

# Manual for Package: mathematics

## Revision 33M

Karl Kästner

July 25, 2023

## Contents

<b>1</b>	<b>calendar</b>	<b>48</b>
1.1	days_per_month . . . . .	48
1.2	isnight . . . . .	48
<b>2</b>	<b>mathematics</b>	<b>48</b>
2.1	cast_byte_to_integer . . . . .	48
<b>3</b>	<b>complex-analysis</b>	<b>48</b>
3.1	complex_exp_product_im_im . . . . .	49
3.2	complex_exp_product_im_re . . . . .	49
3.3	complex_exp_product_re_im . . . . .	49
3.4	complex_exp_product_re_re . . . . .	50
3.5	croots . . . . .	50
3.6	root_complex . . . . .	50
3.7	test_imroots . . . . .	50
<b>4</b>	<b>derivation</b>	<b>50</b>
4.1	derive_acfar1 . . . . .	51
4.2	derive_ar1_spectral_density . . . . .	51
4.3	derive_ar2param . . . . .	51
4.4	derive_arc_length . . . . .	51
4.5	derive_fourier_power . . . . .	51
4.6	derive_fourier_power_exp . . . . .	51
4.7	derive_laplacian_curvilinear . . . . .	51
4.8	derive_laplacian_fourier_piecewise_linear . . . . .	51
4.9	derive_logtripdf . . . . .	51
4.10	derive_phase_drift_inv . . . . .	51
4.11	derive_smooth1d_parametric . . . . .	52
4.12	derive_spectral_density_bandpass_initial_condition . . . . .	52

<b>5</b>	<b>derivation/master</b>	<b>52</b>
5.1	derive_bc_one_sided . . . . .	52
5.2	derive_convergence . . . . .	52
5.3	derive_error_fdm . . . . .	52
5.4	derive_fdm_poly . . . . .	52
5.5	derive_fdm_power . . . . .	52
5.6	derive_fdm_taylor . . . . .	52
5.7	derive_fdm_vargrid . . . . .	52
5.8	derive_fem_2d_mass . . . . .	53
5.9	derive_fem_error_2d . . . . .	53
5.10	derive_fem_error_3d . . . . .	53
5.11	derive_fem_sym_2d . . . . .	53
5.12	derive_grid_constants . . . . .	53
5.13	derive_interpolation . . . . .	53
5.14	derive_laplacian . . . . .	53
5.15	derive_limit . . . . .	53
5.16	derive_nc_1d . . . . .	53
5.17	derive_nc_1d_ . . . . .	53
5.18	derive_nc_2d . . . . .	54
5.19	derive_nonuniform_symmetric . . . . .	54
5.20	derive_richardson . . . . .	54
5.21	derive_sum . . . . .	54
5.22	nn . . . . .	54
5.23	test_derive . . . . .	54
5.24	test_derive_fdm_poly . . . . .	54
5.25	test_filter . . . . .	54
5.26	test_vargrid . . . . .	54
<b>6</b>	<b>derivation</b>	<b>55</b>
6.1	simplify_atan . . . . .	55
<b>7</b>	<b>mathematics</b>	<b>55</b>
7.1	entropy . . . . .	55
<b>8</b>	<b>finance</b>	<b>55</b>
8.1	derive_skewrnd_walsh_paramter . . . . .	55
8.2	gbb_geostd_entire_series . . . . .	55
8.3	gbb_mean . . . . .	55
8.4	gbb_simulate . . . . .	55
8.5	gbb_std . . . . .	55
8.6	gbm_bridge . . . . .	56
8.7	gbm_cdf . . . . .	56
8.8	gbm_fit . . . . .	56
8.9	gbm_fit_old . . . . .	56

