

# Manual for Package: mathematics

## Revision 23M

Karl Kästner

February 16, 2023

## Contents

<b>1</b>	<b>calendar</b>	<b>1</b>
1.1	days_per_month . . . . .	1
1.2	isnight . . . . .	1
<b>2</b>	<b>mathematics</b>	<b>1</b>
2.1	cast_byte_to_integer . . . . .	1
<b>3</b>	<b>complex-analysis</b>	<b>1</b>
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
<b>4</b>	<b>derivation</b>	<b>3</b>
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_brownian_phase_inv . . . . .	4
4.6	derive_fourier_power . . . . .	4
4.7	derive_fourier_power_exp . . . . .	4
4.8	derive_laplacian_curvilinear . . . . .	4
4.9	derive_laplacian_fourier_pieewise_linear . . . . .	4
4.10	derive_logtripdf . . . . .	4
4.11	derive_smooth1d_parametric . . . . .	4
4.12	derive_spectral_density_bandpass_initial_condition . . . . .	5

<b>5</b>	<b>derivation/master</b>	<b>5</b>
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 . . . . .	6
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 . . . . .	7
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
<b>6</b>	<b>derivation</b>	<b>7</b>
6.1	simplify_atan . . . . .	7
<b>7</b>	<b>mathematics</b>	<b>8</b>
7.1	entropy . . . . .	8
<b>8</b>	<b>finance</b>	<b>8</b>
8.1	derive_skewrnd_walsh_paramter . . . . .	8
8.2	gbm_bridge . . . . .	8
8.3	gbm_cdf . . . . .	8
8.4	gbm_fit . . . . .	8
8.5	gbm_fit_old . . . . .	8
8.6	gbm_inv . . . . .	8
8.7	gbm_mean . . . . .	8
8.8	gbm_median . . . . .	9
8.9	gbm_moment2par . . . . .	9

8.10	gbm_pdf . . . . .	9
8.11	gbm_simulate . . . . .	9
8.12	gbm_skewness . . . . .	9
8.13	gbm_std . . . . .	9
8.14	gbm_transform_time_step . . . . .	9
8.15	put_price_black_scholes . . . . .	9
8.16	skewgbm_simulate . . . . .	9
8.17	skewrnd_walsh . . . . .	9
<b>9</b>	<b>finance/test</b>	<b>10</b>
9.1	test_gbm . . . . .	10
9.2	test_gbm_pdf . . . . .	10
9.3	test_skewrnd_walsh . . . . .	10
<b>10</b>	<b>fourier/@STFT</b>	<b>10</b>
10.1	STFT . . . . .	10
10.2	itransform . . . . .	10
10.3	stft_ . . . . .	10
10.4	stftmat . . . . .	10
10.5	transform . . . . .	11
<b>11</b>	<b>fourier</b>	<b>11</b>
11.1	amplitude_from_peak . . . . .	11
11.2	caesaro_weight . . . . .	11
11.3	dftmtx_man . . . . .	11
11.4	example_fourier_window . . . . .	11
11.5	fft2_cartesian2radial . . . . .	12
11.6	fft_man . . . . .	12
11.7	fft_rotate . . . . .	12
11.8	fftsmooth . . . . .	12
11.9	fix_fourier . . . . .	12
11.10	fourier_2d_padd . . . . .	13
11.11	fourier_2d_quadrants . . . . .	13
11.12	fourier_axis . . . . .	13
11.13	fourier_axis_2d . . . . .	13
11.14	fourier_cesaro_correction . . . . .	13
11.15	fourier_coefficient_piecewise_linear . . . . .	14
11.16	fourier_coefficient_piecewise_linear_1 . . . . .	14
11.17	fourier_coefficient_ramp3 . . . . .	14
11.18	fourier_coefficient_ramp_pulse . . . . .	14
11.19	fourier_coefficient_ramp_step . . . . .	14
11.20	fourier_coefficient_square_pulse . . . . .	15
11.21	fourier_complete_negative_half_plane . . . . .	15
11.22	fourier_cubic_interaction_coefficients . . . . .	15

11.23	fourier_derivative . . . . .	15
11.24	fourier_derivative_matrix_1d . . . . .	15
11.25	fourier_derivative_matrix_2d . . . . .	15
11.26	fourier_expand . . . . .	16
11.27	fourier_fit . . . . .	16
11.28	fourier_freq2ind . . . . .	16
11.29	fourier_interpolate . . . . .	16
11.30	fourier_matrix . . . . .	16
11.31	fourier_matrix2 . . . . .	16
11.32	fourier_matrix3 . . . . .	16
11.33	fourier_matrix_exp . . . . .	17
11.34	fourier_multiplicative_interaction_coefficients . . . . .	17
11.35	fourier_power . . . . .	17
11.36	fourier_power_exp . . . . .	17
11.37	fourier_predict . . . . .	17
11.38	fourier_quadratic_interaction_coefficients . . . . .	18
11.39	fourier_random_phase_walk . . . . .	18
11.40	fourier_range . . . . .	18
11.41	fourier_regress . . . . .	18
11.42	fourier_resampled_fit . . . . .	18
11.43	fourier_resampled_predict . . . . .	18
11.44	fourier_signed_square . . . . .	18
11.45	fourier_transform . . . . .	19
11.46	fourier_transform_fractional . . . . .	19
11.47	fourier_truncate_negative_half_plane . . . . .	19
11.48	hyperbolic_fourier_box . . . . .	19
11.49	idftmtx_man . . . . .	19
11.50	laplace_2d_pwlinear . . . . .	19
11.51	mean_fourier_power . . . . .	20
11.52	moments_fourier_power . . . . .	20
11.53	nanfft . . . . .	20
11.54	peaks . . . . .	20
11.55	roots_fourier . . . . .	21
11.56	spectral_density . . . . .	21
11.57	std_fourier_power . . . . .	21
11.58	test_complex_exp_product . . . . .	21
11.59	test_fourier_filter . . . . .	21
11.60	test_idftmtx . . . . .	21
11.61	var_fourier_power . . . . .	21
<b>12</b>	<b>mathematics</b>	<b>22</b>
12.1	gaussfit_quantile . . . . .	22
<b>13</b>	<b>geometry/@Geometry</b>	<b>22</b>

13.1	Geometry . . . . .	22
13.2	arclength . . . . .	22
13.3	arclength_old . . . . .	22
13.4	arclength_old2 . . . . .	22
13.5	base_point . . . . .	22
13.6	base_point_limited . . . . .	23
13.7	centroid . . . . .	23
13.8	cosa_min_max . . . . .	23
13.9	cross2 . . . . .	23
13.10	curvature . . . . .	23
13.11	ddot . . . . .	23
13.12	distance . . . . .	23
13.13	distance2 . . . . .	23
13.14	dot . . . . .	24
13.15	edge_length . . . . .	24
13.16	enclosed_angle . . . . .	24
13.17	enclosing_triangle . . . . .	24
13.18	hexagon . . . . .	24
13.19	inPolygon . . . . .	24
13.20	inTetra . . . . .	24
13.21	inTetra2 . . . . .	24
13.22	inTriangle . . . . .	25
13.23	intersect . . . . .	25
13.24	lineintersect . . . . .	25
13.25	lineintersect1 . . . . .	25
13.26	minimum_distance_lines . . . . .	25
13.27	mittenpunkt . . . . .	25
13.28	nagelpoint . . . . .	25
13.29	onLine . . . . .	25
13.30	orthocentre . . . . .	26
13.31	plumb_line . . . . .	26
13.32	poly_area . . . . .	26
13.33	poly_edges . . . . .	26
13.34	poly_set . . . . .	26
13.35	poly_width . . . . .	26
13.36	polyxpoly . . . . .	26
13.37	project_to_curve . . . . .	27
13.38	quad_isconvex . . . . .	27
13.39	random_disk . . . . .	27
13.40	random_simplex . . . . .	27
13.41	sphere_volume . . . . .	27
13.42	tetra_volume . . . . .	27
13.43	tobarycentric . . . . .	27
13.44	tobarycentric1 . . . . .	27

13.45	tobarycentric2 . . . . .	28
13.46	tobarycentric3 . . . . .	28
13.47	tri_angle . . . . .	28
13.48	tri_area . . . . .	28
13.49	tri_centroid . . . . .	28
13.50	tri_distance_opposit_midpoint . . . . .	28
13.51	tri_edge_length . . . . .	28
13.52	tri_edge_midpoint . . . . .	28
13.53	tri_excircle . . . . .	29
13.54	tri_height . . . . .	29
13.55	tri_incircle . . . . .	29
13.56	tri_isacute . . . . .	29
13.57	tri_isobtuse . . . . .	29
13.58	tri_semiperimeter . . . . .	29
13.59	tri_side_length . . . . .	29
<b>14</b>	<b>geometry</b>	<b>30</b>
14.1	Polygon . . . . .	30
14.2	bounding_box . . . . .	30
14.3	curvature_1d . . . . .	30
14.4	cvt . . . . .	30
14.5	deg_to_frac . . . . .	30
14.6	ellipse . . . . .	31
14.7	ellipseX . . . . .	31
14.8	ellipseY . . . . .	31
14.9	first_intersect . . . . .	31
14.10	golden_ratio . . . . .	31
14.11	hypot3 . . . . .	31
14.12	meanangle . . . . .	31
14.13	meanangle2 . . . . .	31
14.14	meanangle3 . . . . .	32
14.15	meanangle4 . . . . .	32
14.16	medianangle . . . . .	32
14.17	medianangle2 . . . . .	32
14.18	pilim . . . . .	32
14.19	streamline_radius_of_curvature . . . . .	32
<b>15</b>	<b>histogram/@Histogram</b>	<b>33</b>
15.1	2x . . . . .	33
15.2	Histogram . . . . .	33
15.3	bimodes . . . . .	33
15.4	cdf . . . . .	33
15.5	cdfS . . . . .	33
15.6	chi2test . . . . .	33

15.7	cmoment	33
15.8	cmomentS	33
15.9	entropy	33
15.10	entropyS	34
15.11	export_csv	34
15.12	iquantile	34
15.13	kstest	34
15.14	kurtosis	34
15.15	kurtosisS	34
15.16	mean	34
15.17	meanS	34
15.18	median	34
15.19	medianS	34
15.20	mode	35
15.21	modeS	35
15.22	moment	35
15.23	momentS	35
15.24	pdf	35
15.25	quantile	35
15.26	quantileS	35
15.27	resample	35
15.28	setup	35
15.29	skewness	35
15.30	skewnessS	36
15.31	stairs	36
15.32	stairsS	36
15.33	std	36
15.34	stdS	36
15.35	var	36
15.36	varS	36
<b>16</b>	<b>histogram</b>	<b>36</b>
16.1	hist_man	36
16.2	histadapt	36
16.3	histconst	37
16.4	pdf_poly	37
16.5	plotcdf	37
16.6	test_histogram	37
<b>17</b>	<b>mathematics</b>	<b>37</b>
17.1	imrotmat	37
<b>18</b>	<b>linear-algebra</b>	<b>37</b>
18.1	averaging_matrix_2	37

18.2	colnorm . . . . .	37
18.3	condest_ . . . . .	37
18.4	connectivity_matrix . . . . .	38
<b>19</b>	<b>linear-algebra/coordinate-transformation</b>	<b>38</b>
19.1	barycentric2cartesian . . . . .	38
19.2	barycentric2cartesian3 . . . . .	38
19.3	cartesian2barycentric . . . . .	38
19.4	cartesian_to_unit_triangle_basis . . . . .	38
19.5	ellipsoid2geoid . . . . .	38
19.6	example_approximate_utm_conversion . . . . .	38
19.7	latlon2utm . . . . .	38
19.8	latlon2utm_simple . . . . .	39
19.9	lowrance_mercator_to_wgs84 . . . . .	39
19.10	nmea2utm . . . . .	39
19.11	sn2xy . . . . .	39
19.12	unit_triangle_to_cartesian . . . . .	39
19.13	utm2latlon . . . . .	39
19.14	xy2nt . . . . .	39
19.15	xy2sn . . . . .	40
19.16	xy2sn_java . . . . .	40
19.17	xy2sn_old . . . . .	40
<b>20</b>	<b>linear-algebra</b>	<b>40</b>
20.1	det2x2 . . . . .	40
20.2	det3x3 . . . . .	40
20.3	det4x4 . . . . .	40
20.4	diag2x2 . . . . .	40
20.5	down . . . . .	41
20.6	eig2x2 . . . . .	41
<b>21</b>	<b>linear-algebra/eigenvalue</b>	<b>41</b>
21.1	eig_bisection . . . . .	41
21.2	eig_inverse . . . . .	41
21.3	eig_inverse_iteration . . . . .	41
21.4	eig_power_iteration . . . . .	41
<b>22</b>	<b>linear-algebra/eigenvalue/jacobi-davidson</b>	<b>41</b>
22.1	afun_jdm . . . . .	41
22.2	davidson . . . . .	41
22.3	jacobi_davidson . . . . .	41
22.4	jacobi_davidson_qr . . . . .	42
22.5	jacobi_davidson_qz . . . . .	42
22.6	jacobi_davidson_simple . . . . .	42



22.7	jdqr . . . . .	42
22.8	jdqr_sleijpen . . . . .	45
22.9	jdqr_vorst . . . . .	49
22.10	jdqz . . . . .	52
22.11	mfunc_jdm . . . . .	56
22.12	mgs . . . . .	56
22.13	minres_ . . . . .	56
22.14	mv_jacobi_davidson . . . . .	56
<b>23</b>	<b>linear-algebra</b>	<b>57</b>
23.1	first . . . . .	57
23.2	gershgorin_circle . . . . .	57
23.3	haussdorff . . . . .	57
23.4	ieig2x2 . . . . .	57
23.5	inv2x2 . . . . .	57
23.6	inv3x3 . . . . .	57
23.7	inv4x4 . . . . .	57
23.8	kernel2matrix . . . . .	58
<b>24</b>	<b>linear-algebra/lanczos</b>	<b>58</b>
24.1	arnoldi . . . . .	58
24.2	arnoldi_new . . . . .	58
24.3	eigs_lanczos_man . . . . .	58
24.4	lanczos . . . . .	58
24.5	lanczos_ . . . . .	58
24.6	lanczos_biorthogonal . . . . .	58
24.7	lanczos_biorthogonal_improved . . . . .	58
24.8	lanczos_ghep . . . . .	58
24.9	mv_lanczos . . . . .	59
24.10	reorthogonalise . . . . .	59
24.11	test_lanczos . . . . .	59
<b>25</b>	<b>linear-algebra</b>	<b>59</b>
25.1	laplacian_eigenvalue . . . . .	59
25.2	laplacian_eigenvector . . . . .	59
25.3	laplacian_power . . . . .	59
25.4	least_squares_perpendicular_offset . . . . .	59
25.5	left . . . . .	59
<b>26</b>	<b>linear-algebra/linear-systems</b>	<b>59</b>
26.1	gmres_man . . . . .	59
26.2	minres_recycle . . . . .	60
<b>27</b>	<b>linear-algebra</b>	<b>60</b>

27.1	lpmean . . . . .	60
27.2	lpnorm . . . . .	60
27.3	matvec3 . . . . .	60
27.4	max2d . . . . .	60
27.5	mid . . . . .	60
27.6	mpoweri . . . . .	60
27.7	mtimes2x2 . . . . .	60
27.8	mtimes3x3 . . . . .	61
27.9	nannorm . . . . .	61
27.10	nanshift . . . . .	61
27.11	nl . . . . .	61
27.12	normalise . . . . .	61
27.13	normalize1 . . . . .	61
27.14	normrows . . . . .	61
27.15	orth2 . . . . .	62
27.16	orth_man . . . . .	62
27.17	orthogonalise . . . . .	62
27.18	padd2 . . . . .	62
27.19	paddext . . . . .	62
27.20	paddval1 . . . . .	62
27.21	paddval2 . . . . .	62
<b>28</b>	<b>linear-algebra/polynomial</b>	<b>63</b>
28.1	chebychev . . . . .	63
28.2	piecewise_polynomial . . . . .	63
28.3	roots1 . . . . .	63
28.4	roots2 . . . . .	63
28.5	roots2poly . . . . .	63
28.6	roots3 . . . . .	63
28.7	roots4 . . . . .	63
28.8	roots_piecewise_linear . . . . .	63
28.9	test_roots4 . . . . .	64
28.10	vanderi_1d . . . . .	64
<b>29</b>	<b>linear-algebra</b>	<b>64</b>
29.1	randrot . . . . .	64
29.2	right . . . . .	64
29.3	rot2 . . . . .	64
29.4	rot2dir . . . . .	64
29.5	rot3 . . . . .	64
29.6	rotR . . . . .	64
29.7	rownorm . . . . .	65
29.8	simmilarity_matrix . . . . .	65
29.9	spnorm . . . . .	65

29.10	spzeros . . . . .	65
29.11	test_roots3 . . . . .	65
29.12	transform_minmax . . . . .	65
29.13	transpose3 . . . . .	65
29.14	transposeall . . . . .	65
29.15	up . . . . .	65
29.16	vander_nd . . . . .	66
29.17	vanderd_2d . . . . .	66
<b>30</b>	<b>logic</b>	<b>66</b>
30.1	bitor_man . . . . .	66
<b>31</b>	<b>master/plot</b>	<b>66</b>
31.1	attach_boundary_value . . . . .	66
31.2	cartesian_polar . . . . .	66
31.3	img_vargrid . . . . .	66
31.4	plot_basis_functions . . . . .	66
31.5	plot_convergence . . . . .	66
31.6	plot_dof . . . . .	67
31.7	plot_eigenbar . . . . .	67
31.8	plot_error_estimation . . . . .	67
31.9	plot_error_estimation_2 . . . . .	67
31.10	plot_error_fem . . . . .	67
31.11	plot_fdm_kernel . . . . .	67
31.12	plot_fdm_vs_fem . . . . .	67
31.13	plot_fem_accuracy . . . . .	67
31.14	plot_function_and_grid . . . . .	67
31.15	plot_hat . . . . .	67
31.16	plot_hydrogen_wf . . . . .	68
31.17	plot_mesh . . . . .	68
31.18	plot_mesh_2 . . . . .	68
31.19	plot_refine . . . . .	68
31.20	plot_refine_3d . . . . .	68
31.21	plot_runtime . . . . .	68
31.22	plot_spectrum . . . . .	68
31.23	plot_wavefunction . . . . .	68
<b>32</b>	<b>master/ported</b>	<b>68</b>
32.1	assemble_2d_phi_phi . . . . .	68
32.2	assemble_3d_dphi_dphi . . . . .	69
32.3	assemble_3d_phi_phi . . . . .	69
32.4	dV_2d_ . . . . .	69
32.5	derivative_2d . . . . .	69
32.6	derivative_3d . . . . .	69

32.7	element_neighbour_2d . . . . .	69
32.8	prefetch_2d_ . . . . .	69
32.9	promote_2d_3_10 . . . . .	69
32.10	promote_2d_3_15 . . . . .	69
32.11	promote_2d_3_21 . . . . .	69
32.12	promote_2d_3_6 . . . . .	70
32.13	promote_3d_4_10 . . . . .	70
32.14	promote_3d_4_20 . . . . .	70
32.15	promote_3d_4_35 . . . . .	70
32.16	vander_2d . . . . .	70
32.17	vander_3d . . . . .	70
<b>33</b>	<b>mathematics</b>	<b>70</b>
33.1	monotoneous_indices . . . . .	70
33.2	nearest_fractional_timestep . . . . .	70
<b>34</b>	<b>number-theory</b>	<b>71</b>
34.1	ceiln . . . . .	71
34.2	digitsb . . . . .	71
34.3	floorn . . . . .	71
34.4	iseven . . . . .	71
34.5	multichoosek . . . . .	71
34.6	nchoosek_man . . . . .	71
34.7	pythagorean_triple . . . . .	72
34.8	roundn . . . . .	72
<b>35</b>	<b>numerical-methods</b>	<b>72</b>
35.1	advect_analytic . . . . .	72
<b>36</b>	<b>numerical-methods/differentiation</b>	<b>72</b>
36.1	derivative1 . . . . .	72
36.2	derivative2 . . . . .	72
<b>37</b>	<b>numerical-methods</b>	<b>72</b>
37.1	diffuse_analytic . . . . .	72
<b>38</b>	<b>numerical-methods/finite-difference</b>	<b>73</b>
38.1	cdiff . . . . .	73
38.2	cdiffb . . . . .	73
38.3	central_difference . . . . .	73
38.4	cmean . . . . .	73
38.5	cmean2 . . . . .	73
38.6	derivative_matrix_1_1d . . . . .	73
38.7	derivative_matrix_2_1d . . . . .	73
38.8	derivative_matrix_2d . . . . .	74

