# Index

## Numbers

2-norm: *Vol. I* 73

## A

all-to-all broadcast: *Vol. II* 620

all-to-all reduction: *Vol. II* 625

all-to-all rotation: *Vol. II* 609

**all_to_all**: *Vol. II* 611, 613

**all_to_all_broadcast**: *Vol. II* 621

**all_to_all_reduce**: *Vol. II* 626

**all_to_all_setup**: *Vol. II* 611, 613

arbitrary block sparse matrix operations: *Vol. I* 133

arbitrary elementwise sparse matrix operations: *Vol. I* 114

Arnoldi algorithm, implementation: *Vol. I* 413

Arnoldi routines, selected eigenvalues and eigenvectors: *Vol. I* 413

array conversion utilities: *Vol. II* 521

axes, column and row: *Vol. I* 28

## B

banded linear solvers
  algorithms used: *Vol. I* 296
  choosing an algorithm: *Vol. I* 302
  numeric stability: *Vol. I* 303
  setting up data for: *Vol. I* 304
  with pivoting: *Vol. I* 295
  without pivoting: *Vol. I* 322

Basic Linear Algebra Subroutines: *Vol. I* 63

basis transformation: *Vol. I* 359

bi-conjugate gradient algorithm: *Vol. I* 336

BICSTAB algorithm: *Vol. I* 337

**bidiag_svd_singular_values**: *Vol. I* 448

**bidiag_svd_singular_vectors**: *Vol. I* 452

**bidiag_to_gen_left**: *Vol. I* 436

**bidiag_to_gen_right**: *Vol. I* 436

bidiagonal matrices
  singular values of: *Vol. I* 447
  singular vectors of: *Vol. I* 451

BLAS: *Vol. I* 63

block cyclic ordering: *Vol. II* 731

block cyclic permutations, computation of: *Vol. II* 729

block gather utility: *Vol. II* 664

block matrix representation: *Vol. I* 133

block pentadiagonal systems: *Vol. I* 309

block scatter utility: *Vol. II* 664

block sparse matrix operations: *Vol. I* 133

block tridiagonal systems: *Vol. I* 309

**block_gather**: *Vol. II* 664, 667

**block_pentadiag_factor** and related routines: *Vol. I* 323

**block_scatter**: *Vol. II* 664, 667

**block_sparse_mat_gen_mat_mult**: *Vol. I* 133, 147

**block_sparse_matrix_vector_mult**: *Vol. I* 133, 147

**block_sparse_setup**: *Vol. I* 133, 147


*Copyright © 1994 Thinking Machines Corporation*

**block_tridiag_factor** and related routines: *Vol. I* 323
blocking: *Vol. II* 730
blocking factor: *Vol. II* 732
broadcast, all-to-all: *Vol. II* 620
butterfly computations: *Vol. II* 479

## C

CCFFT. *See* FFT
CGS algorithm: *Vol. I* 337
CM Fortran/CMSSL interface: *Vol. I* 53
**CMF$LAYOUT**: *Vol. I* 29
**CMF_ALLOCATE_DETAILED_ARRAY**: *Vol. I* 29
**CMF_NUMBER_OF_PROCESSORS**: *Vol. I* 29
CMOST versions: *Vol. I* 56
CMSSL library, contents: *Vol. I* 2
**cmssl-cmf.h**: *Vol. I* 54
combination of permutations: *Vol. II* 741
**combine_fe_perms**: *Vol. II* 742
**comm_get**: *Vol. II* 708
**comm_send** and related routines: *Vol. II* 708
**comm_set_option**: *Vol. II* 708
**comm_setup** and related routines: *Vol. II* 711
communication compiler: *Vol. II* 707
  how to use: *Vol. II* 709
communication primitives: *Vol. II* 597
compiling: *Vol. I* 54
complex-to-complex FFT: *Vol. II* 479
  *See also* FFT
complex-to-real FFT: *Vol. II* 499
  *See also* FFT
**complex_from_real**: *Vol. II* 501, 509, 522
**compute_fe_block_cyclic_perms**: *Vol. II* 729, 734
conjugate gradient algorithm: *Vol. I* 336
conjugate symmetric axis: *Vol. II* 499
consecutive order: *Vol. II* 733
CRFFT. *See* FFT
cyclic reduction: *Vol. I* 300

