# Manual for Package: mathematics
# Revision 19M

Karl Kästner

June 12, 2022

## Contents

**1 calendar** **1**
1.1 days_per_month . . . . . 1
1.2 isnight . . . . . 1

**2 mathematics** **1**
2.1 cast_byte_to_integer . . . . . 1

**3 complex-analysis** **1**
3.1 complex_exp_product_im_im . . . . . 1
3.2 complex_exp_product_im_re . . . . . 2
3.3 complex_exp_product_re_im . . . . . 2
3.4 complex_exp_product_re_re . . . . . 2
3.5 croots . . . . . 3
3.6 root_complex . . . . . 3
3.7 test_imroots . . . . . 3

**4 derivation** **3**
4.1 derive_acfar1 . . . . . 3
4.2 derive_ar1_spectral_density . . . . . 4
4.3 derive_ar2param . . . . . 4
4.4 derive_arc_length . . . . . 4
4.5 derive_fourier_power . . . . . 4
4.6 derive_fourier_power_exp . . . . . 4
4.7 derive_laplacian_curvilinear . . . . . 4
4.8 derive_laplacian_fourier_piecewise_linear . . . . . 4
4.9 derive_logtripdf . . . . . 4
4.10 derive_smooth1d_parametric . . . . . 4
4.11 derive_spectral_density_bandpass_initial_condition . . . . . 4

**5 derivation/master** **5**

5.1 derive_bc_one_sided . . . 5
5.2 derive_convergence . . . 5
5.3 derive_error_fdm . . . 5
5.4 derive_fdm_poly . . . 5
5.5 derive_fdm_power . . . 5
5.6 derive_fdm_taylor . . . 5
5.7 derive_fdm_vargrid . . . 5
5.8 derive_fem_2d_mass . . . 5
5.9 derive_fem_error_2d . . . 5
5.10 derive_fem_error_3d . . . 6
5.11 derive_fem_sym_2d . . . 6
5.12 derive_grid_constants . . . 6
5.13 derive_interpolation . . . 6
5.14 derive_laplacian . . . 6
5.15 derive_limit . . . 6
5.16 derive_nc_1d . . . 6
5.17 derive_nc_1d_ . . . 6
5.18 derive_nc_2d . . . 6
5.19 derive_nonuniform_symmetric . . . 6
5.20 derive_richardson . . . 7
5.21 derive_sum . . . 7
5.22 nn . . . 7
5.23 test_derive . . . 7
5.24 test_derive_fdm_poly . . . 7
5.25 test_filter . . . 7
5.26 test_vargrid . . . 7

**6 derivation** **7**
6.1 simplify_atan . . . 7

**7 mathematics** **8**
7.1 entropy . . . 8

**8 finance** **8**
8.1 derive_skewrnd_walsh_paramter . . . 8
8.2 gbm_cdf . . . 8
8.3 gbm_fit . . . 8
8.4 gbm_fit_old . . . 8
8.5 gbm_inv . . . 8
8.6 gbm_mean . . . 8
8.7 gbm_median . . . 8
8.8 gbm_pdf . . . 9
8.9 gbm_simulate . . . 9
8.10 gbm_skewness . . . 9

8.11 gbm_std . . . 9
8.12 gbm_transform_time_step . . . 9
8.13 put_price_black_scholes . . . 9
8.14 skewgbm_simulate . . . 9
8.15 skewrnd_walsh . . . 9

**9 finance/test** **9**
9.1 test_gbm . . . 9
9.2 test_gbm_pdf . . . 10
9.3 test_skewrnd_walsh . . . 10

**10 fourier/@STFT** **10**
10.1 STFT . . . 10
10.2 itransform . . . 10
10.3 stft_ . . . 10
10.4 stftmat . . . 10
10.5 transform . . . 10

**11 fourier** **11**
11.1 amplitude_from_peak . . . 11
11.2 dftmtx_man . . . 11
11.3 example_fourier_window . . . 11
11.4 fft_man . . . 11
11.5 fftsmooth . . . 12
11.6 fix_fourier . . . 12
11.7 fourier_axis . . . 12
11.8 fourier_axis_2d . . . 13
11.9 fourier_cesaro_correction . . . 13
11.10 fourier_coefficient_piecewise_linear . . . 13
11.11 fourier_coefficient_piecewise_linear_1 . . . 13
11.12 fourier_coefficient_ramp3 . . . 14
11.13 fourier_coefficient_ramp_pulse . . . 14
11.14 fourier_coefficient_ramp_step . . . 14
11.15 fourier_coefficient_square_pulse . . . 14
11.16 fourier_cubic_interaction_coefficients . . . 14
11.17 fourier_derivative . . . 14
11.18 fourier_derivative_matrix_1d . . . 15
11.19 fourier_derivative_matrix_2d . . . 15
11.20 fourier_expand . . . 15
11.21 fourier_fit . . . 15
11.22 fourier_interpolate . . . 15
11.23 fourier_matrix . . . 15
11.24 fourier_matrix2 . . . 15
11.25 fourier_matrix3 . . . 16

11.26 fourier_matrix_exp . . . 16
11.27 fourier_multiplicative_interaction_coefficients . . . 16
11.28 fourier_power . . . 16
11.29 fourier_power_exp . . . 16
11.30 fourier_predict . . . 17
11.31 fourier_quadratic_interaction_coefficients . . . 17
11.32 fourier_random_phase_walk . . . 17
11.33 fourier_range . . . 17
11.34 fourier_regress . . . 17
11.35 fourier_resampled_fit . . . 17
11.36 fourier_resampled_predict . . . 17
11.37 fourier_signed_square . . . 18
11.38 fourier_transform . . . 18
11.39 fourier_transform_fractional . . . 18
11.40 hyperbolic_fourier_box . . . 18
11.41 idftmtx_man . . . 18
11.42 laplace_2d_pwlinear . . . 19
11.43 mean_fourier_power . . . 19
11.44 moments_fourier_power . . . 19
11.45 nanfft . . . 19
11.46 peaks . . . 19
11.47 roots_fourier . . . 20
11.48 spectral_density . . . 20
11.49 std_fourier_power . . . 20
11.50 test_complex_exp_product . . . 20
11.51 test_fourier_filter . . . 20
11.52 test_idftmtx . . . 20
11.53 var_fourier_power . . . 20

**12 mathematics** **21**
12.1 gaussfit_quantile . . . 21

**13 geometry/@Geometry** **21**
13.1 Geometry . . . 21
13.2 arclength . . . 21
13.3 arclength_old . . . 21
13.4 arclength_old2 . . . 21
13.5 base_point . . . 21
13.6 base_point_limited . . . 22
13.7 centroid . . . 22
13.8 cosa_min_max . . . 22
13.9 cross2 . . . 22
13.10 curvature . . . 22
13.11 ddot . . . 22

13.12 distance . . . 22
13.13 distance2 . . . 22
13.14 dot . . . 23
13.15 edge_length . . . 23
13.16 enclosed_angle . . . 23
13.17 enclosing_triangle . . . 23
13.18 hexagon . . . 23
13.19 inPolygon . . . 23
13.20 inTetra . . . 23
13.21 inTetra2 . . . 23
13.22 inTriangle . . . 24
13.23 intersect . . . 24
13.24 lineintersect . . . 24
13.25 lineintersect1 . . . 24
13.26 minimum_distance_lines . . . 24
13.27 mittenpunkt . . . 24
13.28 nagelpoint . . . 24
13.29 onLine . . . 24
13.30 orthocentre . . . 25
13.31 plumb_line . . . 25
13.32 poly_area . . . 25
13.33 poly_edges . . . 25
13.34 poly_set . . . 25
13.35 poly_width . . . 25
13.36 polyxpoly . . . 25
13.37 project_to_curve . . . 26
13.38 quad_isconvex . . . 26
13.39 random_disk . . . 26
13.40 random_simplex . . . 26
13.41 sphere_volume . . . 26
13.42 tetra_volume . . . 26
13.43 tobarycentric . . . 26
13.44 tobarycentric1 . . . 26
13.45 tobarycentric2 . . . 27
13.46 tobarycentric3 . . . 27
13.47 tri_angle . . . 27
13.48 tri_area . . . 27
13.49 tri_centroid . . . 27
13.50 tri_distance_opposit_midpoint . . . 27
13.51 tri_edge_length . . . 27
13.52 tri_edge_midpoint . . . 27
13.53 tri_excircle . . . 28
13.54 tri_height . . . 28
13.55 tri_incircle . . . 28

13.56 tri_isacute . . . . . 28
13.57 tri_isobtuse . . . . . 28
13.58 tri_semiperimeter . . . . . 28
13.59 tri_side_length . . . . . 28

**14 geometry** **29**
14.1 Polygon . . . . . 29
14.2 bounding_box . . . . . 29
14.3 curvature_1d . . . . . 29
14.4 cvt . . . . . 29
14.5 deg_to_frac . . . . . 29
14.6 ellipse . . . . . 30
14.7 ellipseX . . . . . 30
14.8 ellipseY . . . . . 30
14.9 first_intersect . . . . . 30
14.10 golden_ratio . . . . . 30
14.11 hypot3 . . . . . 30
14.12 meanangle . . . . . 30
14.13 meanangle2 . . . . . 30
14.14 meanangle3 . . . . . 31
14.15 meanangle4 . . . . . 31
14.16 medianangle . . . . . 31
14.17 medianangle2 . . . . . 31
14.18 pilim . . . . . 31
14.19 streamline_radius_of_curvature . . . . . 31

**15 histogram/@Histogram** **32**
15.1 2x . . . . . 32
15.2 Histogram . . . . . 32
15.3 bimodes . . . . . 32
15.4 cdf . . . . . 32
15.5 cdfS . . . . . 32
15.6 chi2test . . . . . 32
15.7 cmoment . . . . . 32
15.8 cmomentS . . . . . 32
15.9 entropy . . . . . 32
15.10 entropyS . . . . . 33
15.11 export_csv . . . . . 33
15.12 iquantile . . . . . 33
15.13 kstest . . . . . 33
15.14 kurtosis . . . . . 33
15.15 kurtosisS . . . . . 33
15.16 mean . . . . . 33
15.17 meanS . . . . . 33

15.18 median . . . . . . . . . . 33
15.19 medianS . . . . . . . . . . 33
15.20 mode . . . . . . . . . . 34
15.21 modeS . . . . . . . . . . 34
15.22 moment . . . . . . . . . . 34
15.23 momentS . . . . . . . . . . 34
15.24 pdf . . . . . . . . . . 34
15.25 quantile . . . . . . . . . . 34
15.26 quantileS . . . . . . . . . . 34
15.27 resample . . . . . . . . . . 34
15.28 setup . . . . . . . . . . 34
15.29 skewness . . . . . . . . . . 34
15.30 skewnessS . . . . . . . . . . 35
15.31 stairs . . . . . . . . . . 35
15.32 stairsS . . . . . . . . . . 35
15.33 std . . . . . . . . . . 35
15.34 stdS . . . . . . . . . . 35
15.35 var . . . . . . . . . . 35
15.36 varS . . . . . . . . . . 35

**16 histogram** **35**
16.1 hist_man . . . . . . . . . . 35
16.2 histadapt . . . . . . . . . . 35
16.3 histconst . . . . . . . . . . 36
16.4 pdf_poly . . . . . . . . . . 36
16.5 plotcdf . . . . . . . . . . 36
16.6 test_histogram . . . . . . . . . . 36

**17 mathematics** **36**
17.1 imrotmat . . . . . . . . . . 36

**18 linear-algebra** **36**
18.1 averaging_matrix_2 . . . . . . . . . . 36
18.2 colnorm . . . . . . . . . . 36
18.3 condest_ . . . . . . . . . . 36
18.4 connectivity_matrix . . . . . . . . . . 37

**19 linear-algebra/coordinate-transformation** **37**
19.1 barycentric2cartesian . . . . . . . . . . 37
19.2 barycentric2cartesian3 . . . . . . . . . . 37
19.3 cartesian2barycentric . . . . . . . . . . 37
19.4 cartesian_to_unit_triangle_basis . . . . . . . . . . 37
19.5 ellipsoid2geoid . . . . . . . . . . 37
19.6 example_approximate_utm_conversion . . . . . . . . . . 37

19.7 latlon2utm . . . 37
19.8 latlon2utm_simple . . . 38
19.9 lowrance_mercator_to_wgs84 . . . 38
19.10 nmea2utm . . . 38
19.11 sn2xy . . . 38
19.12 unit_triangle_to_cartesian . . . 38
19.13 utm2latlon . . . 38
19.14 xy2nt . . . 38
19.15 xy2sn . . . 39
19.16 xy2sn_java . . . 39
19.17 xy2sn_old . . . 39

**20 linear-algebra** **39**
20.1 det2x2 . . . 39
20.2 det3x3 . . . 39
20.3 det4x4 . . . 39
20.4 diag2x2 . . . 39
20.5 down . . . 40
20.6 eig2x2 . . . 40

**21 linear-algebra/eigenvalue** **40**
21.1 eig_bisection . . . 40
21.2 eig_inverse . . . 40
21.3 eig_inverse_iteration . . . 40
21.4 eig_power_iteration . . . 40

**22 linear-algebra/eigenvalue/jacobi-davidson** **40**
22.1 afun_jdm . . . 40
22.2 davidson . . . 40
22.3 jacobi_davidson . . . 40
22.4 jacobi_davidson_qr . . . 41
22.5 jacobi_davidson_qz . . . 41
22.6 jacobi_davidson_simple . . . 41
22.7 jdqr . . . 41
22.8 jdqr_sleijpen . . . 44
22.9 jdqr_vorst . . . 48
22.10 jdqz . . . 51
22.11 mfunc_jdm . . . 55
22.12 mgs . . . 55
22.13 minres_ . . . 55
22.14 mv_jacobi_davidson . . . 55

**23 linear-algebra** **56**
23.1 first . . . 56

23.2 gershgorin_circle . . . . . . . . . . 56
23.3 haussdorff . . . . . . . . . . 56
23.4 ieig2x2 . . . . . . . . . . 56
23.5 inv2x2 . . . . . . . . . . 56
23.6 inv3x3 . . . . . . . . . . 56
23.7 inv4x4 . . . . . . . . . . 56
23.8 kernel2matrix . . . . . . . . . . 57

**24 linear-algebra/lanczos 57**
24.1 arnoldi . . . . . . . . . . 57
24.2 arnoldi_new . . . . . . . . . . 57
24.3 eigs_lanczos_man . . . . . . . . . . 57
24.4 lanczos . . . . . . . . . . 57
24.5 lanczos_ . . . . . . . . . . 57
24.6 lanczos_biorthogonal . . . . . . . . . . 57
24.7 lanczos_biorthogonal_improved . . . . . . . . . . 57
24.8 lanczos_ghep . . . . . . . . . . 57
24.9 mv_lanczos . . . . . . . . . . 58
24.10 reorthogonalise . . . . . . . . . . 58
24.11 test_lanczos . . . . . . . . . . 58

**25 linear-algebra 58**
25.1 left . . . . . . . . . . 58

**26 linear-algebra/linear-systems 58**
26.1 gmres_man . . . . . . . . . . 58
26.2 minres_recycle . . . . . . . . . . 58

**27 linear-algebra 58**
27.1 lpmean . . . . . . . . . . 58
27.2 lpnorm . . . . . . . . . . 58
27.3 matvec3 . . . . . . . . . . 59
27.4 max2d . . . . . . . . . . 59
27.5 mid . . . . . . . . . . 59
27.6 mpoweri . . . . . . . . . . 59
27.7 mtimes2x2 . . . . . . . . . . 59
27.8 mtimes3x3 . . . . . . . . . . 59
27.9 nannorm . . . . . . . . . . 59
27.10 nanshift . . . . . . . . . . 59
27.11 nl . . . . . . . . . . 60
27.12 normalise . . . . . . . . . . 60
27.13 normalize1 . . . . . . . . . . 60
27.14 normrows . . . . . . . . . . 60
27.15 orth2 . . . . . . . . . . 60

27.16 orth_man . . . . . . . . . . 60
27.17 orthogonalise . . . . . . . . . . 60
27.18 padd2 . . . . . . . . . . 61
27.19 paddext . . . . . . . . . . 61
27.20 paddval1 . . . . . . . . . . 61
27.21 paddval2 . . . . . . . . . . 61

**28 linear-algebra/polynomial 61**
28.1 chebychev . . . . . . . . . . 61
28.2 piecewise_polynomial . . . . . . . . . . 61
28.3 roots1 . . . . . . . . . . 61
28.4 roots2 . . . . . . . . . . 62
28.5 roots2poly . . . . . . . . . . 62
28.6 roots3 . . . . . . . . . . 62
28.7 roots4 . . . . . . . . . . 62
28.8 roots_piecewise_linear . . . . . . . . . . 62
28.9 test_roots4 . . . . . . . . . . 62
28.10 vanderi_1d . . . . . . . . . . 62

**29 linear-algebra 62**
29.1 randrot . . . . . . . . . . 62
29.2 right . . . . . . . . . . 63
29.3 rot2 . . . . . . . . . . 63
29.4 rot2dir . . . . . . . . . . 63
29.5 rot3 . . . . . . . . . . 63
29.6 rotR . . . . . . . . . . 63
29.7 rownorm . . . . . . . . . . 63
29.8 simmilarity_matrix . . . . . . . . . . 63
29.9 spnorm . . . . . . . . . . 63
29.10 spzeros . . . . . . . . . . 63
29.11 test_roots3 . . . . . . . . . . 64
29.12 transform_minmax . . . . . . . . . . 64
29.13 transpose3 . . . . . . . . . . 64
29.14 transposeall . . . . . . . . . . 64
29.15 up . . . . . . . . . . 64
29.16 vander_nd . . . . . . . . . . 64
29.17 vanderd_2d . . . . . . . . . . 64

**30 logic 64**
30.1 bitor_man . . . . . . . . . . 64

**31 master/plot 65**
31.1 attach_boundary_value . . . . . . . . . . 65
31.2 cartesian_polar . . . . . . . . . . 65

31.3 img_vargrid . . . 65
31.4 plot_basis_functions . . . 65
31.5 plot_convergence . . . 65
31.6 plot_dof . . . 65
31.7 plot_eigenbar . . . 65
31.8 plot_error_estimation . . . 65
31.9 plot_error_estimation_2 . . . 65
31.10 plot_error_fem . . . 66
31.11 plot_fdm_kernel . . . 66
31.12 plot_fdm_vs_fem . . . 66
31.13 plot_fem_accuracy . . . 66
31.14 plot_function_and_grid . . . 66
31.15 plot_hat . . . 66
31.16 plot_hydrogen_wf . . . 66
31.17 plot_mesh . . . 66
31.18 plot_mesh_2 . . . 66
31.19 plot_refine . . . 66
31.20 plot_refine_3d . . . 67
31.21 plot_runtime . . . 67
31.22 plot_spectrum . . . 67
31.23 plot_wavefunction . . . 67

**32 master/ported 67**
32.1 assemble_2d_phi_phi . . . 67
32.2 assemble_3d_dphi_dphi . . . 67
32.3 assemble_3d_phi_phi . . . 67
32.4 dV_2d_ . . . 67
32.5 derivative_2d . . . 67
32.6 derivative_3d . . . 68
32.7 element_neighbour_2d . . . 68
32.8 prefetch_2d_ . . . 68
32.9 promote_2d_3_10 . . . 68
32.10 promote_2d_3_15 . . . 68
32.11 promote_2d_3_21 . . . 68
32.12 promote_2d_3_6 . . . 68
32.13 promote_3d_4_10 . . . 68
32.14 promote_3d_4_20 . . . 68
32.15 promote_3d_4_35 . . . 68
32.16 vander_2d . . . 69
32.17 vander_3d . . . 69

**33 number-theory 69**
33.1 ceiln . . . 69
33.2 digitsb . . . 69

33.3 floorn . . . 69
33.4 iseven . . . 69
33.5 multichoosek . . . 69
33.6 nchoosek_man . . . 70
33.7 pythagorean_triple . . . 70
33.8 roundn . . . 70

**34 numerical-methods** **70**
34.1 advect_analytic . . . 70

**35 numerical-methods/differentiation** **70**
35.1 derivative1 . . . 70
35.2 derivative2 . . . 70

**36 numerical-methods** **70**
36.1 diffuse_analytic . . . 70

**37 numerical-methods/finite-difference** **71**
37.1 cdiff . . . 71
37.2 cdiffb . . . 71
37.3 central_difference . . . 71
37.4 cmean . . . 71
37.5 cmean2 . . . 71
37.6 derivative_matrix_1_1d . . . 71
37.7 derivative_matrix_2_1d . . . 71
37.8 derivative_matrix_2d . . . 72
37.9 derivative_matrix_curvilinear . . . 72
37.10 derivative_matrix_curvilinear_2 . . . 72
37.11 difference_kernel . . . 72
37.12 diffusion_matrix_2d_anisotropic . . . 72
37.13 diffusion_matrix_2d_anisotropic2 . . . 72
37.14 distmat . . . 72
37.15 downwind_difference . . . 72
37.16 gradpde2d . . . 73
37.17 laplacian . . . 73
37.18 laplacian_fdm . . . 73
37.19 lrmean . . . 73

**38 numerical-methods/finite-difference/master** **73**
38.1 fdm_adaptive_grid . . . 73
38.2 fdm_adaptive_refinement_old . . . 73
38.3 fdm_assemble_d1_2d . . . 73
38.4 fdm_assemble_d2_2d . . . 73
38.5 fdm_confinement . . . 74

38.6 fdm_d_vargrid . . . 74
38.7 fdm_h_unstructured . . . 74
38.8 fdm_hydrogen_vargrid . . . 74
38.9 fdm_mark_unstructured_2d . . . 74
38.10 fdm_plot . . . 74
38.11 fdm_plot_series . . . 74
38.12 fdm_refine_2d . . . 74
38.13 fdm_refine_3d . . . 74
38.14 fdm_refine_unstructured_2d . . . 74
38.15 fdm_schroedinger_2d . . . 75
38.16 fdm_schroedinger_3d . . . 75
38.17 relocate . . . 75

**39 numerical-methods/finite-difference** **75**
39.1 mid . . . 75
39.2 pwmid . . . 75
39.3 ratio . . . 75
39.4 steplength . . . 75
39.5 swapoddeven . . . 75
39.6 test_derivative_matrix_2d . . . 76
39.7 test_derivative_matrix_curvilinear . . . 76
39.8 test_difference_kernel . . . 76
39.9 upwind_difference . . . 76

**40 numerical-methods/finite-element** **76**
40.1 Mesh_2d_java . . . 76
40.2 Tree_2d_java . . . 76
40.3 assemble_1d_dphi_dphi . . . 76
40.4 assemble_1d_phi_phi . . . 76
40.5 assemble_2d_dphi_dphi_java . . . 76
40.6 assemble_2d_phi_phi_java . . . 77
40.7 assemble_3d_dphi_dphi_java . . . 77
40.8 assemble_3d_phi_phi_java . . . 77
40.9 boundary_1d . . . 77
40.10 boundary_2d . . . 77
40.11 boundary_3d . . . 77
40.12 check_area_2d . . . 77
40.13 circmesh . . . 77
40.14 cropradius . . . 77
40.15 display_2d . . . 77
40.16 display_3d . . . 78
40.17 distort . . . 78
40.18 err_2d . . . 78
40.19 estimate_err_2d_3 . . . 78

40.20 example_1d . . . . . . . . . . 78
40.21 example_2d . . . . . . . . . . 78
40.22 explode . . . . . . . . . . 78
40.23 fem_2d . . . . . . . . . . 78
40.24 fem_2d_heuristic_mesh . . . . . . . . . . 78
40.25 fem_get_2d_radial . . . . . . . . . . 78
40.26 fem_interpolation . . . . . . . . . . 79
40.27 fem_plot_1d . . . . . . . . . . 79
40.28 fem_plot_1d_series . . . . . . . . . . 79
40.29 fem_plot_2d . . . . . . . . . . 79
40.30 fem_plot_2d_series . . . . . . . . . . 79
40.31 fem_plot_3d . . . . . . . . . . 79
40.32 fem_plot_3d_series . . . . . . . . . . 79
40.33 fem_plot_confine_series . . . . . . . . . . 79
40.34 fem_radial . . . . . . . . . . 79
40.35 flip_2d . . . . . . . . . . 80
40.36 get_mesh_arrays . . . . . . . . . . 80
40.37 hashkey . . . . . . . . . . 80

**41 numerical-methods/finite-element/int 80**

41.1 int_1d_gauss . . . . . . . . . . 80
41.2 int_1d_gauss_1 . . . . . . . . . . 80
41.3 int_1d_gauss_2 . . . . . . . . . . 80
41.4 int_1d_gauss_3 . . . . . . . . . . 80
41.5 int_1d_gauss_4 . . . . . . . . . . 80
41.6 int_1d_gauss_5 . . . . . . . . . . 80
41.7 int_1d_gauss_6 . . . . . . . . . . 81
41.8 int_1d_gauss_lobatto . . . . . . . . . . 81
41.9 int_1d_gauss_n . . . . . . . . . . 81
41.10 int_1d_nc_2 . . . . . . . . . . 81
41.11 int_1d_nc_3 . . . . . . . . . . 81
41.12 int_1d_nc_4 . . . . . . . . . . 81
41.13 int_1d_nc_5 . . . . . . . . . . 81
41.14 int_1d_nc_6 . . . . . . . . . . 81
41.15 int_1d_nc_7 . . . . . . . . . . 81
41.16 int_1d_nc_7_hardy . . . . . . . . . . 81
41.17 int_2d_gauss_1 . . . . . . . . . . 82
41.18 int_2d_gauss_12 . . . . . . . . . . 82
41.19 int_2d_gauss_13 . . . . . . . . . . 82
41.20 int_2d_gauss_16 . . . . . . . . . . 82
41.21 int_2d_gauss_19 . . . . . . . . . . 82
41.22 int_2d_gauss_25 . . . . . . . . . . 82
41.23 int_2d_gauss_3 . . . . . . . . . . 82
41.24 int_2d_gauss_33 . . . . . . . . . . 82