38.9	derivative_matrix_curvilinear . . . . .	74
38.10	derivative_matrix_curvilinear_2 . . . . .	74
38.11	difference_kernel . . . . .	74
38.12	diffusion_matrix_2d_anisotropic . . . . .	74
38.13	diffusion_matrix_2d_anisotropic2 . . . . .	74
38.14	directional_neighbour . . . . .	74
38.15	distmat . . . . .	74
38.16	downwind_difference . . . . .	75
38.17	gradpde2d . . . . .	75
38.18	laplacian . . . . .	75
38.19	laplacian_fdm . . . . .	75
38.20	lrmean . . . . .	75
<b>39</b>	<b>numerical-methods/finite-difference/master</b>	<b>75</b>
39.1	fdm_adaptive_grid . . . . .	75
39.2	fdm_adaptive_refinement_old . . . . .	75
39.3	fdm_assemble_d1_2d . . . . .	75
39.4	fdm_assemble_d2_2d . . . . .	76
39.5	fdm_confinement . . . . .	76
39.6	fdm_d_vargrid . . . . .	76
39.7	fdm_h_unstructured . . . . .	76
39.8	fdm_hydrogen_vargrid . . . . .	76
39.9	fdm_mark_unstructured_2d . . . . .	76
39.10	fdm_plot . . . . .	76
39.11	fdm_plot_series . . . . .	76
39.12	fdm_refine_2d . . . . .	76
39.13	fdm_refine_3d . . . . .	76
39.14	fdm_refine_unstructured_2d . . . . .	77
39.15	fdm_schroedinger_2d . . . . .	77
39.16	fdm_schroedinger_3d . . . . .	77
39.17	relocate . . . . .	77
<b>40</b>	<b>numerical-methods/finite-difference</b>	<b>77</b>
40.1	mid . . . . .	77
40.2	pwmid . . . . .	77
40.3	ratio . . . . .	77
40.4	steplength . . . . .	77
40.5	swapoddeven . . . . .	78
40.6	test_derivative_matrix_2d . . . . .	78
40.7	test_derivative_matrix_curvilinear . . . . .	78
40.8	test_difference_kernel . . . . .	78
40.9	upwind_difference . . . . .	78
<b>41</b>	<b>numerical-methods/finite-element</b>	<b>78</b>

41.1	Mesh_2d.java . . . . .	78
41.2	Tree_2d.java . . . . .	78
41.3	assemble_1d_dphi_dphi . . . . .	78
41.4	assemble_1d_phi_phi . . . . .	78
41.5	assemble_2d_dphi_dphi_java . . . . .	79
41.6	assemble_2d_phi_phi_java . . . . .	79
41.7	assemble_3d_dphi_dphi_java . . . . .	79
41.8	assemble_3d_phi_phi_java . . . . .	79
41.9	boundary_1d . . . . .	79
41.10	boundary_2d . . . . .	79
41.11	boundary_3d . . . . .	79
41.12	check_area_2d . . . . .	79
41.13	circmesh . . . . .	79
41.14	cropradius . . . . .	79
41.15	display_2d . . . . .	80
41.16	display_3d . . . . .	80
41.17	distort . . . . .	80
41.18	err_2d . . . . .	80
41.19	estimate_err_2d_3 . . . . .	80
41.20	example_1d . . . . .	80
41.21	example_2d . . . . .	80
41.22	explode . . . . .	80
41.23	fem_2d . . . . .	80
41.24	fem_2d_heuristic_mesh . . . . .	80
41.25	fem_get_2d_radial . . . . .	81
41.26	fem_interpolation . . . . .	81
41.27	fem_plot_1d . . . . .	81
41.28	fem_plot_1d_series . . . . .	81
41.29	fem_plot_2d . . . . .	81
41.30	fem_plot_2d_series . . . . .	81
41.31	fem_plot_3d . . . . .	81
41.32	fem_plot_3d_series . . . . .	81
41.33	fem_plot_confine_series . . . . .	81
41.34	fem_radial . . . . .	82
41.35	flip_2d . . . . .	82
41.36	get_mesh_arrays . . . . .	82
41.37	hashkey . . . . .	82
<b>42 numerical-methods/finite-element/int</b>		<b>82</b>
42.1	int_1d_gauss . . . . .	82
42.2	int_1d_gauss_1 . . . . .	82
42.3	int_1d_gauss_2 . . . . .	82
42.4	int_1d_gauss_3 . . . . .	82
42.5	int_1d_gauss_4 . . . . .	82

42.6	int_1d_gauss_5 . . . . .	83
42.7	int_1d_gauss_6 . . . . .	83
42.8	int_1d_gauss_lobatto . . . . .	83
42.9	int_1d_gauss_n . . . . .	83
42.10	int_1d_nc_2 . . . . .	83
42.11	int_1d_nc_3 . . . . .	83
42.12	int_1d_nc_4 . . . . .	83
42.13	int_1d_nc_5 . . . . .	83
42.14	int_1d_nc_6 . . . . .	83
42.15	int_1d_nc_7 . . . . .	83
42.16	int_1d_nc_7_hardy . . . . .	84
42.17	int_2d_gauss_1 . . . . .	84
42.18	int_2d_gauss_12 . . . . .	84
42.19	int_2d_gauss_13 . . . . .	84
42.20	int_2d_gauss_16 . . . . .	84
42.21	int_2d_gauss_19 . . . . .	84
42.22	int_2d_gauss_25 . . . . .	84
42.23	int_2d_gauss_3 . . . . .	84
42.24	int_2d_gauss_33 . . . . .	84
42.25	int_2d_gauss_4 . . . . .	84
42.26	int_2d_gauss_6 . . . . .	85
42.27	int_2d_gauss_7 . . . . .	85
42.28	int_2d_gauss_9 . . . . .	85
42.29	int_2d_nc_10 . . . . .	85
42.30	int_2d_nc_15 . . . . .	85
42.31	int_2d_nc_21 . . . . .	85
42.32	int_2d_nc_3 . . . . .	85
42.33	int_2d_nc_6 . . . . .	85
42.34	int_3d_gauss_1 . . . . .	85
42.35	int_3d_gauss_11 . . . . .	85
42.36	int_3d_gauss_14 . . . . .	86
42.37	int_3d_gauss_15 . . . . .	86
42.38	int_3d_gauss_24 . . . . .	86
42.39	int_3d_gauss_4 . . . . .	86
42.40	int_3d_gauss_45 . . . . .	86
42.41	int_3d_gauss_5 . . . . .	86
42.42	int_3d_nc_11 . . . . .	86
42.43	int_3d_nc_4 . . . . .	86
42.44	int_3d_nc_6 . . . . .	86
42.45	int_3d_nc_8 . . . . .	86
<b>43</b>	<b>numerical-methods/finite-element</b>	<b>87</b>
43.1	interpolation_matrix . . . . .	87
43.2	mark . . . . .	87

43.3	mark_1d . . . . .	87
43.4	mesh_1d_uniform . . . . .	87
43.5	mesh_3d_uniform . . . . .	87
43.6	mesh_interpolate . . . . .	87
43.7	neighbour_1d . . . . .	87
43.8	old . . . . .	87
43.9	pdeeig_1d . . . . .	87
43.10	pdeeig_2d . . . . .	88
43.11	pdeeig_3d . . . . .	88
43.12	polynomial_derivative_1d . . . . .	88
43.13	potential_const . . . . .	88
43.14	potential_coulomb . . . . .	88
43.15	potential_harmonic_oscillator . . . . .	88
43.16	project_circle . . . . .	88
43.17	project_rectangle . . . . .	88
43.18	promote_1d_2_3 . . . . .	88
43.19	promote_1d_2_4 . . . . .	88
43.20	promote_1d_2_5 . . . . .	89
43.21	promote_1d_2_6 . . . . .	89
43.22	quadrilaterate . . . . .	89
43.23	recalculate_regularity_2d . . . . .	89
43.24	refine_1d . . . . .	89
43.25	refine_2d_21 . . . . .	89
43.26	refine_2d_structural . . . . .	89
43.27	regularity_1d . . . . .	89
43.28	regularity_2d . . . . .	89
43.29	regularity_3d . . . . .	90
43.30	relocate_2d . . . . .	90
43.31	test_circmesh . . . . .	90
43.32	test_hermite . . . . .	90
43.33	tri_assign_points . . . . .	90
43.34	triangulation_uniform . . . . .	90
43.35	vander_1d . . . . .	90
43.36	vanderd_1d . . . . .	90
43.37	vanderi_1d . . . . .	90
<b>44</b>	<b>numerical-methods/finite-volume/@Advection</b>	<b>91</b>
44.1	Advection . . . . .	91
44.2	dot_advection . . . . .	91
<b>45</b>	<b>numerical-methods/finite-volume/@Burgers</b>	<b>91</b>
45.1	burgers_split . . . . .	91
45.2	dot_burgers_fdm . . . . .	91
45.3	dot_burgers_fft . . . . .	91



<b>46</b>	<b>numerical-methods/finite-volume/@Finite_Volume</b>	<b>91</b>
46.1	Finite_Volume . . . . .	91
46.2	apply_bc . . . . .	92
46.3	solve . . . . .	92
46.4	step_split_strang . . . . .	92
46.5	step_unsplit . . . . .	92
<b>47</b>	<b>numerical-methods/finite-volume/@Flux_Limiter</b>	<b>92</b>
47.1	Flux_Limiter . . . . .	92
47.2	beam_warming . . . . .	92
47.3	fromm . . . . .	93
47.4	lax_wendroff . . . . .	93
47.5	minmod . . . . .	93
47.6	monotized_central . . . . .	93
47.7	muscl . . . . .	93
47.8	superbee . . . . .	93
47.9	upwind . . . . .	93
47.10	vanLeer . . . . .	94
<b>48</b>	<b>numerical-methods/finite-volume/@KDV</b>	<b>94</b>
48.1	dot_kdv_fdm . . . . .	94
48.2	dot_kdv_fft . . . . .	94
48.3	kdv_split . . . . .	94
<b>49</b>	<b>numerical-methods/finite-volume/@Reconstruct_Average_Evolve</b>	<b>94</b>
49.1	Reconstruct_Average_Evolve . . . . .	94
49.2	advect_highres . . . . .	95
49.3	advect_lowress . . . . .	95
<b>50</b>	<b>numerical-methods/finite-volume</b>	<b>95</b>
50.1	Godunov . . . . .	95
50.2	Lax_Friedrich . . . . .	95
50.3	Measure . . . . .	95
50.4	Roe . . . . .	95
50.5	fv_swe . . . . .	96
50.6	staggered_euler . . . . .	96
50.7	staggered_grid . . . . .	96
<b>51</b>	<b>numerical-methods</b>	<b>96</b>
51.1	grid2quad . . . . .	96
<b>52</b>	<b>numerical-methods/integration</b>	<b>96</b>
52.1	cumintL . . . . .	96
52.2	cumintR . . . . .	96
52.3	cumint_trapezoidal . . . . .	96

52.4	int_trapezoidal . . . . .	97
<b>53</b>	<b>numerical-methods/interpolation/@Kriging</b>	<b>97</b>
53.1	Kriging . . . . .	97
53.2	estimate_semivariance . . . . .	97
53.3	interpolate_ . . . . .	97
<b>54</b>	<b>numerical-methods/interpolation/@RegularizedInterpolator1</b>	<b>97</b>
54.1	RegularizedInterpolator1 . . . . .	97
54.2	init . . . . .	98
<b>55</b>	<b>numerical-methods/interpolation/@RegularizedInterpolator2</b>	<b>98</b>
55.1	RegularizedInterpolator2 . . . . .	98
55.2	init . . . . .	98
<b>56</b>	<b>numerical-methods/interpolation/@RegularizedInterpolator3</b>	<b>98</b>
56.1	RegularizedInterpolator3 . . . . .	98
56.2	init . . . . .	98
<b>57</b>	<b>numerical-methods/interpolation</b>	<b>98</b>
57.1	IDW . . . . .	98
57.2	IPoly . . . . .	98
57.3	IRBM . . . . .	99
57.4	ISparse . . . . .	99
57.5	Inn . . . . .	99
57.6	Interpolator . . . . .	99
57.7	fixnan . . . . .	99
57.8	idw1 . . . . .	99
57.9	idw2 . . . . .	99
57.10	inner2outer . . . . .	100
57.11	inner2outer2 . . . . .	100
57.12	interp1_circular . . . . .	100
57.13	interp1_limited . . . . .	100
57.14	interp1_man . . . . .	100
57.15	interp1_piecewise_linear . . . . .	100
57.16	interp1_save . . . . .	100
57.17	interp1_slope . . . . .	101
57.18	interp1_smooth . . . . .	101
57.19	interp1_unique . . . . .	101
57.20	interp2_man . . . . .	101
57.21	interp_angle . . . . .	101
57.22	interp_fourier . . . . .	101
57.23	interp_fourier_batch . . . . .	101
57.24	interp_sn . . . . .	102

57.25	interp_sn2 . . . . .	102
57.26	interp_sn3 . . . . .	102
57.27	interp_sn_ . . . . .	102
57.28	limit_by_distance_1d . . . . .	102
57.29	resample1 . . . . .	102
57.30	resample_d_min . . . . .	102
57.31	resample_vector . . . . .	103
57.32	test_interp1_limited . . . . .	103
<b>58</b>	<b>numerical-methods</b>	<b>103</b>
58.1	inverse_complex . . . . .	103
58.2	maccormack_step . . . . .	103
58.3	minmod . . . . .	103
<b>59</b>	<b>numerical-methods/multigrid</b>	<b>103</b>
59.1	mg_interpolate . . . . .	103
59.2	mg_restrict . . . . .	103
<b>60</b>	<b>numerical-methods/ode/@BVPS_Characteristic</b>	<b>103</b>
60.1	BVPS_Characteristic . . . . .	103
60.2	assemble1_A . . . . .	104
60.3	assemble1_A_Q . . . . .	104
60.4	assemble2_A . . . . .	104
60.5	assemble_AA . . . . .	104
60.6	assemble_AAA . . . . .	104
60.7	assemble_Ic . . . . .	104
60.8	bvp1c . . . . .	104
60.9	check_arguments . . . . .	104
60.10	couple_junctions . . . . .	105
60.11	derivative . . . . .	105
60.12	init . . . . .	105
60.13	inner2outer_bvp2c . . . . .	105
60.14	reconstruct . . . . .	105
60.15	resample . . . . .	105
60.16	solve . . . . .	105
60.17	test_assemble1_A . . . . .	106
60.18	test_assemble2_A . . . . .	106
<b>61</b>	<b>numerical-methods/ode/@Time_Stepper</b>	<b>106</b>
61.1	Time_Stepper . . . . .	106
61.2	solve . . . . .	106
<b>62</b>	<b>numerical-methods/ode</b>	<b>106</b>
62.1	bvp2fdm . . . . .	106

62.2	bvp2wavetrain . . . . .	107
62.3	bvp2wavetwopass . . . . .	107
62.4	ivp_euler_forward . . . . .	107
62.5	ivp_euler_forward2 . . . . .	107
62.6	ivprk2 . . . . .	107
62.7	ode2_matrix . . . . .	107
62.8	ode2characteristic . . . . .	107
62.9	step_trapezoidal . . . . .	108
62.10	test_bvp2 . . . . .	108
<b>63</b>	<b>numerical-methods/optimisation</b>	<b>108</b>
63.1	aitken_iteration . . . . .	108
63.2	anderson_iteration . . . . .	108
63.3	armijo_stopping_criterion . . . . .	108
63.4	astar . . . . .	108
63.5	binsearch . . . . .	108
63.6	bisection . . . . .	108
63.7	box1 . . . . .	109
63.8	box2 . . . . .	109
63.9	cauchy . . . . .	109
63.10	cauchy2 . . . . .	109
63.11	directional_derivative . . . . .	109
63.12	dud . . . . .	109
63.13	extreme3 . . . . .	109
63.14	extreme_quadratic . . . . .	110
63.15	ftest . . . . .	110
63.16	fzero_bisect . . . . .	110
63.17	fzero_newton . . . . .	110
63.18	grad . . . . .	110
63.19	hessian . . . . .	110
63.20	hessian_from_gradient . . . . .	111
63.21	hessian_projected . . . . .	111
63.22	line_search . . . . .	111
63.23	line_search2 . . . . .	111
63.24	line_search_polynomial . . . . .	111
63.25	line_search_polynomial2 . . . . .	112
63.26	line_search_quadratic . . . . .	112
63.27	line_search_quadratic2 . . . . .	112
63.28	line_search_wolfe . . . . .	112
63.29	ls_bgfs . . . . .	112
63.30	ls_broyden . . . . .	113
63.31	ls_generalized_secant . . . . .	113
63.32	nleg . . . . .	113
63.33	nlls . . . . .	113

63.34	picard . . . . .	113
63.35	poly_extrema . . . . .	114
63.36	quadratic_function . . . . .	114
63.37	quadratic_programming . . . . .	114
63.38	quadratic_step . . . . .	114
63.39	rosenbrock . . . . .	114
63.40	sqrt_heron . . . . .	114
63.41	test_directional_derivative . . . . .	114
63.42	test_dud . . . . .	114
63.43	test_fzero_newton . . . . .	114
63.44	test_line_search_quadratic2 . . . . .	115
63.45	test_ls_generalized_secant . . . . .	115
63.46	test_nlcg_6_order . . . . .	115
63.47	test_nlls . . . . .	115
<b>64</b>	<b>numerical-methods/pde</b>	<b>115</b>
64.1	laplacian2d_fundamental_solution . . . . .	115
<b>65</b>	<b>numerical-methods/piecewise-polynomials</b>	<b>115</b>
65.1	Hermite1 . . . . .	115
65.2	hp2_fit . . . . .	115
65.3	hp2_predict . . . . .	116
65.4	hp_predict . . . . .	116
65.5	hp_regress . . . . .	116
65.6	lp_count . . . . .	116
65.7	lp_predict . . . . .	116
65.8	lp_regress . . . . .	116
65.9	lp_regress_ . . . . .	116
<b>66</b>	<b>numerical-methods</b>	<b>116</b>
66.1	test_adams_bashforth . . . . .	116
<b>67</b>	<b>mathematics</b>	<b>117</b>
67.1	oversampleNZ . . . . .	117
<b>68</b>	<b>regression/@PolyOLS</b>	<b>117</b>
68.1	PolyOLS . . . . .	117
68.2	coefftest . . . . .	117
68.3	detrend . . . . .	117
68.4	fit . . . . .	117
68.5	fit_ . . . . .	117
68.6	predict . . . . .	117
68.7	predict_ . . . . .	118
68.8	slope . . . . .	118