8.10	gbm_geomean . . . . .	56
8.11	gbm_geostd . . . . .	56
8.12	gbm_inv . . . . .	56
8.13	gbm_mean . . . . .	56
8.14	gbm_mean_entire_series . . . . .	56
8.15	gbm_median . . . . .	56
8.16	gbm_moment2par . . . . .	57
8.17	gbm_moment2par_entire_series . . . . .	57
8.18	gbm_pdf . . . . .	57
8.19	gbm_simulate . . . . .	57
8.20	gbm_skewness . . . . .	57
8.21	gbm_std . . . . .	57
8.22	gbm_std_entire_series . . . . .	57
8.23	gbm_transform_time_step . . . . .	57
8.24	put_price_black_scholes . . . . .	57
8.25	skewgbm_simulate . . . . .	57
8.26	skewrnd_walsh . . . . .	58
<b>9</b>	<b>finance/test</b>	<b>58</b>
9.1	test_gbm . . . . .	58
9.2	test_gbm_pdf . . . . .	58
9.3	test_skewrnd_walsh . . . . .	58
<b>10</b>	<b>fourier/@STFT</b>	<b>58</b>
10.1	STFT . . . . .	58
10.2	itransform . . . . .	58
10.3	stft_ . . . . .	58
10.4	stftmat . . . . .	59
10.5	transform . . . . .	59
<b>11</b>	<b>fourier</b>	<b>59</b>
11.1	amplitude_from_peak . . . . .	59
11.2	caesaro_weight . . . . .	59
11.3	dftmtx_man . . . . .	59
11.4	example_fourier_window . . . . .	60
11.5	fft2_cartesian2radial . . . . .	60
11.6	fft_man . . . . .	60
11.7	fft_rotate . . . . .	60
11.8	fftsmooth . . . . .	60
11.9	fix_fourier . . . . .	61
11.10	fourier_2d_padd . . . . .	61
11.11	fourier_2d_quadrants . . . . .	61
11.12	fourier_axis . . . . .	61
11.13	fourier_axis_2d . . . . .	62

11.14	fourier_cesaro_correction . . . . .	62
11.15	fourier_coefficient_piecewise_linear . . . . .	62
11.16	fourier_coefficient_piecewise_linear_1 . . . . .	62
11.17	fourier_coefficient_ramp3 . . . . .	62
11.18	fourier_coefficient_ramp_pulse . . . . .	63
11.19	fourier_coefficient_ramp_step . . . . .	63
11.20	fourier_coefficient_square_pulse . . . . .	63
11.21	fourier_complete_negative_half_plane . . . . .	63
11.22	fourier_cubic_interaction_coefficients . . . . .	63
11.23	fourier_derivative . . . . .	63
11.24	fourier_derivative_matrix_1d . . . . .	64
11.25	fourier_derivative_matrix_2d . . . . .	64
11.26	fourier_expand . . . . .	64
11.27	fourier_fit . . . . .	64
11.28	fourier_freq2ind . . . . .	64
11.29	fourier_interpolate . . . . .	64
11.30	fourier_matrix . . . . .	64
11.31	fourier_matrix2 . . . . .	64
11.32	fourier_matrix3 . . . . .	65
11.33	fourier_matrix_exp . . . . .	65
11.34	fourier_multiplicative_interaction_coefficients . . . . .	65
11.35	fourier_power . . . . .	65
11.36	fourier_power_exp . . . . .	65
11.37	fourier_predict . . . . .	66
11.38	fourier_quadratic_interaction_coefficients . . . . .	66
11.39	fourier_random_phase_walk . . . . .	66
11.40	fourier_range . . . . .	66
11.41	fourier_regress . . . . .	66
11.42	fourier_resampled_fit . . . . .	66
11.43	fourier_resampled_predict . . . . .	66
11.44	fourier_series_signed_square . . . . .	67
11.45	fourier_transform . . . . .	67
11.46	fourier_transform_fractional . . . . .	67
11.47	fourier_truncate_negative_half_plane . . . . .	67
11.48	hyperbolic_fourier_box . . . . .	67
11.49	idftmtx_man . . . . .	68
11.50	laplace_2d_pwlinear . . . . .	68
11.51	mean_fourier_power . . . . .	68
11.52	moments_fourier_power . . . . .	68
11.53	nanfft . . . . .	68
11.54	peaks . . . . .	68
11.55	roots_fourier . . . . .	69
11.56	spectral_density . . . . .	69
11.57	std_fourier_power . . . . .	69

11.58	test_complex_exp_product . . . . .	69
11.59	test_fourier_filter . . . . .	69
11.60	test_idftmtx . . . . .	70
11.61	var_fourier_power . . . . .	70
<b>12</b>	<b>mathematics</b>	<b>70</b>
12.1	gaussfit_quantile . . . . .	70
<b>13</b>	<b>geometry/@Geometry</b>	<b>70</b>
13.1	Geometry . . . . .	70
13.2	arclength . . . . .	70
13.3	arclength_old . . . . .	70
13.4	arclength_old2 . . . . .	71
13.5	base_point . . . . .	71
13.6	base_point_limited . . . . .	71
13.7	centroid . . . . .	71
13.8	cosa_min_max . . . . .	71
13.9	cross2 . . . . .	71
13.10	curvature . . . . .	71
13.11	ddot . . . . .	71
13.12	distance . . . . .	72
13.13	distance2 . . . . .	72
13.14	dot . . . . .	72
13.15	edge_length . . . . .	72
13.16	enclosed_angle . . . . .	72
13.17	enclosing_triangle . . . . .	72
13.18	hexagon . . . . .	72
13.19	inPolygon . . . . .	73
13.20	inTetra . . . . .	73
13.21	inTetra2 . . . . .	73
13.22	inTriangle . . . . .	73
13.23	intersect . . . . .	73
13.24	lineintersect . . . . .	73
13.25	lineintersect1 . . . . .	73
13.26	minimum_distance_lines . . . . .	73
13.27	mittenpunkt . . . . .	74
13.28	nagelpoint . . . . .	74
13.29	onLine . . . . .	74
13.30	orthocentre . . . . .	74
13.31	plumb_line . . . . .	74
13.32	poly_area . . . . .	74
13.33	poly_edges . . . . .	74
13.34	poly_set . . . . .	74
13.35	poly_width . . . . .	75