41.25 int_2d_gauss_4 . . . . . . . . . . . . . . . . . . . . . . . . 82
41.26 int_2d_gauss_6 . . . . . . . . . . . . . . . . . . . . . . . . 82
41.27 int_2d_gauss_7 . . . . . . . . . . . . . . . . . . . . . . . . 83
41.28 int_2d_gauss_9 . . . . . . . . . . . . . . . . . . . . . . . . 83
41.29 int_2d_nc_10 . . . . . . . . . . . . . . . . . . . . . . . . 83
41.30 int_2d_nc_15 . . . . . . . . . . . . . . . . . . . . . . . . 83
41.31 int_2d_nc_21 . . . . . . . . . . . . . . . . . . . . . . . . 83
41.32 int_2d_nc_3 . . . . . . . . . . . . . . . . . . . . . . . . 83
41.33 int_2d_nc_6 . . . . . . . . . . . . . . . . . . . . . . . . 83
41.34 int_3d_gauss_1 . . . . . . . . . . . . . . . . . . . . . . . . 83
41.35 int_3d_gauss_11 . . . . . . . . . . . . . . . . . . . . . . . . 83
41.36 int_3d_gauss_14 . . . . . . . . . . . . . . . . . . . . . . . . 83
41.37 int_3d_gauss_15 . . . . . . . . . . . . . . . . . . . . . . . . 84
41.38 int_3d_gauss_24 . . . . . . . . . . . . . . . . . . . . . . . . 84
41.39 int_3d_gauss_4 . . . . . . . . . . . . . . . . . . . . . . . . 84
41.40 int_3d_gauss_45 . . . . . . . . . . . . . . . . . . . . . . . . 84
41.41 int_3d_gauss_5 . . . . . . . . . . . . . . . . . . . . . . . . 84
41.42 int_3d_nc_11 . . . . . . . . . . . . . . . . . . . . . . . . 84
41.43 int_3d_nc_4 . . . . . . . . . . . . . . . . . . . . . . . . 84
41.44 int_3d_nc_6 . . . . . . . . . . . . . . . . . . . . . . . . 84
41.45 int_3d_nc_8 . . . . . . . . . . . . . . . . . . . . . . . . 84

**42 numerical-methods/finite-element 85**
42.1 interpolation_matrix . . . . . . . . . . . . . . . . . . . . . . . . 85
42.2 mark . . . . . . . . . . . . . . . . . . . . . . . . 85
42.3 mark_1d . . . . . . . . . . . . . . . . . . . . . . . . 85
42.4 mesh_1d_uniform . . . . . . . . . . . . . . . . . . . . . . . . 85
42.5 mesh_3d_uniform . . . . . . . . . . . . . . . . . . . . . . . . 85
42.6 mesh_interpolate . . . . . . . . . . . . . . . . . . . . . . . . 85
42.7 neighbour_1d . . . . . . . . . . . . . . . . . . . . . . . . 85
42.8 old . . . . . . . . . . . . . . . . . . . . . . . . 85
42.9 pdeeig_1d . . . . . . . . . . . . . . . . . . . . . . . . 85
42.10 pdeeig_2d . . . . . . . . . . . . . . . . . . . . . . . . 86
42.11 pdeeig_3d . . . . . . . . . . . . . . . . . . . . . . . . 86
42.12 polynomial_derivative_1d . . . . . . . . . . . . . . . . . . . . . . . . 86
42.13 potential_const . . . . . . . . . . . . . . . . . . . . . . . . 86
42.14 potential_coulomb . . . . . . . . . . . . . . . . . . . . . . . . 86
42.15 potential_harmonic_oscillator . . . . . . . . . . . . . . . . . . . . . . . . 86
42.16 project_circle . . . . . . . . . . . . . . . . . . . . . . . . 86
42.17 project_rectangle . . . . . . . . . . . . . . . . . . . . . . . . 86
42.18 promote_1d_2_3 . . . . . . . . . . . . . . . . . . . . . . . . 86
42.19 promote_1d_2_4 . . . . . . . . . . . . . . . . . . . . . . . . 86
42.20 promote_1d_2_5 . . . . . . . . . . . . . . . . . . . . . . . . 87
42.21 promote_1d_2_6 . . . . . . . . . . . . . . . . . . . . . . . . 87

42.22 quadrilaterate . . . . . 87
42.23 recalculate_regularity_2d . . . . . 87
42.24 refine_1d . . . . . 87
42.25 refine_2d_21 . . . . . 87
42.26 refine_2d_structural . . . . . 87
42.27 regularity_1d . . . . . 87
42.28 regularity_2d . . . . . 87
42.29 regularity_3d . . . . . 88
42.30 relocate_2d . . . . . 88
42.31 test_circmesh . . . . . 88
42.32 test_hermite . . . . . 88
42.33 tri_assign_points . . . . . 88
42.34 triangulation_uniform . . . . . 88
42.35 vander_1d . . . . . 88
42.36 vanderd_1d . . . . . 88
42.37 vanderi_1d . . . . . 88

**43 numerical-methods/finite-volume/@Advection 89**
43.1 Advection . . . . . 89
43.2 dot_advection . . . . . 89

**44 numerical-methods/finite-volume/@Burgers 89**
44.1 burgers_split . . . . . 89
44.2 dot_burgers_fdm . . . . . 89
44.3 dot_burgers_fft . . . . . 89

**45 numerical-methods/finite-volume/@Finite_Volume 89**
45.1 Finite_Volume . . . . . 89
45.2 apply_bc . . . . . 90
45.3 solve . . . . . 90
45.4 step_split_strang . . . . . 90
45.5 step_unsplit . . . . . 90

**46 numerical-methods/finite-volume/@Flux_Limiter 90**
46.1 Flux_Limiter . . . . . 90
46.2 beam_warming . . . . . 90
46.3 fromm . . . . . 91
46.4 lax_wendroff . . . . . 91
46.5 minmod . . . . . 91
46.6 monotized_central . . . . . 91
46.7 muscl . . . . . 91
46.8 superbee . . . . . 91
46.9 upwind . . . . . 91
46.10 vanLeer . . . . . 92

**47 numerical-methods/finite-volume/@KDV** **92**
47.1 dot_kdv_fdm . . . . . . . . . . . . . . . . . . . . . . . . . 92
47.2 dot_kdv_fft . . . . . . . . . . . . . . . . . . . . . . . . . 92
47.3 kdv_split . . . . . . . . . . . . . . . . . . . . . . . . . . 92

**48 numerical-methods/finite-volume/@Reconstruct_Average_Evolve** **92**
48.1 Reconstruct_Average_Evolve . . . . . . . . . . . . . . . . . . 92
48.2 advect_highres . . . . . . . . . . . . . . . . . . . . . . . . 93
48.3 advect_lowress . . . . . . . . . . . . . . . . . . . . . . . . 93

**49 numerical-methods/finite-volume** **93**
49.1 Godunov . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
49.2 Lax_Friedrich . . . . . . . . . . . . . . . . . . . . . . . . . 93
49.3 Measure . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
49.4 Roe . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
49.5 fv_swe . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
49.6 staggered_euler . . . . . . . . . . . . . . . . . . . . . . . . 94
49.7 staggered_grid . . . . . . . . . . . . . . . . . . . . . . . . 94

**50 numerical-methods** **94**
50.1 grid2quad . . . . . . . . . . . . . . . . . . . . . . . . . . 94

**51 numerical-methods/integration** **94**
51.1 cumintL . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
51.2 cumintR . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
51.3 int_trapezoidal . . . . . . . . . . . . . . . . . . . . . . . . 94

**52 numerical-methods/interpolation/@Kriging** **95**
52.1 Kriging . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
52.2 estimate_semivariance . . . . . . . . . . . . . . . . . . . . . 95
52.3 interpolate_ . . . . . . . . . . . . . . . . . . . . . . . . . 95

**53 numerical-methods/interpolation/@RegularizedInterpolator1** **95**
53.1 RegularizedInterpolator1 . . . . . . . . . . . . . . . . . . . 95
53.2 init . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95

**54 numerical-methods/interpolation/@RegularizedInterpolator2** **96**
54.1 RegularizedInterpolator2 . . . . . . . . . . . . . . . . . . . 96
54.2 init . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96

**55 numerical-methods/interpolation/@RegularizedInterpolator3** **96**
55.1 RegularizedInterpolator3 . . . . . . . . . . . . . . . . . . . 96
55.2 init . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96

**56 numerical-methods/interpolation** **96**

56.1 IDW . . . 96
56.2 IPoly . . . 96
56.3 IRBM . . . 97
56.4 ISparse . . . 97
56.5 Inn . . . 97
56.6 Interpolator . . . 97
56.7 fixnan . . . 97
56.8 idw1 . . . 97
56.9 idw2 . . . 97
56.10 inner2outer . . . 98
56.11 inner2outer2 . . . 98
56.12 interp1_circular . . . 98
56.13 interp1_limited . . . 98
56.14 interp1_man . . . 98
56.15 interp1_piecewise_linear . . . 98
56.16 interp1_save . . . 98
56.17 interp1_slope . . . 99
56.18 interp1_smooth . . . 99
56.19 interp1_unique . . . 99
56.20 interp2_man . . . 99
56.21 interp_angle . . . 99
56.22 interp_fourier . . . 99
56.23 interp_fourier_batch . . . 99
56.24 interp_sn . . . 100
56.25 interp_sn2 . . . 100
56.26 interp_sn3 . . . 100
56.27 interp_sn_ . . . 100
56.28 limit_by_distance_1d . . . 100
56.29 resample1 . . . 100
56.30 resample_d_min . . . 100
56.31 resample_vector . . . 101
56.32 test_interp1_limited . . . 101

**57 numerical-methods** **101**
57.1 inverse_complex . . . 101
57.2 maccormack_step . . . 101
57.3 minmod . . . 101

**58 numerical-methods/multigrid** **101**
58.1 mg_interpolate . . . 101
58.2 mg_restrict . . . 101

**59 numerical-methods/ode/@BVPS_Characteristic** **101**
59.1 BVPS_Characteristic . . . 101

59.2 assemble1_A . . . . . . 102
59.3 assemble1_A_Q . . . . . . 102
59.4 assemble2_A . . . . . . 102
59.5 assemble_AA . . . . . . 102
59.6 assemble_AAA . . . . . . 102
59.7 assemble_Ic . . . . . . 102
59.8 bvp1c . . . . . . 102
59.9 check_arguments . . . . . . 102
59.10 couple_junctions . . . . . . 103
59.11 derivative . . . . . . 103
59.12 init . . . . . . 103
59.13 inner2outer_bvp2c . . . . . . 103
59.14 reconstruct . . . . . . 103
59.15 resample . . . . . . 103
59.16 solve . . . . . . 103
59.17 test_assemble1_A . . . . . . 104
59.18 test_assemble2_A . . . . . . 104

**60 numerical-methods/ode/@Time_Stepper 104**
60.1 Time_Stepper . . . . . . 104
60.2 solve . . . . . . 104

**61 numerical-methods/ode 104**
61.1 bvp2fdm . . . . . . 104
61.2 bvp2wavetrain . . . . . . 105
61.3 bvp2wavetwopass . . . . . . 105
61.4 ivp_euler_forward . . . . . . 105
61.5 ivp_euler_forward2 . . . . . . 105
61.6 ivprk2 . . . . . . 105
61.7 ode2_matrix . . . . . . 105
61.8 ode2characteristic . . . . . . 105
61.9 step_trapezoidal . . . . . . 106
61.10 test_bvp2 . . . . . . 106

**62 numerical-methods/optimisation 106**
62.1 aitken_iteration . . . . . . 106
62.2 anderson_iteration . . . . . . 106
62.3 armijo_stopping_criterion . . . . . . 106
62.4 astar . . . . . . 106
62.5 binsearch . . . . . . 106
62.6 bisection . . . . . . 106
62.7 box1 . . . . . . 107
62.8 box2 . . . . . . 107
62.9 cauchy . . . . . . 107

62.10 cauchy2 . . . 107
62.11 directional_derivative . . . 107
62.12 dud . . . 107
62.13 extreme3 . . . 107
62.14 extreme_quadratic . . . 108
62.15 ftest . . . 108
62.16 fzero_bisect . . . 108
62.17 fzero_newton . . . 108
62.18 grad . . . 108
62.19 hessian . . . 108
62.20 hessian_from_gradient . . . 109
62.21 hessian_projected . . . 109
62.22 line_search . . . 109
62.23 line_search2 . . . 109
62.24 line_search_polynomial . . . 109
62.25 line_search_polynomial2 . . . 110
62.26 line_search_quadratic . . . 110
62.27 line_search_quadratic2 . . . 110
62.28 line_search_wolfe . . . 110
62.29 ls_bgfs . . . 110
62.30 ls_broyden . . . 111
62.31 ls_generalized_secant . . . 111
62.32 nlcg . . . 111
62.33 nlls . . . 111
62.34 picard . . . 111
62.35 poly_extrema . . . 112
62.36 quadratic_function . . . 112
62.37 quadratic_programming . . . 112
62.38 quadratic_step . . . 112
62.39 rosenbrock . . . 112
62.40 sqrt_heron . . . 112
62.41 test_directional_derivative . . . 112
62.42 test_dud . . . 112
62.43 test_fzero_newton . . . 112
62.44 test_line_search_quadratic2 . . . 113
62.45 test_ls_generalized_secant . . . 113
62.46 test_nlcg_6_order . . . 113
62.47 test_nlls . . . 113

**63 numerical-methods/pde 113**
63.1 laplacian2d_fundamental_solution . . . 113

**64 numerical-methods/piecewise-polynomials 113**
64.1 Hermite1 . . . 113

64.2 hp2_fit . . . 113
64.3 hp2_predict . . . 114
64.4 hp_predict . . . 114
64.5 hp_regress . . . 114
64.6 lp_count . . . 114
64.7 lp_predict . . . 114
64.8 lp_regress . . . 114
64.9 lp_regress_ . . . 114

**65 numerical-methods 114**
65.1 test_adams_bashforth . . . 114

**66 patterns 115**
66.1 band_pattern . . . 115
66.2 hexagonal_pattern . . . 115

**67 regression/@PolyOLS 115**
67.1 PolyOLS . . . 115
67.2 coefftest . . . 115
67.3 detrend . . . 115
67.4 fit . . . 115
67.5 fit_ . . . 115
67.6 predict . . . 116
67.7 predict_ . . . 116
67.8 slope . . . 116

**68 regression/@PowerLS 116**
68.1 PowerLS . . . 116
68.2 fit . . . 116
68.3 predict . . . 116
68.4 predict_ . . . 116

**69 regression/@Theil 117**
69.1 Theil . . . 117
69.2 detrend . . . 117
69.3 fit . . . 117
69.4 predict . . . 117
69.5 slope . . . 117

**70 regression 117**
70.1 Theil_Multivariate . . . 117
70.2 areg . . . 118
70.3 ginireg . . . 118
70.4 hesssimplereg . . . 118
70.5 l1lin . . . 118

70.6 lsq_sparam . . . 118
70.7 polyfitd . . . 118
70.8 regression_method_of_moments . . . 119
70.9 robustlinreg . . . 119
70.10 theil2 . . . 119
70.11 theil_generalised . . . 119
70.12 total_least_squares . . . 119
70.13 weighted_median_regression . . . 119

**71 set-theory** **120**
71.1 issubset . . . 120

**72 mathematics** **120**
72.1 shuffle_index . . . 120

**73 signal-processing** **120**
73.1 acfar1 . . . 120
73.2 acfar1_2 . . . 120
73.3 acfar2 . . . 120
73.4 acfar2_2 . . . 121
73.5 ar1_cutoff_frequency . . . 121
73.6 ar1_effective_sample_size . . . 121
73.7 ar1_mse_mu_single_sample . . . 121
73.8 ar1_mse_pop . . . 121
73.9 ar1_mse_range . . . 121
73.10 ar1_spectrum . . . 121
73.11 ar1_to_tikhonov . . . 121
73.12 ar1_var_factor . . . 122
73.13 ar1_var_factor_ . . . 122
73.14 ar1_var_range2 . . . 122
73.15 ar1delay . . . 122
73.16 ar1delay_old . . . 122
73.17 ar2_acf2c . . . 122
73.18 ar2conv . . . 123
73.19 ar2dof . . . 123
73.20 ar2param . . . 123
73.21 asymwin . . . 123

**74 signal-processing/autocorr** **123**
74.1 autocorr2 . . . 123
74.2 autocorr_bandpass . . . 123
74.3 autocorr_brownian_phase . . . 123
74.4 autocorr_brownian_phase_2d . . . 123
74.5 autocorr_brownian_phase_across . . . 124

74.6 autocorr_decay_rate . . . . . . 124
74.7 autocorr_effective_sample_size . . . . . . 124
74.8 autocorr_fft . . . . . . 124
74.9 autocorr_genton . . . . . . 124
74.10 autocorr_highpass . . . . . . 124
74.11 autocorr_lowpass . . . . . . 124
74.12 autocorr_periodic_additive_noise . . . . . . 124
74.13 autocorr_periodic_windowed . . . . . . 124
74.14 autocorr_radial . . . . . . 125

**75 signal-processing 125**
75.1 average_wave_shape . . . . . . 125
75.2 bandpass . . . . . . 125
75.3 bandpass1d . . . . . . 125
75.4 bandpass1d_fft . . . . . . 125
75.5 bandpass1d_implicit . . . . . . 125
75.6 bandpass2 . . . . . . 125
75.7 bandpass2d . . . . . . 125
75.8 bandpass2d_2 . . . . . . 126
75.9 bandpass2d_fft . . . . . . 126
75.10 bandpass2d_ideal . . . . . . 126
75.11 bandpass2d_implicit . . . . . . 126
75.12 bandpass2d_iso . . . . . . 126
75.13 bandpass_arg . . . . . . 126
75.14 bandpass_f0_to_rho . . . . . . 126
75.15 bandpass_max . . . . . . 126
75.16 bandpass_max2 . . . . . . 126
75.17 bartlett . . . . . . 127
75.18 bin1d . . . . . . 127
75.19 bin2d . . . . . . 127
75.20 binormrnd . . . . . . 127
75.21 coherence . . . . . . 127
75.22 conv1_man . . . . . . 127
75.23 conv2_man . . . . . . 128
75.24 conv2z . . . . . . 128
75.25 conv30 . . . . . . 128
75.26 conv_ . . . . . . 128
75.27 conv_centered . . . . . . 128
75.28 convz . . . . . . 128
75.29 cosexpdelay . . . . . . 128
75.30 csmooth . . . . . . 128
75.31 daniell_window . . . . . . 129
75.32 danielle_window . . . . . . 129
75.33 db2neper . . . . . . 129

75.34 db2power . . . . . . . . . . . . . . . . . . . . . . . . . . . 129
75.35 derive_bandpass_normalization_and_zeros . . . . . . . . . . 129
75.36 derive_danielle_weight . . . . . . . . . . . . . . . . . . . . . 129
75.37 derive_limit_0_acfar . . . . . . . . . . . . . . . . . . . . . . 129
75.38 detect_peak . . . . . . . . . . . . . . . . . . . . . . . . . . 129
75.39 determine_phase_shift . . . . . . . . . . . . . . . . . . . . . 130
75.40 determine_phase_shift1 . . . . . . . . . . . . . . . . . . . . 130
75.41 digital_low_pass_filter . . . . . . . . . . . . . . . . . . . . . 130
75.42 doublesum_ij . . . . . . . . . . . . . . . . . . . . . . . . . . 130
75.43 effective_sample_size_to_ar1 . . . . . . . . . . . . . . . . . . 130
75.44 filt_hodges_lehman . . . . . . . . . . . . . . . . . . . . . . . 130
75.45 filter1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 130
75.46 filter2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 130
75.47 filter_ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 131
75.48 filter_r_to_f0 . . . . . . . . . . . . . . . . . . . . . . . . . 131
75.49 filter_rho_to_f0 . . . . . . . . . . . . . . . . . . . . . . . . 131
75.50 filter_twosided . . . . . . . . . . . . . . . . . . . . . . . . . 131
75.51 filteriir . . . . . . . . . . . . . . . . . . . . . . . . . . . . 131
75.52 filterp . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 132
75.53 filterp1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 132
75.54 filterstd . . . . . . . . . . . . . . . . . . . . . . . . . . . . 132
75.55 firls_man . . . . . . . . . . . . . . . . . . . . . . . . . . . . 132
75.56 fit_spectral_density . . . . . . . . . . . . . . . . . . . . . . 132
75.57 fit_spectral_density_2d . . . . . . . . . . . . . . . . . . . . . 132
75.58 fit_spectral_density_radial . . . . . . . . . . . . . . . . . . . 132
75.59 flattopwin . . . . . . . . . . . . . . . . . . . . . . . . . . . 133
75.60 frequency_response_boxcar . . . . . . . . . . . . . . . . . . . 133
75.61 freqz_boxcar . . . . . . . . . . . . . . . . . . . . . . . . . . 133
75.62 gaussfilt1 . . . . . . . . . . . . . . . . . . . . . . . . . . . 133
75.63 hanchangewin . . . . . . . . . . . . . . . . . . . . . . . . . . 133
75.64 hanchangewin2 . . . . . . . . . . . . . . . . . . . . . . . . . 133
75.65 hanwin . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 133
75.66 hanwin_ . . . . . . . . . . . . . . . . . . . . . . . . . . . . 133
75.67 high_pass_1d_simple . . . . . . . . . . . . . . . . . . . . . . 134
75.68 highpass . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134
75.69 highpass1d_fft_cos . . . . . . . . . . . . . . . . . . . . . . . 134
75.70 highpass1d_implicit . . . . . . . . . . . . . . . . . . . . . . 134
75.71 highpass2d_fft . . . . . . . . . . . . . . . . . . . . . . . . . 134
75.72 highpass2d_ideal . . . . . . . . . . . . . . . . . . . . . . . . 134
75.73 highpass2d_implicit . . . . . . . . . . . . . . . . . . . . . . 134
75.74 highpass_arg . . . . . . . . . . . . . . . . . . . . . . . . . . 134
75.75 highpass_fc_to_rho . . . . . . . . . . . . . . . . . . . . . . . 134
75.76 jackknife_block . . . . . . . . . . . . . . . . . . . . . . . . . 135
75.77 kaiserwin . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135

75.78 kalman . . . . . . . . . . . . . . . . . . . . . . . . . . 135
75.79 lanczoswin . . . . . . . . . . . . . . . . . . . . . . . . 135
75.80 last . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
75.81 lowpass . . . . . . . . . . . . . . . . . . . . . . . . . . 135
75.82 lowpass1d_fft . . . . . . . . . . . . . . . . . . . . . . . 135
75.83 lowpass1d_implicit . . . . . . . . . . . . . . . . . . . . 135
75.84 lowpass2 . . . . . . . . . . . . . . . . . . . . . . . . . 135
75.85 lowpass2d_2 . . . . . . . . . . . . . . . . . . . . . . . 136
75.86 lowpass2d_anisotropic . . . . . . . . . . . . . . . . . . 136
75.87 lowpass2d_fft . . . . . . . . . . . . . . . . . . . . . . . 136
75.88 lowpass2d_ideal . . . . . . . . . . . . . . . . . . . . . 136
75.89 lowpass2d_implicit . . . . . . . . . . . . . . . . . . . . 136
75.90 lowpass_arg . . . . . . . . . . . . . . . . . . . . . . . . 136
75.91 lowpass_fc_to_rho . . . . . . . . . . . . . . . . . . . . 136
75.92 lowpass_iir . . . . . . . . . . . . . . . . . . . . . . . . 136
75.93 lowpass_iir_symmetric . . . . . . . . . . . . . . . . . . 136
75.94 lowpassfilter2 . . . . . . . . . . . . . . . . . . . . . . . 137
75.95 maxfilt1 . . . . . . . . . . . . . . . . . . . . . . . . . 137
75.96 meanfilt1 . . . . . . . . . . . . . . . . . . . . . . . . . 137
75.97 meanfilt2 . . . . . . . . . . . . . . . . . . . . . . . . . 137
75.98 medfilt1_man . . . . . . . . . . . . . . . . . . . . . . . 137
75.99 medfilt1_man2 . . . . . . . . . . . . . . . . . . . . . . 137
75.100 medfilt1_padded . . . . . . . . . . . . . . . . . . . . . 137
75.101 medfilt1_reduced . . . . . . . . . . . . . . . . . . . . . 137
75.102 mid_term_single_sample . . . . . . . . . . . . . . . . . . 138
75.103 minfilt1 . . . . . . . . . . . . . . . . . . . . . . . . . 138
75.104 mu2ar1 . . . . . . . . . . . . . . . . . . . . . . . . . . 138
75.105 mysmooth . . . . . . . . . . . . . . . . . . . . . . . . . 138
75.106 nanautocorr . . . . . . . . . . . . . . . . . . . . . . . . 138
75.107 nanmedfilt1 . . . . . . . . . . . . . . . . . . . . . . . . 138
75.108 neper2db . . . . . . . . . . . . . . . . . . . . . . . . . 138
75.109 oscillator_noisy . . . . . . . . . . . . . . . . . . . . . . 138
75.110 peaks_man . . . . . . . . . . . . . . . . . . . . . . . . 139
75.111 periodogram . . . . . . . . . . . . . . . . . . . . . . . . 139
75.112 periodogram_2d . . . . . . . . . . . . . . . . . . . . . . 139
75.113 periodogram_annular . . . . . . . . . . . . . . . . . . . 139
75.114 periodogram_bartlett . . . . . . . . . . . . . . . . . . . 139
75.115 periodogram_bootstrap . . . . . . . . . . . . . . . . . . 139
75.116 periodogram_confidence_interval . . . . . . . . . . . . . 139
75.117 periodogram_median . . . . . . . . . . . . . . . . . . . 139
75.118 periodogram_p_value . . . . . . . . . . . . . . . . . . . 139
75.119 periodogram_qq . . . . . . . . . . . . . . . . . . . . . . 140
75.120 periodogram_quantiles . . . . . . . . . . . . . . . . . . . 140
75.121 periodogram_radial . . . . . . . . . . . . . . . . . . . . 140