## D

data types supported: *Vol. I* 30
**deallocate_all_to_all_setup**: *Vol. II* 612, 613
**deallocate_banded**: *Vol. I* 295, 313
**deallocate_banded_solve**: *Vol. I* 323
**deallocate_block_sparse_setup**: *Vol. I* 133, 147
**deallocate_comm_setup**: *Vol. II* 708, 711
**deallocate_fast_rng**: *Vol. II* 557, 562, 569
**deallocate_fft_setup**: *Vol. II* 478, 490, 511
**deallocate_gather_setup**: *Vol. II* 636, 640
**deallocate_gen_bidiag**: *Vol. I* 436
**deallocate_grid_sparse_setup**: *Vol. I* 159, 171
**deallocate_iter_solve**: *Vol. I* 340
**deallocate_ode_rkf_setup**: *Vol. II* 532
**deallocate_part_gather_setup**: *Vol. II* 694
**deallocate_part_scatter_setup**: *Vol. II* 701
**deallocate_pshift_setup**: *Vol. II* 600, 602
**deallocate_scatter_setup**: *Vol. II* 644, 647
**deallocate_sparse_matvec_setup**: *Vol. I* 114, 121
**deallocate_sparse_vecmat_setup**: *Vol. I* 114, 121
**deallocate_sym_lanczos_setup**: *Vol. I* 403, 420
**deallocate_sym_tred**: *Vol. I* 359, 360
**deallocate_vec_gather_setup**: *Vol. II* 654
**deallocate_vec_scatter_setup**: *Vol. II* 660
**deallocate_vp_rng**: *Vol. II* 557, 562, 578



Copyright © 1994 Thinking Machines Corporation

dense simplex: *Vol. II* 541

Detailed FFT: *Vol. II* 477

differential equations, ordinary: *Vol. II* 529

dual connectivity array: *Vol. II* 673

## E

eigensystem analysis: *Vol. I* 353
  accuracy: *Vol. I* 377
  introduction to routines: *Vol. I* 355
  of dense Hermitian matrices: *Vol. I* 377
  using Jacobi rotations: *Vol. I* 388

eigensystem analysis, generalized: *Vol. I* 382
  accuracy: *Vol. I* 383

eigenvalues
  accuracy of routine: *Vol. I* 367
  of real symmetric tridiagonal matrices: *Vol. I* 366
  selected, using *k*-step Arnoldi method: *Vol. I* 413
  selected, using *k*-step Lanczos method: *Vol. I* 394

eigenvectors
  accuracy of routine: *Vol. I* 371
  applicability of routine: *Vol. I* 372
  of real symmetric tridiagonal matrices: *Vol. I* 371
  performance of routine: *Vol. I* 373
  selected, using *k*-step Arnoldi method: *Vol. I* 413
  selected, using *k*-step Lanczos method: *Vol. I* 394

