSVCAL, ZDVCAL**

**Description**

**BLAS Level 1 Extensions Reference**  
**SVCAL DVCAL CVCAL ZVCAL CSVCAL, ZDVCAL**

**incx**

integer\*4

On entry, the increment for the array  $X$ .

If **incx** > 0, vector  $X$  is stored forward in the array, so that



**BLAS Level 1 Extensions Reference**  
**SZAXPY DZAXPY CZAXPY ZZAXPY**

If **incy** = 0, vector



































---















## BLAS Level 2 Reference

2.



























**Arguments**













**BLAS Level 2 Reference**  
**STPMV DTPMV CTPMV ZTPMV**

**diag**  
character\*1  
On entry, specifies whether the matrix

## BLAS Level 2 Reference

**BLAS Level 2 Reference**  
**STRMV DTRMV CTRMV ZTRMV**

**x**  
real\*4 | real\*8 | complex\*8 | complex\*16

**BLAS Level 2 Reference**  
**STRMV DTRMV CTRMV ZTRMV**

**trans**

character\*1

On entry, specifies the operation to be performed:

If **trans** = 'N' or 'n



















**SGEMM DGEMM CGEMM ZGEMM**  
**Matrix-Matrix Product and Addition (Serial and Parallel Versions)**

**Format**

{S,D,C,Z}GEMM ( transa, transb, m, n, k, alpha, a, lda, b, ldb, beta, c, ldc )

**Arguments**

**transa**  
character\*1



**Example**





**Description**





## BLAS Level 3 Reference

**BLAS Level 3 Reference  
SSYRK DSYRK CSYRK ZSYRK**

**Example**

```
REAL*4 A(20,20), B(30,40), C(30,50), ALPHA, BETA  
M = 10
```



**BLAS Level 3 Reference  
CHERK, ZHERK**

On exit, **c** is overwritten; the triangular part of the array **C** is overwritten by the

























**BLAS Level 3 Reference**  
**STRSM DTRSM CTRSM ZTRSM**









---

\_FFT







## Signal Processing Reference

### \_FFT\_APPLY

**stride\_1\_flag**

logical

Specifies the allowed distance between consecutive elements in the input and

















## Signal Processing Reference

















**Signal Processing Reference**  
**\_FFT\_APPLY\_3D**

`fft_struct`



















---

































**Signal Processing Reference**  
**\_CONV\_PERIODIC**



## Signal Processing Reference

### CORR\_PERIODIC

#### Example

```
INCLUDE 'DXMLDEF.FOR'  
INTEGER*4 N_F, N_M, STATUS
```

**Example**















































**Sparse Iterative Solver Reference**  
**DITSOL\_DRIVER**

**Description**











## Sparse Iterative Solver Reference

### DITSOL\_PBCG

- Biconjugate gradient method with right preconditioning:  
This is the bi-conjugate gradient method applied to:

$$* \quad 1 \quad * \quad b$$

where:





## **Sparse Iterative Solver Reference**

### **DITSOL\_PCGS**

For the conjugate gradient squared method, with left preconditioning, the length of the real work space array, defined by the variable











**Sparse Iterative Solver Reference**  
**DMATVEC\_SDIA**

---

**DMATVEC\_SDIATs 1 i ,1.8'á·È 1aKKKQ0 0 1efefeHHH= (It)7'9·iàB**











**nz**  
integer\*4

## Sparse Iterative Solver Reference





























## Sparse Iterative Solver Reference













## Sparse Iterative Solver Reference























**Sparse Direct Solver Reference**  
**DSSKYF**





## Arguments

n  
integer\*4Së-;X .1ÅTj3 -











**Sparse Direct Solver Reference**  
**DSSKYR**

**iwrk**











au

# Sparse Direct Solver Reference

## DSSKYD

iparam(4): nrwrk



## Sparse Direct Solver Reference

**auf**  
real\*8  
On entry, if RPARAM(9) =















































**Sparse Direct Solver Reference**  
**DUSKYC**



## Sparse Direct Solver Reference



# Sparse Direct Solver Reference

## DUSKYR

nbx  
integer\*4

















**Sparse Direct Solver Reference**  
**DUSKYX**







**iparam(5): iounit**

On entry, defines the I/O unit number for printing error messages and information from the routine DUSKYX. The I/O unit must be opened in the calling subprogram. If *iounit*























## **VLIB Reference**

### **VLOG**

#### **Description**

The VEXP function computes the exponential of  $n$  elements of a vector as follows:



## VLIB Reference



## VLIB Reference













**RNG Reference**  
**RANL\_SKIP64**

end

## Example

```
integer nprocs,n  
integer*8 hop
```





**Example**











Sort Reference  
GEN\_SORT

## Sort Reference





**Sort Reference**  
**GEN\_SORTX**

**If type**





---

## Part 4—Appendices

This section contains appendices that provide supplemental information about













## Compaq Tru64 UNIX Considerations



**B**

---









**D**

---

**Bibliography**

## Bibliography

**D.6 Iterative Solvers**







BLAS Level 2 (cont'd)  
DTPMV, Reference-84  
DTPSV, Reference-86









DSSKYC, Reference-246



















Subprograms (cont'd)

DSSKYS, Reference-242

DSSKYX, Reference-256

DSUM, Reference-39

DSUMI, Reference-56



Subprograms (cont'd)

ZVCAL, Reference-40

ZZAXPY, Reference-42

Subroutine, 4-8, 5-6, 12-4