75.122 periodogram_std . . . . . . . . . . . . . . . . . . . . . . . . 140
75.123 periodogram_test_periodicity . . . . . . . . . . . . . . . . . . 140
75.124 periodogram_test_periodicity_2d . . . . . . . . . . . . . . . . 141
75.125 periodogram_test_stationarity . . . . . . . . . . . . . . . . . 141
75.126 periodogram_welsh . . . . . . . . . . . . . . . . . . . . . . . 141
75.127 polyfilt1 . . . . . . . . . . . . . . . . . . . . . . . . . . . 141
75.128 qmedfilt1 . . . . . . . . . . . . . . . . . . . . . . . . . . . 141
75.129 radial_window . . . . . . . . . . . . . . . . . . . . . . . . . 142
75.130 randar1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142
75.131 randar1_dual . . . . . . . . . . . . . . . . . . . . . . . . . 142
75.132 randar2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142
75.133 randarp . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142
75.134 range_window . . . . . . . . . . . . . . . . . . . . . . . . . 142
75.135 rectwin . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142
75.136 recursive_sum . . . . . . . . . . . . . . . . . . . . . . . . . 142
75.137 select_range . . . . . . . . . . . . . . . . . . . . . . . . . 143
75.138 smooth1d_parametric . . . . . . . . . . . . . . . . . . . . . . 143
75.139 smooth2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 143
75.140 smooth_man . . . . . . . . . . . . . . . . . . . . . . . . . . 143
75.141 smooth_parametric . . . . . . . . . . . . . . . . . . . . . . . 143
75.142 smooth_parametric2 . . . . . . . . . . . . . . . . . . . . . . 143
75.143 smooth_with_splines . . . . . . . . . . . . . . . . . . . . . . 143
75.144 smoothfft . . . . . . . . . . . . . . . . . . . . . . . . . . . 143

**76 signal-processing/spectral-density 144**
76.1 spectral_density_ar2 . . . . . . . . . . . . . . . . . . . . . . 144
76.2 spectral_density_area . . . . . . . . . . . . . . . . . . . . . 144
76.3 spectral_density_bandpass2d_ideal . . . . . . . . . . . . . . . 144
76.4 spectral_density_bandpass_2d . . . . . . . . . . . . . . . . . 144
76.5 spectral_density_bandpass_2d_scale . . . . . . . . . . . . . . 144
76.6 spectral_density_bandpass_2d_scale_old . . . . . . . . . . . . 144
76.7 spectral_density_bandpass_continuous . . . . . . . . . . . . . 144
76.8 spectral_density_bandpass_continuous_max . . . . . . . . . . . 145
76.9 spectral_density_bandpass_continuous_max2par . . . . . . . . . 145
76.10 spectral_density_bandpass_continuous_scale . . . . . . . . . . 145
76.11 spectral_density_bandpass_discrete . . . . . . . . . . . . . . 145
76.12 spectral_density_brownian_phase . . . . . . . . . . . . . . . 145
76.13 spectral_density_brownian_phase_2d . . . . . . . . . . . . . . 145
76.14 spectral_density_brownian_phase_across . . . . . . . . . . . . 145
76.15 spectral_density_brownian_phase_mode . . . . . . . . . . . . . 145
76.16 spectral_density_brownian_phase_mode2par . . . . . . . . . . . 146
76.17 spectral_density_brownian_phase_scale . . . . . . . . . . . . . 146
76.18 spectral_density_estimate_2d . . . . . . . . . . . . . . . . . 146
76.19 spectral_density_flat . . . . . . . . . . . . . . . . . . . . . 146

76.20 spectral_density_highpass . . . . . 146
76.21 spectral_density_highpass2d_ideal . . . . . 146
76.22 spectral_density_highpass_2d . . . . . 146
76.23 spectral_density_highpass_cos . . . . . 146
76.24 spectral_density_lorentzian . . . . . 146
76.25 spectral_density_lorentzian_max . . . . . 147
76.26 spectral_density_lorentzian_max2par . . . . . 147
76.27 spectral_density_lorentzian_scale . . . . . 147
76.28 spectral_density_lowpass . . . . . 147
76.29 spectral_density_lowpass2d_ideal . . . . . 147
76.30 spectral_density_lowpass_2d . . . . . 147
76.31 spectral_density_lowpass_one_sided . . . . . 147
76.32 spectral_density_periodic_additive_noise . . . . . 147
76.33 spectral_density_wperiodic . . . . . 147

**77 signal-processing 148**

77.1 spectrogram . . . . . 148
77.2 std_window . . . . . 148
77.3 sum_i_lag . . . . . 148
77.4 sum_ii . . . . . 148
77.5 sum_ii_ . . . . . 148
77.6 sum_ij . . . . . 148
77.7 sum_ij_ . . . . . 148
77.8 sum_ij_partial_ . . . . . 149
77.9 sum_multivar . . . . . 149
77.10 test_acfar1 . . . . . 149
77.11 test_acfar1_2 . . . . . 149
77.12 test_acfar1_3 . . . . . 149
77.13 test_acfar1_4 . . . . . 149
77.14 test_acfar2 . . . . . 149
77.15 test_ar1_var_factor . . . . . 149
77.16 test_ar1_var_factor_2 . . . . . 149
77.17 test_ar1_var_mu_single_sample . . . . . 149
77.18 test_ar1_var_pop . . . . . 150
77.19 test_ar1_var_pop_1 . . . . . 150
77.20 test_ar1delay . . . . . 150
77.21 test_bivariate_covariance_term . . . . . 150
77.22 test_convexity . . . . . 150
77.23 test_lanczoswin . . . . . 150
77.24 test_madcorr . . . . . 150
77.25 test_randar1 . . . . . 150
77.26 test_randar1_multivariate . . . . . 150
77.27 test_randar2 . . . . . 150
77.28 test_sum_ij . . . . . 151

77.29 test_sum_multivar . . . . . . . . . . . . . . . . . . . . . . . . 151
77.30 test_trifilt1 . . . . . . . . . . . . . . . . . . . . . . . . . . . 151
77.31 test_wautocorr . . . . . . . . . . . . . . . . . . . . . . . . . . 151
77.32 test_wavelet_transform . . . . . . . . . . . . . . . . . . . . . 151
77.33 test_wordfilt . . . . . . . . . . . . . . . . . . . . . . . . . . 151
77.34 test_xar1_mid_term . . . . . . . . . . . . . . . . . . . . . . . 151
77.35 tikhonov_to_ar1 . . . . . . . . . . . . . . . . . . . . . . . . . 151
77.36 trapwin . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 151
77.37 trifilt1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
77.38 trifilt2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
77.39 triwin . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
77.40 triwin2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
77.41 varar1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
77.42 welch_spectrogram . . . . . . . . . . . . . . . . . . . . . . . . 152
77.43 wfilt . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
77.44 winbandpass . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
77.45 window2d . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
77.46 window_make_odd . . . . . . . . . . . . . . . . . . . . . . . . . 153
77.47 winfilt0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
77.48 winlength . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
77.49 wmeanfilt . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
77.50 wmedfilt . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
77.51 wordfilt . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
77.52 wordfilt_edgeworth . . . . . . . . . . . . . . . . . . . . . . . . 154
77.53 wrapphase . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
77.54 xar1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
77.55 xcorr_man . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154

**78 sorting 154**
78.1 sort2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
78.2 sort2d . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154

**79 special-functions 154**
79.1 bessel_sphere . . . . . . . . . . . . . . . . . . . . . . . . . . 154
79.2 digamma_man . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
79.3 exp10 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
79.4 hankel_sphere . . . . . . . . . . . . . . . . . . . . . . . . . . 155
79.5 hermite . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
79.6 legendre_man . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
79.7 neumann_sphere . . . . . . . . . . . . . . . . . . . . . . . . . . 155

**80 statistics 155**
80.1 atan_s2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
80.2 beta_kurt . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 156

80.3 beta_mean . . . . . . . . . . 156
80.4 beta_mode_to_parameter . . . . . . . . . . 156
80.5 beta_skew . . . . . . . . . . 156
80.6 beta_std . . . . . . . . . . 156
80.7 chi2_kurt . . . . . . . . . . 156
80.8 chi2_mean . . . . . . . . . . 156
80.9 chi2_skew . . . . . . . . . . 156
80.10 chi2_std . . . . . . . . . . 156
80.11 coefficient_of_determination . . . . . . . . . . 156
80.12 conditional_expectation_normal . . . . . . . . . . 157
80.13 correlation_confidence_pearson . . . . . . . . . . 157

**81 statistics/distributions 157**
81.1 PDF . . . . . . . . . . 157
81.2 binorm_separation_coefficient . . . . . . . . . . 157
81.3 binormcdf . . . . . . . . . . 157
81.4 binormfit . . . . . . . . . . 157
81.5 binormpdf . . . . . . . . . . 157
81.6 edgeworth_cdf . . . . . . . . . . 157
81.7 edgeworth_pdf . . . . . . . . . . 158
81.8 gam_moment2param . . . . . . . . . . 158
81.9 logn_mean . . . . . . . . . . 158
81.10 logn_mode . . . . . . . . . . 158
81.11 logn_moment2param . . . . . . . . . . 158
81.12 logn_param2moment . . . . . . . . . . 158
81.13 logn_std . . . . . . . . . . 158
81.14 lognpdf_ . . . . . . . . . . 158
81.15 pdfsample . . . . . . . . . . 159
81.16 t2cdf . . . . . . . . . . 159
81.17 t2inv . . . . . . . . . . 159

**82 statistics 159**
82.1 example_standard_error_of_sample_quantiles . . . . . . . . . . 159
82.2 f_var_finite . . . . . . . . . . 159
82.3 fisher_mean . . . . . . . . . . 159
82.4 fisher_std . . . . . . . . . . 159
82.5 gam_mean . . . . . . . . . . 159
82.6 gam_std . . . . . . . . . . 160
82.7 gamma_mode_to_parameter . . . . . . . . . . 160
82.8 gamma_stirling . . . . . . . . . . 160
82.9 gaussfit3 . . . . . . . . . . 160
82.10 gaussfit_quantile . . . . . . . . . . 160
82.11 geoserr . . . . . . . . . . 160
82.12 geostd . . . . . . . . . . 160

82.13 hodges_lehmann_correlation . . . . . . . . . . . . . . . . . . 160
82.14 hodges_lehmann_dispersion . . . . . . . . . . . . . . . . . . . 161

**83 statistics/information-theory 161**
83.1 akaike_information_criterion . . . . . . . . . . . . . . . . . . 161
83.2 bayesian_information_criterion . . . . . . . . . . . . . . . . . 161

**84 statistics 161**
84.1 kurtncdf . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
84.2 kurtnpdf . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
84.3 kurtosis_bias_corrected . . . . . . . . . . . . . . . . . . . . 162
84.4 limit . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
84.5 logfactorial . . . . . . . . . . . . . . . . . . . . . . . . . . 162
84.6 loglogpdf . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
84.7 lognfit_quantile . . . . . . . . . . . . . . . . . . . . . . . . 162
84.8 logskewcdf . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
84.9 logskewpdf . . . . . . . . . . . . . . . . . . . . . . . . . . . 162

**85 statistics/logu 162**
85.1 lambertw_numeric . . . . . . . . . . . . . . . . . . . . . . . 162
85.2 logtrialtcdf . . . . . . . . . . . . . . . . . . . . . . . . . . 163
85.3 logtrialtinv . . . . . . . . . . . . . . . . . . . . . . . . . . 163
85.4 logtrialtmean . . . . . . . . . . . . . . . . . . . . . . . . . . 163
85.5 logtrialtpdf . . . . . . . . . . . . . . . . . . . . . . . . . . 163
85.6 logtrialtrnd . . . . . . . . . . . . . . . . . . . . . . . . . . 163
85.7 logtricdf . . . . . . . . . . . . . . . . . . . . . . . . . . . 163
85.8 logtriinv . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
85.9 logtrimean . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
85.10 logtripdf . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
85.11 logtrirnd . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
85.12 logucdf . . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
85.13 logucm . . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
85.14 loguinv . . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
85.15 logumean . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
85.16 logupdf . . . . . . . . . . . . . . . . . . . . . . . . . . . . 165
85.17 logurnd . . . . . . . . . . . . . . . . . . . . . . . . . . . . 165
85.18 loguvar . . . . . . . . . . . . . . . . . . . . . . . . . . . . 165
85.19 medlogu . . . . . . . . . . . . . . . . . . . . . . . . . . . . 165
85.20 test_logurnd . . . . . . . . . . . . . . . . . . . . . . . . . 165
85.21 tricdf . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 165
85.22 triinv . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 165
85.23 trimedian . . . . . . . . . . . . . . . . . . . . . . . . . . . 165
85.24 tripdf . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 166
85.25 trirnd . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 166

**86 statistics** **166**
86.1 max_exprnd . . . 166
86.2 maxnnormals . . . 166
86.3 mean_generalized_gampdf . . . 166
86.4 midrange . . . 166
86.5 minavg . . . 166
86.6 mode_man . . . 167

**87 statistics/moment-statistics** **167**
87.1 autocorr_man3 . . . 167
87.2 autocorr_man4 . . . 167
87.3 autocorr_man5 . . . 167
87.4 blockserr . . . 167
87.5 comoment . . . 167
87.6 corr_man . . . 168
87.7 cov_man . . . 168
87.8 dof . . . 168
87.9 edgeworth_quantile . . . 168
87.10 effective_sample_size . . . 168
87.11 f_correlation . . . 169
87.12 f_finite . . . 169
87.13 lmean . . . 169
87.14 lmoment . . . 169
87.15 maskmean . . . 169
87.16 masknanmean . . . 169
87.17 mean1 . . . 169
87.18 mean_man . . . 169
87.19 mse . . . 170
87.20 nanautocorr_man1 . . . 170
87.21 nanautocorr_man2 . . . 170
87.22 nanautocorr_man4 . . . 170
87.23 nancorr . . . 170
87.24 nancumsum . . . 170
87.25 nanlmean . . . 170
87.26 nanr2 . . . 171
87.27 nanrms . . . 171
87.28 nanrmse . . . 171
87.29 nanserr . . . 171
87.30 nanwmean . . . 171
87.31 nanwstd . . . 171
87.32 nanwvar . . . 171
87.33 nanxcorr . . . 172
87.34 pearson . . . 172
87.35 pearson_to_kendall . . . 172

87.36 pool_samples . . . 172
87.37 qmean . . . 172
87.38 range_mean . . . 172
87.39 rmse_ . . . 172
87.40 serr . . . 172
87.41 serr1 . . . 173
87.42 test_qskew . . . 173
87.43 test_qstd_qskew_optimal_p . . . 173
87.44 wautocorr . . . 173
87.45 wcorr . . . 173
87.46 wcov . . . 173
87.47 wdof . . . 173
87.48 wkurt . . . 174
87.49 wmean . . . 174
87.50 wrms . . . 174
87.51 wserr . . . 174
87.52 wskew . . . 174
87.53 wstd . . . 174
87.54 wvar . . . 174

**88 statistics** **175**
88.1 nangeomean . . . 175
88.2 nangeostd . . . 175

**89 statistics/nonparametric-statistics** **175**
89.1 kernel1d . . . 175
89.2 kernel2d . . . 175

**90 statistics** **175**
90.1 normmoment . . . 175
90.2 normpdf2 . . . 176

**91 statistics/order-statistics** **176**
91.1 hodges_lehmann_location . . . 176
91.2 kendall . . . 176
91.3 kendall_to_pearson . . . 176
91.4 mad2sd . . . 176
91.5 madcorr . . . 176
91.6 median2_holder . . . 177
91.7 median_ci . . . 177
91.8 median_man . . . 177
91.9 mediani . . . 177
91.10 nanmadcorr . . . 177
91.11 nanwmedian . . . 177

91.12 nanwquantile . . . . . . . . . . 177
91.13 oja_median . . . . . . . . . . 178
91.14 qkurtosis . . . . . . . . . . 178
91.15 qmoments . . . . . . . . . . 178
91.16 qskew . . . . . . . . . . 178
91.17 qskewq . . . . . . . . . . 178
91.18 qstdq . . . . . . . . . . 179
91.19 quantile1_optimisation . . . . . . . . . . 179
91.20 quantile2_breckling . . . . . . . . . . 179
91.21 quantile2_chaudhuri . . . . . . . . . . 179
91.22 quantile2_projected . . . . . . . . . . 179
91.23 quantile2_projected2 . . . . . . . . . . 179
91.24 quantile_envelope . . . . . . . . . . 179
91.25 quantile_regression_simple . . . . . . . . . . 179
91.26 ranking . . . . . . . . . . 180
91.27 spatial_median . . . . . . . . . . 180
91.28 spatial_quantile . . . . . . . . . . 180
91.29 spatial_quantile2 . . . . . . . . . . 180
91.30 spatial_quantile3 . . . . . . . . . . 180
91.31 spatial_rank . . . . . . . . . . 180
91.32 spatial_sign . . . . . . . . . . 180
91.33 spatial_signed_rank . . . . . . . . . . 180
91.34 spearman . . . . . . . . . . 181
91.35 spearman_rank . . . . . . . . . . 181
91.36 spearman_to_pearson . . . . . . . . . . 181
91.37 wmedian . . . . . . . . . . 181
91.38 wquantile . . . . . . . . . . 181

**92 statistics** **181**
92.1 qstd . . . . . . . . . . 181
92.2 quantile_extrap . . . . . . . . . . 181
92.3 quantile_sin . . . . . . . . . . 181

**93 statistics/random-number-generation** **182**
93.1 laplacernd . . . . . . . . . . 182
93.2 randc . . . . . . . . . . 182
93.3 skewness2param . . . . . . . . . . 182
93.4 skewpdf_central_moments . . . . . . . . . . 182
93.5 skewrnd . . . . . . . . . . 182
93.6 skewrnd2 . . . . . . . . . . 182

**94 statistics** **182**
94.1 range . . . . . . . . . . 182
94.2 resample_with_replacement . . . . . . . . . . 182

**95 statistics/resampling-statistics/@Jackknife 183**
95.1 Jackknife . . . 183
95.2 estimated_STATIC . . . 183
95.3 matrix1_STATIC . . . 183
95.4 matrix2 . . . 183

**96 statistics/resampling-statistics 184**
96.1 block_jackknife . . . 184
96.2 jackknife_moments . . . 184
96.3 moving_block_jackknife . . . 184
96.4 randblockserr . . . 184
96.5 resample . . . 184

**97 statistics 185**
97.1 scale_quantile_sd . . . 185
97.2 sd_sample_quantiles . . . 185
97.3 skew_generalized_normal_fit . . . 185
97.4 skew_generalized_normpdf . . . 185
97.5 skewcdf . . . 185
97.6 skewparam_to_central_moments . . . 185
97.7 skewpdf . . . 185
97.8 spatialrnd . . . 186
97.9 test_mean_generalized_gampdf . . . 186
97.10 test_skew_generalized_normpdf . . . 186
97.11 trimmed_mean . . . 186
97.12 ttest2_man . . . 186
97.13 ttest_man . . . 186
97.14 ttest_paired . . . 186
97.15 wgeomean . . . 187
97.16 wgeovar . . . 187
97.17 wharmean . . . 187
97.18 wharstd . . . 187
97.19 wharvar . . . 187
97.20 wnormpdf . . . 187

**98 stochastic 187**
98.1 brownian_noise_1d_acf . . . 187
98.2 brownian_noise_1d_fft . . . 187
98.3 brownian_noise_2d_fft . . . 188

**99 mathematics 188**
99.1 ternary_diagram . . . 188

**100 test/master 188**

100.1 dat_test_lanczos_3d_k_20_n_40 . . . . . . . . . . . . . . . . . 188
100.2 poisson2d_blk . . . . . . . . . . . . . . . . . . . . . . . . 188
100.3 qr_implicit_givens_2 . . . . . . . . . . . . . . . . . . . . . 188
100.4 spectral_derivative_2d . . . . . . . . . . . . . . . . . . . . 188
100.5 test_2d_eigensolver_hydrogen . . . . . . . . . . . . . . . . . 188
100.6 test_2d_refine . . . . . . . . . . . . . . . . . . . . . . . . 188
100.7 test_3d_eigensolver_hydrogen . . . . . . . . . . . . . . . . . 189
100.8 test_FEM . . . . . . . . . . . . . . . . . . . . . . . . . . . 189
100.9 test_Mesh_3d . . . . . . . . . . . . . . . . . . . . . . . . . 189
100.10 test_arnoldi . . . . . . . . . . . . . . . . . . . . . . . . . 189
100.11 test_arpackc . . . . . . . . . . . . . . . . . . . . . . . . . 189
100.12 test_assemble . . . . . . . . . . . . . . . . . . . . . . . . 189
100.13 test_assembly_performance . . . . . . . . . . . . . . . . . . 189
100.14 test_bc_one_sided . . . . . . . . . . . . . . . . . . . . . . 189
100.15 test_compare_solvers . . . . . . . . . . . . . . . . . . . . . 189
100.16 test_complete . . . . . . . . . . . . . . . . . . . . . . . . 189
100.17 test_convergence . . . . . . . . . . . . . . . . . . . . . . . 190
100.18 test_convergence_b . . . . . . . . . . . . . . . . . . . . . . 190
100.19 test_df_2d . . . . . . . . . . . . . . . . . . . . . . . . . . 190
100.20 test_eig_algs . . . . . . . . . . . . . . . . . . . . . . . . 190
100.21 test_eig_inverse . . . . . . . . . . . . . . . . . . . . . . . 190
100.22 test_eigs_lanczos . . . . . . . . . . . . . . . . . . . . . . 190
100.23 test_eigs_lanczos_1 . . . . . . . . . . . . . . . . . . . . . 190
100.24 test_eigs_lanczos_2 . . . . . . . . . . . . . . . . . . . . . 190
100.25 test_eigs_lanczos_performance . . . . . . . . . . . . . . . . 190
100.26 test_fdm . . . . . . . . . . . . . . . . . . . . . . . . . . . 190
100.27 test_fdm_d_vargrid . . . . . . . . . . . . . . . . . . . . . . 191
100.28 test_fdm_spectral . . . . . . . . . . . . . . . . . . . . . . 191
100.29 test_fem . . . . . . . . . . . . . . . . . . . . . . . . . . . 191
100.30 test_fem_1d . . . . . . . . . . . . . . . . . . . . . . . . . 191
100.31 test_fem_1d_higher_order . . . . . . . . . . . . . . . . . . . 191
100.32 test_fem_2d_adaptive . . . . . . . . . . . . . . . . . . . . . 191
100.33 test_fem_2d_higher_order . . . . . . . . . . . . . . . . . . . 191
100.34 test_fem_3d_higher_order . . . . . . . . . . . . . . . . . . . 191
100.35 test_fem_3d_refine . . . . . . . . . . . . . . . . . . . . . . 191
100.36 test_fem_b . . . . . . . . . . . . . . . . . . . . . . . . . . 191
100.37 test_fem_derivative . . . . . . . . . . . . . . . . . . . . . 192
100.38 test_fem_quadrature . . . . . . . . . . . . . . . . . . . . . 192
100.39 test_final . . . . . . . . . . . . . . . . . . . . . . . . . . 192
100.40 test_fix_substitution . . . . . . . . . . . . . . . . . . . . 192
100.41 test_forward . . . . . . . . . . . . . . . . . . . . . . . . . 192
100.42 test_get_sparse_arrays . . . . . . . . . . . . . . . . . . . . 192
100.43 test_harmonic_oscillator . . . . . . . . . . . . . . . . . . . 192
100.44 test_high_order_fdm_periodic_bc . . . . . . . . . . . . . . . 192

100.45 test_hydrogen_wf . . . . . . . . . . . . . . . . . . . . . . . . . 192
100.46 test_ichol . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
100.47 test_interpolation . . . . . . . . . . . . . . . . . . . . . . . . . 193
100.48 test_inverse_problem . . . . . . . . . . . . . . . . . . . . . . . . 193
100.49 test_it_vs_exact . . . . . . . . . . . . . . . . . . . . . . . . . . 193
100.50 test_jama . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
100.51 test_jd . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
100.52 test_jdqz . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
100.53 test_lanczos_2 . . . . . . . . . . . . . . . . . . . . . . . . . . 193
100.54 test_lanczos_biorthogonal . . . . . . . . . . . . . . . . . . . . . 193
100.55 test_laplacian . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
100.56 test_laplacian_non_uniform . . . . . . . . . . . . . . . . . . . . 193
100.57 test_laplacian_simple . . . . . . . . . . . . . . . . . . . . . . . 194
100.58 test_mesh_2d_uniform . . . . . . . . . . . . . . . . . . . . . . . 194
100.59 test_mesh_2d_uniform_2 . . . . . . . . . . . . . . . . . . . . . . 194
100.60 test_mesh_circle . . . . . . . . . . . . . . . . . . . . . . . . . . 194
100.61 test_mesh_generation . . . . . . . . . . . . . . . . . . . . . . . 194
100.62 test_mesh_interpolate . . . . . . . . . . . . . . . . . . . . . . . 194
100.63 test_mg . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
100.64 test_minres_recycle . . . . . . . . . . . . . . . . . . . . . . . . 194
100.65 test_multigrid . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
100.66 test_nc . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
100.67 test_nonuniform_symmetric . . . . . . . . . . . . . . . . . . . . . 195
100.68 test_pde . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195
100.69 test_permutation . . . . . . . . . . . . . . . . . . . . . . . . . . 195
100.70 test_poison_fem . . . . . . . . . . . . . . . . . . . . . . . . . . 195
100.71 test_polar . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195
100.72 test_potential . . . . . . . . . . . . . . . . . . . . . . . . . . . 195
100.73 test_powers . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195
100.74 test_precondition . . . . . . . . . . . . . . . . . . . . . . . . . 195
100.75 test_project_rectangle . . . . . . . . . . . . . . . . . . . . . . . 195
100.76 test_qr . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195
100.77 test_quantum_well . . . . . . . . . . . . . . . . . . . . . . . . . 196
100.78 test_radial_adaptive . . . . . . . . . . . . . . . . . . . . . . . . 196
100.79 test_radial_confinement . . . . . . . . . . . . . . . . . . . . . . 196
100.80 test_radial_fixes . . . . . . . . . . . . . . . . . . . . . . . . . . 196
100.81 test_refine_2d . . . . . . . . . . . . . . . . . . . . . . . . . . . 196
100.82 test_refine_2d_b . . . . . . . . . . . . . . . . . . . . . . . . . . 196
100.83 test_refine_3d . . . . . . . . . . . . . . . . . . . . . . . . . . . 196
100.84 test_refine_structural . . . . . . . . . . . . . . . . . . . . . . . 196
100.85 test_regularisation . . . . . . . . . . . . . . . . . . . . . . . . . 196
100.86 test_round_off . . . . . . . . . . . . . . . . . . . . . . . . . . . 196
100.87 test_schrödinger_potentials . . . . . . . . . . . . . . . . . . . . 197
100.88 test_uniform_mesh . . . . . . . . . . . . . . . . . . . . . . . . . 197