<b>69 regression/@PowerLS</b>	<b>118</b>
69.1 PowerLS . . . . .	118
69.2 fit . . . . .	118
69.3 predict . . . . .	118
69.4 predict_ . . . . .	118
<b>70 regression/@Theil</b>	<b>118</b>
70.1 Theil . . . . .	118
70.2 detrend . . . . .	119
70.3 fit . . . . .	119
70.4 predict . . . . .	119
70.5 slope . . . . .	119
<b>71 regression</b>	<b>119</b>
71.1 Theil_Multivariate . . . . .	119
71.2 areg . . . . .	119
71.3 ginireg . . . . .	120
71.4 hesssimplereg . . . . .	120
71.5 llin . . . . .	120
71.6 lsq_sparam . . . . .	120
71.7 polyfitd . . . . .	120
71.8 regression_method_of_moments . . . . .	120
71.9 robustlinreg . . . . .	121
71.10 theil2 . . . . .	121
71.11 theil_generalised . . . . .	121
71.12 total_least_squares . . . . .	121
71.13 weighted_median_regression . . . . .	121
<b>72 set-theory</b>	<b>121</b>
72.1 issubset . . . . .	121
<b>73 mathematics</b>	<b>122</b>
73.1 shuffle_index . . . . .	122
<b>74 signal-processing</b>	<b>122</b>
74.1 asymwin . . . . .	122
<b>75 signal-processing/autocorrelation</b>	<b>122</b>
75.1 acf_radial . . . . .	122
75.2 acfar1 . . . . .	122
75.3 acfar1_2 . . . . .	122
75.4 acfar2 . . . . .	122
75.5 acfar2_2 . . . . .	123
75.6 ar1_cutoff_frequency . . . . .	123
75.7 ar1_effective_sample_size . . . . .	123

75.8	ar1_mse_mu_single_sample . . . . .	123
75.9	ar1_mse_pop . . . . .	123
75.10	ar1_mse_range . . . . .	123
75.11	ar1_spectrum . . . . .	123
75.12	ar1_to_tikhonov . . . . .	123
75.13	ar1_var_factor . . . . .	124
75.14	ar1_var_factor_ . . . . .	124
75.15	ar1_var_range2 . . . . .	124
75.16	ar1delay . . . . .	124
75.17	ar1delay_old . . . . .	124
75.18	ar2_acf2c . . . . .	124
75.19	ar2conv . . . . .	125
75.20	ar2dof . . . . .	125
75.21	ar2param . . . . .	125
75.22	autocorr2 . . . . .	125
75.23	autocorr_angular . . . . .	125
75.24	autocorr_bandpass . . . . .	125
75.25	autocorr_brownian_phase . . . . .	125
75.26	autocorr_brownian_phase_2d . . . . .	125
75.27	autocorr_brownian_phase_across . . . . .	126
75.28	autocorr_decay_rate . . . . .	126
75.29	autocorr_effective_sample_size . . . . .	126
75.30	autocorr_fft . . . . .	126
75.31	autocorr_forest . . . . .	126
75.32	autocorr_genton . . . . .	126
75.33	autocorr_highpass . . . . .	126
75.34	autocorr_lowpass . . . . .	126
75.35	autocorr_periodic_additive_noise . . . . .	126
75.36	autocorr_periodic_windowed . . . . .	127
75.37	autocorr_radial . . . . .	127
75.38	autocorr_radial_hexagonal_pattern . . . . .	127
75.39	autocorrelation_max . . . . .	127
<b>76 signal-processing</b>		<b>127</b>
76.1	average_wave_shape . . . . .	127
76.2	bandpass . . . . .	127
76.3	bandpass_continuous_cdf . . . . .	127
76.4	bartlett . . . . .	127
76.5	bin1d . . . . .	128
76.6	bin2d . . . . .	128
76.7	binormrnd . . . . .	128
76.8	brownian_phase_cdf . . . . .	128
76.9	brownian_phase_inv . . . . .	128
76.10	coherence . . . . .	128

76.11	conv1_man . . . . .	128
76.12	conv2_man . . . . .	128
76.13	conv2z . . . . .	129
76.14	conv30 . . . . .	129
76.15	conv_ . . . . .	129
76.16	conv_centered . . . . .	129
76.17	convz . . . . .	129
76.18	cosexpdelay . . . . .	129
76.19	csmooth . . . . .	129
76.20	daniell_window . . . . .	130
76.21	db2neper . . . . .	130
76.22	db2power . . . . .	130
76.23	derive_bandpass_continuous_scale . . . . .	130
76.24	derive_danielle_weight . . . . .	130
76.25	derive_limit_0_acfar . . . . .	130
76.26	detect_peak . . . . .	130
76.27	determine_phase_shift . . . . .	130
76.28	determine_phase_shift1 . . . . .	131
76.29	doublesum_ij . . . . .	131
76.30	effective_mask_size . . . . .	131
76.31	effective_sample_size_to_ar1 . . . . .	131
76.32	fcut2Lw_gausswin . . . . .	131
76.33	fcut_gausswin . . . . .	131
76.34	filt_hodges_lehman . . . . .	131
<b>77 signal-processing/filters</b>		<b>131</b>
77.1	circfilt2 . . . . .	131
77.2	filter1 . . . . .	132
77.3	filter2 . . . . .	132
77.4	filter_ . . . . .	132
77.5	filter_r_to_f0 . . . . .	132
77.6	filter_rho_to_f0 . . . . .	132
77.7	filter_twosided . . . . .	132
77.8	filteriir . . . . .	132
77.9	filterp . . . . .	133
77.10	filterp1 . . . . .	133
77.11	filterstd . . . . .	133
77.12	gaussfilt2 . . . . .	133
77.13	lowpass_discrete . . . . .	134
77.14	meanfilt2 . . . . .	134
77.15	medfilt1_man . . . . .	134
77.16	medfilt1_man2 . . . . .	134
77.17	medfilt1_padded . . . . .	134
77.18	medfilt1_reduced . . . . .	134



77.19	trifilt1 . . . . .	134
77.20	trifilt2 . . . . .	135
<b>78</b>	<b>signal-processing</b>	<b>135</b>
78.1	firls_man . . . . .	135
78.2	fit_spectral_density . . . . .	135
78.3	fit_spectral_density_2d . . . . .	135
78.4	fit_spectral_density_radial . . . . .	135
78.5	flattopwin . . . . .	135
78.6	frequency_response_boxcar . . . . .	135
78.7	freqz_boxcar . . . . .	135
78.8	gaussfilt1 . . . . .	136
78.9	hanchangewin . . . . .	136
78.10	hanchangewin2 . . . . .	136
78.11	hanwin . . . . .	136
78.12	hanwin_ . . . . .	136
78.13	high_pass_1d_simple . . . . .	136
78.14	kaiserwin . . . . .	136
78.15	kalman . . . . .	136
78.16	lanczoswin . . . . .	137
78.17	last . . . . .	137
78.18	maxfilt1 . . . . .	137
78.19	meanfilt1 . . . . .	137
78.20	mid_term_single_sample . . . . .	137
78.21	minfilt1 . . . . .	137
78.22	minmax . . . . .	137
78.23	mu2ar1 . . . . .	137
78.24	mysmooth . . . . .	138
78.25	nanautocorr . . . . .	138
78.26	nanmedfilt1 . . . . .	138
78.27	neper2db . . . . .	138
78.28	oscillator_noisy . . . . .	138
<b>79</b>	<b>signal-processing/passes</b>	<b>138</b>
79.1	bandpass1d . . . . .	138
79.2	bandpass1d_fft . . . . .	138
79.3	bandpass1d_implicit . . . . .	138
79.4	bandpass2 . . . . .	139
79.5	bandpass2d . . . . .	139
79.6	bandpass2d_2 . . . . .	139
79.7	bandpass2d_fft . . . . .	139
79.8	bandpass2d_ideal . . . . .	139
79.9	bandpass2d_implicit . . . . .	139
79.10	bandpass2d_iso . . . . .	139

79.11	bandpass_arg . . . . .	139
79.12	bandpass_f0_to_rho . . . . .	139
79.13	bandpass_max . . . . .	140
79.14	bandpass_max2 . . . . .	140
79.15	highpass . . . . .	140
79.16	highpass1d_fft_cos . . . . .	140
79.17	highpass1d_implicit . . . . .	140
79.18	highpass2d_fft . . . . .	140
79.19	highpass2d_ideal . . . . .	140
79.20	highpass2d_implicit . . . . .	140
79.21	highpass_arg . . . . .	140
79.22	highpass_fc_to_rho . . . . .	141
79.23	lowpass . . . . .	141
79.24	lowpass1d_fft . . . . .	141
79.25	lowpass1d_implicit . . . . .	141
79.26	lowpass2 . . . . .	141
79.27	lowpass2d_2 . . . . .	141
79.28	lowpass2d_anisotropic . . . . .	141
79.29	lowpass2d_fft . . . . .	141
79.30	lowpass2d_ideal . . . . .	141
79.31	lowpass2d_implicit . . . . .	142
79.32	lowpass_arg . . . . .	142
79.33	lowpass_fc_to_rho . . . . .	142
79.34	lowpass_iir . . . . .	142
79.35	lowpass_iir_symmetric . . . . .	142
79.36	lowpassfilter2 . . . . .	142
<b>80 signal-processing</b>		<b>142</b>
80.1	peaks_man . . . . .	142
<b>81 signal-processing/periodogram</b>		<b>142</b>
81.1	periodogram . . . . .	142
81.2	periodogram_2d . . . . .	143
81.3	periodogram_align . . . . .	143
81.4	periodogram_angular . . . . .	143
81.5	periodogram_bartlett . . . . .	143
81.6	periodogram_bootstrap . . . . .	143
81.7	periodogram_confidence_interval . . . . .	143
81.8	periodogram_filter . . . . .	143
81.9	periodogram_median . . . . .	143
81.10	periodogram_p_value . . . . .	143
81.11	periodogram_qq . . . . .	144
81.12	periodogram_quantiles . . . . .	144
81.13	periodogram_radial . . . . .	144

81.14	periodogram_std . . . . .	144
81.15	periodogram_test_periodicity . . . . .	144
81.16	periodogram_test_periodicity_2d . . . . .	145
81.17	periodogram_test_stationarity . . . . .	145
81.18	periodogram_welsh . . . . .	146
<b>82</b>	<b>signal-processing</b>	<b>146</b>
82.1	polyfilt1 . . . . .	146
82.2	qmedfilt1 . . . . .	146
82.3	quadratfilt1 . . . . .	146
82.4	quadratwin . . . . .	146
82.5	randar1 . . . . .	146
82.6	randar1_dual . . . . .	146
82.7	randar2 . . . . .	146
82.8	randarp . . . . .	147
82.9	rectwin . . . . .	147
82.10	recursive_sum . . . . .	147
82.11	select_range . . . . .	147
82.12	smooth1d_parametric . . . . .	147
82.13	smooth2 . . . . .	147
82.14	smooth_man . . . . .	147
82.15	smooth_parametric . . . . .	147
82.16	smooth_parametric2 . . . . .	148
82.17	smooth_with_splines . . . . .	148
82.18	smoothfft . . . . .	148
<b>83</b>	<b>signal-processing/spectral-density</b>	<b>148</b>
83.1	gampdf_man . . . . .	148
83.2	hex_angular_pdf . . . . .	148
83.3	hex_angular_pdf_max . . . . .	148
83.4	hex_angular_pdf_max2par . . . . .	148
83.5	lognpdf_entropy . . . . .	148
83.6	misespdf . . . . .	148
83.7	normpdf_entropy . . . . .	149
83.8	skewpdf_entropy . . . . .	149
83.9	spectral_density_ar2 . . . . .	149
83.10	spectral_density_area . . . . .	149
83.11	spectral_density_bandpass2d_ideal . . . . .	149
83.12	spectral_density_bandpass_2d . . . . .	149
83.13	spectral_density_bandpass_2d_max2par . . . . .	149
83.14	spectral_density_bandpass_2d_scale . . . . .	149
83.15	spectral_density_bandpass_2d_scale_old . . . . .	149
83.16	spectral_density_bandpass_continuous . . . . .	150
83.17	spectral_density_bandpass_continuous_max . . . . .	150

83.18	spectral_density_bandpass_continuous_max2par . . . . .	150
83.19	spectral_density_bandpass_continuous_scale . . . . .	150
83.20	spectral_density_bandpass_discrete . . . . .	150
83.21	spectral_density_brownian_phase . . . . .	151
83.22	spectral_density_brownian_phase_2d . . . . .	151
83.23	spectral_density_brownian_phase_across . . . . .	151
83.24	spectral_density_brownian_phase_across_max . . . . .	151
83.25	spectral_density_brownian_phase_across_max2par . . . . .	151
83.26	spectral_density_brownian_phase_across_mode2par . . . . .	151
83.27	spectral_density_brownian_phase_mode . . . . .	151
83.28	spectral_density_brownian_phase_mode2par . . . . .	151
83.29	spectral_density_brownian_phase_scale . . . . .	152
83.30	spectral_density_estimate_2d . . . . .	152
83.31	spectral_density_flat . . . . .	152
83.32	spectral_density_forest . . . . .	152
83.33	spectral_density_gausswin . . . . .	152
83.34	spectral_density_highpass . . . . .	152
83.35	spectral_density_highpass2d_ideal . . . . .	152
83.36	spectral_density_highpass_2d . . . . .	152
83.37	spectral_density_highpass_cos . . . . .	152
83.38	spectral_density_lorentzian . . . . .	153
83.39	spectral_density_lorentzian_max . . . . .	153
83.40	spectral_density_lorentzian_max2par . . . . .	153
83.41	spectral_density_lorentzian_scale . . . . .	153
83.42	spectral_density_lowpass2d_ideal . . . . .	153
83.43	spectral_density_lowpass_2d . . . . .	153
83.44	spectral_density_lowpass_continuous . . . . .	153
83.45	spectral_density_lowpass_continuous_scale . . . . .	153
83.46	spectral_density_lowpass_discrete . . . . .	153
83.47	spectral_density_lowpass_one_sided . . . . .	154
83.48	spectral_density_maximum_bias_corrected . . . . .	154
83.49	spectral_density_periodic_additive_noise . . . . .	154
83.50	spectral_density_rectwin . . . . .	154
83.51	spectral_density_wperiodic . . . . .	154

## 84 signal-processing 154

84.1	spectral_density_brownian_phase_reg2par . . . . .	154
84.2	spectrogram . . . . .	154
84.3	sum_i_lag . . . . .	154
84.4	sum_ii . . . . .	155
84.5	sum_ii_ . . . . .	155
84.6	sum_ij . . . . .	155
84.7	sum_ij_ . . . . .	155
84.8	sum_ij_partial_ . . . . .	155

84.9	sum_multivar . . . . .	155
84.10	test_acfar1 . . . . .	155
84.11	tikhonov_to_ar1 . . . . .	155
84.12	trapwin . . . . .	156
84.13	triwin . . . . .	156
84.14	triwin2 . . . . .	156
84.15	tukeywin_man . . . . .	156
84.16	varar1 . . . . .	156
84.17	welch_spectrogram . . . . .	156
84.18	wfilt . . . . .	156
84.19	winbandpass . . . . .	156
<b>85</b>	<b>signal-processing/windows</b>	<b>157</b>
85.1	circwin . . . . .	157
85.2	danielle_window . . . . .	157
85.3	gausswin . . . . .	157
85.4	gausswin1 . . . . .	157
85.5	gausswin2 . . . . .	157
85.6	radial_window . . . . .	157
85.7	range_window . . . . .	157
85.8	rectwin_cutoff_frequency . . . . .	157
85.9	std_window . . . . .	157
85.10	window2d . . . . .	158
85.11	window_make_odd . . . . .	158
<b>86</b>	<b>signal-processing</b>	<b>158</b>
86.1	winfilt0 . . . . .	158
86.2	winlength . . . . .	158
86.3	wmeanfilt . . . . .	158
86.4	wmedfilt . . . . .	158
86.5	wordfilt . . . . .	158
86.6	wordfilt_edgeworth . . . . .	159
86.7	wrapphase . . . . .	159
86.8	xar1 . . . . .	159
86.9	xcorr_man . . . . .	159
<b>87</b>	<b>sorting</b>	<b>159</b>
87.1	sort2 . . . . .	159
87.2	sort2d . . . . .	159
<b>88</b>	<b>spatial-pattern-analysis/@Spatial_Pattern</b>	<b>159</b>
88.1	Spatial_Pattern . . . . .	159
88.2	analyze_grid . . . . .	159
88.3	analyze_transect . . . . .	160

88.4	extract_improfile . . . . .	160
88.5	fit_parametric_densities . . . . .	160
88.6	imread . . . . .	160
88.7	plot . . . . .	160
88.8	plot_transect . . . . .	160
88.9	report . . . . .	160
88.10	resample_functions . . . . .	160
88.11	tabulate . . . . .	160
<b>89</b>	<b>spatial-pattern-analysis</b>	<b>161</b>
89.1	approximate_ratio_distribution . . . . .	161
89.2	banded_pattern . . . . .	161
89.3	brownian_phase_patch_size_distribution . . . . .	161
89.4	hexagonal_pattern . . . . .	161
89.5	patch_size_1d . . . . .	161
89.6	patch_size_2d . . . . .	161
89.7	reconstruct_isotropic_density . . . . .	161
89.8	separate_isotropic_from_anisotropic_density . . . . .	161
89.9	suppress_low_frequency_lobe . . . . .	161
<b>90</b>	<b>special-functions</b>	<b>162</b>
90.1	bessel_sphere . . . . .	162
90.2	besseliln_large_x . . . . .	162
90.3	beta_man . . . . .	162
90.4	betainc_man . . . . .	162
90.5	digamma_man . . . . .	162
90.6	exp10 . . . . .	162
90.7	hankel_sphere . . . . .	162
90.8	hermite . . . . .	162
90.9	legendre_man . . . . .	163
90.10	neumann_sphere . . . . .	163
<b>91</b>	<b>statistics</b>	<b>163</b>
91.1	atan_s2 . . . . .	163
91.2	beta_kurt . . . . .	163
91.3	beta_mean . . . . .	163
91.4	beta_moment2param . . . . .	163
91.5	beta_skew . . . . .	163
91.6	beta_std . . . . .	164
91.7	chi2_kurt . . . . .	164
91.8	chi2_mean . . . . .	164
91.9	chi2_skew . . . . .	164
91.10	chi2_std . . . . .	164
91.11	coefficient_of_determination . . . . .	164

91.12	conditional_expectation_normal . . . . .	164
91.13	correlation_confidence_pearson . . . . .	164
<b>92</b>	<b>statistics/distributions</b>	<b>164</b>
92.1	PDF . . . . .	164
92.2	binorm_separation_coefficient . . . . .	165
92.3	binormcdf . . . . .	165
92.4	binormfit . . . . .	165
92.5	binormpdf . . . . .	165
92.6	edgeworth_cdf . . . . .	165
92.7	edgeworth_pdf . . . . .	165
92.8	exppdf_max2par . . . . .	165
92.9	gam_moment2param . . . . .	166
92.10	logn_mean . . . . .	166
92.11	logn_mode . . . . .	166
92.12	logn_mode2param . . . . .	166
92.13	logn_moment2param . . . . .	166
92.14	logn_param2moment . . . . .	166
92.15	logn_std . . . . .	166
92.16	lognpdf_ . . . . .	166
92.17	normpdf_wrapped . . . . .	166
92.18	normpdf_wrapped_mode . . . . .	167
92.19	normpdf_wrapped_mode2par . . . . .	167
92.20	pdfsample . . . . .	167
92.21	t2cdf . . . . .	167
92.22	t2inv . . . . .	167
<b>93</b>	<b>statistics</b>	<b>167</b>
93.1	error_propagation_fraction . . . . .	167
93.2	error_propagation_product . . . . .	167
93.3	example_standard_error_of_sample_quantiles . . . . .	167
93.4	f_var_finite . . . . .	168
93.5	fisher_mean . . . . .	168
93.6	fisher_moment2param . . . . .	168
93.7	fisher_std . . . . .	168
93.8	gam_mean . . . . .	168
93.9	gam_std . . . . .	168
93.10	gamma_mode . . . . .	168
93.11	gamma_mode2par . . . . .	168
93.12	gamma_moments_to_parameter . . . . .	168
93.13	gamma_stirling . . . . .	169
93.14	gaussfit3 . . . . .	169
93.15	gaussfit_quantile . . . . .	169
93.16	geoserr . . . . .	169