13.36	polyxpoly . . . . .	75
13.37	project_to_curve . . . . .	75
13.38	quad_isconvex . . . . .	75
13.39	random_disk . . . . .	75
13.40	random_simplex . . . . .	75
13.41	sphere_volume . . . . .	75
13.42	tetra_volume . . . . .	76
13.43	tobarycentric . . . . .	76
13.44	tobarycentric1 . . . . .	76
13.45	tobarycentric2 . . . . .	76
13.46	tobarycentric3 . . . . .	76
13.47	tri_angle . . . . .	76
13.48	tri_area . . . . .	76
13.49	tri_centroid . . . . .	76
13.50	tri_distance_opposit_midpoint . . . . .	77
13.51	tri_edge_length . . . . .	77
13.52	tri_edge_midpoint . . . . .	77
13.53	tri_excircle . . . . .	77
13.54	tri_height . . . . .	77
13.55	tri_incircle . . . . .	77
13.56	tri_isacute . . . . .	77
13.57	tri_isobtuse . . . . .	77
13.58	tri_semiperimeter . . . . .	78
13.59	tri_side_length . . . . .	78
<b>14 geometry</b>		<b>78</b>
14.1	Polygon . . . . .	78
14.2	bounding_box . . . . .	78
14.3	curvature_1d . . . . .	78
14.4	cvt . . . . .	78
14.5	deg_to_frac . . . . .	79
14.6	ellipse . . . . .	79
14.7	ellipseX . . . . .	79
14.8	ellipseY . . . . .	79
14.9	first_intersect . . . . .	79
14.10	golden_ratio . . . . .	79
14.11	hypot3 . . . . .	79
14.12	meanangle . . . . .	79
14.13	meanangle2 . . . . .	80
14.14	meanangle3 . . . . .	80
14.15	meanangle4 . . . . .	80
14.16	medianangle . . . . .	80
14.17	medianangle2 . . . . .	80
14.18	pilim . . . . .	80

14.19	streamline_radius_of_curvature . . . . .	81
<b>15</b>	<b>histogram/@Histogram</b>	<b>81</b>
15.1	2x . . . . .	81
15.2	Histogram . . . . .	81
15.3	bimodes . . . . .	81
15.4	cdf . . . . .	81
15.5	cdfS . . . . .	81
15.6	chi2test . . . . .	81
15.7	cmoment . . . . .	81
15.8	cmomentS . . . . .	81
15.9	entropy . . . . .	82
15.10	entropyS . . . . .	82
15.11	export_csv . . . . .	82
15.12	iquantile . . . . .	82
15.13	kstest . . . . .	82
15.14	kurtosis . . . . .	82
15.15	kurtosisS . . . . .	82
15.16	mean . . . . .	82
15.17	meanS . . . . .	82
15.18	median . . . . .	82
15.19	medianS . . . . .	83
15.20	mode . . . . .	83
15.21	modeS . . . . .	83
15.22	moment . . . . .	83
15.23	momentS . . . . .	83
15.24	pdf . . . . .	83
15.25	quantile . . . . .	83
15.26	quantileS . . . . .	83
15.27	resample . . . . .	83
15.28	setup . . . . .	83
15.29	skewness . . . . .	84
15.30	skewnessS . . . . .	84
15.31	stairs . . . . .	84
15.32	stairsS . . . . .	84
15.33	std . . . . .	84
15.34	stdS . . . . .	84
15.35	var . . . . .	84
15.36	varS . . . . .	84
<b>16</b>	<b>histogram</b>	<b>84</b>
16.1	hist_man . . . . .	84
16.2	histadapt . . . . .	85
16.3	histconst . . . . .	85