element nodes array: *Vol. II* 672

elementwise consecutive order: *Vol. II* 733

elementwise sparse matrix operations: *Vol. I* 114

example code: *Vol. I* 60

executing CMSSL programs: *Vol. I* 56

extract and deposit vector: *Vol. II* 724

## F

fast Fourier transform. *See* FFT

Fast RNG: *Vol. II* 555
  period of: *Vol. II* 562

**fast_rng**: *Vol. II* 557, 569

**fast_rng_residue**: *Vol. II* 563, 569

**fast_rng_state_field**: *Vol. II* 563, 569

FastGraph: *Vol. II* 715

FFT
  array conversion utilities: *Vol. II* 521
  bit ordering: *Vol. II* 479
  bit reversal: *Vol. II* 479
  complex-to-complex: *Vol. II* 479
  complex-to-real: *Vol. II* 499
  data orderings (RCFFT and CRFFT): *Vol. II* 500
  Detailed: *Vol. II* 477
  implementation (complex-to-complex): *Vol. II* 483
  implementation (RCFFT and CRFFT): *Vol. II* 505
  introduction to: *Vol. II* 476
  library calls: *Vol. II* 477
  multidimensional (complex-to-complex): *Vol. II* 481
  multidimensional (RCFFT and CRFFT): *Vol. II* 503
  multiple-instance (complex-to-complex): *Vol. II* 481
  multiple-instance (RCFFT and CRFFT): *Vol. II* 503
  optimization (complex-to-complex): *Vol. II* 480
  performance (complex-to-complex): *Vol. II* 483
  performing (CRFFT and RCFFT): *Vol. II* 509
  real-to-complex: *Vol. II* 499
  Simple: *Vol. II* 477
  twiddle factors: *Vol. II* 479

**fft**: *Vol. II* 477, 490


*Copyright © 1994 Thinking Machines Corporation*

**fft_detailed**: *Vol. II* 477, 478, 490, 509, 511

**fft_setup**: *Vol. II* 477, 478, 490, 509, 511

finite element numbering: *Vol. II* 674

Fourier transform. *See* FFT

# G

gather operation
  defined: *Vol. II* 636
  examples: *Vol. II* 637

gather utility: *Vol. II* 636

gather, block: *Vol. II* 664

gather, partitioned: *Vol. II* 693

gather, vector: *Vol. II* 651

Gauss-Jordan system solver: *Vol. I* 264, 270–291
  stability and performance: *Vol. I* 266

Gaussian elimination: *Vol. I* 183
  numerical stability: *Vol. I* 191
  with external storage: *Vol. I* 273
  with pairwise pivoting: *Vol. I* 299