93.17	geostd . . . . .	169
93.18	hodges_lehmann_correlation . . . . .	169
93.19	hodges_lehmann_dispersion . . . . .	169
<b>94</b>	<b>statistics/information-theory</b>	<b>170</b>
94.1	akaike_information_criterion . . . . .	170
94.2	bayesian_information_criterion . . . . .	170
<b>95</b>	<b>statistics</b>	<b>170</b>
95.1	jackknife_block . . . . .	170
95.2	kurtncdf . . . . .	170
95.3	kurtnpdf . . . . .	170
95.4	kurtosis_bias_corrected . . . . .	170
95.5	limit . . . . .	171
95.6	logfactorial . . . . .	171
95.7	loglogpdf . . . . .	171
95.8	lognfit_quantile . . . . .	171
95.9	logskewcdf . . . . .	171
95.10	logskewpdf . . . . .	171
<b>96</b>	<b>statistics/logu</b>	<b>171</b>
96.1	lambertw_numeric . . . . .	171
96.2	logtrialtcdf . . . . .	171
96.3	logtrialtinv . . . . .	172
96.4	logtrialtmean . . . . .	172
96.5	logtrialtpdf . . . . .	172
96.6	logtrialtrnd . . . . .	172
96.7	logtricdf . . . . .	172
96.8	logtriinv . . . . .	172
96.9	logtrimean . . . . .	173
96.10	logtripdf . . . . .	173
96.11	logtirnd . . . . .	173
96.12	logucdf . . . . .	173
96.13	logucm . . . . .	173
96.14	loguinv . . . . .	173
96.15	logumean . . . . .	173
96.16	logupdf . . . . .	173
96.17	logurnd . . . . .	174
96.18	loguvar . . . . .	174
96.19	medlogu . . . . .	174
96.20	test_logurnd . . . . .	174
96.21	tricdf . . . . .	174
96.22	triinv . . . . .	174
96.23	trimediam . . . . .	174



96.24	tripdf . . . . .	174
96.25	trirnd . . . . .	175
<b>97</b>	<b>statistics</b>	<b>175</b>
97.1	max_exprnd . . . . .	175
97.2	maxnormals . . . . .	175
97.3	mean_angle . . . . .	175
97.4	mean_generalized_gampdf . . . . .	175
97.5	midrange . . . . .	175
97.6	minavg . . . . .	175
97.7	mode_man . . . . .	176
<b>98</b>	<b>statistics/moment-statistics</b>	<b>176</b>
98.1	autocorr_man3 . . . . .	176
98.2	autocorr_man4 . . . . .	176
98.3	autocorr_man5 . . . . .	176
98.4	blockserr . . . . .	176
98.5	comoment . . . . .	176
98.6	corr_man . . . . .	177
98.7	cov_man . . . . .	177
98.8	dof . . . . .	177
98.9	edgeworth_quantile . . . . .	177
98.10	effective_sample_size . . . . .	177
98.11	f_correlation . . . . .	178
98.12	f_finite . . . . .	178
98.13	lmean . . . . .	178
98.14	lmoment . . . . .	178
98.15	maskmean . . . . .	178
98.16	masknanmean . . . . .	178
98.17	mean1 . . . . .	178
98.18	mean_man . . . . .	178
98.19	mse . . . . .	179
98.20	nanautocorr_man1 . . . . .	179
98.21	nanautocorr_man2 . . . . .	179
98.22	nanautocorr_man4 . . . . .	179
98.23	nancorr . . . . .	179
98.24	nancumsum . . . . .	179
98.25	nanlmean . . . . .	179
98.26	nanr2 . . . . .	180
98.27	nanrms . . . . .	180
98.28	nanrmse . . . . .	180
98.29	nanserr . . . . .	180
98.30	nanwmean . . . . .	180
98.31	nanwstd . . . . .	180

98.32	nanwvar . . . . .	180
98.33	nanxcorr . . . . .	181
98.34	pearson . . . . .	181
98.35	pearson_to_kendall . . . . .	181
98.36	pool_samples . . . . .	181
98.37	qmean . . . . .	181
98.38	range_mean . . . . .	181
98.39	rmse_ . . . . .	181
98.40	serr . . . . .	181
98.41	serr1 . . . . .	182
98.42	test_kskew . . . . .	182
98.43	test_qstd_kskew_optimal_p . . . . .	182
98.44	wautocorr . . . . .	182
98.45	wcorr . . . . .	182
98.46	wcov . . . . .	182
98.47	wdof . . . . .	182
98.48	wkurt . . . . .	183
98.49	wmean . . . . .	183
98.50	wrms . . . . .	183
98.51	wserr . . . . .	183
98.52	wskew . . . . .	183
98.53	wstd . . . . .	183
98.54	wvar . . . . .	183
<b>99 statistics</b>		<b>184</b>
99.1	nangeomean . . . . .	184
99.2	nangeostd . . . . .	184
<b>100statistics/nonparametric-statistics</b>		<b>184</b>
100.1	kernel1d . . . . .	184
100.2	kernel2d . . . . .	184
<b>101statistics</b>		<b>184</b>
101.1	normmoment . . . . .	184
101.2	normpdf2 . . . . .	185
<b>102statistics/order-statistics</b>		<b>185</b>
102.1	hodges_lehmann_location . . . . .	185
102.2	kendall . . . . .	185
102.3	kendall_to_pearson . . . . .	185
102.4	mad2sd . . . . .	185
102.5	madcorr . . . . .	185
102.6	median2_holder . . . . .	186
102.7	median_ci . . . . .	186

102.8	median_man . . . . .	186
102.9	mediani . . . . .	186
102.10	nanmadcorr . . . . .	186
102.11	nanwmedian . . . . .	186
102.12	nanwquantile . . . . .	186
102.13	oja_median . . . . .	187
102.14	qkurtosis . . . . .	187
102.15	qmoments . . . . .	187
102.16	qskew . . . . .	187
102.17	qskewq . . . . .	187
102.18	qstdq . . . . .	188
102.19	quantile1_optimisation . . . . .	188
102.20	quantile2_breckling . . . . .	188
102.21	quantile2_chaudhuri . . . . .	188
102.22	quantile2_projected . . . . .	188
102.23	quantile2_projected2 . . . . .	188
102.24	quantile_envelope . . . . .	188
102.25	quantile_regression_simple . . . . .	188
102.26	ranking . . . . .	189
102.27	spatial_median . . . . .	189
102.28	spatial_quantile . . . . .	189
102.29	spatial_quantile2 . . . . .	189
102.30	spatial_quantile3 . . . . .	189
102.31	spatial_rank . . . . .	189
102.32	spatial_sign . . . . .	189
102.33	spatial_signed_rank . . . . .	189
102.34	spearman . . . . .	190
102.35	spearman_rank . . . . .	190
102.36	spearman_to_pearson . . . . .	190
102.37	wmedian . . . . .	190
102.38	wquantile . . . . .	190
<b>103statistics</b>		<b>190</b>
103.1	qstd . . . . .	190
103.2	quantile_extrap . . . . .	190
103.3	quantile_sin . . . . .	190
<b>104statistics/random-number-generation</b>		<b>191</b>
104.1	laplacernd . . . . .	191
104.2	randc . . . . .	191
104.3	skewness2param . . . . .	191
104.4	skewpdf_central_moments . . . . .	191
104.5	skewrnd . . . . .	191

<b>105</b>	<b>statistics</b>	<b>191</b>
105.1	range . . . . .	191
105.2	resample_with_replacement . . . . .	191
<b>106</b>	<b>statistics/resampling-statistics/@Jackknife</b>	<b>192</b>
106.1	Jackknife . . . . .	192
106.2	estimated_STATIC . . . . .	192
106.3	matrix1_STATIC . . . . .	192
106.4	matrix2 . . . . .	192
<b>107</b>	<b>statistics/resampling-statistics</b>	<b>193</b>
107.1	block_jackknife . . . . .	193
107.2	jackknife_moments . . . . .	193
107.3	moving_block_jackknife . . . . .	193
107.4	randblockserr . . . . .	193
107.5	resample . . . . .	193
<b>108</b>	<b>statistics</b>	<b>194</b>
108.1	scale_quantile_sd . . . . .	194
108.2	sd_sample_quantiles . . . . .	194
108.3	skew_generalized_normal_fit . . . . .	194
108.4	skew_generalized_normpdf . . . . .	194
108.5	skewcdf . . . . .	194
108.6	skewparam_to_central_moments . . . . .	194
108.7	skewpdf . . . . .	194
108.8	spatialrnd . . . . .	195
108.9	test_mean_generalized_gampdf . . . . .	195
108.10	test_skew_generalized_normpdf . . . . .	195
108.11	trimmed_mean . . . . .	195
108.12	ttest2_man . . . . .	195
108.13	ttest_man . . . . .	195
108.14	ttest_paired . . . . .	195
108.15	uniformnpdf . . . . .	196
108.16	wgeomean . . . . .	196
108.17	wgeovar . . . . .	196
108.18	wharmean . . . . .	196
108.19	wharstd . . . . .	196
108.20	wharvar . . . . .	196
108.21	wnormpdf . . . . .	196
<b>109</b>	<b>stochastic</b>	<b>196</b>
109.1	brownian_drift_hitting_probability . . . . .	196
109.2	brownian_drift_hitting_probability2 . . . . .	197
109.3	brownian_field . . . . .	197

109.4	brownian_motion_1d_acf . . . . .	197
109.5	brownian_motion_1d_cov . . . . .	197
109.6	brownian_motion_1d_fft . . . . .	197
109.7	brownian_motion_1d_fourier . . . . .	197
109.8	brownian_motion_1d_interleave . . . . .	197
109.9	brownian_motion_1d_laplacian . . . . .	197
109.10	brownian_motion_2d_cov . . . . .	197
109.11	brownian_motion_2d_fft . . . . .	198
109.12	brownian_motion_2d_fft_old . . . . .	198
109.13	brownian_motion_2d_fourier . . . . .	198
109.14	brownian_motion_2d_interleave . . . . .	198
109.15	brownian_motion_2d_interleaving . . . . .	198
109.16	brownian_motion_2d_kahunen . . . . .	198
109.17	brownian_motion_2d_laplacian . . . . .	198
109.18	brownian_motion_with_drift_hitting_probability . . . . .	198
<b>110mathematics</b>		<b>198</b>
110.1	ternary_diagram . . . . .	198
<b>111test/fourier</b>		<b>199</b>
111.1	test_fourier_freq2ind . . . . .	199
<b>112test/master</b>		<b>199</b>
112.1	dat_test_lanczos_3d_k_20_n_40 . . . . .	199
112.2	poisson2d_blk . . . . .	199
112.3	qr_implicit_givens_2 . . . . .	199
112.4	spectral_derivative_2d . . . . .	199
112.5	test_2d_eigensolver_hydrogen . . . . .	199
112.6	test_2d_refine . . . . .	199
112.7	test_3d_eigensolver_hydrogen . . . . .	199
112.8	test_FEM . . . . .	199
112.9	test_Mesh_3d . . . . .	200
112.10	test_arnoldi . . . . .	200
112.11	test_arpackc . . . . .	200
112.12	test_assemble . . . . .	200
112.13	test_assembly_performance . . . . .	200
112.14	test_bc_one_sided . . . . .	200
112.15	test_compare_solvers . . . . .	200
112.16	test_complete . . . . .	200
112.17	test_convergence . . . . .	200
112.18	test_convergence_b . . . . .	200
112.19	test_df_2d . . . . .	201
112.20	test_eig_algs . . . . .	201
112.21	test_eig_inverse . . . . .	201

112.22	test_eigs_lanczos . . . . .	201
112.23	test_eigs_lanczos_1 . . . . .	201
112.24	test_eigs_lanczos_2 . . . . .	201
112.25	test_eigs_lanczos_performance . . . . .	201
112.26	test_fdm . . . . .	201
112.27	test_fdm_d_vargrid . . . . .	201
112.28	test_fdm_spectral . . . . .	201
112.29	test_fem . . . . .	202
112.30	test_fem_1d . . . . .	202
112.31	test_fem_1d_higher_order . . . . .	202
112.32	test_fem_2d_adaptive . . . . .	202
112.33	test_fem_2d_higher_order . . . . .	202
112.34	test_fem_3d_higher_order . . . . .	202
112.35	test_fem_3d_refine . . . . .	202
112.36	test_fem_b . . . . .	202
112.37	test_fem_derivative . . . . .	202
112.38	test_fem_quadrature . . . . .	202
112.39	test_final . . . . .	203
112.40	test_fix_substitution . . . . .	203
112.41	test_forward . . . . .	203
112.42	test_get_sparse_arrays . . . . .	203
112.43	test_harmonic_oscillator . . . . .	203
112.44	test_high_order_fdm_periodic_bc . . . . .	203
112.45	test_hydrogen_wf . . . . .	203
112.46	test_ichol . . . . .	203
112.47	test_interpolation . . . . .	203
112.48	test_inverse_problem . . . . .	203
112.49	test_it_vs_exact . . . . .	204
112.50	test_jama . . . . .	204
112.51	test_jd . . . . .	204
112.52	test_jdqz . . . . .	204
112.53	test_lanczos_2 . . . . .	204
112.54	test_lanczos_biorthogonal . . . . .	204
112.55	test_laplacian . . . . .	204
112.56	test_laplacian_non_uniform . . . . .	204
112.57	test_laplacian_simple . . . . .	204
112.58	test_mesh_2d_uniform . . . . .	204
112.59	test_mesh_2d_uniform_2 . . . . .	205
112.60	test_mesh_circle . . . . .	205
112.61	test_mesh_generation . . . . .	205
112.62	test_mesh_interpolate . . . . .	205
112.63	test_mg . . . . .	205
112.64	test_minres_recycle . . . . .	205
112.65	test_multigrid . . . . .	205

112.66	test_nc . . . . .	205
112.67	test_nonuniform_symmetric . . . . .	205
112.68	test_pde . . . . .	205
112.69	test_permutation . . . . .	206
112.70	test_poison_fem . . . . .	206
112.71	test_polar . . . . .	206
112.72	test_potential . . . . .	206
112.73	test_powers . . . . .	206
112.74	test_precondition . . . . .	206
112.75	test_project_rectangle . . . . .	206
112.76	test_qr . . . . .	206
112.77	test_quantum_well . . . . .	206
112.78	test_radial_adaptive . . . . .	206
112.79	test_radial_confinement . . . . .	207
112.80	test_radial_fixes . . . . .	207
112.81	test_refine_2d . . . . .	207
112.82	test_refine_2d_b . . . . .	207
112.83	test_refine_3d . . . . .	207
112.84	test_refine_structural . . . . .	207
112.85	test_regularisation . . . . .	207
112.86	test_round_off . . . . .	207
112.87	test_schrödinger_potentials . . . . .	207
112.88	test_uniform_mesh . . . . .	207
112.89	test_vargrid . . . . .	208
<b>113test/signal-processing/autocorrelation</b>		<b>208</b>
113.1	test_acf . . . . .	208
113.2	test_acf_bias . . . . .	208
113.3	test_acf_brownian_phase . . . . .	208
113.4	test_acfar1_2 . . . . .	208
113.5	test_acfar1_3 . . . . .	208
113.6	test_acfar1_4 . . . . .	208
113.7	test_acfar2 . . . . .	208
113.8	test_ar1_var_factor . . . . .	208
113.9	test_ar1_var_factor_2 . . . . .	209
113.10	test_ar1_var_mu_single_sample . . . . .	209
113.11	test_ar1_var_pop . . . . .	209
113.12	test_ar1_var_pop_1 . . . . .	209
113.13	test_ar1delay . . . . .	209
113.14	test_ar2 . . . . .	209
<b>114test/signal-processing/passes</b>		<b>209</b>
114.1	test_bandpass2d_ideal . . . . .	209
114.2	test_lowpass1d_fft . . . . .	209

114.3	test_lowpass1d_implicit . . . . .	209
114.4	test_lowpass2d_anisotropic . . . . .	210
114.5	test_lowpass2d_fft . . . . .	210
114.6	test_lowpass2d_rho . . . . .	210
<b>115</b>	<b>test/signal-processing/periodogram</b>	<b>210</b>
115.1	test_periodicity_test_2d . . . . .	210
115.2	test_periodogram_bartlett_se . . . . .	210
115.3	test_periodogram_gauss . . . . .	210
115.4	test_periodogram_radial . . . . .	210
115.5	test_periodogram_test . . . . .	210
115.6	test_periodogram_test_periodicity_2d . . . . .	210
115.7	test_periogogram_significance . . . . .	211
<b>116</b>	<b>test/signal-processing/spectral-density</b>	<b>211</b>
116.1	test_spectral_density_2 . . . . .	211
116.2	test_spectral_density_bandpass_2d . . . . .	211
116.3	test_spectral_density_bandpass_2d_max2par . . . . .	211
116.4	test_spectral_density_bandpass_continuous . . . . .	211
116.5	test_spectral_density_bandpass_continuous_1 . . . . .	211
116.6	test_spectral_density_bandpass_maximum . . . . .	211
116.7	test_spectral_density_bandpass_scale . . . . .	211
116.8	test_spectral_density_bp . . . . .	211
116.9	test_spectral_density_bp_2d . . . . .	212
116.10	test_spectral_density_bp_approx . . . . .	212
116.11	test_spectral_density_brownian_phase . . . . .	212
116.12	test_spectral_density_brownian_phase_2d . . . . .	212
116.13	test_spectral_density_brownian_phase_across . . . . .	212
116.14	test_spectral_density_brownian_phase_across_mode2par . . . . .	212
116.15	test_spectral_density_brownian_phase_mode . . . . .	212
116.16	test_spectral_density_brownian_phase_mode2par . . . . .	212
116.17	test_spectral_density_brownian_phase_scale . . . . .	212
116.18	test_spectral_density_flat . . . . .	212
116.19	test_spectral_density_hp_cos . . . . .	213
116.20	test_spectral_density_lorentzian_max . . . . .	213
116.21	test_spectral_density_lorentzian_scale . . . . .	213
116.22	test_spectral_density_lowpass . . . . .	213
116.23	test_spectral_density_lowpass_continuous . . . . .	213
116.24	test_spectral_density_lowpass_continuous_1 . . . . .	213
116.25	test_spectral_density_maxiumum_bias_corrected . . . . .	213
<b>117</b>	<b>test/signal-processing</b>	<b>213</b>
117.1	test_autocorrelation_max . . . . .	213
117.2	test_cdf_bandpass_continuous . . . . .	213



117.3	test_cdf_brownian_phase . . . . .	214
117.4	test_fit_spectral_density . . . . .	214
<b>118</b>	<b>test/spatial-pattern-analysis</b>	<b>214</b>
118.1	test_approximate_ratio_distribution . . . . .	214
118.2	test_approximate_ratio_quantile . . . . .	214
118.3	test_separate_isotropic_density . . . . .	214
<b>119</b>	<b>test/statistics/distributions</b>	<b>214</b>
119.1	test_normpdf_wrapped . . . . .	214
<b>120</b>	<b>test/statistics/moment-statistics</b>	<b>214</b>
120.1	test_wmean . . . . .	214
<b>121</b>	<b>test/statistics</b>	<b>214</b>
121.1	test_fisher_moment2param . . . . .	214
121.2	test_gamma_mode . . . . .	215
<b>122</b>	<b>test/stochastics</b>	<b>215</b>
122.1	test_brownian_surface . . . . .	215
<b>123</b>	<b>test</b>	<b>215</b>
123.1	test_S . . . . .	215
123.2	test_advect_analytic . . . . .	215
123.3	test_asymbp . . . . .	215
123.4	test_bandpass2d . . . . .	215
123.5	test_bandwidth . . . . .	215
123.6	test_bartlett_angle . . . . .	215
123.7	test_bartlett_distribution . . . . .	215
123.8	test_bartlett_expansion . . . . .	216
123.9	test_beta . . . . .	216
123.10	test_betainc . . . . .	216
123.11	test_bivariate_covariance_term . . . . .	216
123.12	test_brownian_drift_hitting_probability . . . . .	216
123.13	test_brownian_drift_hitting_probability2 . . . . .	216
123.14	test_brownian_motion_1d . . . . .	216
123.15	test_brownian_motion_2d_cov . . . . .	216
123.16	test_brownian_motion_2d_fft . . . . .	216
123.17	test_brownian_noise_1d . . . . .	216
123.18	test_brownian_noise_2d . . . . .	217
123.19	test_brownian_noise_interleave . . . . .	217
123.20	test_coherence . . . . .	217
123.21	test_combined_spectral_density . . . . .	217
123.22	test_continuous_fourier_transform . . . . .	217
123.23	test_convexity . . . . .	217