16.4	pdf_poly . . . . .	85
16.5	plotcdf . . . . .	85
16.6	test_histogram . . . . .	85
<b>17</b>	<b>mathematics</b>	<b>85</b>
17.1	imrotmat . . . . .	85
<b>18</b>	<b>linear-algebra</b>	<b>85</b>
18.1	averaging_matrix_2 . . . . .	85
18.2	colnorm . . . . .	85
18.3	condest_ . . . . .	86
18.4	connectivity_matrix . . . . .	86
<b>19</b>	<b>linear-algebra/coordinate-transformation</b>	<b>86</b>
19.1	barycentric2cartesian . . . . .	86
19.2	barycentric2cartesian3 . . . . .	86
19.3	cartesian2barycentric . . . . .	86
19.4	cartesian_to_unit_triangle_basis . . . . .	86
19.5	ellipsoid2geoid . . . . .	86
19.6	example_approximate_utm_conversion . . . . .	86
19.7	latlon2utm . . . . .	87
19.8	latlon2utm_simple . . . . .	87
19.9	lowrance_mercator_to_wgs84 . . . . .	87
19.10	nmea2utm . . . . .	87
19.11	sn2xy . . . . .	87
19.12	unit_triangle_to_cartesian . . . . .	87
19.13	utm2latlon . . . . .	87
19.14	xy2nt . . . . .	88
19.15	xy2sn . . . . .	88
19.16	xy2sn_java . . . . .	88
19.17	xy2sn_old . . . . .	88
<b>20</b>	<b>linear-algebra</b>	<b>88</b>
20.1	deflation_matrix . . . . .	88
20.2	det2x2 . . . . .	88
20.3	det3x3 . . . . .	89
20.4	det4x4 . . . . .	89
20.5	diag2x2 . . . . .	89
20.6	down . . . . .	89
20.7	eig2x2 . . . . .	89
<b>21</b>	<b>linear-algebra/eigenvalue</b>	<b>89</b>
21.1	eig_bisection . . . . .	89
21.2	eig_inverse . . . . .	89



21.3	eig_inverse_iteration . . . . .	89
21.4	eig_power_iteration . . . . .	90
<b>22</b>	<b>linear-algebra/eigenvalue/jacobi-davidson</b>	<b>90</b>
22.1	afun_jdm . . . . .	90
22.2	davidson . . . . .	90
22.3	jacobi_davidson . . . . .	90
22.4	jacobi_davidson_qr . . . . .	90
22.5	jacobi_davidson_qz . . . . .	90
22.6	jacobi_davidson_simple . . . . .	90
22.7	jdqr . . . . .	90
22.8	jdqr_sleijpen . . . . .	94
22.9	jdqr_vorst . . . . .	97
22.10	jdqz . . . . .	100
22.11	mfunc_jdm . . . . .	105
22.12	mgs . . . . .	105
22.13	minres_ . . . . .	105
22.14	mv_jacobi_davidson . . . . .	105
<b>23</b>	<b>linear-algebra</b>	<b>105</b>
23.1	first . . . . .	105
23.2	gershgorin_circle . . . . .	105
23.3	haussdorff . . . . .	105
23.4	ieig2x2 . . . . .	105
23.5	inv2x2 . . . . .	106
23.6	inv3x3 . . . . .	106
23.7	inv4x4 . . . . .	106
23.8	kernel2matrix . . . . .	106
<b>24</b>	<b>linear-algebra/lanczos</b>	<b>106</b>
24.1	arnoldi . . . . .	106
24.2	arnoldi_new . . . . .	106
24.3	eigs_lanczos_man . . . . .	106
24.4	lanczos . . . . .	106
24.5	lanczos_ . . . . .	106
24.6	lanczos.biorthogonal . . . . .	107
24.7	lanczos.biorthogonal_improved . . . . .	107
24.8	lanczos_ghep . . . . .	107
24.9	mv_lanczos . . . . .	107
24.10	reorthogonalise . . . . .	107
24.11	test_lanczos . . . . .	107
<b>25</b>	<b>linear-algebra</b>	<b>107</b>
25.1	laplacian_eigenvalue . . . . .	107