100.89 test_vargrid . . . . . . . . . . . . . . . . . . . . . . . . . . 197

**101test** **197**

101.1 test_bandpass2d . . . . . . . . . . . . . . . . . . . . . . . . . 197
101.2 test_gaussfit3 . . . . . . . . . . . . . . . . . . . . . . . . . . 197
101.3 test_geoserr . . . . . . . . . . . . . . . . . . . . . . . . . . . 197
101.4 test_hexagonal_pattern . . . . . . . . . . . . . . . . . . . . . . 197
101.5 test_lognfit_quantile . . . . . . . . . . . . . . . . . . . . . . . 197
101.6 test_lowpass1d_implicit . . . . . . . . . . . . . . . . . . . . . . 197
101.7 test_lowpass2d_anisotropic . . . . . . . . . . . . . . . . . . . . 198
101.8 test_lowpass2d_fft . . . . . . . . . . . . . . . . . . . . . . . . 198
101.9 test_max_normal . . . . . . . . . . . . . . . . . . . . . . . . . . 198
101.10 test_moments_fourier_power . . . . . . . . . . . . . . . . . . . 198
101.11 test_mtimes3x3 . . . . . . . . . . . . . . . . . . . . . . . . . . 198

**102wavelet** **198**

102.1 contiuous_wavelet_transform . . . . . . . . . . . . . . . . . . . 198
102.2 cwt_man . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 198
102.3 example_wavelets . . . . . . . . . . . . . . . . . . . . . . . . . 198
102.4 phasewrap . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 199
102.5 test_cwt_man . . . . . . . . . . . . . . . . . . . . . . . . . . . . 199
102.6 test_phasewrap . . . . . . . . . . . . . . . . . . . . . . . . . . . 199
102.7 test_wavelet . . . . . . . . . . . . . . . . . . . . . . . . . . . . 199
102.8 test_wavelet2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 199
102.9 test_wavelet_analysis . . . . . . . . . . . . . . . . . . . . . . . 199
102.10 test_wavelet_reconstruct . . . . . . . . . . . . . . . . . . . . . 199
102.11 test_wtc . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 199
102.12 wavelet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 199
102.13 wavelet_reconstruct . . . . . . . . . . . . . . . . . . . . . . . . 200
102.14 wavelet_transform . . . . . . . . . . . . . . . . . . . . . . . . . 200

# 1 calendar

## 1.1 days_per_month

## 1.2 isnight

# 2 mathematics

```
mathematical functions of various kind
```

## 2.1 cast_byte_to_integer

```
cast byte to integer
```

# 3 complex-analysis

```
operations on complex numbers
```

## 3.1 complex_exp_product_im_im

```
 product of the imaginary part of two complex exponentials

the product has two frequency components

input :
      c : complex amplitudes
      o : frequencies
output :
      cp : amplitude of the product
      op : frequencies of the product
```

## 3.2 complex_exp_product_im_re

```
 product of the imaginary part of one and the real part of a second
 complex exponential

the product has two frequency components

input :
      c : complex amplitudes
      o : frequencies
output :
      cp : amplitude of the product
      op : frequencies of the product
```

## 3.3 complex_exp_product_re_im

```
the product has two frequency components

 product of the imaginary part of one and the real part of a second
 complex exponential

input :
      c : complex amplitudes
      o : frequencies
output :
      cp : amplitude of the product
      op : frequencies of the product
```

### 3.4 complex_exp_product_re_re

```
 product of the real part of two complex exponentials

re(c1 exp(io1x))*re(c2 exp(io2x)) =
      1/2*(    real(c1*c2*exp(i*(n1+n2)*o*x)) ...
            + real(conj(c1)*c2*exp(i*(n2-n1)*o*x)) )

the product has two frequency components

input :
      c : complex amplitudes
      o : frequencies
output :
      cp : amplitude of the product
      op : frequencies of the product
```

### 3.5 croots

```
nth-roots of a complex number

input:
c : complex number
n : order of root
   n must be rational, to obtain n solutions
   otherwise no finite set of solutions exists

r : roots of the complex number
```

### 3.6 root_complex

```
root of a complex number
```

### 3.7 test_imroots

## 4 derivation

```
derivation of several functions by means of symbolic computation
```

### 4.1 derive_acfar1

### 4.2 derive_ar1_spectral_density

### 4.3 derive_ar2param

### 4.4 derive_arc_length

### 4.5 derive_fourier_power

### 4.6 derive_fourier_power_exp

### 4.7 derive_laplacian_curvilinear

### 4.8 derive_laplacian_fourier_piecewise_linear

### 4.9 derive_logtripdf

### 4.10 derive_smooth1d_parametric

### 4.11 derive_spectral_density_bandpass_initial_condition

## 5 derivation/master

### 5.1 derive_bc_one_sided

### 5.2 derive_convergence

### 5.3 derive_error_fdm

### 5.4 derive_fdm_poly

### 5.5 derive_fdm_power

## 5.6 derive_fdm_taylor

## 5.7 derive_fdm_vargrid

## 5.8 derive_fem_2d_mass

## 5.9 derive_fem_error_2d

## 5.10 derive_fem_error_3d

## 5.11 derive_fem_sym_2d

## 5.12 derive_grid_constants

## 5.13 derive_interpolation

## 5.14 derive_laplacian

## 5.15 derive_limit

## 5.16 derive_nc_1d

## 5.17 derive_nc_1d_

## 5.18 derive_nc_2d

## 5.19 derive_nonuniform_symmetric

%

## 5.20 derive_richardson

## 5.21 derive_sum

## 5.22 nn

## 5.23 test_derive

## 5.24 test_derive_fdm_poly

## 5.25 test_filter

### 5.26 test_vargrid

## 6 derivation

derivation of several functions by means of symbolic computation

### 6.1 simplify_atan

symbolic simplification of the arcus tangent

## 7 mathematics

mathematical functions of various kind

### 7.1 entropy

## 8 finance

### 8.1 derive_skewrnd_walsh_paramter

### 8.2 gbm_cdf

### 8.3 gbm_fit

### 8.4 gbm_fit_old

### 8.5 gbm_inv

### 8.6 gbm_mean

### 8.7 gbm_median

### 8.8 gbm_pdf

### 8.9 gbm_simulate

### 8.10 gbm_skewness

### 8.11 gbm_std

### 8.12 gbm_transform_time_step

### 8.13 put_price_black_scholes

### 8.14 skewgbm_simulate

### 8.15 skewrnd_walsh

# 9 finance/test

### 9.1 test_gbm

### 9.2 test_gbm_pdf

### 9.3 test_skewrnd_walsh

# 10 fourier/@STFT

### 10.1 STFT

```
class for short time fourier transform

Note: the interval Ti should be set to at leat 2*max(T), as
    otherwise coefficients
      tend to oscillate in the presence of noise
Note: for convenience, the independent variable is labeled as time
    (t),
      but the independent variable is arbitrary, so it works
          likewise in space
```

### 10.2 itransform

```
inverse of the short time fourier transform
```

### 10.3 stft_

```
static wrapper for STFT
```

### 10.4 stftmat

```
transformation matrix for the short time fourier transform
```

### 10.5 transform

```
short time fourier transform
```

## 11 fourier

```
support and analysis functions both for the discrete (fast) fourier
   transform (dft/fft)
and continuous fourier analysis (fourier series)
```

### 11.1 amplitude_from_peak

```
amplitude and standard deviation of the amplitude of a frequency
    component
represented by a peak in the fourier domain
input :
h : peak height
w : peak width at half height

output:
a : amplitude in real space
s : standard deviation of the frequency (!)
```

### 11.2 dftmtx_man

```
fourier matrix in matlab style with a limited number of rows,
columns of higher frequencies are omitted

input :
n  : number of samples
nr : number of columns

output :
F : fourier matrix
```

## 11.3 example_fourier_window

## 11.4 fft_man

```
fast fourier transform for complex input data

input:
F : data in real space

output :

F : fourier transformation of F
```

## 11.5 fftsmooth

```
smooth the fourier transform and determine upper and lower bound
    confidence intervals

input :
f :
sfunc  : a smoothing function (for example fir convolution with
    rectangular window)
        returns filtered (mean) value and normalized fir window
nf     : window length
nsigma : number of standard deviations for confidnce intervals


output :
ff : filtered fourier transform
l : lower bound
u : upper bound
```

## 11.6 fix_fourier

```
fill gaps (missing data) by means of fourier extrapolation

fix periodic data series with fourier interpolation
longest gap should not exceed 1/2 of the shortest time span of
    interest (1/cutoff frequency)
 note: this limit equals the position of first side lobe of the ft
     of a rectangular window with gap length
```

## 11.7 fourier_axis

```
return axis of frequencies and periods for the discrete fourier
    transform
as computed by fft (matlab-style)

input:
X : sample locations (equal interval)
L : length of samples
n : number of samples

output :
f    : frequencies
T    : periods
mask : mask for 1/2 of the fourier transform
       (as both halves are complex conjugates)
N    : frequency id
```

## 11.8 fourier_axis_2d

## 11.9 fourier_cesaro_correction

## 11.10 fourier_coefficient_piecewise_linear

```
fourier series coefficients of a piecewise linear function
(not coefficient of discrete fourier transform)
function can be discontinuous between intervals
scales domain length to 2pi

input :
l,r : end points of piecewise linear function
lval, rval : values at end points
L : length of domain
n : number of samples/highest frequency

output :
a, b : coefficients for frequency components
```

### 11.11 fourier_coefficient_piecewise_linear_1

```
fourier series coefficients of a piecewise linear function
(not coefficient of discrete fourier transform)
function can be discontinuous between intervals
scales domain length to 2pi

input :
X : end points of piecewise linear function
Y : values at end points

output :
ab : coefficients for frequency components
```

### 11.12 fourier_coefficient_ramp3

```
fourier series coefficient of a ramp
```

### 11.13 fourier_coefficient_ramp_pulse

```
fourier series coefficient of a ramp pules
```

### 11.14 fourier_coefficient_ramp_step

```
fourier coefficient of a ramp-step
```

### 11.15 fourier_coefficient_square_pulse

```
fourier series coefficients of a square pulse
```

### 11.16 fourier_cubic_interaction_coefficients

### 11.17 fourier_derivative

```
derivative via fourier transform
exponential convergence for periodic functions
results in spurious oscillations for aperiodic functions

input:
x : data, sampled in equal intervals
k : order of the derivative

dx : kth-derivative of x

note : 1) the derivative converges with spectral accuracy, i.e. is
         exact up to rounding condition for L sufficiently large
           and x being periodic
        2) the derivative converges with order p, when x has only
           p-continous derivatives, including discontinuous
               derivatives
           over the boundary
        3) discontinuous derivatives result in gibbs phenomenon
```

### 11.18 fourier_derivative_matrix_1d

### 11.19 fourier_derivative_matrix_2d

### 11.20 fourier_expand

```
expand values of fourier series
```

### 11.21 fourier_fit

```
fit a fourier series to a set of sample points that are not spaced
    in
equal intervals
```

### 11.22 fourier_interpolate

```
interpolate samples y sampled at moments (location) t to locations
    ti
```

### 11.23 fourier_matrix

```
transformation matrix for a continuous fourier series
(not for the discrete dft/fft)
```

### 11.24 fourier_matrix2

```
transformation matrix for a continuous fourier series
(not for the discrete dft/fft)
```

### 11.25 fourier_matrix3

```
transformation matrix for the continous fourier transform
this is a matrix with (2*n+1) real columns
```

### 11.26 fourier_matrix_exp

```
transformation matrix for a continuous fourier series
(not for the discrete dft/fft)
```

### 11.27 fourier_multiplicative_interaction_coefficients

### 11.28 fourier_power

```
powers of a continuous fourier series in sin/cos form

powers of a^p = (ur + u1 sin(ot) + u2 sin(ot+dp))^p
phase of first component assumed 0

frequencies higher than 2-omega ignored in input
frequencies higher than 3-omega not computed
```

### 11.29 fourier_power_exp

```
powers of the continuous fourier series
      a^p = (ur + u1 sin(ot) + u2 sin(ot+dp))^p
phase of first component assumed 0

higher orders than 2 ignored input
higher order than 3 not computed in output

y = a_0 + sum (a_j sin(jot) + b_j cos(jot))
  = Real(sum_{i=0}^inf c_i exp(1i*omega), c_i = a_i + b_i

NOT the alternative sum_{i=-inf}^inf \tilde c_i, tile c_j = 1/2 a_j
     + 1/2i b_j
```

### 11.30 fourier_predict

```
expand a continous fourier series at times t
```

### 11.31 fourier_quadratic_interaction_coefficients

### 11.32 fourier_random_phase_walk

```
evaluete fourier series where the phase undergoes a brownian motion
```

### 11.33 fourier_range

```
approximate range of a continous Fourier series with 2 components
range(y) = max(y) - min(y)
```

### 11.34 fourier_regress

```
fit a continous fourier series to a set of sample points not
    sampled
at equal intervals
```

### 11.35 fourier_resampled_fit

```
fits coefficients of a continuous fourier transform,
but stores them as resampled values
```

### 11.36 fourier_resampled_predict

```
interpolates a continuous fourier series that has been stored as
    values
at their support points
```

### 11.37 fourier_signed_square

```
coefficients of the fourier series of | cos a + cos t | (cos a +
    cos t)
in general
          cos a is midrange
          cos t is tidal variation
c.f Dronkers
```

### 11.38 fourier_transform

```
continuous fourier transformation of y
(not discrete fourier transformation dft/fft)

input:
      b : data sampled at equal intervals
      T : length of data in time or space, i.e. position of last
          sample if
          position of first sample is 0
     T_max : maximum period to include

output :
      A  : fourier matrix
     p  : fourier transformation of b
      tt : TODO
```

### 11.39 fourier_transform_fractional

## 11.40 hyperbolic_fourier_box

## 11.41 idftmtx_man

```
inverse matrix for the discrete fourier transform in matlab style
with a limited number of columns, thus ignoring higher frequencies
keep 2nc+1 columns (mean and conj-complex pairs of nc frequencies)
```

## 11.42 laplace_2d_pwlinear

```
solution to the Laplacian in two dimensions for a finite
    rectangular domain
with piecewise constant boundary conditions
linear system with 4 unknowns per freqency component
these are coefficients of s,c,sh,ch
       (pu*(s + c) + qu*(s' + c'))*(shu + chu) = ru      % upper bc
       (pd*(s + c) + qd*(s' + c'))*(shd + chd) = rd      % lower bc
       ( (sl + cl)*( pl*(shl + chl) + ql*(shl' + chl')) = rl % left
           bc
       ( (sr + cr)*( pr*(shr + chr) + qr*(shr' + chr')) = rr % right
            bc

 least squares with piecewise integration
[x0,p,q,r] piecewise linear polynomials at the boundaries
```

## 11.43 mean_fourier_power

## 11.44 moments_fourier_power

## 11.45 nanfft

```
discrete fourier transform of a data series with gaps
```

### 11.46 peaks

```
peaks of the power spectrum of a disctrete fourier transform

rule for peaks: there is no higher value left or right of the "peak
    "
              until the signal drops to p*y_peak, p = 0.5

works best, when spectrum has been smoothened

input :
f : frequency
y : absolute value of fourier transform (power spectrum)
L : length in space or time of series

output :

a0 : amplitude
s0 : standard deviation (error?) of amplitude
w0 : width of peak
lambda = wave length (period?)
pdx : index of peak
f : frequency (if not given as input)
```

### 11.47 roots_fourier

```
zeros of continuous fourier series series

      f  = a_0 + sum_j=^n a_i cos(j x) + b_i sin(j x)
```

### 11.48 spectral_density

```
spectral density
```

### 11.49 std_fourier_power

### 11.50 test_complex_exp_product

### 11.51 test_fourier_filter

### 11.52 test_idftmtx

### 11.53 var_fourier_power

## 12 mathematics

```
mathematical functions of various kind
```

### 12.1 gaussfit_quantile

## 13 geometry/@Geometry

### 13.1 Geometry

### 13.2 arclength

```
 arc length of a two dimensional curve

 8th order accurate
 does not require the segments length to vary smoothly

 note: the curve can be considered parametric, e.g. x = x(t), y=y(t)
      and
        and t = t(s), but the error term contains derivatives of t,
      thus a non smooth t (strongly varying distance between points)
      requires the scaling as done below
```

### 13.3 arclength_old

```
arc length of a two dimensional function
```

### 13.4 arclength_old2

```
arc length of a two dimensional function
```

### 13.5 base_point

```
base point (fusspunkt), i.e. point on a line with shortest distance
to another point
```

### 13.6 base_point_limited

```
base point (Fusspunkt) of a point on a line
```

### 13.7 centroid

```
centroid pf a polygone
```

### 13.8 cosa_min_max

### 13.9 cross2

```
cross product in two dimensions
```

### 13.10 curvature

```
curvature of a function in two dimensions
```

### 13.11 ddot

```
sum of squares of cos of inner angles of triangle
```

### 13.12 distance

```
equclidan distance between two points
```

### 13.13 distance2

```
euclidean distance between two points
this function requires a and be of equal dimensions, or the least
    the first pair or second pair to be a scalar
```

### 13.14 dot

```
dot product
```

### 13.15 edge_length

```
edge length
```

### 13.16 enclosed_angle

```
angle enclosed between two lines
```

### 13.17 enclosing_triangle

```
smallest enclosing equilateral triangle with bottom site paralle to
     X axis
```

### 13.18 hexagon

```
coordinates of a hexagon, scaled and rotated
```

### 13.19 inPolygon

```
flag points contained in a polygon
much faster than matlab internal function
```

### 13.20 inTetra

```
flag points contained in tetrahedron
```

### 13.21 inTetra2

```
flag points contained in tetrahedron
```

### 13.22 inTriangle

```
flag points contained in triangle
function [flag, c] = inTriangle(P1,P2,P3,P0)
```

### 13.23 intersect

```
intersect between two lines
```

### 13.24 lineintersect

```
intersect of two lines
```

### 13.25 lineintersect1

```
intersect of two lines
```

### 13.26 minimum_distance_lines

```
minimum distance of two lines in three dimensions
```

### 13.27 mittenpunkt

```
mittenpunkt of a triangle
```

### 13.28 nagelpoint

```
nagelpoint of a triangle
```

### 13.29 onLine

### 13.30 orthocentre

```
orthocentre of triangle
```

### 13.31 plumb_line

### 13.32 poly_area

```
area of a polygon
function A = poly_area(x,y)
```

### 13.33 poly_edges

```
edges of a polygon
```

### 13.34 poly_set

```
associate point at arbitary location with a polygon it is contained
     in
and assign the value of the polygon to it
```

## 13.35 poly_width

```
width of polygon width holes by surface normals
holes / islands separated with NaN
order of points of outer boundary must be cw
order of points of holes must be ccw
note that this function does not give the true width for expanding
    sections
use voronoi polygons for this
```

## 13.36 polyxpoly

```
intersections of two polygons
```

## 13.37 project_to_curve

```
closest point on a curve with respect to a point at distance to the
     curve
```

## 13.38 quad_isconvex

## 13.39 random_disk

```
draw random points on the unit disk
```

## 13.40 random_simplex

```
random point inside of a triangle
```

## 13.41 sphere_volume

```
volume of a sphere
function v = sphere_volume(r)
```

### 13.42 tetra_volume

```
volume of a tetrahedron
```

### 13.43 tobarycentric

```
cartesian to barycentric coordinates
```

### 13.44 tobarycentric1

```
cartesian to barycentric coordinates
```

### 13.45 tobarycentric2

```
cartesian to barycentric coordinates
```

### 13.46 tobarycentric3

```
cartesian to barycentric coordinates
```

### 13.47 tri_angle

```
cos of angles of a triangle
```

### 13.48 tri_area

```
angle of a triangle
```

### 13.49 tri_centroid

```
centroid of a triangle
```

## 13.50 tri_distance_opposit_midpoint

```
distance between corner of a triangle and its opposing mid-point
```

## 13.51 tri_edge_length

```
edge length of a triangle
```

## 13.52 tri_edge_midpoint

```
mid point of a triangle
```

## 13.53 tri_excircle

```
excircle of a triangle
```

## 13.54 tri_height

```
height of a triangle
```

## 13.55 tri_incircle

```
incircle of a triangle
```

## 13.56 tri_isacute

```
flag acute triangles
```

## 13.57 tri_isobtuse

```
flag obntuse triangles
```

### 13.58 tri_semiperimeter

```
semiperimeter of a triangle
```

### 13.59 tri_side_length

```
edge lenght of triangle
```

## 14 geometry

### 14.1 Polygon

```
Simple 2D polygon class

  Polygon properties:
      x - x coordinates of polygon
      y - y coordinates of polygon
      nnodes - number of nodes in the polygon

  Polygon methods:
      in - checks whether given points lie inside, on the edge, or
          outside of the polygon
      area - returns the area of the polygon
      centerline - computes the centerline of the river
      iscw - check whether polygon is clockwise
      reverse - reverse the order of the polygon
```

### 14.2 bounding_box

```
bounding box of X
```

### 14.3 curvature_1d

```
curvature of a sampled parametric curve in two dimensions
```

### 14.4 cvt

```
centroidal voronoi tesselation
```

### 14.5 deg_to_frac

```
degree, minutes and seconds to fractions
```

### 14.6 ellipse

```
return points on an ellipse
n : number of points
ci : confidence interval, i.e. for 1 sigma
```

### 14.7 ellipseX

```
x-coordinates of y-coordinates of an ellipse
```

### 14.8 ellipseY

### 14.9 first_intersect

```
get first intersection between lines in A and B
```

### 14.10 golden_ratio

```
golden ratio
```

### 14.11 hypot3

```
hypothenuse in 3D
```

### 14.12 meanangle

```
weighted mean of angles
```

## 14.13 meanangle2

```
mean angle
```

## 14.14 meanangle3

```
mean angle
```

## 14.15 meanangle4

```
mean angle
```

## 14.16 medianangle

```
median angle
angle, that has the smallest squared distance to all others
```

## 14.17 medianangle2

```
median angle

input
alpha : x*m, [rad] angle

ouput
ma    : 1*m, [rad] median angle
sa    : 1*m, [rad] standard error of median angle for uncorrelated
    error
```

## 14.18 pilim

```
limit to +- pi
```

### 14.19 streamline_radius_of_curvature

```
streamline radius of curvature
simplifies when rotatate to streamwise coordinates to R = 1/dv/ds *
    u
```

## 15 histogram/@Histogram

### 15.1 2x

### 15.2 Histogram

### 15.3 bimodes

### 15.4 cdf

### 15.5 cdfS

### 15.6 chi2test

### 15.7 cmoment

### 15.8 cmomentS

### 15.9 entropy

### 15.10 entropyS

### 15.11 export_csv

### 15.12 iquantile

### 15.13 kstest

### 15.14 kurtosis

### 15.15 kurtosisS

### 15.16 mean

### 15.17 meanS

### 15.18 median

### 15.19 medianS

### 15.20 mode

### 15.21 modeS

### 15.22 moment

### 15.23 momentS

### 15.24 pdf

### 15.25 quantile

### 15.26 quantileS

### 15.27 resample

### 15.28 setup

### 15.29 skewness

### 15.30 skewnessS

### 15.31 stairs

### 15.32 stairsS

### 15.33 std

### 15.34 stdS

### 15.35 var

### 15.36 varS

## 16 histogram

### 16.1 hist_man

### 16.2 histadapt

### 16.3 histconst

### 16.4 pdf_poly

### 16.5 plotcdf

### 16.6 test_histogram

## 17 mathematics

```
mathematical functions of various kind
```

### 17.1 imrotmat

## 18 linear-algebra

### 18.1 averaging_matrix_2

### 18.2 colnorm

```
 norms of columns
```

### 18.3 condest_

estimation of the condition number

### 18.4 connectivity_matrix

## 19 linear-algebra/coordinate-transformation

### 19.1 barycentric2cartesian

barycentric to cartesian coordinates

### 19.2 barycentric2cartesian3

convert barycentric to cartesian coordinates

### 19.3 cartesian2barycentric

cartesian to barycentric coordinates

### 19.4 cartesian_to_unit_triangle_basis

transform coodinates into unit triangle

### 19.5 ellipsoid2geoid

### 19.6 example_approximate_utm_conversion

### 19.7 latlon2utm

```
transform latitude and longitude to WGS84 UTM
```

### 19.8 latlon2utm_simple

### 19.9 lowrance_mercator_to_wgs84

```
convert lowrance coordinates to wgs84
```

```
based on spreadsheet by D Whitney King and Patty B at Lowrance
```

### 19.10 nmea2utm

```
convert nmea messages to utm coordinates
```

### 19.11 sn2xy

```
convert sn to xy coordinates
```

### 19.12 unit_triangle_to_cartesian

```
transform coordinates in unit triangle to cartesian coordinates
```

### 19.13 utm2latlon

```
convert wgs84 utm to latitute and longitude
```

### 19.14 xy2nt

```
project all points onto the cross section and assign them nz-
    coordinates

transform coordinate into N-T reference
rotate coordinate, so that cross section goes along x-axis
then x and y are n and t respectively scaled by width
N and T coordinates
```

### 19.15 xy2sn

```
convert cartesian to streamwise coordiantes
```

### 19.16 xy2sn_java

```
use java port for speed up
```

### 19.17 xy2sn_old

```
transform points from cartesian into streamwise coordinates

NOTE : prefer the java version, this has some problems with round
    off
```