123.24	test_d2	217
123.25	test_determine_phase_shift	217
123.26	test_diffuse_analytic	217
123.27	test_diffusion_matrix	217
123.28	test_ellipse	218
123.29	test_error_propagation_fraction	218
123.30	test_f	218
123.31	test_f2	218
123.32	test_fit_2d_spectral_density	218
123.33	test_fourier	218
123.34	test_fourier_derivative	218
123.35	test_fourier_derivative_1	218
123.36	test_fourier_integral	218
123.37	test_fourier_mask_covariance_matrix	218
123.38	test_ft_bp	219
123.39	test_gam	219
123.40	test_gamma_distribution	219
123.41	test_gampdf_man	219
123.42	test_gaussfit3	219
123.43	test_gaussian_flat	219
123.44	test_geoserr	219
123.45	test_hexagonal_pattern	219
123.46	test_iafrate	219
123.47	test_implicit_ode	219
123.48	test_imrotmat	220
123.49	test_integration	220
123.50	test_ivp	220
123.51	test_jacobian	220
123.52	test_lanczoswin	220
123.53	test_laplacian_power	220
123.54	test_lognfit_quantile	220
123.55	test_ls_perpendicular_offset	220
123.56	test_madcorr	220
123.57	test_mask	220
123.58	test_max_normal	221
123.59	test_moments	221
123.60	test_moments_fourier_power	221
123.61	test_mtimes3x3	221
123.62	test_noisy_oscillator	221
123.63	test_nonperiodic_pattern	221
123.64	test_normalization	221
123.65	test_ols	221
123.66	test_parcorr	221
123.67	test_positivity_preserving	221

123.68	test_randar1 . . . . .	222
123.69	test_randar1_multivariate . . . . .	222
123.70	test_randar2 . . . . .	222
123.71	test_ratio_distributions . . . . .	222
123.72	test_sd_rectwin . . . . .	222
123.73	test_spatialrnd . . . . .	222
123.74	test_spectrum_additivity . . . . .	222
123.75	test_stationarity . . . . .	222
123.76	test_stationarity2 . . . . .	222
123.77	test_sum_ij . . . . .	222
123.78	test_sum_multivar . . . . .	223
123.79	test_trifilt1 . . . . .	223
123.80	test_wautocorr . . . . .	223
123.81	test_wavelet_transform . . . . .	223
123.82	test_whittle . . . . .	223
123.83	test_window . . . . .	223
123.84	test_wordfilt . . . . .	223
123.85	test_xar1_mid_term . . . . .	223
<b>124</b>	<b>mathematics</b>	<b>223</b>
124.1	trapezoidal_fixed . . . . .	223
<b>125</b>	<b>wavelet</b>	<b>224</b>
125.1	contiuous_wavelet_transform . . . . .	224
125.2	cwt_man . . . . .	224
125.3	cwt_man2 . . . . .	224
125.4	example_wavelets . . . . .	224
125.5	phasewrap . . . . .	224
125.6	test_cwt_man . . . . .	224
125.7	test_phasewrap . . . . .	224
125.8	test_wavelet . . . . .	224
125.9	test_wavelet2 . . . . .	225
125.10	test_wavelet_analysis . . . . .	225
125.11	test_wavelet_reconstruct . . . . .	225
125.12	test_wtc . . . . .	225
125.13	wavelet . . . . .	225
125.14	wavelet_reconstruct . . . . .	225
125.15	wavelet_transform . . . . .	225

## 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

$$\begin{aligned} \text{re}(c_1 \exp(i\omega_1 x)) * \text{re}(c_2 \exp(i\omega_2 x)) = \\ \frac{1}{2} * ( \text{real}(c_1 * c_2 * \exp(i * (n_1 + n_2) * o * x)) \dots \\ + \text{real}(\text{conj}(c_1) * c_2 * \exp(i * (n_2 - n_1) * o * x)) ) \end{aligned}$$

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\_brownian\_phase\_inv

### 4.6 derive\_fourier\_power

### 4.7 derive\_fourier\_power\_exp

4.8 `derive_laplacian_curvilinear`

4.9 `derive_laplacian_fourier_piecewise_linear`

4.10 `derive_logtripdf`

4.11 `derive_smooth1d_parametric`

4.12 `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\_bridge

8.3 gbm\_cdf

8.4 `gbm_fit`

8.5 `gbm_fit_old`

8.6 `gbm_inv`

8.7 `gbm_mean`

8.8 `gbm_median`

8.9 `gbm_moment2par`

8.10 `gbm_pdf`

8.11 `gbm_simulate`

8.12 `gbm_skewness`

8.13 `gbm_std`

8.14 `gbm_transform_time_step`

8.15 `put_price_black_scholes`

8.16 `skewgbm_simulate`

8.17 `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 least `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 caesaro\_weight

## 11.3 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

## 11.4 example\_fourier\_window

## 11.5 fft2\_cartesian2radial

## 11.6 fft\_man

fast fourier transform for complex input data

input:  
F : data in real space

output :  
F : fourier transformation of F

## 11.7 fft\_rotate

## 11.8 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 confidence intervals

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

## 11.9 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

## 11.10 fourier\_2d\_padd

## 11.11 fourier\_2d\_quadrants

## 11.12 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

```

### 11.13 `fourier_axis_2d`

### 11.14 `fourier_cesaro_correction`

### 11.15 `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  $2\pi$

```

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

```

### 11.16 `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  $2\pi$



input :  
X : end points of piecewise linear function  
Y : values at end points  
  
output :  
ab : coefficients for frequency components

### 11.17 `fourier_coefficient_ramp3`

fourier series coefficient of a ramp

### 11.18 `fourier_coefficient_ramp_pulse`

fourier series coefficient of a ramp pules

### 11.19 `fourier_coefficient_ramp_step`

fourier coefficient of a ramp-step

### 11.20 `fourier_coefficient_square_pulse`

fourier series coefficients of a square pulse

### 11.21 `fourier_complete_negative_half_plane`

### 11.22 `fourier_cubic_interaction_coefficients`

### 11.23 `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-continuous derivatives, including discontinuous  
derivatives  
over the boundary  
3) discontinuous derivatives result in gibbs phenomenon

### 11.24 `fourier_derivative_matrix_1d`

### 11.25 `fourier_derivative_matrix_2d`

### 11.26 `fourier_expand`

expand values of fourier series

### 11.27 `fourier_fit`

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

### 11.28 `fourier_freq2ind`

### 11.29 `fourier_interpolate`

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

### 11.30 `fourier_matrix`

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

### 11.31 `fourier_matrix2`

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

### 11.32 `fourier_matrix3`

transformation matrix for the continuous fourier transform  
this is a matrix with  $(2n+1)$  real columns

### 11.33 `fourier_matrix_exp`

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

### 11.34 `fourier_multiplicative_interaction_coefficients`

### 11.35 fourier\_power

powers of a continuous fourier series in sin/cos form

$a^p = (u_r + u_1 \sin(\omega t) + u_2 \sin(\omega t + \delta))$ <sup>p</sup>  
phase of first component assumed 0

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

### 11.36 fourier\_power\_exp

powers of the continuous fourier series

$a^p = (u_r + u_1 \sin(\omega t) + u_2 \sin(\omega t + \delta))$ <sup>p</sup>  
phase of first component assumed 0

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

$y = a_0 + \sum (a_j \sin(j\omega t) + b_j \cos(j\omega t))$   
 $= \text{Real}(\sum_{i=0}^{\infty} c_i \exp(i\omega t)), c_i = a_i + b_i$

NOT the alternative  $\sum_{i=-\infty}^{\infty} \tilde{c}_i$ , tile  $c_j = 1/2 a_j$   
 $+ 1/2i b_j$

### 11.37 fourier\_predict

expand a continuous fourier series at times t

### 11.38 fourier\_quadratic\_interaction\_coefficients

### 11.39 fourier\_random\_phase\_walk

evaluate fourier series where the phase undergoes a brownian motion

#### 11.40 `fourier_range`

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

#### 11.41 `fourier_regress`

fit a continuous Fourier series to a set of sample points not  
sampled  
at equal intervals

#### 11.42 `fourier_resampled_fit`

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

#### 11.43 `fourier_resampled_predict`

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

#### 11.44 `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

#### 11.45 `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

```

#### 11.46 fourier\_transform\_fractional

#### 11.47 fourier\_truncate\_negative\_half\_plane

#### 11.48 hyperbolic\_fourier\_box

#### 11.49 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)

#### 11.50 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 frequency component  
these are coefficients of  $s, c, sh, ch$

$$\begin{aligned}
 (pu*(s + c) + qu*(s' + c'))*(shu + chu) &= ru && \% \text{ upper bc} \\
 (pd*(s + c) + qd*(s' + c'))*(shd + chd) &= rd && \% \text{ lower bc} \\
 ((sl + cl)*(pl*(shl + chl) + ql*(shl' + chl'))) &= rl && \% \text{ left} \\
 bc \\
 ((sr + cr)*(pr*(shr + chr) + qr*(shr' + chr'))) &= rr && \% \text{ right} \\
 bc
 \end{aligned}$$

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

### 11.51 mean\_fourier\_power

### 11.52 moments\_fourier\_power

### 11.53 nanfft

discrete fourier transform of a data series with gaps

### 11.54 peaks

peaks of the power spectrum of a discrete fourier transform

rule for peaks: there is no higher value left or right of the "peak"

until the signal drops to  $p \cdot y_{\text{peak}}$ ,  $p = 0.5$

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)

### 11.55 roots\_fourier

zeros of continuous fourier series series

$$f = a_0 + \sum_{j=1}^n a_j \cos(j x) + b_j \sin(j x)$$

### 11.56 spectral\_density

spectral density

### 11.57 std\_fourier\_power

### 11.58 test\_complex\_exp\_product

### 11.59 test\_fourier\_filter

### 11.60 test\_idftmtx

### 11.61 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 of 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 side parallel 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 arbitrary 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 obtuse triangles

### 13.58 tri\_semiperimeter

semiperimeter of a triangle

### 13.59 tri\_side\_length

edge length 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 tessellation

## 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

hypotenuse 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 coordinates 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 latitude 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,rnorm/snorm]];
% 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,rnorm/snorm]];
% 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,rnorm/snorm]];

```

```

% 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,rnm/snm]];
% 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
% precondition and project r
% solve preconditioned system
% no preconditioning
% solve two-sided expl. preconditioned system
% compute vectors and matrices for skew projection
% precondition 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 precondition=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,rnorm/snorm]];
% 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,rnorm/snorm]];
% 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 rectangles 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 eig2x2

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 laplacian\_eigenvalue

### 25.2 laplacian\_eigenvector

### 25.3 laplacian\_power

### 25.4 least\_squares\_perpendicular\_offset

### 25.5 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 quadratic 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 orthogonal 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 extrapolation

### 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  
 $c_1 x^2 + c_2 x + c_3 = 0$

### 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** `mathematics`

mathematical functions of various kind

**33.1** `monotoneous_indices`

**33.2** `nearest_fractional_timestep`

## **34** `number-theory`

**34.1** `ceiln`

floor to leading n-digits

**34.2** `digitsb`

number of digits with respect to specified base

### 34.3 floorn

floor to n-digits

### 34.4 iseven

true for even numbers in X

### 34.5 multichoosek

all combinations of length 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

### 34.6 nchoosek\_man

vectorised binomial coefficient

$b = N!/K!(N-K)!$

### 34.7 pythagorean\_triple

pythagorean triple

### 34.8 roundn

round to n digits

## 35 numerical-methods

### 35.1 advect\_analytic

## **36 numerical-methods/differentiation**

### **36.1 derivative1**

first derivative on variable mesh  
second order accurate

### **36.2 derivative2**

second derivative on a variable mesh

## **37 numerical-methods**

### **37.1 diffuse\_analytic**

## **38 numerical-methods/finite-difference**

### **38.1 cdiff**

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

### **38.2 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

### **38.3 central\_difference**

### 38.4 cmean

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

### 38.5 cmean2

### 38.6 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)
```

### 38.7 derivative\_matrix\_2\_1d

finite derivative matrix of second derivative in one dimension

### 38.8 derivative\_matrix\_2d

finite difference derivative matrix in two dimensions

### 38.9 derivative\_matrix\_curvilinear

derivative matrix on a curvilinear grid

### 38.10 derivative\_matrix\_curvilinear\_2

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

### 38.11 difference\_kernel

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

### 38.12 diffusion\_matrix\_2d\_anisotropic

### 38.13 diffusion\_matrix\_2d\_anisotropic2

### 38.14 directional\_neighbour

### 38.15 distmat

distance matrix for a 2 dimensional rectangular matrix

### 38.16 downwind\_difference

### 38.17 gradpde2d

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

### 38.18 laplacian

### 38.19 laplacian\_fdm

finite difference matrix of the laplacian  
BC

### 38.20 lrmean

mean of the left and right element

## 39 numerical-methods/finite-difference/master

### 39.1 fdm\_adaptive\_grid

### 39.2 fdm\_adaptive\_refinement\_old

### 39.3 fdm\_assemble\_d1\_2d

### 39.4 fdm\_assemble\_d2\_2d

### 39.5 fdm\_confinement

### 39.6 fdm\_d\_vargrid

### 39.7 fdm\_h\_unstructured

39.8 `fdm_hydrogen_vargrid`

39.9 `fdm_mark_unstructured_2d`

39.10 `fdm_plot`

39.11 `fdm_plot_series`

39.12 `fdm_refine_2d`

39.13 `fdm_refine_3d`

39.14 `fdm_refine_unstructured_2d`

39.15 `fdm_schroedinger_2d`

39.16 `fdm_schroedinger_3d`

39.17 `relocate`

## 40 numerical-methods/finite-difference

### 40.1 mid

mid point between neighbouring vector elements

### 40.2 pwmid

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

### 40.3 ratio

ratio of two subsequent values

### 40.4 steplength

step length of a vector if it were equispaced

### 40.5 swapoddeven

swap odd and even elements in a vector

### 40.6 test\_derivative\_matrix\_2d

### 40.7 test\_derivative\_matrix\_curvilinear

### 40.8 test\_difference\_kernel



40.9 upwind\_difference

## 41 numerical-methods/finite-element

41.1 Mesh\_2d.java

41.2 Tree\_2d.java

41.3 assemble\_1d\_dphi\_dphi

41.4 assemble\_1d\_phi\_phi

41.5 assemble\_2d\_dphi\_dphi.java

41.6 assemble\_2d\_phi\_phi.java

41.7 assemble\_3d\_dphi\_dphi.java

41.8 assemble\_3d\_phi\_phi.java

41.9    `boundary_1d`

41.10   `boundary_2d`

41.11   `boundary_3d`

41.12   `check_area_2d`

41.13   `circmesh`

41.14   `cropradius`

41.15   `display_2d`

41.16   `display_3d`

41.17   `distort`

41.18   `err_2d`

41.19 `estimate_err_2d_3`

41.20 `example_1d`

41.21 `example_2d`

41.22 `explode`

41.23 `fem_2d`

41.24 `fem_2d_heuristic_mesh`

41.25 `fem_get_2d_radial`

41.26 `fem_interpolation`

41.27 `fem_plot_1d`

41.28 `fem_plot_1d_series`

41.29 fem\_plot\_2d

41.30 fem\_plot\_2d\_series

41.31 fem\_plot\_3d

41.32 fem\_plot\_3d\_series

41.33 fem\_plot\_confine\_series

41.34 fem\_radial

adaptive grid  
constant grid

41.35 flip\_2d

41.36 get\_mesh\_arrays

41.37 hashkey

## 42 numerical-methods/finite-element/int

### 42.1 int\_1d\_gauss

### 42.2 int\_1d\_gauss\_1

### 42.3 int\_1d\_gauss\_2

### 42.4 int\_1d\_gauss\_3

### 42.5 int\_1d\_gauss\_4

### 42.6 int\_1d\_gauss\_5

### 42.7 int\_1d\_gauss\_6

### 42.8 int\_1d\_gauss\_lobatto

### 42.9 int\_1d\_gauss\_n

42.10 int\_1d\_nc\_2

42.11 int\_1d\_nc\_3

42.12 int\_1d\_nc\_4

42.13 int\_1d\_nc\_5

42.14 int\_1d\_nc\_6

42.15 int\_1d\_nc\_7

42.16 int\_1d\_nc\_7\_hardy

42.17 int\_2d\_gauss\_1

42.18 int\_2d\_gauss\_12

42.19 int\_2d\_gauss\_13

42.20 int\_2d\_gauss\_16

42.21 int\_2d\_gauss\_19

42.22 int\_2d\_gauss\_25

42.23 int\_2d\_gauss\_3

42.24 int\_2d\_gauss\_33

42.25 int\_2d\_gauss\_4

42.26 int\_2d\_gauss\_6

42.27 int\_2d\_gauss\_7

42.28 int\_2d\_gauss\_9

42.29 int\_2d\_nc\_10

42.30 int\_2d\_nc\_15

42.31 int\_2d\_nc\_21

42.32 int\_2d\_nc\_3

42.33 int\_2d\_nc\_6

42.34 int\_3d\_gauss\_1

42.35 int\_3d\_gauss\_11

42.36 int\_3d\_gauss\_14

42.37 int\_3d\_gauss\_15

42.38 int\_3d\_gauss\_24

42.39 int\_3d\_gauss\_4



42.40 int\_3d\_gauss\_45

42.41 int\_3d\_gauss\_5

42.42 int\_3d\_nc\_11

42.43 int\_3d\_nc\_4

42.44 int\_3d\_nc\_6

42.45 int\_3d\_nc\_8

## 43 numerical-methods/finite-element

43.1 interpolation\_matrix

43.2 mark

43.3 mark\_1d

43.4 mesh\_1d\_uniform

43.5 mesh\_3d\_uniform

43.6 mesh\_interpolate

43.7 neighbour\_1d

43.8 old

43.9 pdeeig\_1d

43.10 pdeeig\_2d

43.11 pdeeig\_3d

43.12 polynomial\_derivative\_1d

43.13 potential\_const

43.14 potential\_coulomb

43.15 potential\_harmonic\_oscillator

43.16 project\_circle

43.17 project\_rectangle

43.18 promote\_1d\_2\_3

43.19 promote\_1d\_2\_4

43.20 promote\_1d\_2\_5

43.21 promote\_1d\_2\_6

43.22 quadrilaterate

43.23 recalculate\_regularity\_2d

43.24 refine\_1d

43.25 refine\_2d\_21

43.26 refine\_2d\_structural

43.27 regularity\_1d

43.28 regularity\_2d

43.29 regularity\_3d

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

43.30 relocate\_2d

43.31 test\_circmesh

43.32 test\_hermite

### 43.33 tri\_assign\_points

### 43.34 triangulation\_uniform

### 43.35 vander\_1d

van der Monde matrix

### 43.36 vanderd\_1d

### 43.37 vanderi\_1d

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

### 44.1 Advection

FVM treatment of the Advection equation

### 44.2 dot\_advection

advection equation

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

### 45.1 burgers\_split

viscous Burgers' equation,  
mixed analytic and numerical derivative in frequency space  
by splitting scheme  
 $u_t = -(0.5*u^2)_x + c*u_{xx}$

## 45.2 dot\_burgers\_fdm

viscous burgers' equation  
 $u_t = -d/dx (1/2*u^2) + c d^2/dx^2 u_{xx}$

## 45.3 dot\_burgers\_fft

viscous Burgers' equation in frequency space  
 $u_t + (0.5*u^2)_x = c*u_{xx}$

# 46 numerical-methods/finite-volume/@Finite\_Volume

## 46.1 Finite\_Volume

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

## 46.2 apply\_bc

apply boundary conditions

## 46.3 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);
```

## 46.4 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

## 46.5 step\_unsplit

step in time, without splitting the inhomogeneous term

## 47 numerical-methods/finite-volume/@Flux\_Limiter

### 47.1 Flux\_Limiter

class of flux limiters

### 47.2 beam\_warming

beam warming scheme

low resolution

note: works only if sign of eigenvalues point into the same  
direction according to RL