Gaussian elimination, pipelined: *Vol. I* 297

**gbl_gen_2_norm**: *Vol. I* 74

**gbl_gen_inner_product** and related routines: *Vol. I* 66

**gen_2_norm**: *Vol. I* 74

**gen_arnoldi** and related routines: *Vol. I* 413, 420

**gen_banded_factor**: *Vol. I* 295, 313

**gen_banded_solve**: *Vol. I* 295, 313

**gen_bidiag** and related routines: *Vol. I* 436

**gen_bidiag_singular_system**: *Vol. I* 457

**gen_gj_invert**: *Vol. I* 264, 267

**gen_gj_solve**: *Vol. I* 264, 270

**gen_infinity_norm**: *Vol. I* 93

**gen_inner_product** and related routines: *Vol. I* 66

**gen_iter_solve** and related routines: *Vol. I* 340

**gen_lu_factor** and related routines: *Vol. I* 193

**gen_lu_factor_ext**: *Vol. I* 274

**gen_lu_setup_ext**: *Vol. I* 274

**gen_lu_solve_ext**: *Vol. I* 274

**gen_mat_block_sparse_mat_mult**: *Vol. I* 133, 147

**gen_mat_grid_sparse_mat_mult**: *Vol. I* 159, 171

**gen_mat_sparse_mat_mult**: *Vol. I* 114, 121

**gen_matrix_mult** and related routines: *Vol. I* 97

**gen_matrix_mult_ext**: *Vol. I* 104

**gen_matrix_transpose**: *Vol. II* 633

**gen_matrix_vector_mult** and related routines: *Vol. I* 83

**gen_outer_product** and related routines: *Vol. I* 78

**gen_pentadiag_factor** and related routines: *Vol. I* 323

**gen_qr_factor** and related routines: *Vol. I* 239

**gen_qr_factor_ext**: *Vol. I* 285

**gen_qr_solve_ext**: *Vol. I* 285

**gen_simplex**: *Vol. II* 546

**gen_to_bidiag_left**: *Vol. I* 436

**gen_to_bidiag_right**: *Vol. I* 436

**gen_tridiag_factor** and related routines: *Vol. I* 323

**gen_vector_matrix_mult** and related routines: *Vol. I* 88

generalized eigensystem analysis: *Vol. I* 382

generalized minimal residual algoithm: *Vol. I* 336

**generate_dual**: *Vol. II* 683

global axis: *Vol. I* 30

GMRES algorithm: *Vol. I* 336

grid sparse matrix operations: *Vol. I* 159



*Version 3.2, April 1994*
*Copyright © 1994 Thinking Machines Corporation*

grid sparse matrix representation: *Vol. I* 160

**grid_sparse_mat_gen_mat_mult**: *Vol. I* 159, 171

**grid_sparse_matrix_vector_mult**: *Vol. I* 159, 171

**grid_sparse_setup**: *Vol. I* 159, 171

## H

header file: *Vol. I* 54

histogram: *Vol. II* 589
  how to: *Vol. II* 590
**histogram**: *Vol. II* 592

**histogram_range**: *Vol. II* 589, 594

**histogram_range_cm**: *Vol. II* 589, 594

Householder transformations: *Vol. I* 215

## I

ill-conditioned systems: *Vol. I* 233

infinity norm: *Vol. I* 92

**initialize_fast_rng**: *Vol. II* 557, 569

**initialize_vp_rng**: *Vol. II* 557, 578

inner product: *Vol. I* 64

inverse iteration algorithm: *Vol. I* 371

iterative solvers: *Vol. I* 335

## J

Jacobi rotations: *Vol. I* 388

## L

Lanczos algorithm: *Vol. I* 394
  convergence properties: *Vol. I* 396
  data layout guidelines: *Vol. I* 401, 418
  implementation: *Vol. I* 394
  input arguments and data structures: *Vol. I* 395, 414
  reverse communication interface: *Vol. I* 397, 415

Lanczos routines, selected eigenvalues and eigenvectors: *Vol. I* 394

LAPACK, equivalent CMSSL routines: *Vol. II* 751

least squares decomposition, with external storage: *Vol. I* 284

least squares solution, with external storage: *Vol. I* 285

linking: *Vol. I* 54

load balancing: *Vol. II* 730

local array elements: *Vol. I* 29

local axis: *Vol. I* 30

look-ahead Lanczos algorithm: *Vol. I* 337

*LU* decomposition: *Vol. I* 183
  with external storage: *Vol. I* 273

*LU* factors: *Vol. I* 186

*LU* state, saving and restoring: *Vol. I* 191

## M

man pages, on-line: *Vol. I* 60

matrix inversion: *Vol. I* 264, 267
  stability and performance: *Vol. I* 266

matrix multiplication: *Vol. I* 96
  with external storage: *Vol. I* 103

matrix transpose: *Vol. II* 631

matrix vector multiplication: *Vol. I* 82

mesh, unstructured, partitioning of: *Vol. II* 671

multiple instances: *Vol. I* 33
  all-to-all rotation: *Vol. I* 42
  Fast Fourier transforms: *Vol. I* 41
  how to specify: *Vol. I* 36
  matrix vector multiplication example: *Vol. I* 39