## 20 linear-algebra

### 20.1 det2x2

```
2x2 matrix inverse of 2x2 matrices stacked along dim 3
```

### 20.2 det3x3

```
determinant of stacked 3x3 matrices
```

### 20.3 det4x4

```
determinant of stacked 4x4 matrices
```

### 20.4 diag2x2

```
diagonal of stacked 2x2 matrices
```

### 20.5 down

### 20.6 eig2x2

```
eigenvalues of stacked 2x2 matrices
```

## 21 linear-algebra/eigenvalue

### 21.1 eig_bisection

### 21.2 eig_inverse

### 21.3 eig_inverse_iteration

### 21.4 eig_power_iteration

# 22 linear-algebra/eigenvalue/jacobi-davidson

## 22.1 afun_jdm

## 22.2 davidson

## 22.3 jacobi_davidson

## 22.4 jacobi_davidson_qr

## 22.5 jacobi_davidson_qz

## 22.6 jacobi_davidson_simple

## 22.7 jdqr

```
% Read/set parameters
% Initiate global variables
% Return if eigenvalueproblem is trivial
% Initialize V, W:
%   V,W orthonormal, A*V=W*R+Qschur*E, R upper triangular
% The JD loop (Standard)
%    V orthogonal, V orthogonal to Qschur
%    V*V=eye(j), Qschur'*V=0,
%    W=A*V, M=V'*W
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```
%
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
% The JD loop (Harmonic Ritz values)
%    Both V and W orthonormal and orthogonal w.r.t. Qschur
%    V*V=eye(j), Qschur'*V=0, W'*W=eye(j), Qschur'*W=0
%    (A*V-tau*V)=W*R+Qschur*E, E=Qschur'*(A*V-tau*V), M=W'*V
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
% The JD loop (Harmonic Ritz values)
%    V W AV.
%    Both V and W orthonormal and orthogonal w.r.t. Qschur, AV=A*V-
    tau*V
%    V*V=eye(j), W'*W=eye(j), Qschur'*V=0, Qschur'*W=0,
%    (I-Qschur*Qschur')*AV=W*R, M=W'*V; R=W'*AV;
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
```

```
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
% The JD loop (Harmonic Ritz values)
%    W orthonormal, V and W orthogonal to Qschur,
%    W'*W=eye(j), Qschur'*V=0, Qschur'*W=0
%    W=(A*V-tau*V)-Qschur*E, E=Qschur'*(A*V-tau*V),
%    M=W'*V
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
 W=V*Q; V=V(:,1:j)/R; E=E/R; R=eye(j); M=Q(1:j,:)'/R;
 W=V*H; V(:,j+1)=[];R=R'*R; M=H(1:j,:)';
%======== ARNOLDI (for initializing spaces)
    ===============================
%=========== END ARNOLDI
    ============================================
% not accurate enough M=Rw'\(M/Rv);
%=========== COMPUTE SORTED JORDAN FORM
    ==================================
% compute vectors and matrices for skew projection
% solve preconditioned system
% 0 step of bicgstab eq. 1 step of bicgstab
```

```
% Then x is a multiple of b
% HIST=[0,1];
 explicit preconditioning
% compute norm in l-space
%  HIST=[HIST;[nmv,rnrm/snrm]];
% sufficient accuracy. No need to update r,u
 implicit preconditioning
% collect the updates for x in l-space
% but, do the orth to Q implicitly
% compute norm in l-space
% HIST=[HIST;[nmv,rnrm/snrm]];
% sufficient accuracy. No need to update r,u
% Do the orth to Q explicitly
% In exact arithmetic not needed, but
% appears to be more stable.
%  plot(HIST(:,1),log10(HIST(:,2)+eps),'*'), drawnow, pause
% 0 step of gmres eq. 1 step of gmres
% Then x is a multiple of b
%=======================================================================

% 0 step of gmres eq. 1 step of gmres
% Then x is a multiple of b
 HIST=1;
% Lucky break-down
 HIST=[HIST;(gamma~=0)/sqrt(rho)];
% Lucky break-down
% solve in least square sense
 HIST=log10(HIST+eps); J=[0:size(HIST,1)-1]';
 plot(J,HIST(:,1),'*'); drawnow,% pause
 r=r/rho; rho=1;
%  HIST=rho;
%  HIST=[HIST;rho];
 HIST=log10(HIST+eps); J=[0:size(HIST,1)-1]';
 plot(J,HIST(:,1),'*'); drawnow,% pause
% HIST = rho;
%  HIST=[HIST;rho];
 HIST=log10(HIST+eps); J=[0:size(HIST,1)-1]';
 plot(J,HIST(:,1),'*'); drawnow, pause
% HIST = rho;
%  HIST=[HIST;rho];
 HIST=log10(HIST+eps); J=[0:size(HIST,1)-1]';
 plot(J,HIST(:,1),'*'); drawnow, pause
%------ compute schur form -------------
 A*Q=Q*S, Q'*Q=eye(size(A));
% transform real schur form to complex schur form
%------ find order eigenvalues ---------------
%------ reorder schur form ----------------
%------ compute qz form ----------------
%------ sort eigenvalues ---------------
```

```
%------ sort qz form -------------------
% i>j, move ith eigenvalue to position j
% determine dimension
% defaults
%% 'v'
```

## 22.8 jdqr_sleijpen

```
% Read/set parameters
% Initiate global variables
% Return if eigenvalueproblem is trivial
% Initialize V, W:
%   V,W orthonormal, A*V=W*R+Qschur*E, R upper triangular
% The JD loop (Standard)
%    V orthogonal, V orthogonal to Qschur
%    V*V=eye(j), Qschur'*V=0,
%    W=A*V, M=V'*W
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
% The JD loop (Harmonic Ritz values)
%    Both V and W orthonormal and orthogonal w.r.t. Qschur
%    V*V=eye(j), Qschur'*V=0, W'*W=eye(j), Qschur'*W=0
%    (A*V-tau*V)=W*R+Qschur*E, E=Qschur'*(A*V-tau*V), M=W'*V
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
% The JD loop (Harmonic Ritz values)
%    V W AV.
%    Both V and W orthonormal and orthogonal w.r.t. Qschur, AV=A*V-
    tau*V
%    V*V=eye(j), W'*W=eye(j), Qschur'*V=0, Qschur'*W=0,
%    (I-Qschur*Qschur')*AV=W*R, M=W'*V; R=W'*AV;
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
% The JD loop (Harmonic Ritz values)
%    W orthonormal, V and W orthogonal to Qschur,
%    W'*W=eye(j), Qschur'*V=0, Qschur'*W=0
%    W=(A*V-tau*V)-Qschur*E, E=Qschur'*(A*V-tau*V),
%    M=W'*V
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
 W=V*Q; V=V(:,1:j)/R; E=E/R; R=eye(j); M=Q(1:j,:)'/R;
 W=V*H; V(:,j+1)=[];R=R'*R; M=H(1:j,:)';
%======== ARNOLDI (for initializing spaces)
    ===============================
%=========== END ARNOLDI
    ============================================
% not accurate enough M=Rw'\(M/Rv);
%=========== COMPUTE SORTED JORDAN FORM
    ==================================
% compute vectors and matrices for skew projection
% solve preconditioned system
% 0 step of bicgstab eq. 1 step of bicgstab
% Then x is a multiple of b
% HIST=[0,1];
 explicit preconditioning
% compute norm in l-space
%  HIST=[HIST;[nmv,rnrm/snrm]];
% sufficient accuracy. No need to update r,u
 implicit preconditioning
% collect the updates for x in l-space
% but, do the orth to Q implicitly
% compute norm in l-space
% HIST=[HIST;[nmv,rnrm/snrm]];
% sufficient accuracy. No need to update r,u
% Do the orth to Q explicitly
% In exact arithmetic not needed, but
% appears to be more stable.
%  plot(HIST(:,1),log10(HIST(:,2)+eps),'*'), drawnow, pause
% 0 step of gmres eq. 1 step of gmres
% Then x is a multiple of b
%=======================================================================

% 0 step of gmres eq. 1 step of gmres
% Then x is a multiple of b
 HIST=1;
% Lucky break-down
 HIST=[HIST;(gamma~=0)/sqrt(rho)];
% Lucky break-down
```

```
% solve in least square sense
 HIST=log10(HIST+eps); J=[0:size(HIST,1)-1]';
 plot(J,HIST(:,1),'*'); drawnow,% pause
 r=r/rho; rho=1;
%  HIST=rho;
%  HIST=[HIST;rho];
 HIST=log10(HIST+eps); J=[0:size(HIST,1)-1]';
 plot(J,HIST(:,1),'*'); drawnow,% pause
% HIST = rho;
%  HIST=[HIST;rho];
 HIST=log10(HIST+eps); J=[0:size(HIST,1)-1]';
 plot(J,HIST(:,1),'*'); drawnow, pause
% HIST = rho;
%  HIST=[HIST;rho];
 HIST=log10(HIST+eps); J=[0:size(HIST,1)-1]';
 plot(J,HIST(:,1),'*'); drawnow, pause
%------ compute schur form -------------
 A*Q=Q*S, Q'*Q=eye(size(A));
% transform real schur form to complex schur form
%------ find order eigenvalues ---------------
%------ reorder schur form ----------------
%------ compute qz form ----------------
%------ sort eigenvalues ---------------
%------ sort qz form -------------------
% i>j, move ith eigenvalue to position j
% determine dimension
% defaults
%% 'v'
```

### 22.9 jdqr_vorst

```
% Read/set parameters
% Initiate global variables
% Return if eigenvalueproblem is trivial
% Initialize V, W:
%   V,W orthonormal, A*V=W*R+Qschur*E, R upper triangular
% The JD loop (Standard)
%    V orthogonal, V orthogonal to Qschur
%    V*V=eye(j), Qschur'*V=0,
%    W=A*V, M=V'*W
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
% The JD loop (Harmonic Ritz values)
%    Both V and W orthonormal and orthogonal w.r.t. Qschur
%    V*V=eye(j), Qschur'*V=0, W'*W=eye(j), Qschur'*W=0
%    (A*V-tau*V)=W*R+Qschur*E, E=Qschur'*(A*V-tau*V), M=W'*V
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
% The JD loop (Harmonic Ritz values)
%    V W AV.
%    Both V and W orthonormal and orthogonal w.r.t. Qschur, AV=A*V-
    tau*V
%    V*V=eye(j), W'*W=eye(j), Qschur'*V=0, Qschur'*W=0,
%    (I-Qschur*Qschur')*AV=W*R, M=W'*V; R=W'*AV;
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
% The JD loop (Harmonic Ritz values)
%   W orthonormal, V and W orthogonal to Qschur,
%   W'*W=eye(j), Qschur'*V=0, Qschur'*W=0
%   W=(A*V-tau*V)-Qschur*E, E=Qschur'*(A*V-tau*V),
%   M=W'*V
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
 Rschur=[[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
 W=V*Q; V=V(:,1:j)/R; E=E/R; R=eye(j); M=Q(1:j,:)'/R;
 W=V*H; V(:,j+1)=[];R=R'*R; M=H(1:j,:)';
%======== ARNOLDI (for initializing spaces)
    ==============================
%=========== END ARNOLDI
    ============================================
% not accurate enough M=Rw'\(M/Rv);
%=========== COMPUTE SORTED JORDAN FORM
    =================================
% accepted separation between eigenvalues:
% no preconditioning
% solve left preconditioned system
% compute vectors and matrices for skew projection
% precondion and project r
% solve preconditioned system
```

```
% no preconditioning
% solve two-sided expl. precond. system
% compute vectors and matrices for skew projection
% precondion and project r
% solve preconditioned system
% "unprecondition" solution
%%%% u(:,j+1)=Atilde*u(:,j)
%%%% r(:,j+1)=Atilde*r(:,j)
%------ compute schur form -------------
 A*Q=Q*S, Q'*Q=eye(size(A));
% transform real schur form to complex schur form
%------ find order eigenvalues ---------------
%------ reorder schur form ----------------
%------ compute qz form ----------------
%------ sort eigenvalues ---------------
%------ sort qz form -------------------
% i>j, move ith eigenvalue to position j
% determine dimension
% defaults
```

### 22.10 jdqz

```
% Read/set parameters
% Return if eigenvalueproblem is trivial
% Initialize target, test space and interaction matrices
% V=RepGS(Qschur,V); [AV,BV]=MV(V); %%% more stability??
% W=RepGS(Zschur,eval(testspace)); %%% dangerous if sigma~lambda
% Solve the preconditioned correction equation
% Expand the subspaces and the interaction matrices
% Check for stagnation
% Solve projected eigenproblem
% Compute approximate eigenpair and residual
%=== an alternative, but less stable way of computing z =====
% display history
% save history
% check convergence
% EXPAND Schur form
% Expand preconditioned Schur matrix MinvZ=M\Zschur
% check for conjugate pair
% To detect whether another eigenpair is accurate enough
% restart if dim(V)> jmax
% Initialize target, test space and interaction matrices
% additional stabilisation. May not be needed
% V=RepGS(Zschur,V); [AV,BV]=MV(V);
% end add. stab.
% Solve the preconditioned correction equation
% expand the subspaces and the interaction matrices
```

```
% Check for stagnation
% compute approximate eigenpair
% Compute approximate eigenpair and residual
% display history
% save history
% check convergence
% expand Schur form
% ZastQ=Z'*Q0
% the final Qschur
% check for conjugate pair
% t perp Zschur, t in span(Q0,imag(q))
% To detect whether another eigenpair is accurate enough
% restart if dim(V)> jmax
%======== END JDQZ
    ====================================================
%=======================================================================

%======== PREPROCESSING
    ===============================================
%=======================================================================

%======== ARNOLDI (for initial spaces)
    ================================
%% then precond=I and target = 0: apply Arnoldi with A
%======== END ARNOLDI
    =================================================
%=======================================================================

%======== POSTPROCESSING
    ==============================================
%=======================================================================

%======== SORT QZ DECOMPOSITION INTERACTION MATRICES
    ==================
%======== COMPUTE SORTED JORDAN FORM
    ==================================
%======== END JORDAN FORM
    =============================================
%======== OUTPUT
    ======================================================
%=======================================================================

%======== UPDATE PRECONDITIONED SCHUR VECTORS
    =========================
%=======================================================================

%=======================================================================
```

```
%======== SOLVE CORRECTION EQUATION
    ===================================
%========================================================================

% solve preconditioned system
%========================================================================

%======== LINEAR SOLVERS
    ==============================================
%========================================================================

% [At,Bt]=MV(x); At=theta(2)*At-theta(1)*Bt;
% xtol=norm(r-At+Z*(Z'*At))/norm(r);
%===== Iterative methods
    ==============================================
% 0 step of bicgstab eq. 1 step of bicgstab
% Then x is a multiple of b
% HIST=[0,1];
 explicit preconditioning
% compute norm in l-space
% HIST=[HIST;[nmv,rnrm/snrm]];
% sufficient accuracy. No need to update r,u
 implicit preconditioning
% collect the updates for x in l-space
% but, do the orth to Z implicitly
% compute norm in l-space
% HIST=[HIST;[nmv,rnrm/snrm]];
% sufficient accuracy. No need to update r,u
% Do the orth to Z explicitly
% In exact arithmetic not needed, but
% appears to be more stable.
% plot(HIST(:,1),log10(HIST(:,2)+eps),'*'), drawnow
% 0 step of gmres eq. 1 step of gmres
% Then x is a multiple of b
%========================================================================

% 0 step of gmres eq. 1 step of gmres
% Then x is a multiple of b
 HIST=1;
% Lucky break-down
 HIST=[HIST;(gamma~=0)/sqrt(rho)];
% Lucky break-down
% solve in least square sense
 HIST=log10(HIST+eps); J=[0:size(HIST,1)-1]';
 plot(J,HIST(:,1),'*'); drawnow
%======== END SOLVE CORRECTION EQUATION
    ===============================
%========================================================================
```

```
%======== BASIC OPERATIONS
    ============================================
%========================================================================

 y(1:5,1), pause
%======== COMPUTE r AND z
    =============================================
% E*u=Q*sigma, sigma(1,1)>sigma(2,2)
%======== END computation r and z
    =====================================
%========================================================================

%======== Orthogonalisation
    ===========================================
%========================================================================

%======== END Orthogonalisation
    ======================================
%========================================================================

%======== Sorts Schur form
    ============================================
%========================================================================

   kappa=max(norm(A,inf)/max(norm(B,inf),1.e-12),1);
   kappa=2^(round(log2(kappa)));
%------ compute the qz factorization -------
%------ scale the eigenvalues --------------
%------ sort the eigenvalues ---------------
%------ swap the qz form -------------------
% repeat SwapQZ if angle is too small
%========================================================================

%========================================================================

% i>j, move ith eigenvalue to position j
% compute q s.t. C*q=(t(i,1)*S-s(i,1)*T)*q=0
 C*P=Q*R
 check whether last but one diag. elt r nonzero
 C*q
% end computation q
%======== END sort QZ decomposition interaction matrices
    ==============
%========================================================================

%======== INITIALIZATION
    ==============================================
%========================================================================
```

```
%===========================================================================

% defaults            %%%% search for 'xx' in fieldnames
%% 'ma'
%% 'sch'
%% 'to'
%% 'di'
% jmin=nselect+p0 %%%% 'jmi'
% jmax=jmin+p1  %%%% 'jma'
%% 'te'
%% 'pai'
%% 'av'
%% 'tr'
%% 'fix'
%% 'ns'
%% 'ch'
%% 'lso'
%% 'ls_m'
%% 'ls_t'
%% 'ls_e'
%% 'ty'
%% 'l_'
%% 'u_'
%% 'p_'
%% 'sca'
%% 'v0'
 initiation
 'standard'
 'harmonic'
 'searchspace'
%===========================================================================

% or Operator_Form=3 or Operator_Form=5???
%===========================================================================

%========= DISPLAY FUNCTIONS
    ============================================
%===========================================================================

%===========================================================================

%===========================================================================

%===========================================================================
```

### 22.11 mfunc_jdm

### 22.12 mgs

### 22.13 minres_

### 22.14 mv_jacobi_davidson

## 23 linear-algebra

### 23.1 first

### 23.2 gershgorin_circle

```
range of eigenvalues determined by the gershgorin circle theorem
```

### 23.3 haussdorff

```
haussdorf dimension
box counting: count cectangles passed through by line (covered by
    polygon)

Koch snow flake 3:4 -> 1.2619
Kantor set     2:3, (4:9) -> 0.6309
quadrat        4:2, 9:3, 16:4 -> 2
```

### 23.4 ieig2x2

```
reconstruct matrix from eigenvalue decomposition
```

### 23.5 inv2x2

```
2x2 inverse of stacked matrices
```

### 23.6 inv3x3

### 23.7 inv4x4

```
inverse of stacked 4x4 matrices
```

### 23.8 kernel2matrix

## 24 linear-algebra/lanczos

### 24.1 arnoldi

### 24.2 arnoldi_new

### 24.3 eigs_lanczos_man

### 24.4 lanczos

### 24.5 lanczos_

### 24.6 lanczos_biorthogonal

### 24.7 lanczos_biorthogonal_improved

### 24.8 lanczos_ghep

### 24.9 mv_lanczos

### 24.10 reorthogonalise

### 24.11 test_lanczos

## 25 linear-algebra

### 25.1 left

```
left element of vector, leftmost column is extrapolated
```

## 26 linear-algebra/linear-systems

### 26.1 gmres_man

```
break on convergence
```

### 26.2 minres_recycle

## 27 linear-algebra

### 27.1 lpmean

```
mean of pth-power of a
```

### 27.2 lpnorm

```
norm of lth-power of a
```

### 27.3 matvec3

```
matrix-vector product of stacked matrices and vectors
```

### 27.4 max2d

```
maximum value and i-j index for matrix
```

### 27.5 mid

```
mid point between neighbouring vector elements
```

### 27.6 mpoweri

```
approximation of A^p, where p is not integer by quadtratic
    interpolation
```

### 27.7 mtimes2x2

### 27.8 mtimes3x3

```
product of stacked 3x3 matrices
```

### 27.9 nannorm

```
norm of a vector, skips nan-values
```

### 27.10 nanshift

```
shift vector, but set out of range values to NaN
```

### 27.11 nl

```
number rows (lines) of a matrix

analogue to unix nl command
```

### 27.12 normalise

```
normalise a vector or the columns of a matrix
note that the columns are independently normalised, and hence not
    necessarily
orthogonal to each other use the gram schmidt algorithm for this (
    qr or orth)
```

### 27.13 normalize1

```
normalize columns in x to [-1,1]
```

### 27.14 normrows

### 27.15 orth2

```
make matrix A orhogonal to B
```

### 27.16 orth_man

```
orthogonalize the columns of A
```

### 27.17 orthogonalise

```
make x orthogonal to Y
```

### 27.18 padd2

```
padd values around a 2d (image) matrix, constant exprapolation
```

### 27.19 paddext

```
padd values to vactor
not suitable for noisy data
order = 0 : constant extrapolation (hold)
order = 1 : linear extrapolation
```

### 27.20 paddval1

```
padd values at end of x
```

### 27.21 paddval2

```
padd values to x
```

# 28 linear-algebra/polynomial

## 28.1 chebychev

```
chebycheff polynomials
```

## 28.2 piecewise_polynomial

```
evaluate piecewise polynomial
```

## 28.3 roots1

```
roots of linear functions
```

## 28.4 roots2

```
roots of quadratic function
c1 x^2 + c2 x + c3 = 0
```

## 28.5 roots2poly

## 28.6 roots3

## 28.7 roots4

## 28.8 roots_piecewise_linear

### 28.9 test_roots4

### 28.10 vanderi_1d

```
vandermonde matrix of an integral
```

## 29 linear-algebra

### 29.1 randrot

```
random rotation matrix
```

### 29.2 right

```
get right column by shifting columns to left
extrapolate rightmost column
```

### 29.3 rot2

```
rotation matrix from angle
```

### 29.4 rot2dir

```
rotation matrix from direction vector
```

### 29.5 rot3

### 29.6 rotR

### 29.7 rownorm

### 29.8 simmilarity_matrix

### 29.9 spnorm

```
frobenius norm
```

### 29.10 spzeros

```
allocate a sparze matrix of zeros
```

### 29.11 test_roots3

### 29.12 transform_minmax

### 29.13 transpose3

```
transpose stacked 3x3 matrices
```

### 29.14 transposeall

### 29.15 up

### 29.16 vander_nd

### 29.17 vanderd_2d

## 30 logic

```
bitwise operations on integers
```

### 30.1 bitor_man

```
 bitwise OR of the numbers of the columns of A

 input:
       A (positive integer)
```

## 31 master/plot

### 31.1 attach_boundary_value

### 31.2 cartesian_polar

### 31.3 img_vargrid

### 31.4 plot_basis_functions

### 31.5 plot_convergence

### 31.6 plot_dof

### 31.7 plot_eigenbar

### 31.8 plot_error_estimation

### 31.9 plot_error_estimation_2

### 31.10 plot_error_fem

### 31.11 plot_fdm_kernel

### 31.12 plot_fdm_vs_fem

### 31.13 plot_fem_accuracy

### 31.14 plot_function_and_grid

### 31.15 plot_hat

### 31.16 plot_hydrogen_wf

### 31.17 plot_mesh

### 31.18 plot_mesh_2

### 31.19 plot_refine

### 31.20 plot_refine_3d

### 31.21 plot_runtime

### 31.22 plot_spectrum

### 31.23 plot_wavefunction

## 32 master/ported

### 32.1 assemble_2d_phi_phi

### 32.2 assemble_3d_dphi_dphi

### 32.3 assemble_3d_phi_phi

### 32.4 dV_2d_

### 32.5 derivative_2d

### 32.6 derivative_3d

### 32.7 element_neighbour_2d

### 32.8 prefetch_2d_

### 32.9 promote_2d_3_10

### 32.10 promote_2d_3_15

### 32.11 promote_2d_3_21

### 32.12 promote_2d_3_6

### 32.13 promote_3d_4_10

### 32.14 promote_3d_4_20

### 32.15 promote_3d_4_35

### 32.16 vander_2d

### 32.17 vander_3d

## 33 number-theory

### 33.1 ceiln

```
floor to leading n-digits
```

### 33.2 digitsb

```
number of digits with respect to specified base
```

### 33.3 floorn

```
floor to n-digits
```

### 33.4 iseven

```
true for even numbers in X
```

### 33.5 multichoosek

```
all combinations of lenght k from set values with repetitions
c.f. nchoosek, combinations without repetition

input :
      x : scalar integer or vector of arbitrary numbers
      k : length of subsets
output :
      if x scalar : number of combinations
      if x vector : the exact combinations
```

### 33.6 nchoosek_man

```
vecotrised binomial coefficient
b = N!/K!(N-K)!
```

### 33.7 pythagorean_triple

```
pythagorean triple
```

### 33.8 roundn

```
round to n digits
```

## 34 numerical-methods

### 34.1 advect_analytic

# 35 numerical-methods/differentiation

## 35.1 derivative1

```
first derivative on variable mesh
second order accurate
```

## 35.2 derivative2

```
second derivative on a variable mesh
```

# 36 numerical-methods

## 36.1 diffuse_analytic

# 37 numerical-methods/finite-difference

## 37.1 cdiff

```
differences of columns of X
degree = 1 : central first order differences
degreee = 2 : central second order differences
```

## 37.2 cdiffb

```
differences of columns of X
degree = 1 : central first order differences
degreee = 2 : central second order differences
TODO use difference matrix function for simplicity
```

## 37.3 central_difference

### 37.4 cmean

```
single gaussian smoothing step with kernel 1/4*[1,2,1]
```

### 37.5 cmean2

### 37.6 derivative_matrix_1_1d

```
finite difference matrix of first derivative in one dimensions
n : number of grid points
h = L/(n+1) constant step with

function [D1, d1] = derivative_matrix_1d(n,L,order)
```

### 37.7 derivative_matrix_2_1d

```
finite derivative matrix of second derivative in one dimension
```

### 37.8 derivative_matrix_2d

```
finite difference derivative matrix in two dimensions
```

### 37.9 derivative_matrix_curvilinear

```
derivative matrix on a curvilinear grid
```

### 37.10 derivative_matrix_curvilinear_2

```
derivative matrix on a two dimensional curvilinear grid
the grid has not necessarily to be orthogonal
```

## 37.11 difference_kernel

```
difference kernels for equispaced grids
c.f. Computing the Spectrum of the Confined Hydrogen Atom, Kastner,
     2012
```