### 47.3 fromm

fromme limiter

low res

### 47.4 lax\_wendroff

lax wendroff scheme

second order accurate, but no tvd

this is effectively not a limiter

eq. 6.39 in randall, leveque

### 47.5 minmod

min-mod schock limiter

## 47.6 monotized\_central

monotonized central flux limiter

## 47.7 muscl

muscl flux limiter

## 47.8 superbee

superbee limiter

## 47.9 upwind

godunov scheme  
godunov, first order accurate

## 47.10 vanLeer

van Leer limiter

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

## 48.1 dot\_kdv\_fdm

korteweg de vries equation  
 $u_t + (0.5*u^2)_x = c*u_{xxx}$

## 48.2 dot\_kdv\_fft

korteweg de vries equation  
compute derivatives in frequency space  
 $u_t + (0.5*u^2)_x = c*u_{xxx}$



### 48.3 kdv\_split

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

## 49 numerical-methods/finite-volume/@Reconstruct\_Average\_Evolve

### 49.1 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;
```

### 49.2 advect\_highres

single time step for the reconstruct evolve algorithm

### 49.3 advect\_lowress

single time step

low resolution

## 50 numerical-methods/finite-volume

### 50.1 Godunov

Godunov, upwind method for systems of pdes

## 50.2 Lax-Friedrich

Lax-Friedrich-Method  
for hyperbolic conservation laws  
 $\text{err} = O(\text{dt}) + O(\text{dx})$   
 $|a \text{ dt/dx}| < 1$

## 50.3 Measure

## 50.4 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

## 50.5 fv\_swe

wrapper for solving SWE

## 50.6 staggered\_euler

forward euler method with staggered grid

## 50.7 staggered\_grid

staggered grid approximation to the SWE

# 51 numerical-methods

## 51.1 grid2quad

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

## 52 numerical-methods/integration

### 52.1 cumintL

cumulative integral from left to right

### 52.2 cumintR

cumulative integral from right to left

### 52.3 cumint\_trapezoidal

integrate y along x with the trapezoidal rule

### 52.4 int\_trapezoidal

integrate y along x with the trapezoidal rule

## 53 numerical-methods/interpolation/@Kriging

### 53.1 Kriging

class for Kriging interpolation

### 53.2 estimate\_semivariance

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

### 53.3 interpolate\_

interpolate with Krieking 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 : target point coordinates  
Vt : value at target points  
E2t : squared interpolation error at target points

## 54 numerical-methods/interpolation/@RegularizedInterpolator

### 54.1 RegularizedInterpolator1

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

#### 54.2 init

initialize the interpolator with a set of sampling points

## 55 numerical-methods/interpolation/@RegularizedInterpolator

### 55.1 RegularizedInterpolator2

class for regularized interpolation on an unstructures mesh (  
interpolation)

#### 55.2 init

initialize the interpolator with a set of point samples

## 56 numerical-methods/interpolation/@RegularizedInterpolator3

### 56.1 RegularizedInterpolator3

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

#### 56.2 init

initialize the interpolator with a set of sampling points

## 57 numerical-methods/interpolation

### 57.1 IDW

spatial averaging by inverse distance weighting

### 57.2 IPoly

polynomial interpolation class

### 57.3 IRBM

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

### 57.4 ISparse

sparse interpolation class

### 57.5 Inn

nearest neighbour interpolation

## 57.6 Interpolator

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

## 57.7 fixnan

fill nan-values in vector with gaps

## 57.8 idw1

spatial average by inverse distance weighting

## 57.9 idw2

spatial average by inverse distance weighting

## 57.10 inner2outer

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

## 57.11 inner2outer2

interpolate from element (segment) centres to edge points

## 57.12 interp1\_circular

### 57.13 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

### 57.14 interp1\_man

interpolate

### 57.15 interp1\_piecewise\_linear

### 57.16 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

### 57.17 interp1\_slope

quadratic interpolation returning value and derivative(s)

### 57.18 interp1\_smooth

### 57.19 interp1\_unique

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

## 57.20 `interp2_man`

nearest neighbour interpolation in two dimensions

## 57.21 `interp_angle`

interpolate an angle

## 57.22 `interp_fourier`

interpolation by the fourier method

## 57.23 `interp_fourier_batch`

batch interpolation by the fourier interpolation

## 57.24 `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_);`

## 57.25 `interp_sn2`

interpolation in streamwise coordinates

## 57.26 `interp_sn3`

## 57.27 `interp_sn_`



### 57.28 limit\_by\_distance\_1d

smooth subsequent values along a curve such that  
 $v(x_0+dx) < v(x_0) + (\text{ratio}-1)*dx$   
if  $v$  is the edge length in a resampled polygon, then  $v_i/v_{(i+1)} <$   
ratio  
 $\text{ratio}^1 = \exp(a*1)$

### 57.29 resample1

interpolation along a parametric curve with variable step width

### 57.30 resample\_d\_min

resample a function

### 57.31 resample\_vector

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

### 57.32 test\_interp1\_limited

## 58 numerical-methods

### 58.1 inverse\_complex

### 58.2 maccormack\_step

### 58.3 minmod

## **59 numerical-methods/multigrid**

### **59.1 mg\_interpolate**

### **59.2 mg\_restrict**

## **60 numerical-methods/ode/@BVPS\_Characteristic**

### **60.1 BVPS\_Characteristic**

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

### **60.2 assemble1\_A**

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

### **60.3 assemble1\_A\_Q**

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

### **60.4 assemble2\_A**

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

### **60.5 assemble\_AA**

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

## 60.6 assemble\_AAA

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

## 60.7 assemble\_Ic

## 60.8 bvp1c

## 60.9 check\_arguments

## 60.10 couple\_junctions

## 60.11 derivative

## 60.12 init

## 60.13 inner2outer\_bvp2c

## 60.14 reconstruct

## 60.15 resample

## 60.16 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])

## 60.17 test\_assemble1\_A

## 60.18 test\_assemble2\_A

# 61 numerical-methods/ode/@Time\_Stepper

## 61.1 Time\_Stepper

## 61.2 solve

## 62 numerical-methods/ode

### 62.1 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])

### 62.2 bvp2wavetrain

solve second order boundary value problem by repeated integration

### 62.3 bvp2wavetwopass

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

### 62.4 ivp\_euler\_forward

solve initial value problem by the euler forward method

### 62.5 ivp\_euler\_forward2

## 62.6 ivprk2

solve initial value problem by the two step runge kutta method

## 62.7 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]
```

## 62.8 ode2characteristic

second order odes  
transmitted and reflected wave

## 62.9 step\_trapezoidal

single trapezoidal step

## 62.10 test\_bvp2

# 63 numerical-methods/optimisation

## 63.1 aitken\_iteration

## 63.2 anderson\_iteration

### 63.3 armijo\_stopping\_criterion

armijo stopping criterion for optimizations

### 63.4 astar

astar path finding algorithm

### 63.5 binsearch

binary search on a line

### 63.6 bisection

bisection

### 63.7 box1

test objective function for optimisation routines

### 63.8 box2

### 63.9 cauchy

### 63.10 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

### 63.11 directional\_derivative

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

### 63.12 dud

optimization by the dud algorithm

### 63.13 extreme3

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

input  
t : sampling time (uniformly spaced)  
v : values at sampling times  
output:  
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

### 63.14 extreme\_quadratic

### 63.15 ftest

### 63.16 fzero\_bisect



### 63.17 `fzero_newton`

### 63.18 `grad`

numerical gradient

### 63.19 `hessian`

numerical hessian

### 63.20 `hessian_from_gradient`

numerical hessian from gradient

### 63.21 `hessian_projected`

numerical hessian projected to one dimension

### 63.22 `line_search`

bisection routine

### 63.23 `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`

### 63.24 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
```

### 63.25 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
```

### 63.26 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
```

### 63.27 line\_search\_quadratic2

```
quadratic line search
```

### 63.28 line\_search\_wolfe

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

### 63.29 ls\_bgfs

least squares by the bgfs method

### 63.30 ls\_broyden

least squares by the broyden method  
for rectangular / non symmetric systems  
Numerical Optimization nokedal  
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

### 63.31 ls\_generalized\_secant

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

### 63.32 nlcg

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

opt : struct options  
fdx : gradient constraint

### **63.33 nlls**

non-linear least squares

### **63.34 picard**

picard iteration

### **63.35 poly\_extrema**

extrema of a polynomial

### **63.36 quadratic\_function**

evaluate quadratic function in higher dimensions

### **63.37 quadratic\_programming**

optimize by quadratic programming

### **63.38 quadratic\_step**

single step of the quadratic programming

### **63.39 rosenbrock**

rosenbrock test function

#### 63.40 `sqrt_heron`

Heron's method for the square root

#### 63.41 `test_directional_derivative`

#### 63.42 `test_dud`

#### 63.43 `test_fzero_newton`

#### 63.44 `test_line_search_quadratic2`

#### 63.45 `test_ls_generalized_secant`

#### 63.46 `test_nlcg_6_order`

#### 63.47 `test_nlls`

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

### 64 numerical-methods/pde

#### 64.1 `laplacian2d_fundamental_solution`

## 65 numerical-methods/piecewise-polynomials

### 65.1 Hermite1

hermite polynomial interpolation in 1d

### 65.2 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)

### 65.3 hp2\_predict

prediction with pw hermite polynomial  
c are values at support points

### 65.4 hp\_predict

predict with piecewise hermite polynomial

### 65.5 hp\_regress

fit piecewise hermite polynomial  
coefficients are values and derivatives

### 65.6 lp\_count

lagrangian basis for interpolation  
count number of valid samples

## 65.7 lp\_predict

lagrangian basis piecwie interpolation, predicator

## 65.8 lp\_regress

## 65.9 lp\_regress\_

# 66 numerical-methods

## 66.1 test\_adams\_bashforth

# 67 mathematics

mathematical functions of various kind

## 67.1 oversampleNZ

# 68 regression/@PolyOLS

## 68.1 PolyOLS

class for polynomial least squares

## 68.2 coefftest

### 68.3 detrend

detrending by polynomial regression

### 68.4 fit

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

### 68.5 fit\_

fit a polynomial function

### 68.6 predict

predict polynomial function values

### 68.7 predict\_

### 68.8 slope

slope by linear regression

## 69 regression/@PowerLS

### 69.1 PowerLS

class for power law regression



## 69.2 fit

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

## 69.3 predict

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

## 69.4 predict\_

# 70 regression/@Theil

## 70.1 Theil

Kendal-Theil-Sen robust regression

## 70.2 detrend

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

## 70.3 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

## 70.4 predict

predict values and confidence intervals with the Theil-Sen method

## 70.5 slope

fit the slope with the Theil-Sen method

## 71 regression

linear and non-linear regression

### 71.1 Theil\_Multivariate

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

### 71.2 areg

regression using the pth-fraction of samples with smallest residual

### 71.3 ginireg

gini regression

### 71.4 hesssimplereg

hessian, gradient and objective function value of the simple  
regression  
 $\text{rhs} = p(1) + p(2) x + \text{eps}$

### 71.5 l1lin

solve  $\|Ax - b\|_{L1}$  by means of linear programming

## 71.6 lsq\_sparam

parameter covariance of the least squares regression

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

## 71.7 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

## 71.8 regression\_method\_of\_moments

fit linear function  $\|a \ b \ x = y\|_{L2}$  by the method of moments  
 $y + \epsilon = \alpha + \beta x$

## 71.9 robustlinreg

fit a linear function by splitting the x-values at their median  
 $(\text{med}(y_{\text{left}}) - \text{med}(y_{\text{right}})) / (\text{med}(x_{\text{left}}) - \text{med}(x_{\text{right}}))$   
this approach performs poorly compared to the theil-senn operator

## 71.10 theil2

Theil senn-estimator for two dimensions (glm)

## 71.11 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

## 71.12 total\_least\_squares

total least squares

## 71.13 weighted\_median\_regression

weighted median regression  
c.f. Scholz, 1978

## 72 set-theory

### 72.1 issubset

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

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

## 73 mathematics

mathematical functions of various kind

### 73.1 shuffle\_index

## 74 signal-processing

### 74.1 asymwin

creates asymmetrical filter windows  
filter will always have negative weights

## 75 signal-processing/autocorrelation

### 75.1 acf\_radial

### 75.2 acfar1

Autocorrelation function of the finite AR1 process

$$\begin{aligned} a_k &= 1/(n-k) \sum x_{i+k} x_i + (x_i + x_{i+k})\mu + \mu^2 \\ &= r^k + 1/n \sum_{i=j} x_i + 1/n \\ &\quad \text{pause} \end{aligned}$$

### 75.3 acfar1\_2

autocorrelation of the ar1 process

### 75.4 acfar2

impulse response of the ar2 process

### 75.5 acfar2\_2

autocorrelation of the ar2 process  
 $X_i + a_1 X_{i-1} + a_2 X_{i-2} = 0$

### 75.6 ar1\_cutoff\_frequency

### 75.7 ar1\_effective\_sample\_size

effective sample size correction for autocorrelated series

## 75.8 ar1\_mse\_mu\_single\_sample

standard error of a single sample of an ar1 correlated process

## 75.9 ar1\_mse\_pop

variance of the population mean of a single realisation around zero

$$E[(\mu_N - 0)^2] = E[\mu_N^2]$$

## 75.10 ar1\_mse\_range

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

## 75.11 ar1\_spectrum

spectrum of the ar1 process

## 75.12 ar1\_to\_tikhonov

convert ar1 correlation to tikhonovs lambda

## 75.13 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

## 75.14 ar1\_var\_factor\_

variance of an autocorrelated finite process

### 75.15 ar1\_var\_range2

variance of sub sample starting at the end of the series  
from the finite length first order autocorrelated process  
$$s2 = 1/m^2 \sum_i^m \sum_j^m \rho^{|i-j|}$$

### 75.16 ar1delay

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

### 75.17 ar1delay\_old

autocorrelation of the residual

### 75.18 ar2\_acf2c

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

### 75.19 ar2conv

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

### 75.20 ar2dof

effective samples size for the ar2 process

## 75.21 ar2param

ar2 parameter estimation from first two terms of acf

acf = [1 a1 a2 ...]

## 75.22 autocorr2

## 75.23 autocorr\_angular

## 75.24 autocorr\_bandpass

## 75.25 autocorr\_brownian\_phase

## 75.26 autocorr\_brownian\_phase\_2d

## 75.27 autocorr\_brownian\_phase\_across

## 75.28 autocorr\_decay\_rate

estimate exponential decay of the autocorrelation

## 75.29 autocorr\_effective\_sample\_size

effective sample size from acf



### 75.30 autocorr\_fft

estimate sample autocorrelation function

### 75.31 autocorr\_forest

### 75.32 autocorr\_genton

autocorrelation function

### 75.33 autocorr\_highpass

### 75.34 autocorr\_lowpass

### 75.35 autocorr\_periodic\_additive\_noise

### 75.36 autocorr\_periodic\_windowed

### 75.37 autocorr\_radial

### 75.38 autocorr\_radial\_hexagonal\_pattern

### 75.39 autocorrelation\_max

## 76 signal-processing

### 76.1 average\_wave\_shape

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

### 76.2 bandpass

bandpass filter

### 76.3 bandpass\_continuous\_cdf

### 76.4 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  $a = \rho^k = \exp(-k/L)$ ,  $T = 2L$

### 76.5 bin1d

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

## 76.6 bin2d

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

## 76.7 binormrnd

generate two correlated normally distributed vectors

## 76.8 brownian\_phase\_cdf

## 76.9 brownian\_phase\_inv

## 76.10 coherence

## 76.11 conv1\_man

convolutions with padding

## 76.12 conv2\_man

convolution in 2d

## 76.13 conv2z

#### 76.14 conv30

convolve with rectangular window of length n  
circular boundaries

#### 76.15 conv\_

convolution of a with b

#### 76.16 conv\_centered

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

#### 76.17 convz

#### 76.18 cosexpdelay

#### 76.19 csmooth

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

#### 76.20 daniell\_window

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

### 76.21 db2neper

convert decibel to neper

### 76.22 db2power

power ratio from db

### 76.23 derive\_bandpass\_continuous\_scale

### 76.24 derive\_danielle\_weight

### 76.25 derive\_limit\_0\_acfar

### 76.26 detect\_peak

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

### 76.27 determine\_phase\_shift

### 76.28 determine\_phase\_shift1

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

## 76.29 doublesum\_ij

double sum of  $r^i$

## 76.30 effective\_mask\_size

## 76.31 effective\_sample\_size\_to\_ar1

convert effective sample size to ar1 correlation

## 76.32 fcut2Lw\_gausswin

## 76.33 fcut\_gausswin

## 76.34 filt\_hodges\_lehman

# 77 signal-processing/filters

## 77.1 circfilt2

Mon 19 Dec 17:03:02 CET 2022  
smooth (filter) the 2D image z with a circular disk of radius nf  
apply periodic boundary conditions

## 77.2 filter1

filter along one dimension

### 77.3 filter2

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

### 77.4 filter\_

invalidate values that exceed n-times the robust standard deviation

### 77.5 filter\_r\_to\_f0

### 77.6 filter\_rho\_to\_f0

### 77.7 filter\_twosided

### 77.8 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 position in the column (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

## 77.9 filterp

### 77.10 filterp1

fir filter with some fancy extras

### 77.11 filterstd

### 77.12 gaussfilt2

smooth (filter) the 2D image z with a gaussian window

apply periodic boundary conditions

### 77.13 lowpass\_discrete

design coefficients of a low pass filter with specified cut of frequency

and sampling period

alalogue low pass with pole at  $s = -\omega_c = 1/\tau = 1/RC$

$$H_a = \tau / (\tau + s) = 1 / (1 + \omega_c s)$$



#### 77.14 meanfilt2

filter with a rectangular window along both dimensions

#### 77.15 medfilt1\_man

moving median filter, supports columnwise operation

#### 77.16 medfilt1\_man2

moving median filter with special treatment of boundaries

#### 77.17 medfilt1\_padded

median filter with padding

#### 77.18 medfilt1\_reduced

median filter with padding

#### 77.19 trfilt1

filter with triangular window

trifilt1 is ident to twice applying rectfilt1 (meanfilt1) with half  
the domain size

note : infinitely many convolution yield a gaussian

#### 77.20 trifilt2

filter with a triangular window along both dimensions

## 78 signal-processing

### 78.1 firls\_man

design finite impulse response filter by the least squares method

### 78.2 fit\_spectral\_density

fit (spectral) densities

### 78.3 fit\_spectral\_density\_2d

fit spectral densities

### 78.4 fit\_spectral\_density\_radial

fit spectral densities

### 78.5 flattopwin

the flat top window

### 78.6 frequency\_response\_boxcar

frequency response of a boxcar filter

### 78.7 freqz\_boxcar

frequency response of a boxcar filter

### 78.8 gaussfilt1

filter data series with a gaussian window, assumes periodic bc

## 78.9 hanchangewin

hanning window for change point detection

## 78.10 hanchangewin2

hanning window for change point detection

## 78.11 hanwin

hanning filter window

## 78.12 hanwin\_

hanning filter window

## 78.13 high\_pass\_1d\_simple

## 78.14 kaiserwin

kaiser filter window

## 78.15 kalman

Kalman filter

## 78.16 lanczoswin

Lanczos window

### 78.17 last

lake tail, but for matrices

### 78.18 maxfilt1

### 78.19 meanfilt1

moving average filter with special treatment of the boundaries

### 78.20 mid\_term\_single\_sample

variance of single sample, mid term

### 78.21 minfilt1

### 78.22 minmax

### 78.23 mu2ar1

error variance of the mean of the finite length ar1 process

$(\mu)^2 = (\sum \text{epsi})^2 = \sum_i \sum_j \text{epsi}_i \text{epsi}_j = \sum_{ii}(\text{rho}, n)/n^2$   
this has the limit  $s^2$  for  $\text{rho} \rightarrow 1$

### 78.24 mysmooth

### **78.25 nanautocorr**

autocorrelation with nan-values

### **78.26 nanmedfilt1**

medfilt1, skipping nans

### **78.27 neper2db**

convert neper to db

### **78.28 oscillator\_noisy**

## **79 signal-processing/passes**

### **79.1 bandpass1d**

### **79.2 bandpass1d\_fft**

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

### **79.3 bandpass1d\_implicit**

### **79.4 bandpass2**

bandpass filter

**79.5**   `bandpass2d`

**79.6**   `bandpass2d_2`

**79.7**   `bandpass2d_fft`

**79.8**   `bandpass2d_ideal`

**79.9**   `bandpass2d_implicit`

**79.10**   `bandpass2d_iso`

**79.11**   `bandpass_arg`

determine correlation coefficient from frequency of mode for the  
symmetric

**79.12**   `bandpass_f0_to_rho`

correlation coefficient for the pth-order symmetric bandpass filter  
with  
maximum at  $f_0$  (when  $\rho_{lp} = \rho_{hp}$ )

**79.13**   `bandpass_max`

**79.14** `bandpass_max2`

**79.15** `highpass`

high pass filter

**79.16** `highpass1d_fft_cos`

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

**79.17** `highpass1d_implicit`

**79.18** `highpass2d_fft`

**79.19** `highpass2d_ideal`

**79.20** `highpass2d_implicit`

**79.21** `highpass_arg`

**79.22** `highpass_fc_to_rho`

### 79.23 lowpass

low pass filter

### 79.24 lowpass1d\_fft

### 79.25 lowpass1d\_implicit

### 79.26 lowpass2

design low pass filter with cutoff-frequency f1

### 79.27 lowpass2d\_2

### 79.28 lowpass2d\_anisotropic

### 79.29 lowpass2d\_fft

### 79.30 lowpass2d\_ideal

### 79.31 lowpass2d\_implicit



### **79.32 lowpass\_arg**

### **79.33 lowpass\_fc\_to\_rho**

### **79.34 lowpass\_iir**

iir-low pass

### **79.35 lowpass\_iir\_symmetric**

two-sided iir low pass filter (for symmetry)

### **79.36 lowpassfilter2**

low-pass filter of data

## **80 signal-processing**

### **80.1 peaks\_man**

peaks of a periodogram

## **81 signal-processing/periodogram**

### **81.1 periodogram**

compute the normalized periodogram

### **81.2 periodogram\_2d**

compute the normalized periodogram in two dimensions

### 81.3 periodogram\_align

### 81.4 periodogram\_angular

### 81.5 periodogram\_bartlett

estimate the spectral density nonparametrically with Bartlett's  
method

### 81.6 periodogram\_bootstrap

### 81.7 periodogram\_confidence\_interval

confidence interval for periodogram values

### 81.8 periodogram\_filter

### 81.9 periodogram\_median

### 81.10 periodogram\_p\_value

### 81.11 periodogram\_qq

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

## 81.12 periodogram\_quantiles

quantiles of a periodogram

## 81.13 periodogram\_radial

## 81.14 periodogram\_std

standard deviation of a periodogram

## 81.15 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
```