*Copyright © 1994 Thinking Machines Corporation*

polyshift: *Vol. I* 42
*QR* solver example: *Vol. I* 39
RNGs: *Vol. I* 42

## N

NEWS-to-send reordering: *Vol. II* 746

**news_to_send**: *Vol. II* 747

node, processing: *Vol. I* 29

non-local axis: *Vol. I* 30

numeric stability: *Vol. I* 42
definition: *Vol. I* 42

numerical complexity: *Vol. I* 43

## O

**ode_rkf** and related routines: *Vol. II* 532

ODEs: *Vol. II* 529

on-line examples: *Vol. I* 60

on-line man pages: *Vol. I* 60

ordinary differential equations: *Vol. II* 529

out-of-core *LU* routines: *Vol. I* 273

out-of-core matrix multiplication: *Vol. I* 103

out-of-core *QR* routines: *Vol. I* 284

outer product: *Vol. I* 77

## P

parallel bisection algorithm: *Vol. I* 366

**part_gather** and related routines: *Vol. II* 694

**part_scatter** and related routines: *Vol. II* 701

**part_vector_gather** and related routines: *Vol. II* 694

**part_vector_scatter** and related routines: *Vol. II* 701

**partition_mesh**: *Vol. II* 683

partitioned gather utility: *Vol. II* 693

partitioned scatter utility: *Vol. II* 700

partitioning of unstructured mesh: *Vol. II* 671

partitioning permutation: *Vol. II* 679

pentadiagonal systems: *Vol. I* 304

permutation, along an axis: *Vol. II* 737

permutations
block cyclic: *Vol. II* 729
combination of: *Vol. II* 741

**permute_cm_matrix_axis_from_fe**: *Vol. II* 737, 738

pipelined Gaussian elimination: *Vol. I* 297

pointers, renumbering of: *Vol. II* 681

pointers, reordering of: *Vol. II* 671, 680

polyshift operation. *See* PSHIFT

processing element: *Vol. I* 29

processing node: *Vol. I* 29

PSHIFT
operation: *Vol. II* 599
optimization recommendations: *Vol. II* 601

**pshift** and related routines: *Vol. II* 599

**pshift_setup**: *Vol. II* 599, 602

**pshift_setup_looped**: *Vol. II* 599, 602

## Q

QMR algorithm: *Vol. I* 336

*QR*, backsolve strategy: *Vol. I* 229

*QR* factorization
with external storage: *Vol. I* 284, 285

*QR* factors: *Vol. I* 223

*QR* routines: *Vol. I* 215
blocking and load balancing: *Vol. I* 222
Householder algorithm: *Vol. I* 219
numerical stability: *Vol. I* 232
pivoting option: *Vol. I* 233

*QR* state, saving and restoring: *Vol. I* 237




*Copyright © 1994 Thinking Machines Corporation*

quasi-minimal residual algorithm: *Vol. I* 336

# R

random number generators: *Vol. II* 555
range histogram: *Vol. II* 590
RCFFT. *See* FFT
real-to-complex FFT: *Vol. II* 499
  *See also* FFT
**real_from_complex**: *Vol. II* 522
reduction to tridiagonal form: *Vol. I* 359
reduction, all-to-all: *Vol. II* 625
**reinitialize_fast_rng**: *Vol. II* 569
**reinitialize_vp_rng**: *Vol. II* 578
**renumber_pointers**: *Vol. II* 683
renumbering of pointers: *Vol. II* 681
**reorder_pointers**: *Vol. II* 683
reordering of pointers: *Vol. II* 671, 680
restarted generalized minimal residual algorithm: *Vol. I* 336
**restore_fast_rng_temps**: *Vol. II* 557, 569
**restore_gen_lu**: *Vol. I* 193
**restore_gen_lu_fms**: *Vol. I* 193
**restore_gen_qr**: *Vol. I* 239
**restore_gen_qr_fms**: *Vol. I* 239
**restore_vp_rng_temps**: *Vol. II* 557, 578
reverse communication interface: *Vol. I* 397, 415
RNG
  alternate stream checkpointing: *Vol. II* 558, 565
  checkpointing: *Vol. II* 562
  Fast: *Vol. II* 555
  fast and VP compared: *Vol. II* 556
  implementation: *Vol. II* 558
  period of a: *Vol. II* 562
  safety checkpointing: *Vol. II* 557, 564
  saving and restoring: *Vol. II* 564
  state tables: *Vol. II* 559
  VP: *Vol. II* 555
Runge-Kutta method: *Vol. II* 529

# S

sample code: *Vol. I* 60
**save_fast_rng_temps**: *Vol. II* 557, 563, 569
**save_gen_lu**: *Vol. I* 193
**save_gen_lu_fms**: *Vol. I* 193
**save_gen_qr**: *Vol. I* 239
**save_gen_qr_fms**: *Vol. I* 239
**save_vp_rng_temps**: *Vol. II* 557, 563, 578
scaling: *Vol. I* 236
scatter operation
  defined: *Vol. II* 644
  example: *Vol. II* 645
scatter utility: *Vol. II* 644
scatter, block: *Vol. II* 664
scatter, partitioned: *Vol. II* 700
scatter, vector: *Vol. II* 658
send-to-NEWS reordering: *Vol. II* 746
**send_to_news**: *Vol. II* 747
Simple FFT: *Vol. II* 477
simplex: *Vol. II* 541
  algorithm: *Vol. II* 541
  degeneracy: *Vol. II* 544
  reinversion: *Vol. II* 543
  vertices and bases: *Vol. II* 542
singular value decomposition: *Vol. I* 433, 456
singular values of bidiagonal matrices: *Vol. I* 447
singular vectors of bidiagonal matrices: *Vol. I* 451
SPARC processing node: *Vol. I* 29
sparse gather utility: *Vol. II* 636
sparse matrices, storage of: *Vol. I* 115