## 37.12 diffusion_matrix_2d_anisotropic

## 37.13 diffusion_matrix_2d_anisotropic2

## 37.14 distmat

```
distance matrix for a 2 dimensional rectangular matrix
```

## 37.15 downwind_difference

## 37.16 gradpde2d

```
objective function gradiend on two dimensional regular grid
numeric gradient for non-linear least squares optimisation
of a PDE on a rectangular grid
x_* = min(f(x))
f = (v(x) - v(x_*))^2 = f(x) + A dx + O(dx^2)
a_ij = df_i/dx_j
```

## 37.17 laplacian

## 37.18 laplacian_fdm

```
finite difference matrix of the laplacian
BC
```

### 37.19 lrmean

```
mean of the left and right element
```

# 38 numerical-methods/finite-difference/master

### 38.1 fdm_adaptive_grid

### 38.2 fdm_adaptive_refinement_old

### 38.3 fdm_assemble_d1_2d

### 38.4 fdm_assemble_d2_2d

### 38.5 fdm_confinement

### 38.6 fdm_d_vargrid

### 38.7 fdm_h_unstructured

### 38.8 fdm_hydrogen_vargrid

### 38.9 fdm_mark_unstructured_2d

### 38.10 fdm_plot

### 38.11 fdm_plot_series

### 38.12 fdm_refine_2d

### 38.13 fdm_refine_3d

### 38.14 fdm_refine_unstructured_2d

### 38.15 fdm_schroedinger_2d

### 38.16 fdm_schroedinger_3d

### 38.17 relocate

# 39 numerical-methods/finite-difference

## 39.1 mid

```
mid point between neighbouring vector elements
```

## 39.2 pwmid

```
segment end point to segment mid point transformation for regular 1
    d grids
```

## 39.3 ratio

```
ratio of two subsequent values
```

## 39.4 steplength

```
step length of a vector if it were equispaced
```

## 39.5 swapoddeven

```
swap odd and even elements in a vector
```

## 39.6 test_derivative_matrix_2d

## 39.7 test_derivative_matrix_curvilinear

## 39.8 test_difference_kernel

### 39.9 upwind_difference

## 40 numerical-methods/finite-element

### 40.1 Mesh_2d_java

### 40.2 Tree_2d_java

### 40.3 assemble_1d_dphi_dphi

### 40.4 assemble_1d_phi_phi

### 40.5 assemble_2d_dphi_dphi_java

### 40.6 assemble_2d_phi_phi_java

### 40.7 assemble_3d_dphi_dphi_java

### 40.8 assemble_3d_phi_phi_java

## 40.9 boundary_1d

## 40.10 boundary_2d

## 40.11 boundary_3d

## 40.12 check_area_2d

## 40.13 circmesh

## 40.14 cropradius

## 40.15 display_2d

## 40.16 display_3d

## 40.17 distort

## 40.18 err_2d

### 40.19 estimate_err_2d_3

### 40.20 example_1d

### 40.21 example_2d

### 40.22 explode

### 40.23 fem_2d

### 40.24 fem_2d_heuristic_mesh

### 40.25 fem_get_2d_radial

### 40.26 fem_interpolation

### 40.27 fem_plot_1d

### 40.28 fem_plot_1d_series

### 40.29 fem_plot_2d

### 40.30 fem_plot_2d_series

### 40.31 fem_plot_3d

### 40.32 fem_plot_3d_series

### 40.33 fem_plot_confine_series

### 40.34 fem_radial

```
adaptive grid
constant grid
```

### 40.35 flip_2d

### 40.36 get_mesh_arrays

### 40.37 hashkey

# 41 numerical-methods/finite-element/int

## 41.1 int_1d_gauss

## 41.2 int_1d_gauss_1

## 41.3 int_1d_gauss_2

## 41.4 int_1d_gauss_3

## 41.5 int_1d_gauss_4

## 41.6 int_1d_gauss_5

## 41.7 int_1d_gauss_6

## 41.8 int_1d_gauss_lobatto

## 41.9 int_1d_gauss_n

### 41.10 int_1d_nc_2

### 41.11 int_1d_nc_3

### 41.12 int_1d_nc_4

### 41.13 int_1d_nc_5

### 41.14 int_1d_nc_6

### 41.15 int_1d_nc_7

### 41.16 int_1d_nc_7_hardy

### 41.17 int_2d_gauss_1

### 41.18 int_2d_gauss_12

### 41.19 int_2d_gauss_13

### 41.20 int_2d_gauss_16

### 41.21 int_2d_gauss_19

### 41.22 int_2d_gauss_25

### 41.23 int_2d_gauss_3

### 41.24 int_2d_gauss_33

### 41.25 int_2d_gauss_4

### 41.26 int_2d_gauss_6

### 41.27 int_2d_gauss_7

### 41.28 int_2d_gauss_9

### 41.29 int_2d_nc_10

### 41.30 int_2d_nc_15

### 41.31 int_2d_nc_21

### 41.32 int_2d_nc_3

### 41.33 int_2d_nc_6

### 41.34 int_3d_gauss_1

### 41.35 int_3d_gauss_11

### 41.36 int_3d_gauss_14

### 41.37 int_3d_gauss_15

### 41.38 int_3d_gauss_24

### 41.39 int_3d_gauss_4

### 41.40 int_3d_gauss_45

### 41.41 int_3d_gauss_5

### 41.42 int_3d_nc_11

### 41.43 int_3d_nc_4

### 41.44 int_3d_nc_6

### 41.45 int_3d_nc_8

## 42 numerical-methods/finite-element

### 42.1 interpolation_matrix

### 42.2 mark

### 42.3 mark_1d

### 42.4 mesh_1d_uniform

### 42.5 mesh_3d_uniform

### 42.6 mesh_interpolate

### 42.7 neighbour_1d

### 42.8 old

### 42.9 pdeeig_1d

### 42.10 pdeeig_2d

### 42.11 pdeeig_3d

### 42.12 polynomial_derivative_1d

### 42.13 potential_const

### 42.14 potential_coulomb

### 42.15 potential_harmonic_oscillator

### 42.16 project_circle

### 42.17 project_rectangle

### 42.18 promote_1d_2_3

### 42.19 promote_1d_2_4

### 42.20 promote_1d_2_5

### 42.21 promote_1d_2_6

### 42.22 quadrilaterate

### 42.23 recalculate_regularity_2d

### 42.24 refine_1d

### 42.25 refine_2d_21

### 42.26 refine_2d_structural

### 42.27 regularity_1d

### 42.28 regularity_2d

### 42.29 regularity_3d

```
{       T = [1 2 3 4];
}
```

### 42.30 relocate_2d

### 42.31 test_circmesh

### 42.32 test_hermite

### 42.33 tri_assign_points

### 42.34 triangulation_uniform

### 42.35 vander_1d

```
van der Monde matrix
```

### 42.36 vanderd_1d

### 42.37 vanderi_1d

## 43 numerical-methods/finite-volume/@Advection

### 43.1 Advection

```
FVM treatment of the Advection equation
```

### 43.2 dot_advection

```
advection equation
```

## 44 numerical-methods/finite-volume/@Burgers

### 44.1 burgers_split

```
viscous Burgers' equation,
mixed analytic and numerical derivative in frequency space
by splitting sheme
u_t = -(0.5*u^2)_x + c*u_xx
```

### 44.2 dot_burgers_fdm

```
viscous burgers' equation
u_t = -d/dx (1/2*u^2) + c d^2/dx^2 u_xx
```

### 44.3 dot_burgers_fft

```
viscous Burgers' equation in frequency space
u_t + (0.5*u^2)_x = c*u_xx
```

## 45 numerical-methods/finite-volume/@Finite_Volume

### 45.1 Finite_Volume

```
finite volume method for partial differential equations 1+1
    dimensions
(time and space)
```

### 45.2 apply_bc

```
apply boundary conditions
```

### 45.3 solve

```
solve the the PDE by successively stepping in time
this is a trivial implmentation with constant step length
severity of diffusive error depends on dt/dx-ratio
stability depends on wave height
                     printf('Progress %2.1f%% %2.1fs\n',100*(t-Ti
                         (1))/(Ti(2)-Ti(1)),t_real);
```

### 45.4 step_split_strang

```
step in time, treat inhomogeneous part by Strang splitting
this scheme is not suitable for stationary solutions, for example
steady shallow water flow
```

### 45.5 step_unsplit

```
step in time, without splitting the inhomogeneous term
```

## 46 numerical-methods/finite-volume/@Flux_Limiter

### 46.1 Flux_Limiter

```
class of flux limiters
```

### 46.2 beam_warming

```
beam warming sheme
low resolution
note: works only if sign of eigenvalues point into the same
    direction according to RL
```

### 46.3 fromm

```
fromme limiter
low res
```

### 46.4 lax_wendroff

```
lax wendroff scheme
second order accurate, but no tvd
this is effectively not a limiter
eq. 6.39 in randall, leveque
```

### 46.5 minmod

```
min-mod schock limiter
```

### 46.6 monotized_central

```
monotonized central flux limiter
```

### 46.7 muscl

```
muscl flux limiter
```

### 46.8 superbee

```
superbee limiter
```

### 46.9 upwind

```
godunov scheme
godunov, first order accurate
```

### 46.10 vanLeer

```
van Leer limiter
```

## 47 numerical-methods/finite-volume/@KDV

### 47.1 dot_kdv_fdm

```
korteweg de vries equation
u_t + (0.5*u^2)_x = c*u_xxx
```

### 47.2 dot_kdv_fft

```
korteweg de vries equation
compute derivatives in frequency space
u_t + (0.5*u^2)_x = c*u_xxx
```

### 47.3 kdv_split

```
korteweg de vries equation in frequency space,
derivative treated by splitting scheme
```

## 48 numerical-methods/finite-volume/@Reconstruct_Average_Evolve

### 48.1 Reconstruct_Average_Evolve

```
Reconstruct Average Evolve Finite Volume Method for treatment of
    1+1D pdes

McCronack Scheme
err = O(dt^2) + O(dx^2), except as discontinuities
error:
      h_xxx(3:end-2) = 1/dx^3*( -0.5*h(1:end-4) + h(2:end-3) - h(4:
          end-1) + 0.5*h(5:end) );
      th = -1/6*dx^2*qh_.*(1 - (qh_*dt/dx).^2).*h_xxx;
```

### 48.2 advect_highres

```
single time step for the reconstruct evolve algorithm
```

### 48.3 advect_lowress

```
single time step
low resolution
```

## 49 numerical-methods/finite-volume

### 49.1 Godunov

```
Godunov, upwind method for systems of pdes
```

### 49.2 Lax_Friedrich

```
Lax-Friedrich-Method
for hyperbolic conservation laws
err = O(dt) + O(dx)
|a dt/dx| < 1
```

### 49.3 Measure

### 49.4 Roe

```
non linear roe solver for the SWE (randall, leveque 15.3.1)

The roe solver guarantess:
- A is diagonalisable with real eigenvalues (15.12)
- can be determined by a closed formula
- is an efficient replacement for true Rieman solver
```

### 49.5 fv_swe

```
wrapper for solving SWE
```

### 49.6 staggered_euler

```
forward euler method with staggered grid
```

### 49.7 staggered_grid

```
staggered grid approximation to the SWE
```

## 50 numerical-methods

### 50.1 grid2quad

```
extract rectangular elements of a structured grid
in form of an unstructured quad-mesh format
```

# 51 numerical-methods/integration

## 51.1 cumintL

```
cumulative integral from left to right
```

## 51.2 cumintR

```
cumulative integral from right to left
```

## 51.3 int_trapezoidal

```
integrate y along x with the trapezoidal rule
```

# 52 numerical-methods/interpolation/@Kriging

## 52.1 Kriging

```
class for Kriging interpolation
```

## 52.2 estimate_semivariance

```
estimate the parameter of the semivariance model for Kriging
    interpolation
          % set up the regression matrix and solve for
              parameters
```

## 52.3 interpolate_

```
interpolate with Krieging method

this function may interpolate several quantities per coordinate,
using the same variogram, if the semivariance of the quantities
    differs,
the user may prefer to estimate the semivariance and interpolate
    each quantity
individually
```

```
Xs  : source point coordinates
Vs  : value at source points
Xt  : targe point coordinates
Vt  : value at target points
E2t : squared interpolation error at target points
```

# 53 numerical-methods/interpolation/@RegularizedInterpolator

## 53.1 RegularizedInterpolator1

```
class for regularized interpolation (Thikonov) on a 1D mesh
```

## 53.2 init

```
initialize the interpolator with a set of sampling points
```

# 54 numerical-methods/interpolation/@RegularizedInterpolator

## 54.1 RegularizedInterpolator2

```
class for regularized interpolation on an unstructures mesh (
    interpolation)
```

## 54.2 init

```
initialize the interpolator with a set of point samples
```

# 55 numerical-methods/interpolation/@RegularizedInterpolator

## 55.1 RegularizedInterpolator3

```
class for regularized interpolation (Tikhonov) on a triangulation
(unstructured mesh)
```

### 55.2 init

```
initialize the interpolator with a set of sampling points
```

## 56 numerical-methods/interpolation

### 56.1 IDW

```
spatial averaging by inverse distance weighting
```

### 56.2 IPoly

```
polynomial interpolation class
```

### 56.3 IRBM

```
interpolate by the radial basis function method
                      fprintf(1,'Progress IRBM: %d%%\n',round(100*
                          idx/size(Xi,1)));
```

### 56.4 ISparse

```
sparse interpolation class
```

### 56.5 Inn

```
nearest neighbour interpolation
```

### 56.6 Interpolator

```
interpolator super-class
                              fprintf(1,'Progress: %f%% %fs\n',100*
                                  idx/size(Xt,1),t);
```

## 56.7 fixnan

```
fill nan-values in vector with gaps
```

## 56.8 idw1

```
spatial average ny inverse distance weighting
```

## 56.9 idw2

```
spatial average by inverse distance weighting
```

## 56.10 inner2outer

```
linear interpolation of segment mit point to grid points at segment
     ends
assumes equal grid spacing
```

## 56.11 inner2outer2

```
interpolate from element (segment) centres to edge points
```

## 56.12 interp1_circular

## 56.13 interp1_limited

```
interpolate values, but not beyond a certain distance
this function is idempotent, i.e. it will not extrapolate over into
     gaps
exceedint the limit and thus not spuriously extend the series when
    called a second time on the same data
```

### 56.14 interp1_man

```
interpolate
```

### 56.15 interp1_piecewise_linear

### 56.16 interp1_save

```
make interpolation save to round off errors
the matlab internal interpolation suffers from rounding errors,
    which
are unacceptable when values of X and Y are large (for example UTm
    coordinates)
this normalization prevents this
```

### 56.17 interp1_slope

```
quadratic interpolation returning value and derivative(s)
```

### 56.18 interp1_smooth

### 56.19 interp1_unique

```
matlab fails to interpolate, when x values are not unique
this function makes the values unique before use
```

### 56.20 interp2_man

```
nearest neighbour interpolation in two dimensions
```

## 56.21 interp_angle

```
interpolate an angle
```

## 56.22 interp_fourier

```
interpolation by the fourier method
```

## 56.23 interp_fourier_batch

```
batch interpolation by the fourier interpolation
```

## 56.24 interp_sn

```
interpolate along streamwise coordinates
This gives similar result to setting aspect ratio for sN to
    infinity,
but not quite,as the input point set is not dense (scale for sN to
    infinity does not work)
              sdx  = sdx(sdx_);
```

## 56.25 interp_sn2

```
interpolation in streamwise coordinates
```

## 56.26 interp_sn3

## 56.27 interp_sn_

### 56.28 limit_by_distance_1d

```
smooth subsequent values along a curve such that
      v(x0+dx) < v(x0) + (ratio-1)*dx
if v is the edge length in a resampled polygon, then v_i/v_(i+1) <
    ratio
      ratio^1 = exp(a*1)
```

### 56.29 resample1

```
interpolation along a parametric curve with variable step width
```

### 56.30 resample_d_min

```
resample a function
```

### 56.31 resample_vector

```
resample a track so that velocity vectors do not run into each
    other
```

### 56.32 test_interp1_limited

## 57 numerical-methods

### 57.1 inverse_complex

### 57.2 maccormack_step

### 57.3 minmod

# 58 numerical-methods/multigrid

## 58.1 mg_interpolate

## 58.2 mg_restrict

# 59 numerical-methods/ode/@BVPS_Characteristic

## 59.1 BVPS_Characteristic

```
solve coupled first- and second-order 1D boundary-value problems
```

## 59.2 assemble1_A

```
assemble the discretisation matrix for a first order ode
(mean component, zero frequency)
```

## 59.3 assemble1_A_Q

```
assemble the discretisation matrix for a first order ode
(mean component, zero frequency)
```

## 59.4 assemble2_A

```
assemble the discretisation matrix for a second-order ode
(non-zero frequency component)
```

## 59.5 assemble_AA

```
assemble the discretisation matrix for each channel
iteratively calls assembly for each frequency components
```

### 59.6 assemble_AAA

```
assemble the discretisation matrix for the entire network
iteratively calls assembly for each channel
```

### 59.7 assemble_Ic

### 59.8 bvp1c

### 59.9 check_arguments

### 59.10 couple_junctions

### 59.11 derivative

### 59.12 init

### 59.13 inner2outer_bvp2c

### 59.14 reconstruct

### 59.15 resample

### 59.16 solve

```
solve system of non-linear second order odes (in more than one
    variable)
as boundary value problems

odefun provides ode coefficients c:
c(x,1) y''(x) + c(x,2) y'(x) + c(x,3) y = c(x,4)
   c_1 y"    + c_2 y'     + c_3 y + c_4 = c_4

subject to the boundary conditions
bcfun provides v and p and optionally q, so that:

b_1 y + b_2 y' = f
   q(x,1)*( p(x,1) y_l(x) + p(x,2) y_l'(x)
 + q(x,2)*( p(x,1) y_r(x) + p(x,2) y_r'(x) = v(x)
where q weighs the waves travelling from left to right and right to
     left (default [1 1])
```

### 59.17 test_assemble1_A

### 59.18 test_assemble2_A

## 60 numerical-methods/ode/@Time_Stepper

### 60.1 Time_Stepper

### 60.2 solve

# 61 numerical-methods/ode

## 61.1 bvp2fdm

```
solve system of non-linear second order odes (in more than one
    variable)
as boundary value problems by the finite difference method

odefun provides ode coefficients c:
c(x,1) y''(x) + c(x,2) y'(x) + c(x,3) y = c(x,4)
 c_1 y" + c_2 y' + c_3 y + c_4 = 0

subject to the boundary conditions
bcfun provides v and p and optionally q, so that:

b_1 y + b_2 y' = f
   q(x,1)*( p(x,1) y_l(x) + p(x,2) y_l'(x)
 + q(x,2)*( p(x,1) y_r(x) + p(x,2) y_r'(x) = v(x)
where q weighs the waves travelling from left to right and right to
     left (default [1 1])
```

## 61.2 bvp2wavetrain

```
solve second order boundary value problem by repeated integration
```

## 61.3 bvp2wavetwopass

```
two pass solution for the linearised wave equation
solve first for the wave number k, and then for y
```

## 61.4 ivp_euler_forward

```
solve intial value problem by the euler forward method
```

## 61.5 ivp_euler_forward2

### 61.6 ivprk2

```
solve initial value problem by the two step runge kutta method
```

### 61.7 ode2_matrix

```
transformation matrix of second order ode
to left and right going wave

c = odefun(x)
c1 y'' + c2' y + c3 y == 0
y = y_p + y_m, left and right going wave
d/dx [y_p, y_m] = A*[y_m, y_p]
```

### 61.8 ode2characteristic

```
second order odes
transmittded and reflected wave
```

### 61.9 step_trapezoidal

```
single trapezoidal step
```

### 61.10 test_bvp2

## 62 numerical-methods/optimisation

### 62.1 aitken_iteration

### 62.2 anderson_iteration

## 62.3 armijo_stopping_criterion

```
armijo stopping criterion for optimizations
```

## 62.4 astar

```
astar path finding alforithm
```

## 62.5 binsearch

```
binary search on a line
```

## 62.6 bisection

```
bisection
```

## 62.7 box1

```
test objective function for optimisation routines
```

## 62.8 box2

## 62.9 cauchy

## 62.10 cauchy2

```
solve non-linear system by cuachy's method
slower than quadratic optimisation, but does not require a hessian
fun : objective function, returns
      f  : scalar, objective function value
      g  : nx1, gradient
x   : nx1, initial position
opt : options
```

### 62.11 directional_derivative

```
directional (projected) derivative
d : derivative, highest first
p : series expansion around x0
```

### 62.12 dud

```
optimization by the dud algorithm
```

### 62.13 extreme3

```
extract maxima by quadratic approximation from sampled function val
    (t)
intended to be called after [mval, mid] = max(val) for refinement
    of
locatian and maximum

input
t    : sampling time (uniformly spaced)
v    : values at sampling times
ouput:
tdx  : index where extremum should be computed
t0   : location of the extremum
val0 : value of extremum

v'(dt0) = 0 and v''(dt0) determines type of extremum
```

### 62.14 extreme_quadratic

### 62.15 ftest

### 62.16 fzero_bisect

### 62.17 fzero_newton

### 62.18 grad

```
numerical gradient
```

### 62.19 hessian

```
numerical hessian
```

### 62.20 hessian_from_gradient

```
numerical hessian from gradient
```

### 62.21 hessian_projected

```
numerical hessian projected to one dimenstion
```

### 62.22 line_search

```
bisection routine
```

### 62.23 line_search2

```
bisection method

fun : objective funct
x0  : start value
f0  : objective function value at x0
g   : gradient at x0
p   : search direction from x0 (p = g for steepest descend)
h   : initial step length (default 1)
lb  : lower bound for x
up  : upper bound for x
```

### 62.24 line_search_polynomial

```
polynomial line search
fun : objective funct
x0  : start value
f0  : objective function value at x0
g   : gradient at x0
dir : search direction from x0 (p = g for steepest descend)
h   : initial step length (default 1)
lb  : lower bound for x
up  : upper bound for x
```

### 62.25 line_search_polynomial2

```
cubic line search
fun : objective funct
x0  : start value
f0  : objective function value at x0
g   : gradient at x0
dir : search direction from x0 (p = g for steepest descend)
h   : initial step length (default 1)
lb  : lower bound for x
up  : upper bound for x
```

### 62.26 line_search_quadratic

```
quadratic line search
fun : objective funct
x0  : start value
f0  : objective function value at x0
g   : gradient at x0
dir : search direction from x0 (p = g for steepest descend)
h   : initial step length (default 1)
lb  : lower bound for x
up  : upper bound for x
```

### 62.27 line_search_quadratic2

```
quadratic line search
```

### 62.28 line_search_wolfe

```
line search by wolfe method
c.f.: OPTIMIZATION THEORY AND METHODS - Nonlinear Programming, Sun,
     Yuan
```

### 62.29 ls_bgfs

```
least squares by the bgfs method
```

### 62.30 ls_broyden

```
least squares by the broyden method
for rectangular / non symmetric systems
      Numerical Optimization nocedal
      Practical Methods of  Optimization fletcher
c.f. gerber 1981
c.f. fletcher 1978 (more advanced, not used here)
c.f. Kelley 1999 ch. 4

BGFS:
Broyden 1965
Fletcher 1970
Goldfarb 1970
Shanno 1970
```

### 62.31 ls_generalized_secant

```
least squares by the secant method
Barnes, 1965
Wolfe, 1959
Fletcher 1980, 6.3
seber 2003
gerber
```

### 62.32 nlcg

```
non-linear conjugate gradient
input:
x   : nx1 start vectort
```

```
opt : struct options
fdx : gradient constraint
```

### 62.33 nlls

```
non-linear least squares
```

### 62.34 picard

```
picard iteration
```

### 62.35 poly_extrema

```
extrema of a polynomial
```

### 62.36 quadratic_function

```
evaluate quadratic function in higher dimensions
```

### 62.37 quadratic_programming

```
optimize by quadratic programming
```

### 62.38 quadratic_step

```
single step of the quadratic programming
```

### 62.39 rosenbrock

```
rosenbrock test function
```

### 62.40 sqrt_heron

```
Heron's method for the square root
```

### 62.41 test_directional_derivative

### 62.42 test_dud

### 62.43 test_fzero_newton

### 62.44 test_line_search_quadratic2

### 62.45 test_ls_generalized_secant

### 62.46 test_nlcg_6_order

### 62.47 test_nlls

```
f = w'*(p*abs(x-1).^4) + w'*(1-p)*abs(x-1).^2;
```

## 63 numerical-methods/pde

### 63.1 laplacian2d_fundamental_solution

# 64 numerical-methods/piecewise-polynomials

## 64.1 Hermite1

```
hermite polynomial interpolation in 1d
```

## 64.2 hp2_fit

```
fit a hermite polynomial
coefficients are derivative free
x0  : left point of first segment
x1  : right point of last segment
n   : number of segments
x   : sample x-value
val : sample y-value
c   : coefficients (values at points, no derivatives)
```

## 64.3 hp2_predict

```
prediction with pw hermite polynomial
c are values at support points
```

## 64.4 hp_predict

```
predict with piecewise hermite polynomial
```

## 64.5 hp_regress

```
fit piecewise hermite polynomial
coefficients are values and derivatives
```

## 64.6 lp_count

```
lagrangian basis for interpolation
count number of valid samples
```

### 64.7 lp_predict

```
lagrangian basis piecwie interpolation, predicor
```

### 64.8 lp_regress

### 64.9 lp_regress_

## 65 numerical-methods

### 65.1 test_adams_bashforth

## 66 patterns

### 66.1 band_pattern

### 66.2 hexagonal_pattern

## 67 regression/@PolyOLS

### 67.1 PolyOLS

```
class for polynomial least squares
```

### 67.2 coefftest

### 67.3 detrend

```
detrending by polynomial regression
```

### 67.4 fit

```
fit a polynomial function
like polyfit, but returns parameter error estimates
TODO automatically activate scaleflag
```