TODO pass L and not fx

## 81.16 periodogram\_test\_periodicity\_2d

test a periodogram for hidden periodic frequency components

```
[p,stat,ratio] = periodogram_test_periodicity_2d(b, L, nf, frmin,
    frmax)
```

input:

- fx : frequengcies
- b : image to test for presence of hidden periodicities,  
i.e. periodicities where the frequency is not known a  
priori
- nf : radius of circular disk (in number of bins) used for  
smoothing  
the periodogram to estimate the spectral density
- bmask : mask determining parts of the image to include in the  
analysis  
default is entire image
- fmin, fmax : (radial) frequency range limits to test (fmask)
- 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

influence of masking the input file:

- the root-mean-square energy of the ordinates is  
proportional  
to the number of unmasked points
- values in the periodogram are not any more linearly  
independent  
so that the dof of the filter window is not  $nf^2$

TODO make frmin, frmax an fmask

### 81.17 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

### 81.18 periodogram\_welsh

## 82 signal-processing

### 82.1 polyfilt1

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

### 82.2 qmedfilt1

medfilt1, after fitting a quadratic polynomial

### 82.3 quadratfilt1

### 82.4 quadratwin

### 82.5 randar1

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

### 82.6 randar1\_dual

draw random variables of two correlated ar1 processes

### 82.7 randar2

generate ar2 process

### 82.8 randarp

randomly generate the instance of an ar-p process

## 82.9 rectwin

rectangular window

## 82.10 recursive\_sum

## 82.11 select\_range

## 82.12 smooth1d\_parametric

smooth position of  $p_0=x_0, y_0$  between  $p_1=x_1, y_1$  and  $p_2=x_2, y_2$ ,  
so that distance to  $p_1$  and  $p_2$  becomes equal  
and the chord length remains the same

## 82.13 smooth2

smooth vectos of X

## 82.14 smooth\_man

## 82.15 smooth\_parametric

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

## 82.16 smooth\_parametric2

parametrically smooth the curve

82.17 `smooth_with_splines`

82.18 `smoothfft`

`filter with fast fourier transform`

## 83 `signal-processing/spectral-density`

83.1 `gampdf_man`

83.2 `hex_angular_pdf`

83.3 `hex_angular_pdf_max`

83.4 `hex_angular_pdf_max2par`

83.5 `lognpdf_entropy`

83.6 `misespdf`

83.7 `normpdf_entropy`

### 83.8 skewpdf\_entropy

### 83.9 spectral\_density\_ar2

### 83.10 spectral\_density\_area

integrate the spectral density

### 83.11 spectral\_density\_bandpass2d\_ideal

### 83.12 spectral\_density\_bandpass\_2d

### 83.13 spectral\_density\_bandpass\_2d\_max2par

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

### 83.14 spectral\_density\_bandpass\_2d\_scale

### 83.15 spectral\_density\_bandpass\_2d\_scale\_old



### 83.16 spectral\_density\_bandpass\_continuous

```
function [S_bp,Sc] = spectral_density_bandpass_continuous(fx,fc,
    order,normalize,pp)

output :
S_bp : spectral density of the bandpass filter in continuous space
      limit case of the discrete bandpass for  $dx \rightarrow 0$ 
Sc   : scale factor to normalize area to 1, if normalize = true

input :
f     : frequency (abszissa)
fc    : central frequency, location of maximum on abszissa
order : number of times filter is applied iteratively, not
       necessarily integer
normalize : normalize area under curve  $\int_0^\infty S(f) df = 1$ , if
          not maximum  $S(fc) = 1$ 
pp    : powers for recombination of the lowpass filter
```

### 83.17 spectral\_density\_bandpass\_continuous\_max

maximum of the bandpass spectral density

### 83.18 spectral\_density\_bandpass\_continuous\_max2par

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

### 83.19 spectral\_density\_bandpass\_continuous\_scale

normalization scale of the spatial bandpass density

### 83.20 spectral\_density\_bandpass\_discrete

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

### 83.21 spectral\_density\_brownian\_phase

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

### 83.22 spectral\_density\_brownian\_phase\_2d

### 83.23 spectral\_density\_brownian\_phase\_across

### 83.24 spectral\_density\_brownian\_phase\_across\_max

### 83.25 spectral\_density\_brownian\_phase\_across\_max2par

### 83.26 spectral\_density\_brownian\_phase\_across\_mode2par

### 83.27 spectral\_density\_brownian\_phase\_mode

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

### 83.28 spectral\_density\_brownian\_phase\_mode2par

transform mode to parameters of the brownian phase spectral density

### 83.29 `spectral_density_brownian_phase_scale`

normalization scale of the brownian phase spectral density

### 83.30 `spectral_density_estimate_2d`

### 83.31 `spectral_density_flat`

flat spectral density of a random vector with iid elements

### 83.32 `spectral_density_forest`

### 83.33 `spectral_density_gausswin`

### 83.34 `spectral_density_highpass`

### 83.35 `spectral_density_highpass2d_ideal`

### 83.36 `spectral_density_highpass_2d`

### 83.37 `spectral_density_highpass_cos`

consine shaped spectral density of a highpass filter

### 83.38 `spectral_density_lorentzian`

lorentzian spectral density

### 83.39 `spectral_density_lorentzian_max`

mode (maximum) of the lorentzian spectral density

### 83.40 `spectral_density_lorentzian_max2par`

transform maximum of the lorentzian spectral density to its  
distribution parameters

### 83.41 `spectral_density_lorentzian_scale`

normalization scale of the lorentzian spectral density

### 83.42 `spectral_density_lowpass2d_ideal`

### 83.43 `spectral_density_lowpass_2d`

### 83.44 `spectral_density_lowpass_continuous`

### 83.45 `spectral_density_lowpass_continuous_scale`

### 83.46 `spectral_density_lowpass_discrete`

83.47 spectral\_density\_lowpass\_one\_sided

83.48 spectral\_density\_maximum\_bias\_corrected

83.49 spectral\_density\_periodic\_additive\_noise

83.50 spectral\_density\_rectwin

83.51 spectral\_density\_wperiodic

## 84 signal-processing

84.1 spectral\_density\_brownian\_phase\_reg2par

84.2 spectrogram

spectrogram

84.3 sum\_i\_lag

sum of ar1 matrix with lag  
 $\sum_{i=1}^n \rho^{|i-k|}$

#### 84.4 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

#### 84.5 sum\_ii\_

#### 84.6 sum\_ij

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

#### 84.7 sum\_ij\_

#### 84.8 sum\_ij\_partial\_

#### 84.9 sum\_multivar

sum of matrix entries of bivariate ar1 process

#### 84.10 test\_acfar1

#### 84.11 tikhonov\_to\_ar1

convert coefficient of the tikhonov regularization to correlation  
of the ar1 process

## 84.12 trapwin

trapezoidal filter window

## 84.13 triwin

triangular filter window

## 84.14 triwin2

triangular filter window

## 84.15 tukeywin\_man

## 84.16 varar1

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

## 84.17 welch\_spectrogram

welch spectrogram

## 84.18 wfilt

filter with window

## 84.19 winbandpass

filter with bandpass

## 85 signal-processing/windows

### 85.1 circwin

### 85.2 danielle\_window

danielle fourier window

### 85.3 gausswin

### 85.4 gausswin1

### 85.5 gausswin2

### 85.6 radial\_window

radial filter window in the 2d-frequency domain

### 85.7 range\_window

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

### 85.8 rectwin\_cutoff\_frequency

### 85.9 std\_window

moving block standard deviation



### 85.10 window2d

### 85.11 window\_make\_odd

## 86 signal-processing

### 86.1 winfilt0

filter with window

### 86.2 winlength

window length for desired cutoff frequency  
power at  $f_c$  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

### 86.3 wmeanfilt

mean filter with window

### 86.4 wmedfilt

median filter with window

### 86.5 wordfilt

weighted order filter

## 86.6 wordfilt\_edgeworth

weighed order filter

## 86.7 wrapphase

## 86.8 xar1

## 86.9 xcorr\_man

cross correlation of two sampled ar1 processes

## 87 sorting

### 87.1 sort2

sort two numbers

### 87.2 sort2d

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

## 88 spatial-pattern-analysis/@Spatial\_Pattern

### 88.1 Spatial\_Pattern

### 88.2 analyze\_grid

88.3   analyze\_transect

88.4   extract\_improfile

88.5   fit\_parametric\_densities

88.6   imread

88.7   plot

88.8   plot\_transect

88.9   report

88.10   resample\_functions

88.11   tabulate

## 89 spatial-pattern-analysis

### 89.1 approximate\_ratio\_distribution

### 89.2 banded\_pattern

### 89.3 brownian\_phase\_patch\_size\_distribution

### 89.4 hexagonal\_pattern

### 89.5 patch\_size\_1d

### 89.6 patch\_size\_2d

### 89.7 reconstruct\_isotropic\_density

### 89.8 separate\_isotropic\_from\_anisotropic\_density

### 89.9 suppress\_low\_frequency\_lobe

## 90 special-functions

### 90.1 `bessel_sphere`

spherical Bessel function of the first kind

### 90.2 `besseliln_large_x`

### 90.3 `beta_man`

### 90.4 `betainc_man`

### 90.5 `digamma_man`

### 90.6 `exp10`

### 90.7 `hankel_sphere`

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

### 90.8 `hermite`

probabilistic's hermite polynomial by recurrence relation

input :  
n : order  
x : value

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

## 90.9 legendre\_man

legendre polynomials

## 90.10 neumann\_sphere

spherical Neumann function  
Bessel function of the second kind

# 91 statistics

## 91.1 atan\_s2

stadard deviation of the arcus tangens by means of taylor expansion

## 91.2 beta\_kurt

## 91.3 beta\_mean

## 91.4 beta\_moment2param

transform central moments (mean and sd) to parameters of the beta  
function

## 91.5 beta\_skew

**91.6**   `beta_std`

**91.7**   `chi2_kurt`

**91.8**   `chi2_mean`

**91.9**   `chi2_skew`

**91.10**   `chi2_std`

**91.11**   `coefficient_of_determination`

**91.12**   `conditional_expectation_normal`

**91.13**   `correlation_confidence_pearson`

confidence intervals of the correlation coefficient  
c.f. Fischer 1921

## **92**   `statistics/distributions`

**92.1**   `PDF`

class for quasi-distributions from a set of sampling points

## 92.2 `binorm_separation_coefficient`

separation coefficient of a bimodal normal distribution

## 92.3 `binormcdf`

bio-modal gaussian distribution

## 92.4 `binormfit`

fit sum of to normal distribution to a histogram

## 92.5 `binormpdf`

## 92.6 `edgeworth_cdf`

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

## 92.7 `edgeworth_pdf`

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

## 92.8 `exppdf_max2par`



## 92.9 gam\_moment2param

## 92.10 logn\_mean

## 92.11 logn\_mode

mode (maximum) of the log-normal density

## 92.12 logn\_mode2param

## 92.13 logn\_moment2param

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

## 92.14 logn\_param2moment

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

## 92.15 logn\_std

## 92.16 lognpdf\_

log normal distribution called by modes rather than parameters

## 92.17 normpdf\_wrapped

**92.18** `normpdf_wrapped_mode`

**92.19** `normpdf_wrapped_mode2par`

**92.20** `pdfsample`

pdf from sample distribution  
Note: better use kernel density estimates

**92.21** `t2cdf`

Hotelling's T-squared cumulative distribution

**92.22** `t2inv`

inverse of Hotelling's T-squared cumulative distribution

## **93** `statistics`

**93.1** `error_propagation_fraction`

**93.2** `error_propagation_product`

**93.3** `example_standard_error_of_sample_quantiles`

#### 93.4 f\_var\_finite

reduction of variance when sampling from a finite population  
without replacement

#### 93.5 fisher\_mean

#### 93.6 fisher\_moment2param

#### 93.7 fisher\_std

#### 93.8 gam\_mean

#### 93.9 gam\_std

#### 93.10 gamma\_mode

#### 93.11 gamma\_mode2par

#### 93.12 gamma\_moments\_to\_parameter

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

### 93.13 gamma\_stirling

### 93.14 gaussfit3

### 93.15 gaussfit\_quantile

### 93.16 geoserr

### 93.17 geostd

### 93.18 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

### 93.19 hodges\_lehmann\_dispersion

dispersion determined by the hodges lehman method  
asymptotic efficiency of dispersion estimates:  
standard deviation:  $E(s - \hat{s})/s = 2/\sqrt{2n} \sim 0.707/\sqrt{n}$   
(100%)  
hodges lehmann dispersion  $E(s - \hat{s})/s = (\pi/3)^2 / (\sqrt{2n}) \sim$   
 $0.775/\sqrt{n}$  (91%)  
mad  $E(s - \hat{s})/s \sim 1.17 s/\sqrt{n}$   
(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

## 94 statistics/information-theory

### 94.1 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

### 94.2 bayesian\_information\_criterion

bayesian information criterion

## 95 statistics

### 95.1 jackknife\_block

### 95.2 kurtncdf

### 95.3 kurtnpdf

### 95.4 kurtosis\_bias\_corrected

bias corrected kurtosis

## 95.5 limit

limit a by lower and upper bound

## 95.6 logfactorial

approximate log of the factorial

## 95.7 loglogpdf

## 95.8 lognfit\_quantile

## 95.9 logskewcdf

## 95.10 logskewpdf

# 96 statistics/logu

## 96.1 lambertw\_numeric

lambert-w function

## 96.2 logtrialtcdf

pdf of a logarithmic triangular distribution

### 96.3 logtrialtinv

inverse of the logarithmic triangular distribution

$$= (d F \log(a) \log(b) + a \log(b) - b \log(a) - d F \log(a) \log(c) - a \log(c) + d F \log(b) \log(c) + b \log(c) - d F \log^2(b)) / ((\log(a) - \log(b)) W((a^{-1/(\log(a) - \log(b))}) (b^{-\log(c)/\log(a)} - 1/\log(a)) c^{-\log(a)/(\log(a) - \log(b))}) (-d F \log^2(b) + a \log(b) + d F \log(a) \log(b) + d F \log(c) \log(b) - b \log(a) - a \log(c) + b \log(c) - d F \log(a) \log(c))) / (\log(a) - \log(b)))$$
$$x = (d F \log(a) \log(b) + a \log(b) - b \log(a) - d F \log(a) \log(c) - a \log(c) + d F \log(b) \log(c) + b \log(c) - d F \log^2(b)) / ((\log(a) - \log(b)) W((a^{-1/(\log(a) - \log(b))}) (b^{-\log(c)/\log(a)} - 1/\log(a)) c^{-\log(a)/(\log(a) - \log(b))}) (-d F \log^2(b) + a \log(b) + d F \log(a) \log(b) + d F \log(c) \log(b) - b \log(a) - a \log(c) + b \log(c) - d F \log(a) \log(c))) / (\log(a) - \log(b)))$$

### 96.4 logtrialtmean

mean of the logarithmic triangular distribution

### 96.5 logtrialtpdf

density of the logarithmic triangular distribution

### 96.6 logtrialtrnd

### 96.7 logtricdf

cumulative distribution of the logarithmic triangular distribution

### 96.8 logtriinv

invere of the logarithmic triangular distribution

## **96.9 logtrimean**

mean of the logarithmic triangular distribution

## **96.10 logtripdf**

probability density of the logarithmic triangular distribution

## **96.11 logtirnd**

## **96.12 logucdf**

probability density of the logarithmic uniform distribution

## **96.13 logucm**

central moments of the log-uniform distribution

## **96.14 loguinv**

inverse of the log-uniform distribution

## **96.15 logumean**

mean of the log-uniform distribution

## **96.16 logupdf**

pdf of the log uniform distribution



### **96.17 logurnd**

random numbers following a log-uniform distribution

### **96.18 loguvar**

variance of the log-uniform distribution

### **96.19 medlogu**

median of the log-uniform distribution

### **96.20 test\_logurnd**

### **96.21 tricdf**

cumulative distribution of the log-triangular distribution

### **96.22 triinv**

inverse of the triangular distribution

### **96.23 trimedian**

median of the triangular distribution

### **96.24 tripdf**

probability density of the triangular distribution

## 96.25 trirnd

random numbers of the triangular distribution

## 97 statistics

### 97.1 max\_exprnd

### 97.2 maxnnormals

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

### 97.3 mean\_angle

### 97.4 mean\_generalized\_gampdf

### 97.5 midrange

mid range of columns of X

### 97.6 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

## 97.7 mode\_man

## 98 statistics/moment-statistics

### 98.1 autocorr\_man3

autocorrelation of the columns of X

### 98.2 autocorr\_man4

autocorrelation for x if x is a vector, or individually 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

### 98.3 autocorr\_man5

autocorrellation of the columns of X

### 98.4 blockserr

estimate the standard error of potentially sequentially 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