*Copyright © 1994 Thinking Machines Corporation*

sparse matrix operations
  arbitrary block: *Vol. I* 133
  arbitrary elementwise: *Vol. I* 114
  arbitrary sparse matrices: *Vol. I* 109
  gathering and scattering: *Vol. I* 111
  grid: *Vol. I* 159
  optimization recommendations: *Vol. I* 112
  storage representations: *Vol. I* 110
sparse scatter utility: *Vol. II* 644
sparse vector scatter utility: *Vol. II* 658
**sparse_mat_gen_mat_mult**: *Vol. I* 114, 121
**sparse_matvec_mult**: *Vol. I* 114, 121
**sparse_matvec_setup**: *Vol. I* 114, 121
**sparse_util_gather**: *Vol. II* 636, 640
**sparse_util_gather_setup**: *Vol. II* 636, 640
**sparse_util_scatter**: *Vol. II* 644, 647
**sparse_util_scatter_setup**: *Vol. II* 644, 647
**sparse_util_vec_gather**: *Vol. II* 654
**sparse_util_vec_gather_setup**: *Vol. II* 654
**sparse_util_vec_scatter**: *Vol. II* 660
**sparse_util_vec_scatter_setup**: *Vol. II* 660
**sparse_vecmat_mult**: *Vol. I* 114, 121
**sparse_vecmat_setup**: *Vol. I* 114, 121
stability, definition: *Vol. I* 42
statistical analysis: *Vol. II* 589–596
subgrid: *Vol. I* 29
substructuring: *Vol. I* 299
SVD. *See* singular value decomposition
**sym_jacobi_eigensystem**: *Vol. I* 388, 390
**sym_lanczos** and related routines: *Vol. I* 394, 403
**sym_to_tridiag**: *Vol. I* 359, 360
**sym_tred**: *Vol. I* 359, 360
**sym_tred_eigensystem**: *Vol. I* 377, 378
**sym_tred_gen_eigensystem**: *Vol. I* 382, 384
**sym_tridiag_eigenvalues**: *Vol. I* 366, 369
**sym_tridiag_eigenvectors**: *Vol. I* 371, 374

# T

transpose operation: *Vol. II* 631
transpose, matrix: *Vol. II* 631
**tridiag_to_sym**: *Vol. I* 359, 360
tridiagonal form, reduction to: *Vol. I* 359
tridiagonal systems: *Vol. I* 304
twiddle factors for FFT: *Vol. II* 479
two-norm: *Vol. I* 73

# U

unstructured mesh, partitioning of: *Vol. II* 671

# V

vector gather operation
  defined: *Vol. II* 651
  examples: *Vol. II* 651
vector gather utility: *Vol. II* 651
vector matrix multiplication: *Vol. I* 87
vector move: *Vol. II* 724
vector scatter operation
  defined: *Vol. II* 658
  examples: *Vol. II* 658
vector unit: *Vol. I* 29
**vector_block_sparse_matrix_mult**: *Vol. I* 133, 147
**vector_grid_sparse_matrix_mult**: *Vol. I* 159, 171
**vector_move**: *Vol. II* 725
**vector_move_utils**: *Vol. II* 725
VP RNG: *Vol. II* 555
  period of: *Vol. II* 562
**vp_rng**: *Vol. II* 557, 578
**vp_rng_residue**: *Vol. II* 563, 578
**vp_rng_state_field**: *Vol. II* 563, 578




*Copyright © 1994 Thinking Machines Corporation*

# Z

**zero_elements**: *Vol. II* 745
zeroing routine: *Vol. II* 744


*Copyright © 1994 Thinking Machines Corporation*