### 67.5 fit_

```
fit a polynomial function
```

### 67.6 predict

```
predict polynomial function values
```

### 67.7 predict_

### 67.8 slope

```
slope by linear regression
```

## 68 regression/@PowerLS

### 68.1 PowerLS

```
class for power law regression
```

### 68.2 fit

```
fit a power law
like polyfit, but returns parameter error estimates
```

### 68.3 predict

```
predict with power law
      S2 = diag((A*obj.C)*A');
      L  = Y - S;
      U  = Y + S;
```

### 68.4 predict_

## 69 regression/@Theil

### 69.1 Theil

```
Kendal-Theil-Sen robust regression
```

### 69.2 detrend

```
linear detrending of a set of samples by the Theil-Senn Slope
```

### 69.3 fit

```
fit slope and intercept to a set of sample with the Theil-Sen
    method

c    : confidence interval c = 2*ns*normcdf(1) for ns-sigma
    intervals
param : itercept and slope
P : confidence interval
```

### 69.4 predict

```
predict values and confidence intervals with the Theil-Sen method
```

### 69.5 slope

```
fit the slope with the Theil-Sen method
```

## 70 regression

```
linear and non-linear regression
```

### 70.1 Theil_Multivariate

```
extension of the Theil-Senn regression to higher dimensions by
means of the Gauss-Seidel iteration
```

### 70.2 areg

```
regression using the pth-fraction of samples with smallest residual
```

### 70.3 ginireg

```
gini regression
```

### 70.4 hesssimplereg

```
hessian, gradient and objective function value of the simple
    regression
rhs = p(1) + p(2) x + eps
```

### 70.5 l1lin

```
solve ||Ax - b||_L1 by means of linear programming
```

## 70.6 lsq_sparam

```
parameter covariance of the least squares regression

fun : model function for predtiction
b   : sample values
f(p) = b
p   : parameter at point of evaluation (preferably optimum)
```

## 70.7 polyfitd

```
fit a polynomial of order n to a set of sampled values and sampled
    values
of the derivative

x0 must contain at least for conditioning as otherwise the
    intercept
cannot be determined
```

## 70.8 regression_method_of_moments

```
fit linear function ||a b x = y||_L2 by the method of moments
y+eps = alpha + beta*x
```

## 70.9 robustlinreg

```
fit a linear function by splitting the x-values at their median
      (med(y_left) - med(y_right))/(med(x_left)-med(x_right)
this approach performs poorly compared to the theil-senn operator
```

## 70.10 theil2

```
Theil senn-estimator for two dimensions (glm)
```

## 70.11 theil_generalised

```
generalization of the Theil-Senn operator to higher dimensions,
for arbitrary functions such as polynomials and multivariate
    regression
either higher order polynomials or glm
c.f. "On theil's fitting method", Pegoraro, 1991
```

### 70.12 total_least_squares

```
total least squares
```

### 70.13 weighted_median_regression

```
weighted median regression
c.f. Scholz, 1978
```

## 71 set-theory

### 71.1 issubset

```
test if set B is subset of A in O(n)-runtime

A : first set
B : second set
P : set of primes (auxiliary)
```

## 72 mathematics

mathematical functions of various kind

### 72.1 shuffle_index

## 73 signal-processing

### 73.1 acfar1

```
Autocorrelation function of the finite AR1 process

a_k = 1/(n-k)sum x_ix_i+1 + (xi + xi+k)mu + mu^2
    = r^k + 1/n sum_ij + 1/n
       pause
```

### 73.2 acfar1_2

```
autocorrelation of the ar1 process
```

### 73.3 acfar2

```
impulse response of the ar2 process
```

### 73.4 acfar2_2

```
autocorrelation of the ar2 process
X_i + a1 X_i-1 + a2 X_i-2 = 0
```

### 73.5 ar1_cutoff_frequency

### 73.6 ar1_effective_sample_size

```
effective sample size correction for autocorrelated series
```

### 73.7 ar1_mse_mu_single_sample

```
standard error of a single sample of an ar1 correlated process
```

### 73.8 ar1_mse_pop

```
variance of the population mean of a single realisation around zero

E[(mu_N-0)^2] = E[mu_N^]
```

## 73.9 ar1_mse_range

```
mean standard error of the mean of a range of values taken from an
    ar1 process
```

## 73.10 ar1_spectrum

```
spectrum of the ar1 process
```

## 73.11 ar1_to_tikhonov

```
convert ar1 correlation to tikhonovs lambda
```

## 73.12 ar1_var_factor

```
variance correction factor for an autocorrelated finite process
n   : [1 .. inf] population size
m   : [1 .. n]  samples size
rho : [ -1 < rho < 1 (for convergence) ] correlation of samples
```

## 73.13 ar1_var_factor_

```
variance of an autocorrelated finite process
```

## 73.14 ar1_var_range2

```
variance of sub sample starting at the end of the series

from the finite length first order autocorrelated process

s2 = 1/m^2 sum_i^m sum_j^m rho^-|i-j|
```

## 73.15 ar1delay

```
approximate acf by the ar1 process
acf: autocovariance or autocorrerlation function
nf : skip first samples (for mixed geometric-arithmetic series (
    ARMA)
```

## 73.16 ar1delay_old

```
autocorrelation of the residual
```

## 73.17 ar2_acf2c

```
determine coefficients of the ar2 process from the first two lags
    of the
autocorrelation function
```

## 73.18 ar2conv

```
coefficients of the ar2 process determined from the two leading
    correlations
of the acf [1,r1,r2,...]
```

## 73.19 ar2dof

```
effective samples size for the ar2 process
```

## 73.20 ar2param

```
ar2 parameter estimation from first two terms of acf

acf = [1 a1 a2 ...]
```

## 73.21 asymwin

```
creates asymmetrical filter windows
filter will always have negative weights
```

# 74 signal-processing/autocorr

## 74.1 autocorr2

## 74.2 autocorr_bandpass

## 74.3 autocorr_brownian_phase

## 74.4 autocorr_brownian_phase_2d

## 74.5 autocorr_brownian_phase_across

## 74.6 autocorr_decay_rate

```
estimate exponential decay of the autocorrelation
```

## 74.7 autocorr_effective_sample_size

```
effective sample size from acf
```

## 74.8 autocorr_fft

```
estimate sample autocorrelation function
```

## 74.9 autocorr_genton

```
autocorrelation function
```

### 74.10 autocorr_highpass

### 74.11 autocorr_lowpass

### 74.12 autocorr_periodic_additive_noise

### 74.13 autocorr_periodic_windowed

### 74.14 autocorr_radial

## 75 signal-processing

### 75.1 average_wave_shape

```
extract waves with varying length from a wave train and and average
     their shape
```

### 75.2 bandpass

```
bandpass filter
```

### 75.3 bandpass1d

### 75.4 bandpass1d_fft

```
filter input vector with a spatial (two-sided) bandpass in fourier
    space
```

### 75.5 bandpass1d_implicit

### 75.6 bandpass2

```
bandpass filter
```

### 75.7 bandpass2d

### 75.8 bandpass2d_2

### 75.9 bandpass2d_fft

### 75.10 bandpass2d_ideal

### 75.11 bandpass2d_implicit

### 75.12 bandpass2d_iso

## 75.13 bandpass_arg

```
determine correlation coefficient from frequency of mode for the
    symmetric
```

## 75.14 bandpass_f0_to_rho

```
correlation coefficient for the pth-order symmetric bandpass filter
     with
maximum at f0 (when rho_lp = rho_hp)
```

## 75.15 bandpass_max

## 75.16 bandpass_max2

## 75.17 bartlett

```
Effective sample size factor for bartlett window
c.f. thiebaux
c.f spectral analysis-jenkins, eq. (6.3.27)
c = acf
note: results seams always to be 1 tac too low
T : reduction factor for dof
for ar1 with a = rho^k = exp(-k/L), T = 2L
```

## 75.18 bin1d

```
bin values of v sampled at x into bins bounded by "edges"
apply function v to it
```

### 75.19 bin2d

```
bin values of V sampled at X and Y into the grid structured grid ex
    ,ey
apply function func to all walues in the bin
func = mean : default
func = sum : non-normalized frequency histogram in 2D
```

### 75.20 binormrnd

```
generate two correlated normally distributed vectors
```

### 75.21 coherence

### 75.22 conv1_man

```
convolutions with padding
```

### 75.23 conv2_man

```
convolution in 2d
```

### 75.24 conv2z

### 75.25 conv30

```
convolve with rectangular window of lenght n
circular boundaries
```

### 75.26 conv_

```
convolution of a with b
```

## 75.27 conv_centered

```
convolve x with filter window f
when length of f is even, this guarantees a symmetric result (no
    off by on
displacement) by making the lenght of f odd at first
```

## 75.28 convz

## 75.29 cosexpdelay

## 75.30 csmooth

```
smooth recursively with [1,2,1]/4 kernel
function x = csmooth(x,n,p,circ)
```

## 75.31 daniell_window

```
Daniell window for smoothing the power spectrum
c.f. Daniell 1946
Bloomfield 2000
meko 2015
```

## 75.32 danielle_window

```
danielle fourier window
```

## 75.33 db2neper

```
convert decibel to neper
```

## 75.34 db2power

```
power ratio from db
```

## 75.35 derive_bandpass_normalization_and_zeros

## 75.36 derive_danielle_weight

## 75.37 derive_limit_0_acfar

## 75.38 detect_peak

```
detect peaks in a vector
requires function value to fall to p*max before new value is
    allowed
```

## 75.39 determine_phase_shift

## 75.40 determine_phase_shift1

```
average phase and phase shift per time step of a train of waves
```

## 75.41 digital_low_pass_filter

```
design coefficients of a low pass filter with specified cut of
    frequency
and sampling period
alalogue low pass with pole at s=-omega_c=1/tau=1/RC
Ha = tau/(tau + s) = 1/(1 + omega_c*s)
```

### 75.42 doublesum_ij

```
double sum of r^i
```

### 75.43 effective_sample_size_to_ar1

```
convert effective sample size to ar1 correlation
```

### 75.44 filt_hodges_lehman

### 75.45 filter1

```
filter along one dimension
```

### 75.46 filter2

```
filter columns of x (matlab does only support vector input)
```

### 75.47 filter_

```
invalidate values that exceed n-times the robust standard deviation
```

### 75.48 filter_r_to_f0

### 75.49 filter_rho_to_f0

### 75.50 filter_twosided

## 75.51 filteriir

```
filter adcp t-n data over time

v : nz,nt  : values to be filtered
H : nt,1   : depth of ensemble
last : nt,1 : last bin above bottom that can be sampled without
    side lobe interference
nf : scalar : number of reweighted iterations

when samples
- distance to bed is reference (advantageous for near-bed suspended
     transport)
TODO for wash load: distance to surface is more relevant
interpolate depending on z

when depth changes, neighbouring indices do not correspond to same
    relative position in the water column
relative poisition in the colum (s-coordinate) smoothes values
near the bed: absolute distance to bed is chosen
near surface: absolute distance to surface is chosen
-> cubic transformation of index

faster and avoid alising (smoothing along z)
      resample ensemble to same number of bins in S -> filter ->
          resample back
      use nonlinear transform z-s coordinates
-> resampling has to be local (Hi -> H-filtered)

filtered profile coordinates to sample coordinates
      zf -> zi (special transform)
corresponding indices and fractions
filtration step (update of hf and vf)
sample coordinates to updated profile coordinates
(the inverse step is actually not necessary)
write filtered value
```

## 75.52 filterp

## 75.53 filterp1

```
fir filter with some fancy extras
```

### 75.54 filterstd

### 75.55 firls_man

```
design finite impulse response filter by the least squares method
```

### 75.56 fit_spectral_density

```
fit spectral densities
```

### 75.57 fit_spectral_density_2d

```
fit spectral densities
```

### 75.58 fit_spectral_density_radial

```
fit spectral densities
```

### 75.59 flattopwin

```
the flat top window
```

### 75.60 frequency_response_boxcar

```
frquency response of a boxcar filter
```

### 75.61 freqz_boxcar

```
frequncy response of a boxcar filter
```

### 75.62 gaussfilt1

```
filter data series with a gaussian window
```

### 75.63 hanchangewin

```
hanning window for change point detection
```

### 75.64 hanchangewin2

```
nanning window for chage point detection
```

### 75.65 hanwin

```
hanning filter window
```

### 75.66 hanwin_

```
hanning filter window
```

### 75.67 high_pass_1d_simple

### 75.68 highpass

```
high pass filter
```

### 75.69 highpass1d_fft_cos

```
filter the input vector with a cosine-shaped highpass in frequency
    space
```

## 75.70 highpass1d_implicit

## 75.71 highpass2d_fft

## 75.72 highpass2d_ideal

## 75.73 highpass2d_implicit

## 75.74 highpass_arg

## 75.75 highpass_fc_to_rho

## 75.76 jackknife_block

## 75.77 kaiserwin

```
kaiser filter window
```

## 75.78 kalman

```
Kalman filter
```

## 75.79 lanczoswin

```
Lanczos window
```

## 75.80 last

```
lake tail, but for matrices
```

## 75.81 lowpass

```
low pass filter
```

## 75.82 lowpass1d_fft

## 75.83 lowpass1d_implicit

## 75.84 lowpass2

```
design low pass filter with cutoff-frequency f1
```

## 75.85 lowpass2d_2

## 75.86 lowpass2d_anisotropic

## 75.87 lowpass2d_fft

### 75.88 lowpass2d_ideal

### 75.89 lowpass2d_implicit

### 75.90 lowpass_arg

### 75.91 lowpass_fc_to_rho

### 75.92 lowpass_iir

```
iir-low pass
```

### 75.93 lowpass_iir_symmetric

```
two-sided iir low pass filter (for symmetry)
```

### 75.94 lowpassfilter2

```
low-pass filter of data
```

### 75.95 maxfilt1

### 75.96 meanfilt1

```
moving average filter with special treatment of the boundaries
```

### 75.97 meanfilt2

```
filter with a rectangular window along both dimensions
```

### 75.98 medfilt1_man

```
moving median filter, supports columnwise operation
```

### 75.99 medfilt1_man2

```
moving median filter with special treatment of boundaries
```

### 75.100 medfilt1_padded

```
median filter with padding
```

### 75.101 medfilt1_reduced

```
median filter with padding
```

### 75.102 mid_term_single_sample

```
variance of single sample, mid term
```

### 75.103 minfilt1

### 75.104 mu2ar1

```
error variance of the mean of the finite length ar1 process
```

```
(mu)^2 = (sum epsi)^2 = sum_i sum_j eps_i eps_j = sum_ii(rho,n)/n^2
this has the limit s^2 for rho->1
```

### 75.105 mysmooth

### 75.106 nanautocorr

```
autocorrelation with nan-values
```

### 75.107 nanmedfilt1

```
medfilt1, skipping nans
```

### 75.108 neper2db

```
convert neper to db
```

### 75.109 oscillator_noisy

### 75.110 peaks_man

```
peaks of a periodogram
```

### 75.111 periodogram

```
compute the normalized periodogram
```

### 75.112 periodogram_2d

```
compute the normalized periodogram in two dimensions
```

### 75.113 periodogram_annular

### 75.114 periodogram_bartlett

```
estimate the spectral density nonparametrically with Bartlett's
    method
```

### 75.115 periodogram_bootstrap

### 75.116 periodogram_confidence_interval

```
confidence interval for periodogram values
```

### 75.117 periodogram_median

### 75.118 periodogram_p_value

### 75.119 periodogram_qq

```
qq-plot of a spectral density estimate by smoothing against the
    expected
beta-density
```

### 75.120 periodogram_quantiles

```
quantiles of a periodogram
```

### 75.121 periodogram_radial

## 75.122 periodogram_std

```
standard deviation of a periodogram
```

## 75.123 periodogram_test_periodicity

```
test a periodogram for hidden periodic frequency components

function [p,ratio,maxShat,mdx,fdx,S] = periodogram_test_periodicity
    (fx,Shat,nf,fmin,fmax,S,mode)
input:
       fx : frequengcies
       Shat : corresponding periodogram values
     nf   : number of bins to test for periodicity, ignored when S
         is given
       fmin, fmax : frequency range limits to test
       S    : exact (a priori known theoretical spectral density,
           must not be estimated from the periodogram)
       mode : automatically set to "exact", when S given
             inclusive : estimate density by smoothing including the
                  central bin
             exclusive : estimate density by smoothing excluding the
                  central bin
       note: inclusive and exclusive lead to different distribution
            but identical p-values
```

## 75.124 periodogram_test_periodicity_2d

```
test a periodogram for hidden periodic frequency components

function [p,ratio,maxShat,mdx,fdx,S] = periodogram_test_periodicity
    (fx,Shat,nf,fmin,fmax,S,mode)
input:
       fx : frequengcies
       Shat : corresponding periodogram values
     nf   : number of bins to test for periodicity, ignored when S
         is given
       fmin, fmax : frequency range limits to test
       S    : exact (a priori known theoretical spectral density,
           must not be estimated from the periodogram)
       mode : automatically set to "exact", when S given
             inclusive : estimate density by smoothing including the
                  central bin
```

```
        exclusive : estimate density by smoothing excluding the
             central bin
  note: inclusive and exclusive lead to different distribution
        but identical p-values
```

### 75.125 periodogram_test_stationarity

```
test a periodogram for statoinarity
note : the method works, but is of little practical use,
as it requires about 50 periods and a small dx to detect a
    frequency change by a factor of 2
```

### 75.126 periodogram_welsh

### 75.127 polyfilt1

```
polynomial filter,
can be achieved by iteratively processing the data with
a mean (zero-order) filter
```

### 75.128 qmedfilt1

```
medfilt1, after fitting a quadratic polynomial
```

### 75.129 radial_window

```
radial filter window in the 2d-frequency domain
```

### 75.130 randar1

```
generate random ar1 process
e1 = randar1(sigma,p,n,m)
```

### 75.131 randar1_dual

```
draw random variables of two corrlated ar1 processes
```

### 75.132 randar2

```
generate ar2 process
```

### 75.133 randarp

```
randomly generate the instance of an ar-p process
```

### 75.134 range_window

```
range of values within a certain range of indices (window)
```

### 75.135 rectwin

```
rectangular window
```

### 75.136 recursive_sum

### 75.137 select_range

### 75.138 smooth1d_parametric

```
smooth position of p0=x0,y0 between p1=x1,y1 and p2=x2,y2,
so that distance to p1 and p2 becomes equal
and the chord length remains the same
```

### 75.139 smooth2

```
smooth vectos of X
```

### 75.140 smooth_man

### 75.141 smooth_parametric

```
smooth a parametric function given in x-y coordinates
      matvec2x2(R,[dxc;dyc])
```

### 75.142 smooth_parametric2

```
parametrically smooth the curve
```

### 75.143 smooth_with_splines

### 75.144 smoothfft

```
filter with fast fourier transform
```

## 76 signal-processing/spectral-density

### 76.1 spectral_density_ar2

### 76.2 spectral_density_area

```
integrate the spectral density
```

## 76.3 spectral_density_bandpass2d_ideal

## 76.4 spectral_density_bandpass_2d

## 76.5 spectral_density_bandpass_2d_scale

## 76.6 spectral_density_bandpass_2d_scale_old

## 76.7 spectral_density_bandpass_continuous

```
S : spectral density of the bandpass filter in continuos space
    limit case of the discrete bandpass for dx -> 0

function [S,IS] = spectral_density_bandpass_continous(fx,fc,order)
f     : frequency (abszissa)
fc    : central frequncy, location of maximum on abszissa
order : number of times filter is applied iteratively, not
    necessarily integer
```

## 76.8 spectral_density_bandpass_continuous_max

```
maximum of the bandpass spectral density
```

## 76.9 spectral_density_bandpass_continuous_max2par

```
transform mode (maxima) of the bandpass spectral density into the
    paramter
of the underlying distribution
```

### 76.10 spectral_density_bandpass_continuous_scale

```
normaliztation scale of the spatial bandpass density
```

### 76.11 spectral_density_bandpass_discrete

```
spectral density of the discrete spatial (two-sided) bandpass
    filter
```

### 76.12 spectral_density_brownian_phase

```
spectral density of a fourier series where the phase undergoes
    brownian motion
with standard deviation s per unit distance
```

### 76.13 spectral_density_brownian_phase_2d

### 76.14 spectral_density_brownian_phase_across

### 76.15 spectral_density_brownian_phase_mode

```
mode (maximum) of the spectral density of the fourier series with
    brownian phase
```

### 76.16 spectral_density_brownian_phase_mode2par

```
transform mode to parameters of the brownian phase spectral density
```

### 76.17 spectral_density_brownian_phase_scale

```
normalization scale of the brownian phase spectral density
```

## 76.18 spectral_density_estimate_2d

## 76.19 spectral_density_flat

```
flat spectral density of a random vector woth iid elements
```

## 76.20 spectral_density_highpass

## 76.21 spectral_density_highpass2d_ideal

## 76.22 spectral_density_highpass_2d

## 76.23 spectral_density_highpass_cos

```
consine shaped spectral density of a highpass filter
```

## 76.24 spectral_density_lorentzian

```
lorentzian spectral density
```

## 76.25 spectral_density_lorentzian_max

```
mode (maximum) of the lorentzian spectral density
```

## 76.26 spectral_density_lorentzian_max2par

```
transform maximum of the lorentzian spectral density to its
    distribution parameters
```

### 76.27 spectral_density_lorentzian_scale

```
normalization scale of the lorentzian spectral density
```

### 76.28 spectral_density_lowpass

### 76.29 spectral_density_lowpass2d_ideal

### 76.30 spectral_density_lowpass_2d

### 76.31 spectral_density_lowpass_one_sided

### 76.32 spectral_density_periodic_additive_noise

### 76.33 spectral_density_wperiodic

## 77 signal-processing

### 77.1 spectrogram

```
spectrogram
```

### 77.2 std_window

```
moving block standard deviation
```

### 77.3 sum_i_lag

```
sum of ar1 matrix with lag
sum_i=1^n rho^|i-k|
```

### 77.4 sum_ii

```
sum of ar1 matrix
sum_i=1^n sum_j=1^n rho^|i-j|
this is for the variance, take square root for the standard
    deviation factor
```

### 77.5 sum_ii_

### 77.6 sum_ij

```
sum of ar1 matrix
sum_{i=1}^n sum_{j=1}^m r^|i-j|
```

### 77.7 sum_ij_

### 77.8 sum_ij_partial_

### 77.9 sum_multivar

```
sum of matrix entries of bivariate ar1 process
```

### 77.10 test_acfar1

### 77.11 test_acfar1_2

### 77.12 test_acfar1_3

### 77.13 test_acfar1_4

### 77.14 test_acfar2

### 77.15 test_ar1_var_factor

### 77.16 test_ar1_var_factor_2

### 77.17 test_ar1_var_mu_single_sample

### 77.18 test_ar1_var_pop

### 77.19 test_ar1_var_pop_1

### 77.20 test_ar1delay

### 77.21 test_bivariate_covariance_term

### 77.22 test_convexity

### 77.23 test_lanczoswin

### 77.24 test_madcorr

### 77.25 test_randar1

### 77.26 test_randar1_multivariate

### 77.27 test_randar2

### 77.28 test_sum_ij

### 77.29 test_sum_multivar

### 77.30 test_trifilt1

### 77.31 test_wautocorr

### 77.32 test_wavelet_transform

### 77.33 test_wordfilt

### 77.34 test_xar1_mid_term

### 77.35 tikhonov_to_ar1

```
convert coefficient of the tikhonov regularization to correlatioon
    of the ar1 process
```

### 77.36 trapwin

```
trapezoidal filter window
```

### 77.37 trifilt1

```
filter with triangular window
trifilt1 is ident to twice applying rectfilt1 (meanfilt1) with half
     the domain size
note : inifnitely many convolution yield a gaussian
```

### 77.38 trifilt2

```
filter with a triangular window along both dimensions
```

## 77.39 triwin

```
triangular filter window
```

## 77.40 triwin2

```
triangular filter window
```

## 77.41 varar1

```
error variance of a single sample of a finite length ar1 process
with respect to the mean, averaged over the population
```

## 77.42 welch_spectrogram

```
welch spectrogram
```

## 77.43 wfilt

```
filter with window
```

## 77.44 winbandpass

```
filter with bandpass
```

## 77.45 window2d

## 77.46 window_make_odd

### 77.47 winfilt0

```
filter with window
```

### 77.48 winlength

```
window length for desired cutoff frequency
power at fc is halved
H(wf) = 1/sqrt(2) H(f)
if the filter window were used as a low pass filter
note: the user should prefer a windowed ideal low pass filter
TODO, relate this to DOF
```

### 77.49 wmeanfilt

```
mean filter with window
```

### 77.50 wmedfilt

```
median filter with window
```

### 77.51 wordfilt

```
weighted order filter
```

### 77.52 wordfilt_edgeworth

```
weighed order filter
```

### 77.53 wrapphase

### 77.54 xar1

### 77.55 xcorr_man

```
cross correlation of two sampled ar1 processes
```

## 78 sorting

### 78.1 sort2

```
sort two numbers
```

### 78.2 sort2d

```
sort elements of matrix in X
returns row and column index of sorted values
```

## 79 special-functions

### 79.1 bessel_sphere

```
spherical Bessel function of the first kind
```

### 79.2 digamma_man

### 79.3 exp10

### 79.4 hankel_sphere

```
spherical Hankel function for the far field (incident plane wave)
first kind
```

### 79.5 hermite

```
probabilistic's hermite polynomial by recurrence relation

input :
n : order
x : value

output:
f : H_n(x)
df : d/dx H_n(x)
```

### 79.6 legendre_man

```
legendre polynomials
```

### 79.7 neumann_sphere

```
spherical Neumann function
Bessel function of the second kind
```

## 80 statistics

### 80.1 atan_s2

```
stadard deviation of the arcus tangens by means of taylor expansion
```

### 80.2 beta_kurt

### 80.3 beta_mean

### 80.4 beta_mode_to_parameter

```
transform modes (mean and sd) to paramets of the beta function
```

### 80.5 beta_skew

### 80.6 beta_std

### 80.7 chi2_kurt

### 80.8 chi2_mean

### 80.9 chi2_skew

### 80.10 chi2_std

### 80.11 coefficient_of_determination

### 80.12 conditional_expectation_normal

### 80.13 correlation_confidence_pearson

```
confience intervals of the correlation coefficient
c.f. Fischer 1921
```

# 81 statistics/distributions

## 81.1 PDF

```
class for quasi-distributions from a set of sampling points
```

## 81.2 binorm_separation_coefficient

```
separation coefficient of a bimodal normal distribution
```

## 81.3 binormcdf

```
bio-modal gaussian distribution
```

## 81.4 binormfit

```
fit sum of to normal distribution to a histogram
```

## 81.5 binormpdf

## 81.6 edgeworth_cdf

```
edgeworth expansion of an unknown cumulative distribution
with mean mu, standard deviation sigma, and third and fourth
    cumulants
c.f. Rao 2010
```