### 98.5 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  $x^4 c_{ii}^2$ , the square seems to be missing

mu : nx1 mean vector

C : nxn covariance matrix

k : nx1 powers of variables in moments

## 98.6 corr\_man

correlation of two vectors

## 98.7 cov\_man

covariance matrix of two vectors

## 98.8 dof

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

## 98.9 edgeworth\_quantile

inverse edgeworth expansion

c.f. cornis fisher 1937

c.f. Rao 2010

c.f. 2.50 in hall

CHERNOZHUKOV 3.3

## 98.10 effective\_sample\_size

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

### 98.11 `f_correlation`

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

### 98.12 `f_finite`

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

### 98.13 `lmean`

mean of  $x.^l$ , not of  $\text{abs}$

### 98.14 `lmoment`

$l$ -moment of vector  $x$

### 98.15 `maskmean`

mean of the masked values of  $X$

### 98.16 `masknanmean`

### 98.17 `mean1`

mean of  $x$

### 98.18 `mean_man`

mean and standard error of  $X$

### 98.19 mse

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

### 98.20 nanautocorr\_man1

autocorrelation of a vector with nan-values

### 98.21 nanautocorr\_man2

autocorrelation of a vector with nan-values

### 98.22 nanautocorr\_man4

compute autocorrelation for x if x is a vector, or individually  
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

### 98.23 nancorr

(co)-correlation matrix when samples a NaN

### 98.24 nancumsum

cumulative sum, setting nan values to zero

### 98.25 nanlmean

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

## 98.26 nanr2

coefficient of determination when samples are invalid

## 98.27 nanrms

root mean square value when sample contains nan-values

## 98.28 nanrmse

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

## 98.29 nanserr

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

## 98.30 nanwmean

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

## 98.31 nanwstd

weighed standard deviation

## 98.32 nanwvar

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

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

### 98.33 nanxcorr

### 98.34 pearson

pearson correlation coefficient

### 98.35 pearson\_to\_kendall

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

### 98.36 pool\_samples

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

### 98.37 qmean

trimmed mean

### 98.38 range\_mean

### 98.39 rmse\_

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

### 98.40 serr

standard error of the mean of a set of uncorrelated samples



**98.41**    **serr1**

**98.42**    **test\_qskew**

**98.43**    **test\_qstd\_qskew\_optimal\_p**

**98.44**    **wautocorr**

autocorrelation for  $x$  if  $x$  is a vector, or individually 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

**98.45**    **wcorr**

correlation of two vectors when samples are weighted

**98.46**    **wcov**

covariance of two vectors when samples are weighted

**98.47**    **wdof**

effective degrees of freedom for weighted samples

#### 98.48 wkurt

kurtosis with weighted samples

#### 98.49 wmean

weighted mean

$\min_x \sum w (x - \mu)^2 \Rightarrow \mu = \sum(wx) / \sum(w)$

varargin can be dim

function [mu serr] = wmean(w,x)

#### 98.50 wrms

weighted root mean square

#### 98.51 wserr

weighted root mean square error

#### 98.52 wskew

skewness of a weighted set of samples

function sk = wskew(w,x)

#### 98.53 wstd

weighed standard deviation

#### 98.54 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)  
 $s^2 = \text{sum}(w^2(x - \text{sum}(wx))^2)$   
s2\_mu : error of mean, s2\_mu : sd of prediction

## 99 statistics

### 99.1 nangeomean

### 99.2 nangeostd

geometric standard deviation ignoring nan-values

## 100 statistics/nonparametric-statistics

### 100.1 kernel1d

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

### 100.2 kernel2d

kernel density estimate in two dimensions

## 101 statistics

### 101.1 normmmoment

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

## 101.2 normpdf2

pdf of the bivariate normal distribution

## 102 statistics/order-statistics

### 102.1 hodges\_lehmann\_location

hodges lehman location estimator

Asymptotic rms efficiency of location estimate:

mean:  $1 s/\sqrt{n}$   
hodges lehman:  $\sqrt{\pi/3} * s \sim 1.0233 s/\sqrt{n}$   
median:  $\pi/2 s/\sqrt{n} \sim 1.25 s / \sqrt{n}$

### 102.2 kendall

kendall correlation coefficient

### 102.3 kendall\_to\_pearson

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

c.f. Kruskal, 1958, p. 823

### 102.4 mad2sd

transform median absolute deviation to standard deviation  
for normal distributed values

### 102.5 madcorr

proxy correlation by median absolute deviation

## 102.6 median2\_holder

## 102.7 median\_ci

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

## 102.8 median\_man

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

## 102.9 mediani

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

## 102.10 nanmadcorr

proxy correlation by median absolute deviation

## 102.11 nanwmedian

weighted median, skips nan-values

## 102.12 nanwquantile

weighted quantile, skips nan values

### 102.13 oja\_median

two dimensional oja median

note: the multivariate median is not unique

oja 1983, for extension to multivariate function, see chaudhri

### 102.14 qkurtosis

kurtosis 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

### 102.15 qmoments

moments estimated from quantiles

### 102.16 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

### 102.17 qskewq

skewness estimated by quantiles

## 102.18 qstdq

proxy standard deviation determined by quantiles

## 102.19 quantile1\_optimisation

## 102.20 quantile2\_breckling

qunatile regression

## 102.21 quantile2\_chaudhuri

quantile regression

## 102.22 quantile2\_projected

quantile in two dimensions

## 102.23 quantile2\_projected2

spatial qunatile for chosen direction

## 102.24 quantile\_envelope

## 102.25 quantile\_regression\_simple

simple quantile regression

## 102.26 ranking

ranking for spearman statistics

## 102.27 spatial\_median

c.f. Oja 2008

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

## 102.28 spatial\_quantile

spatial quantile

## 102.29 spatial\_quantile2

spatial quantile

## 102.30 spatial\_quantile3

spatial quantile

## 102.31 spatial\_rank

unsigned rank

## 102.32 spatial\_sign

spatial sign

## 102.33 spatial\_signed\_rank

signed rank

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



### 102.34 spearman

spearman's product moment coefficient

### 102.35 spearman\_rank

### 102.36 spearman\_to\_pearson

conversion of spearman rank to person product moment correlation coefficient

### 102.37 wmedian

weighted median

### 102.38 wquantile

weighted quantile

## 103 statistics

### 103.1 qstd

### 103.2 quantile\_extrap

### 103.3 quantile\_sin

## **104 statistics/random-number-generation**

### **104.1 laplacernd**

random number of laplace distribution

### **104.2 randc**

correlate to correlated standard normally distributed vectors

### **104.3 skewness2param**

### **104.4 skewpdf\_central\_moments**

### **104.5 skewrnd**

random numbers of the skew normal distribution

## **105 statistics**

### **105.1 range**

range and mid range of input

### **105.2 resample\_with\_replacement**

## 106 statistics/resampling-statistics/@Jackknife

### 106.1 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 jackknife, 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 jackknife performs iferior to the d1  
jackknife

### 106.2 estimated\_STATIC

jackknife 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

### 106.3 matrix1\_STATIC

matrix of estimation for leaving out two samples at a time

### 106.4 matrix2

matrix of estimations for jacknive with two samples left out

## 107 statistics/resampling-statistics

### 107.1 block\_jackknife

### 107.2 jackknife\_moments

moments determined by the jackknife

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

### 107.3 moving\_block\_jackknife

blocked Jackknife for autocorrelated data  
sliding block, statistically more efficient but computationally  
expensive  
note, number of blocks must be sufficiently large  $h \sim \sqrt{n}$ ?  $\ll n$

### 107.4 randblockterr

standard error of sequentially 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

### 107.5 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

## 108 statistics

### 108.1 scale\_quantile\_sd

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

### 108.2 sd\_sample\_quantiles

### 108.3 skew\_generalized\_normal\_fit

### 108.4 skew\_generalized\_normpdf

### 108.5 skewcdf

### 108.6 skewparam\_to\_central\_moments

### 108.7 skewpdf

skew-normal distribution  
c.f. Azzalini 1985

## 108.8 spatialrnd

## 108.9 test\_mean\_generalized\_gampdf

## 108.10 test\_skew\_generalized\_normpdf

## 108.11 trimmed\_mean

trimmed mean

## 108.12 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

## 108.13 ttest\_man

two-sample t-test  
unequal sample size  
equal variance

## 108.14 ttest\_paired

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

### 108.15 uniformnpdf

### 108.16 wgeomean

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

### 108.17 wgeovar

variance of the weighted geometric mean

### 108.18 wharmean

weighted harmonic mean

### 108.19 wharstd

### 108.20 wharvar

### 108.21 wnormpdf

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

## 109 stochastic

### 109.1 brownian\_drift\_hitting\_probability

## 109.2 brownian\_drift\_hitting\_probability2

## 109.3 brownian\_field

```
simulate Fractional Brownian field on unit disk, with Hurst
parameter 'H';
Reference:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## 109.4 brownian\_motion\_1d\_acf

## 109.5 brownian\_motion\_1d\_cov

## 109.6 brownian\_motion\_1d\_fft

## 109.7 brownian\_motion\_1d\_fourier

## 109.8 brownian\_motion\_1d\_interleave

## 109.9 brownian\_motion\_1d\_laplacian

## 109.10 brownian\_motion\_2d\_cov



109.11 brownian\_motion\_2d\_fft

109.12 brownian\_motion\_2d\_fft\_old

109.13 brownian\_motion\_2d\_fourier

109.14 brownian\_motion\_2d\_interleave

109.15 brownian\_motion\_2d\_interleaving

109.16 brownian\_motion\_2d\_kahunen

109.17 brownian\_motion\_2d\_laplacian

109.18 brownian\_motion\_with\_drift\_hitting\_probability

## 110 mathematics

mathematical functions of various kind

110.1 ternary\_diagram

## 111 test/fourier

111.1 test\_fourier\_freq2ind

## 112 test/master

112.1 dat\_test\_lanczos\_3d\_k\_20\_n\_40

112.2 poisson2d\_blk

112.3 qr\_implicit\_givens\_2

112.4 spectral\_derivative\_2d

112.5 test\_2d\_eigensolver\_hydrogen

112.6 test\_2d\_refine

112.7 test\_3d\_eigensolver\_hydrogen

112.8 test\_FEM

112.9 test\_Mesh\_3d

112.10 test\_arnoldi

112.11 test\_arpackc

112.12 test\_assemble

112.13 test\_assembly\_performance

112.14 test\_bc\_one\_sided

112.15 test\_compare\_solvers

112.16 test\_complete

112.17 test\_convergence

112.18 test\_convergence\_b

112.19 test\_df\_2d

112.20 test\_eig\_algs

112.21 test\_eig\_inverse

112.22 test\_eigs\_lanczos

112.23 test\_eigs\_lanczos\_1

112.24 test\_eigs\_lanczos\_2

112.25 test\_eigs\_lanczos\_performance

112.26 test\_fdm

112.27 test\_fdm\_d\_vargrid

112.28 test\_fdm\_spectral

112.29 test\_fem

112.30 test\_fem\_1d

112.31 test\_fem\_1d\_higher\_order

112.32 test\_fem\_2d\_adaptive

112.33 test\_fem\_2d\_higher\_order

112.34 test\_fem\_3d\_higher\_order

112.35 test\_fem\_3d\_refine

112.36 test\_fem\_b

112.37 test\_fem\_derivative

112.38 test\_fem\_quadrature

112.39 test\_final

112.40 test\_fix\_substitution

112.41 test\_forward

112.42 test\_get\_sparse\_arrays

112.43 test\_harmonic\_oscillator

112.44 test\_high\_order\_fdm\_periodic\_bc

112.45 test\_hydrogen\_wf

112.46 test\_ichol

112.47 test\_interpolation

112.48 test\_inverse\_problem

112.49 test\_it\_vs\_exact

112.50 test\_jama

112.51 test\_jd

112.52 test\_jdqz

112.53 test\_lanczos\_2

112.54 test\_lanczos\_biorthogonal

112.55 test\_laplacian

112.56 test\_laplacian\_non\_uniform

112.57 test\_laplacian\_simple

112.58 test\_mesh\_2d\_uniform

112.59 test\_mesh\_2d\_uniform\_2

112.60 test\_mesh\_circle

112.61 test\_mesh\_generation

112.62 test\_mesh\_interpolate

112.63 test\_mg

112.64 test\_minres\_recycle

112.65 test\_multigrid

112.66 test\_nc

112.67 test\_nonuniform\_symmetric

112.68 test\_pde



112.69 test\_permutation

112.70 test\_poisson\_fem

112.71 test\_polar

112.72 test\_potential

112.73 test\_powers

112.74 test\_precondition

112.75 test\_project\_rectangle

112.76 test\_qr

112.77 test\_quantum\_well

112.78 test\_radial\_adaptive

112.79 test\_radial\_confinement

112.80 test\_radial\_fixes

112.81 test\_refine\_2d

112.82 test\_refine\_2d\_b

112.83 test\_refine\_3d

112.84 test\_refine\_structural

112.85 test\_regularisation

112.86 test\_round\_off

112.87 test\_schrödinger\_potentials

112.88 test\_uniform\_mesh

112.89 test\_vargrid

## 113 test/signal-processing/autocorrelation

113.1 test\_acf

113.2 test\_acf\_bias

113.3 test\_acf\_brownian\_phase

113.4 test\_acfar1\_2

113.5 test\_acfar1\_3

113.6 test\_acfar1\_4

113.7 test\_acfar2

113.8 test\_ar1\_var\_factor

113.9 test\_ar1\_var\_factor\_2

113.10 test\_ar1\_var\_mu\_single\_sample

113.11 test\_ar1\_var\_pop

113.12 test\_ar1\_var\_pop\_1

113.13 test\_ar1delay

113.14 test\_ar2

114 test/signal-processing/passes

114.1 test\_bandpass2d\_ideal

114.2 test\_lowpass1d\_fft

114.3 test\_lowpass1d\_implicit

114.4 test\_lowpass2d\_anisotropic

114.5 test\_lowpass2d\_fft

114.6 test\_lowpass2d\_rho

115 test/signal-processing/periodogram

115.1 test\_periodicity\_test\_2d

115.2 test\_periodogram\_bartlett\_se

115.3 test\_periodogram\_gauss

115.4 test\_periodogram\_radial

115.5 test\_periodogram\_test

115.6 test\_periodogram\_test\_periodicity\_2d

115.7 test\_periogram\_significance

116 test/signal-processing/spectral-density

116.1 test\_spectral\_density\_2

116.2 test\_spectral\_density\_bandpass\_2d

116.3 test\_spectral\_density\_bandpass\_2d\_max2par

116.4 test\_spectral\_density\_bandpass\_continuous

```
title(sprintf('n %d L %g %g%%', [n,L,1e2*rmse(idx,jdx)]));
```

116.5 test\_spectral\_density\_bandpass\_continuous\_1

116.6 test\_spectral\_density\_bandpass\_maximum

116.7 test\_spectral\_density\_bandpass\_scale

116.8 test\_spectral\_density\_bp

116.9    `test_spectral_density_bp_2d`

116.10   `test_spectral_density_bp_approx`

116.11   `test_spectral_density_brownian_phase`

116.12   `test_spectral_density_brownian_phase_2d`

116.13   `test_spectral_density_brownian_phase_across`

116.14   `test_spectral_density_brownian_phase_across_mode2par`

116.15   `test_spectral_density_brownian_phase_mode`

116.16   `test_spectral_density_brownian_phase_mode2par`

116.17   `test_spectral_density_brownian_phase_scale`

116.18   `test_spectral_density_flat`

116.19 test\_spectral\_density\_hp\_cos

116.20 test\_spectral\_density\_lorentzian\_max

116.21 test\_spectral\_density\_lorentzian\_scale

116.22 test\_spectral\_density\_lowpass

116.23 test\_spectral\_density\_lowpass\_continuous

116.24 test\_spectral\_density\_lowpass\_continuous\_1

116.25 test\_spectral\_density\_maxiumum\_bias\_corrected

117 test/signal-processing

117.1 test\_autocorrelation\_max

117.2 test\_cdf\_bandpass\_continuous



117.3 test\_cdf\_brownian\_phase

117.4 test\_fit\_spectral\_density

118 test/spatial-pattern-analysis

118.1 test\_approximate\_ratio\_distribution

118.2 test\_approximate\_ratio\_quantile

118.3 test\_separate\_isotropic\_density

119 test/statistics/distributions

119.1 test\_normpdf\_wrapped

120 test/statistics/moment-statistics

120.1 test\_wmean

121 test/statistics

121.1 test\_fisher\_moment2param

121.2 test\_gamma\_mode

122 test/stochastics

122.1 test\_brownian\_surface

123 test

123.1 test\_S

123.2 test\_advect\_analytic

123.3 test\_asymbp

123.4 test\_bandpass2d

123.5 test\_bandwidth

123.6 test\_bartlett\_angle

123.7 test\_bartlett\_distribution

123.8 test\_bartlett\_expansion

123.9 test\_beta

123.10 test\_betainc

123.11 test\_bivariate\_covariance\_term

123.12 test\_brownian\_drift\_hitting\_probability

123.13 test\_brownian\_drift\_hitting\_probability2

123.14 test\_brownian\_motion\_1d

123.15 test\_brownian\_motion\_2d\_cov

123.16 test\_brownian\_motion\_2d\_fft

123.17 test\_brownian\_noise\_1d

123.18 test\_brownian\_noise\_2d

123.19 test\_brownian\_noise\_interleave

123.20 test\_coherence

123.21 test\_combined\_spectral\_density

123.22 test\_continuous\_fourier\_transform

123.23 test\_convexity

123.24 test\_d2

123.25 test\_determine\_phase\_shift

123.26 test\_diffuse\_analytic

123.27 test\_diffusion\_matrix

123.28 test\_ellipse

123.29 test\_error\_propagation\_fraction

123.30 test\_f

123.31 test\_f2

123.32 test\_fit\_2d\_spectral\_density

123.33 test\_fourier

123.34 test\_fourier\_derivative

123.35 test\_fourier\_derivative\_1

123.36 test\_fourier\_integral

123.37 test\_fourier\_mask\_covariance\_matrix

123.38 test\_ft\_bp

123.39 test\_gam

123.40 test\_gamma\_distribution

123.41 test\_gampdf\_man

123.42 test\_gaussfit3

123.43 test\_gaussian\_flat

123.44 test\_geoserr

123.45 test\_hexagonal\_pattern

123.46 test\_iafrate

123.47 test\_implicit\_ode

123.48 test\_imrotmat

123.49 test\_integration

123.50 test\_ivp

123.51 test\_jacobian

123.52 test\_lanczoswin

123.53 test\_laplacian\_power

123.54 test\_lognfit\_quantile

123.55 test\_ls\_perpendicular\_offset

123.56 test\_madcorr

123.57 test\_mask

123.58 test\_max\_normal

123.59 test\_moments

123.60 test\_moments\_fourier\_power

123.61 test\_mtimes3x3

123.62 test\_noisy\_oscillator

123.63 test\_nonperiodic\_pattern

123.64 test\_normaliztation

123.65 test\_ols

123.66 test\_parcorr

123.67 test\_positivity\_preserving



123.68 test\_randar1

123.69 test\_randar1\_multivariate

123.70 test\_randar2

123.71 test\_ratio\_distributions

123.72 test\_sd\_rectwin

123.73 test\_spatialrnd

123.74 test\_spectrum\_additivity

123.75 test\_stationarity

123.76 test\_stationarity2

123.77 test\_sum\_ij

123.78 test\_sum\_multivar

123.79 test\_trifilt1

123.80 test\_wautocorr

123.81 test\_wavelet\_transform

123.82 test\_whittle

123.83 test\_window

123.84 test\_wordfilt

123.85 test\_xar1\_mid\_term

## 124 mathematics

mathematical functions of various kind

124.1 trapezoidal\_fixed

## 125 wavelet

### 125.1 continuous\_wavelet\_transform

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

### 125.2 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

### 125.3 cwt\_man2

### 125.4 example\_wavelets

### 125.5 phasewrap

wrap the phase to +/- pi

### 125.6 test\_cwt\_man

### 125.7 test\_phasewrap

### 125.8 test\_wavelet

**125.9 test\_wavelet2**

**125.10 test\_wavelet\_analysis**

**125.11 test\_wavelet\_reconstruct**

**125.12 test\_wtc**

**125.13 wavelet**

wavelet windows

**125.14 wavelet\_reconstruct**

iverses wavelet transform for single frequency  
(reconstruction of time series)  
n : window lengths in multiples of filter period  $1/f_0$

**125.15 wavelet\_transform**

wavelet transform for single frequency  
n : window lengths in multiples of filter period  $1/f_0$