

Meschach

Matrix Computations in C

1	Tutorial	1
1.1	The data structures and some basic operations	1
1.2	How to manage memory	5
1.2.1	No deallocation	6
1.2.2	Allocate and deallocate	6
1.2.3	Resize on demand	6
1.2.4	Registration of workspace	7
1.3	Simple vector operations: An RK4 routine	8
1.4	Using routines for lists of arguments	14
1.5	A least squares problem	15
1.6	A sparse matrix example	18
1.7	How do I ?	20
1.7.1 solve a system of linear equations	20
1.7.2 solve a least-squares problem	20
1.7.3 find all the eigenvalues (and eigenvectors) of a general matrix	20
1.7.4 solve a large, sparse, positive definite system of equations	21
2	Data structures	23
2.1	General principles	23
2.2	Vectors	24
2.2.1	Integer vectors	25
2.2.2	Complex vectors	25
2.3	Matrices	26
2.3.1	Complex matrices	27
2.3.2	Band matrices	27
2.4	Permutations	28
2.5	Basic sparse operations and structures	29
2.6	The sparse data structures	30
2.7	Sparse matrix factorisation	33
2.8	Iterative techniques	34
2.9	Other data structures	36
3	Numerical Linear Algebra	37
3.1	What numerical linear algebra is about	37
3.2	Complex conjugates and adjoints	38
3.3	Vector and matrix norms	38
3.4	“Ill conditioning” or intrinsically bad problems	39

3.5	Least squares and pseudo-inverses	41
3.5.1	Singular Value Decompositions	42
3.5.2	Pseudo-inverses	42
3.5.3	QR factorisations and least squares	43
3.6	Eigenvalues and eigenvectors	44
3.7	Sparse matrix operations	46
4	Basic Dense Matrix Operations	49
5	Dense Matrix Factorisation Operations	115
6	Sparse Matrix & Iterative Operations	148
7	Installation and copyright	181
7.1	Installation	181
7.1.1	Installation on non-Unix systems	183
7.1.2	makefile	184
7.1.3	machine.h	184
7.1.4	machine.c	185
7.2	Backward compatibility	187
7.3	Copyright	187
8	Designing numerical libraries in C	189
8.1	Numerical programming in C	189
8.1.1	On efficient compilers	190
8.1.2	Strategies for using C	190
8.1.3	Non-C programmers start here!	191
8.2	The data structures	195
8.2.1	Pointers to struct's	195
8.2.2	Really basic operations	196
8.2.3	Output	198
8.2.4	Copying	199
8.2.5	Input	200
8.2.6	Resizing	202
8.3	How to implement routines	203
8.3.1	Design for debugging	203
8.3.2	Workspace	204
8.3.3	Incorporating user-defined types into Meschach	206
8.3.4	Output and object resizing	210
8.4	User-defined functions	211
8.5	Building the library	213
8.5.1	Numerical aspects	214
8.6	Debugging	215
8.6.1	Memory allocation bugs	216
8.6.2	If all else fails	217

8.7	Suggestions for enthusiasts	218
8.8	Pride and Prejudice	218
8.8.1	What about Fortran 90?	218
8.8.2	Why should people writing numerical code care about good software?	218
	For further reading . . .	220
	Index	221
	Function index	229