## 81.7 edgeworth_pdf

```
probability density of and unknown distribution
with mean mu, standard deviation sigma, and third and fourth
    cumulants
c.f. Rao 2010
```

## 81.8 gam_moment2param

## 81.9 logn_mean

## 81.10 logn_mode

```
mode (maximum) of the log-normal density
```

## 81.11 logn_moment2param

```
transform the mode (mu,sd) to parameters of the log normal
    distribution
```

## 81.12 logn_param2moment

```
transform parameters to mode (mu, sd) for the log normal
    distribution
```

## 81.13 logn_std

## 81.14 lognpdf_

```
log normal distribution called by modes rather than parameters
```

## 81.15 pdfsample

```
pdf from sample distribution
Note: better use kernal density estimates
```

### 81.16 t2cdf

```
Hotelling's T-squared cumulative distribution
```

### 81.17 t2inv

```
inverse of Hotelling's T-squared cumulative distribution
```

## 82 statistics

### 82.1 example_standard_error_of_sample_quantiles

### 82.2 f_var_finite

```
reduction of variance when sampling from a finite population
without replacement
```

### 82.3 fisher_mean

### 82.4 fisher_std

### 82.5 gam_mean

### 82.6 gam_std

### 82.7 gamma_mode_to_parameter

```
transform modes (mu,sd) to parameters of the gamma distribution
```

### 82.8 gamma_stirling

### 82.9 gaussfit3

### 82.10 gaussfit_quantile

### 82.11 geoserr

### 82.12 geostd

### 82.13 hodges_lehmann_correlation

```
hodges_lehmann correlatoon coefficient
c.f. Shamos 1976
c.f. Bickel and Lehmann 1976
c.f. rousseeuw 1993
c.f. Shevlyakov 2011
```

### 82.14 hodges_lehmann_dispersion

```
dispersion determined by the hodges lehman method
asymptotic efficiency of dispersion estimates:
standard deviation:   E(s - hat s)/s = 2/sqrt(2 n) ~ 0.707/sqrt(n)
               (100%)
hodges lehmann dispersion E(s-\hat s)/s = (pi/3)^2 /(sqrt(2 n)) ~
    0.775/sqrt(n) (91%)
mad                   E(s-\hat s)/s ~ 1.17 s/sqrt(n)
                   (60%)
c.f. Shamos 1976
c.f. Bickel and Lehmann 1976
c.f. rousseeuw 1993
nb: rousseeuw uses the 25th percentile, which is more efficient for
     small sample sizes
```

## 83 statistics/information-theory

### 83.1 akaike_information_criterion

```
akaike information criterion

serr : rmse of model prediction
n : effective sample size
k : number of parameters

c.f. akaike (1974)
c.f. sugiura 1978
```

### 83.2 bayesian_information_criterion

```
bayesian information criterion
```

## 84 statistics

### 84.1 kurtncdf

### 84.2 kurtnpdf

### 84.3 kurtosis_bias_corrected

```
bias corrected kurtosis
```

### 84.4 limit

```
limit a by lower and upper bound
```

### 84.5 logfactorial

```
approximate log of the factorial
```

### 84.6 loglogpdf

### 84.7 lognfit_quantile

### 84.8 logskewcdf

### 84.9 logskewpdf

## 85 statistics/logu

### 85.1 lambertw_numeric

```
lambert-w function
```

## 85.2 logtrialtcdf

```
pdf of a logarithmic triangular distribution
```

## 85.3 logtrialtinv

```
 inverse of the logarithmic triangular distribution
 = (d F log(a) log(b) + a log(b) - b log(a) - d F log(a) log(c) - a
     log(c) + d F log(b) log(c) + b log(c) - d F log^2(b))/((log(a)
     - log(b)) W((a^(-1/(log(a) - log(b))) (b^(-log(c)/log(a) - 1/
     log(a)) c)^(-log(a)/(log(a) - log(b))) (-d F log^2(b) + a log(b
     ) + d F log(a) log(b) + d F log(c) log(b) - b log(a) - a log(c)
      + b log(c) - d F log(a) log(c)))/(log(a) - log(b))))
x = (d F log(a) log(b) + a log(b) - b log(a) - d F log(a) log(c) - a
     log(c) + d F log(b) log(c) + b log(c) - d F log^2(b))/((log(a)
    - log(b)) W((a^(-1/(log(a) - log(b))) (b^(-log(c)/log(a) - 1/log
    (a)) c)^(-log(a)/(log(a) - log(b))) (-d F log^2(b) + a log(b) +
    d F log(a) log(b) + d F log(c) log(b) - b log(a) - a log(c) + b
    log(c) - d F log(a) log(c)))/(log(a) - log(b))))
```

## 85.4 logtrialtmean

```
mean of the logarithmic triangular distribution
```

## 85.5 logtrialtpdf

```
density of the logarithmic triangular distribution
```

## 85.6 logtrialtrnd

## 85.7 logtricdf

```
cumulative distribution of the logarithmic triangular distribution
```

### 85.8 logtriinv

```
invere of the logarithmic triangular distribution
```

### 85.9 logtrimean

```
mean of the logarithmic triangular distribution
```

### 85.10 logtripdf

```
probability density of the logarithmic triangular distribution
```

### 85.11 logtrirnd

### 85.12 logucdf

```
probability density of the logarithmic uniform distribution
```

### 85.13 logucm

```
central moments of the log-uniform distribution
```

### 85.14 loguinv

```
inverse of the log-uniform distribution
```

### 85.15 logumean

```
mean of the log-uniform distribution
```

## 85.16 logupdf

```
pdf of the log uniform distribution
```

## 85.17 logurnd

```
random numbers following a log-uniform distribution
```

## 85.18 loguvar

```
variance of the log-uniform distribution
```

## 85.19 medlogu

```
median of the log-uniform distribution
```

## 85.20 test_logurnd

## 85.21 tricdf

```
cumulative distribution of the log-triangular distribution
```

## 85.22 triinv

```
inverse of the triangular distribution
```

## 85.23 trimedian

```
median of the triangular distribution
```

### 85.24 tripdf

```
probability density of the triangular distribution
```

### 85.25 trirnd

```
random numbers of the triangular distribution
```

## 86 statistics

### 86.1 max_exprnd

### 86.2 maxnnormals

```
expected maximum of n normal variables
c.f. Wolperts
this is the median, not the mean of the maximum!
see median of gumbel
```

### 86.3 mean_generalized_gampdf

### 86.4 midrange

```
mid range of columns of X
```

### 86.5 minavg

```
solution of the minimum variance problem
minimise the variance of the weighted sum of n-independent
random variables with equal mean and individual variance
```

### 86.6 mode_man

## 87 statistics/moment-statistics

### 87.1 autocorr_man3

```
autoccorrelation of the columns of X
```

### 87.2 autocorr_man4

```
autocorrelation for x if x is a vector, or indivvidually for the
columns of x if x is a matrix

c.f. box jenkins 2008 eq. 2.1.12

Note that it is faster to compute the acf in frequency space
as done in the matlab internal function
```

### 87.3 autocorr_man5

```
autocorrellation of the columns of X
```

### 87.4 blockserr

```
estimate the standard error of potetially sequentilly correlated
    data
by blocking
block length should be sufficiently larger than correlation length
and sufficiently smaller than data length
this uses a sliding block approach, which reduces the variation of
    the error estimate
```

### 87.5 comoment

```
non-central higher order moments of the multivariate normal
    distribution

c.f. Moments and cumulants of the multivariate real and complex
    Gaussian distributions

note : there seem to be some typos in the original paper,
       for x^4 cii^2, the square seems to be missing
mu : nx1 mean vector
C  : nxn covariance matrix
k  : nx1 powers of variables in moments
```

## 87.6 corr_man

```
correlation of two vectors
```

## 87.7 cov_man

```
covariance matrix of two vectors
```

## 87.8 dof

```
mininum number of support points
for a polynomial of degree order in dim dimensions
```

## 87.9 edgeworth_quantile

```
inverse edgeworth expansion
c.f. cornis fisher 1937
c.f. Rao 2010
c.f. 2.50 in hall
CHERNOZHUKOV 3.3
```

## 87.10 effective_sample_size

```
effective sample size of the weighted mean of uncorrelated data
c.f. Kish
```

## 87.11 f_correlation

```
correction factor for standard error of the mean of n ar1-
    correlated iid samples
```

## 87.12 f_finite

```
reduction factor of standard error for sampling from a finite
    distribution
without replacement
```

## 87.13 lmean

```
mean of x.^l, not of abs
```

## 87.14 lmoment

```
l-moment of vector x
```

## 87.15 maskmean

```
mean of the masked values of X
```

## 87.16 masknanmean

## 87.17 mean1

```
mean of x
```

## 87.18 mean_man

```
mean and standard error of X
```

### 87.19 mse

```
mean squared error of residual vector res
this is de-facto the std for an unbiased residual
```

### 87.20 nanautocorr_man1

```
autocorrelation of a vector with nan-values
```

### 87.21 nanautocorr_man2

```
autocorrelation of a vector with nan-values
```

### 87.22 nanautocorr_man4

```
compute autocorrelation for x if x is a vector, or indivvidually
    for the
columns of x if x is a matrix
box jenkins 2008 eq. 2.1.12
TODO nan is problematic!
Note that it is faster to compute the acf in frequency space
as done in the matlab internal function
```

### 87.23 nancorr

```
(co)-correlation matrix when samples a NaN
```

### 87.24 nancumsum

```
cumulative sum, setting nan values to zero
```

### 87.25 nanlmean

```
mean of the l-th power of the absolute value of x
```

### 87.26 nanr2

```
coefficient of determination when samples are invalid
```

### 87.27 nanrms

```
root mean square value when sample contains nan-values
```

### 87.28 nanrmse

```
root mean square error from vector of residuals
this is de-facto the std for an unbiased residual
```

### 87.29 nanserr

```
standard error of x with respect to mean when x contains nan values
```

### 87.30 nanwmean

```
weighted mean
min_x sum w (x-mu)^2 => mu = sum(wx)/sum(w)
varargin can be dim
function [mu serr] = nanwmean(w,x)
```

### 87.31 nanwstd

```
weighed standard deviation
```

### 87.32 nanwvar

```
weighted variance of columns, corrected for degrees of freedom (
    bessel)

s^2 = sum(w*(x-sum(wx)/sum(w))^2)/sum(w)
```

### 87.33 nanxcorr

### 87.34 pearson

```
pearson correlation coefficient
```

### 87.35 pearson_to_kendall

```
conversion of pearson to kendall correlation coefficient
c.f. Kruskal 1958
```

### 87.36 pool_samples

```
pooled mean and standard deviation of several groups of different
    size, mean and standard deviation
```

### 87.37 qmean

```
trimmed mean
```

### 87.38 range_mean

### 87.39 rmse_

```
root mean square error computed from a residual vector
this is de-facto the std for an unbiased residual
```

### 87.40 serr

```
standard error of the mean of a set of uncorrelated samples
```

## 87.41 serr1

## 87.42 test_qskew

## 87.43 test_qstd_qskew_optimal_p

## 87.44 wautocorr

```
autocorrelation for x if x is a vector, or indivvidually for the
columns of x if x is a matrix
samples can be weighted

c.f. box jenkins 2008 eq. 2.1.12

c.f. autocorr_man4

Note that it is faster to compute the acf in frequency space
as done in the matlab internal function
```

## 87.45 wcorr

```
correlation of two vectors when samples are weighted
```

## 87.46 wcov

```
covariance of two vectors when samples are weighted
```

## 87.47 wdof

```
effective degrees of freedom for weighted samples
```

### 87.48 wkurt

```
kurtosis with weighted samples
```

### 87.49 wmean

```
weighted mean

min_x sum w (x-mu)^2 => mu = sum(wx)/sum(w)

varargin can be dim
function [mu serr] = wmean(w,x)
```

### 87.50 wrms

```
weighted root mean square
```

### 87.51 wserr

```
weighted root mean square error
```

### 87.52 wskew

```
skewness of a weighted set of samples
function sk = wskew(w,x)
```

### 87.53 wstd

```
weighed standard deviation
```

### 87.54 wvar

```
weighted variance of columns, corrected for degrees of freedom (
    bessel)
variance of the weighted sample mean of samples with same mean (but
     not necessarily same variance)
s^2 = sum (w^2(x-sum(wx)^2))

s2_mu : error of mean, s2_mu : sd of prediction
```

# 88 statistics

## 88.1 nangeomean

## 88.2 nangeostd

```
geometric standard deviation ignoring nan-values
```

# 89 statistics/nonparametric-statistics

## 89.1 kernel1d

```
X   : ouput x axis bins
xi  : samples along x
m   : number of bins in X
fun : kernel function
pdf : propability density of xi
```

## 89.2 kernel2d

```
kernel density estimate in two dimensions
```

# 90 statistics

## 90.1 normmoment

```
expected norm of x.^n, when values x in x are iid normal with mu
    and sigma
```

### 90.2 normpdf2

```
pdf of the bivariate normal distribution
```

## 91 statistics/order-statistics

### 91.1 hodges_lehmann_location

```
hodges lehman location estimator

Asymptotic rms efficency of location estimte:
    mean:         1 s/sqrt(n)
    hodges lehman: sqrt(pi/3)*s ~ 1.0233 s/sqrt(n)
    median:       pi/2 s/sqrt(n) ~ 1.25 s / sqrt(n)
```

### 91.2 kendall

```
kendall correlation coefficient
```

### 91.3 kendall_to_pearson

```
convert kendall rank correlation coefficient to the person product
    moment
correlation coefficient

c.f. Kruska, 1985
```

### 91.4 mad2sd

```
transform median absolute deviation to standard deviation
for normal distributed values
```

### 91.5 madcorr

```
proxy correlation by median absolute deviation
```

### 91.6 median2_holder

### 91.7 median_ci

```
median and its confidence intervals under assumption of normality
se_me = sqrt(1/2 pi) 1.25331 * sd/sqrt(n)
```

### 91.8 median_man

```
median and confidence intervals
c is a P value for the confidence interval,
default is 0.95 (2-sigma)
median of the colums of X
```

### 91.9 mediani

```
index of median, if median is not unique, any of the values is
    chosen
```

### 91.10 nanmadcorr

```
proxy correlation by median absolute deviation
```

### 91.11 nanwmedian

```
weighted median, skips nan-values
```

### 91.12 nanwquantile

```
weighted quantile, skips nan values
```

### 91.13 oja_median

```
two dimensional oja median
note: the multivariate median is not unique

oja 1983, for extension to multivariate function, see chaudhri
```

### 91.14 qkurtosis

```
kurosis computed for quantiles

Note : this is a measurement of shape-tailedness and yields the
    same value for the
      normal distribution as "kurtosis"
      However, this is a separate statistic and hence requires
          different
      methods for calculating P-values and hypothesis testing
```

### 91.15 qmoments

```
moments estimated from quantiles
```

### 91.16 qskew

```
skewness estimated from quantiles

Note : this is a measurement of shape-symmetry and yields the same
    value for the
      skew-normal distribution as "skewness"
      However, this is an own statistic and hence requires
          different
      methods for calculating P-values and hypothesis testing
```

### 91.17 qskewq

```
skewness estimated by quantiles
```

### 91.18 qstdq

proxy standard deviation determined by quantiles

### 91.19 quantile1_optimisation

### 91.20 quantile2_breckling

qunatile regression

### 91.21 quantile2_chaudhuri

quantile regression

### 91.22 quantile2_projected

quantile in two dimensions

### 91.23 quantile2_projected2

spatial qunatile for chosen direction

### 91.24 quantile_envelope

### 91.25 quantile_regression_simple

simple quantile regression

## 91.26 ranking

```
ranking for spearman statistics
```

## 91.27 spatial_median

```
c.f. Oja 2008
is this the same as the oja simplex median (c.f. small 1990)?
```

## 91.28 spatial_quantile

```
spatial quantile
```

## 91.29 spatial_quantile2

```
spatial quantile
```

## 91.30 spatial_quantile3

```
spatial quantile
```

## 91.31 spatial_rank

```
unsigned rank
```

## 91.32 spatial_sign

```
spatial sign
```

## 91.33 spatial_signed_rank

```
signed rank
Note: this is only a true rank if X is normal with zero mean,
    abitrary variance
```

### 91.34 spearman

```
spearman's product moment coefficient
```

### 91.35 spearman_rank

### 91.36 spearman_to_pearson

```
conversion of spearman rank to person product moment correlation
    coefficient
```

### 91.37 wmedian

```
weighted median
```

### 91.38 wquantile

```
weighted quantile
```

## 92 statistics

### 92.1 qstd

### 92.2 quantile_extrap

### 92.3 quantile_sin

# 93 statistics/random-number-generation

## 93.1 laplacernd

```
random number of laplace distribution
```

## 93.2 randc

```
correlate to correlated standard normally distributed vectors
```

## 93.3 skewness2param

## 93.4 skewpdf_central_moments

## 93.5 skewrnd

```
random numbers of the skew normal distribution
```

## 93.6 skewrnd2

```
random numbers of the skew normal distribution
```

# 94 statistics

## 94.1 range

```
range and mid range of input
```

## 94.2 resample_with_replacement

# 95 statistics/resampling-statistics/@Jackknife

## 95.1 Jackknife

```
class for leave out 1 (delete 1) Jackknife estimates

note 1 : the 1-delete jackknife does not yield consistend estimates
     for all functions,
       in particular it will perform poorly on robust estimation
           functions
       this is overcome by the d-delete jacknife, where d has to
           exceed the breakdown point
       of the estimating function, for example sqrt(n) for the
           median
       as this leads to unreasonably large number of repetitions,
           bootstrap
       is recommended for large sample cases (or blocking for
           sequential data)
note 2 : as a linearisation, jackknife underestimates the error
    variance in case of
        dependence in the data
note 3 : studentisation and the leave out 1 jackknife are related
note 4 : the double 1 sample jacknife performs iferior to the d1
    jacknife
```

## 95.2 estimated_STATIC

```
jacknife estimate of mean, bias and standard error
theta0 : estimate from all samples
thetad : set of estimates obtained by leaving out one data point
    each
        last dimension of theta is assumed to be the jackknife
            dimension
```

## 95.3 matrix1_STATIC

```
matrix of estimation for leaving out two samples at a time
```

## 95.4 matrix2

```
matrix of estimations for jacknive with two samples left out
```

# 96 statistics/resampling-statistics

## 96.1 block_jackknife

## 96.2 jackknife_moments

```
moments determined by the jacknife

func : function of interest on the samples (e.g. mean)
A    : parameter matrix
       columns : parameters
       rows    : samples of the parameter sets
d   : number of samples left out
```

## 96.3 moving_block_jackknife

```
blocked Jacknfife for autocorrelated data
sliding block, statistically more efficient but computationally
    expensive
note, number of blocks must be sufficiently large h ~ sqrt(n)? << n
```

## 96.4 randblockserr

```
standard error of sequentilly correlated data by blocking
block length should be sufficiently larger than correlation length
and sufficiently smaller than data length
this uses a sliding block approach, which reduces the variation of
    the error estimate
TODO this does not work, randomly picking samples does not reveal
    the correlation
```

## 96.5 resample

```
resample a vector and apply function to it

TODO, should be with replacement

n  : number of samples
```

```
m  : number of subsamples
cx : maximum number of combinations
```

# 97 statistics

## 97.1 scale_quantile_sd

```
scale factor for the standard deviation
of the asymtpotic distibution of sample quantiles
(for normal distribution)
see cadwell, 1952
```

## 97.2 sd_sample_quantiles

## 97.3 skew_generalized_normal_fit

## 97.4 skew_generalized_normpdf

## 97.5 skewcdf

## 97.6 skewparam_to_central_moments

## 97.7 skewpdf

```
skew-normal distribution
c.f. Azzalini 1985
```

### 97.8 spatialrnd

### 97.9 test_mean_generalized_gampdf

### 97.10 test_skew_generalized_normpdf

### 97.11 trimmed_mean

```
trimmed mean
```

### 97.12 ttest2_man

```
two-sample t-test
here posix return value standard: h = 0 accepted, h = 1 failed
note: the matlab logic is inverse : h = 1 accepted, h = 0 failed
two sided univariate t-test
```

### 97.13 ttest_man

```
two-sample t-test
unequal sample size
equal variance
```

### 97.14 ttest_paired

```
paired t-test
unequal sample size
equal variance
more powerfull than unpaired test, as long as correlation between
    x1 and x2 > 0
```

### 97.15 wgeomean

```
weighted geometric mean
function mu = wgeomean(w,x)
```

### 97.16 wgeovar

```
variance of the weighted geometric mean
```

### 97.17 wharmean

```
weighted harmonic mean
```

### 97.18 wharstd

### 97.19 wharvar

### 97.20 wnormpdf

```
wrapped normal distribution to the unit circle
c.f. stephens
```

## 98 stochastic

### 98.1 brownian_noise_1d_acf

### 98.2 brownian_noise_1d_fft

### 98.3 brownian_noise_2d_fft

## 99 mathematics

`mathematical functions of various kind`

### 99.1 ternary_diagram

## 100 test/master

### 100.1 dat_test_lanczos_3d_k_20_n_40

### 100.2 poisson2d_blk

### 100.3 qr_implicit_givens_2

### 100.4 spectral_derivative_2d

### 100.5 test_2d_eigensolver_hydrogen

### 100.6 test_2d_refine

### 100.7 test_3d_eigensolver_hydrogen

### 100.8 test_FEM

### 100.9 test_Mesh_3d

### 100.10 test_arnoldi

### 100.11 test_arpackc

### 100.12 test_assemble

### 100.13 test_assembly_performance

### 100.14 test_bc_one_sided

### 100.15 test_compare_solvers

### 100.16 test_complete

## 100.17 test_convergence

## 100.18 test_convergence_b

## 100.19 test_df_2d

## 100.20 test_eig_algs

## 100.21 test_eig_inverse

## 100.22 test_eigs_lanczos

## 100.23 test_eigs_lanczos_1

## 100.24 test_eigs_lanczos_2

## 100.25 test_eigs_lanczos_performance

## 100.26 test_fdm

## 100.27 test_fdm_d_vargrid

## 100.28 test_fdm_spectral

## 100.29 test_fem

## 100.30 test_fem_1d

## 100.31 test_fem_1d_higher_order

## 100.32 test_fem_2d_adaptive

## 100.33 test_fem_2d_higher_order

## 100.34 test_fem_3d_higher_order

## 100.35 test_fem_3d_refine

## 100.36 test_fem_b

### 100.37 test_fem_derivative

### 100.38 test_fem_quadrature

### 100.39 test_final

### 100.40 test_fix_substitution

### 100.41 test_forward

### 100.42 test_get_sparse_arrays

### 100.43 test_harmonic_oscillator

### 100.44 test_high_order_fdm_periodic_bc

### 100.45 test_hydrogen_wf

### 100.46 test_ichol

## 100.47 test_interpolation

## 100.48 test_inverse_problem

## 100.49 test_it_vs_exact

## 100.50 test_jama

## 100.51 test_jd

## 100.52 test_jdqz

## 100.53 test_lanczos_2

## 100.54 test_lanczos_biorthogonal

## 100.55 test_laplacian

## 100.56 test_laplacian_non_uniform

## 100.57 test_laplacian_simple

## 100.58 test_mesh_2d_uniform

## 100.59 test_mesh_2d_uniform_2

## 100.60 test_mesh_circle

## 100.61 test_mesh_generation

## 100.62 test_mesh_interpolate

## 100.63 test_mg

## 100.64 test_minres_recycle

## 100.65 test_multigrid

## 100.66 test_nc

## 100.67 test_nonuniform_symmetric

## 100.68 test_pde

## 100.69 test_permutation

## 100.70 test_poison_fem

## 100.71 test_polar

## 100.72 test_potential

## 100.73 test_powers

## 100.74 test_precondition

## 100.75 test_project_rectangle

## 100.76 test_qr

### 100.77 test_quantum_well

### 100.78 test_radial_adaptive

### 100.79 test_radial_confinement

### 100.80 test_radial_fixes

### 100.81 test_refine_2d

### 100.82 test_refine_2d_b

### 100.83 test_refine_3d

### 100.84 test_refine_structural

### 100.85 test_regularisation

### 100.86 test_round_off

### 100.87 test_schrödinger_potentials

### 100.88 test_uniform_mesh

### 100.89 test_vargrid

## 101 test

### 101.1 test_bandpass2d

### 101.2 test_gaussfit3

### 101.3 test_geoserr

### 101.4 test_hexagonal_pattern

### 101.5 test_lognfit_quantile

### 101.6 test_lowpass1d_implicit

### 101.7 test_lowpass2d_anisotropic

### 101.8 test_lowpass2d_fft

### 101.9 test_max_normal

### 101.10 test_moments_fourier_power

### 101.11 test_mtimes3x3

## 102 wavelet

### 102.1 contiuous_wavelet_transform

```
continuous wavelet transform
follows "The Illustrated Wavelet Transform Handbook: Introductory
    Theory and ..."
```

### 102.2 cwt_man

```
continuous fourier transform
as of time of implmentation, the matlab interal cwt is affected by
serious round-off errors and has issues with the scaling,
which is not the case here
```

### 102.3 example_wavelets

### 102.4 phasewrap

```
wrap the phase to +/- pi
```

### 102.5 test_cwt_man

### 102.6 test_phasewrap

### 102.7 test_wavelet

### 102.8 test_wavelet2

### 102.9 test_wavelet_analysis

### 102.10 test_wavelet_reconstruct

### 102.11 test_wtc

### 102.12 wavelet

```
wavelet windows
```

## 102.13 wavelet_reconstruct

```
iverses wavelet transform for single frequency
(reconstruction of time series)
n : window lengths in multiples of filter period 1/f0
```

## 102.14 wavelet_transform

```
wavelet transform for single frequency
n : window lengths in multiples of filter period 1/f0
```