25.2	laplacian_eigenvector . . . . .	107
25.3	laplacian_power . . . . .	107
25.4	least_squares_perpendicular_offset . . . . .	108
25.5	left . . . . .	108
<b>26</b>	<b>linear-algebra/linear-systems</b>	<b>108</b>
26.1	gmres_man . . . . .	108
26.2	minres_recycle . . . . .	108
<b>27</b>	<b>linear-algebra</b>	<b>108</b>
27.1	lpmean . . . . .	108
27.2	lpnorm . . . . .	108
27.3	matvec3 . . . . .	108
27.4	max2d . . . . .	108
27.5	mid . . . . .	109
27.6	mpoweri . . . . .	109
27.7	mtimes2x2 . . . . .	109
27.8	mtimes3x3 . . . . .	109
27.9	nannorm . . . . .	109
27.10	nanshift . . . . .	109
27.11	nl . . . . .	109
27.12	normalise . . . . .	110
27.13	normalize1 . . . . .	110
27.14	normrows . . . . .	110
27.15	orth2 . . . . .	110
27.16	orth_man . . . . .	110
27.17	orthogonalise . . . . .	110
27.18	padd2 . . . . .	110
27.19	paddext . . . . .	111
27.20	paddval1 . . . . .	111
27.21	paddval2 . . . . .	111
<b>28</b>	<b>linear-algebra/polynomial</b>	<b>111</b>
28.1	chebychev . . . . .	111
28.2	piecewise_polynomial . . . . .	111
28.3	roots1 . . . . .	111
28.4	roots2 . . . . .	111
28.5	roots2poly . . . . .	112
28.6	roots3 . . . . .	112
28.7	roots4 . . . . .	112
28.8	roots_piecewise_linear . . . . .	112
28.9	test_roots4 . . . . .	112
28.10	vanderi_1d . . . . .	112

<b>29</b>	<b>linear-algebra</b>	<b>112</b>
29.1	randrot . . . . .	112
29.2	right . . . . .	112
29.3	rot2 . . . . .	113
29.4	rot2dir . . . . .	113
29.5	rot3 . . . . .	113
29.6	rotR . . . . .	113
29.7	rownorm . . . . .	113
29.8	simmilarity_matrix . . . . .	113
29.9	spnorm . . . . .	113
29.10	spzeros . . . . .	113
29.11	test_roots3 . . . . .	113
29.12	transform_minmax . . . . .	114
29.13	transpose3 . . . . .	114
29.14	transposeall . . . . .	114
29.15	up . . . . .	114
29.16	vander_nd . . . . .	114
29.17	vanderd_2d . . . . .	114
<b>30</b>	<b>logic</b>	<b>114</b>
30.1	bitor_man . . . . .	114
<b>31</b>	<b>master/plot</b>	<b>115</b>
31.1	attach_boundary_value . . . . .	115
31.2	cartesian_polar . . . . .	115
31.3	img_vargrid . . . . .	115
31.4	plot_basis_functions . . . . .	115
31.5	plot_convergence . . . . .	115
31.6	plot_dof . . . . .	115
31.7	plot_eigenbar . . . . .	115
31.8	plot_error_estimation . . . . .	115
31.9	plot_error_estimation_2 . . . . .	115
31.10	plot_error_fem . . . . .	116
31.11	plot_fdm_kernel . . . . .	116
31.12	plot_fdm_vs_fem . . . . .	116
31.13	plot_fem_accuracy . . . . .	116
31.14	plot_function_and_grid . . . . .	116
31.15	plot_hat . . . . .	116
31.16	plot_hydrogen_wf . . . . .	116
31.17	plot_mesh . . . . .	116
31.18	plot_mesh_2 . . . . .	116
31.19	plot_refine . . . . .	116
31.20	plot_refine_3d . . . . .	117
31.21	plot_runtime . . . . .	117

31.22	plot_spectrum . . . . .	117
31.23	plot_wavefunction . . . . .	117
<b>32</b>	<b>master/ported</b>	<b>117</b>
32.1	assemble_2d_phi_phi . . . . .	117
32.2	assemble_3d_dphi_dphi . . . . .	117
32.3	assemble_3d_phi_phi . . . . .	117
32.4	dV_2d_ . . . . .	117
32.5	derivative_2d . . . . .	117
32.6	derivative_3d . . . . .	118
32.7	element_neighbour_2d . . . . .	118
32.8	prefetch_2d_ . . . . .	118
32.9	promote_2d_3_10 . . . . .	118
32.10	promote_2d_3_15 . . . . .	118
32.11	promote_2d_3_21 . . . . .	118
32.12	promote_2d_3_6 . . . . .	118
32.13	promote_3d_4_10 . . . . .	118
32.14	promote_3d_4_20 . . . . .	118
32.15	promote_3d_4_35 . . . . .	118
32.16	vander_2d . . . . .	119
32.17	vander_3d . . . . .	119
<b>33</b>	<b>mathematics</b>	<b>119</b>
33.1	monotoneous_indices . . . . .	119
33.2	nearest_fractional_timestep . . . . .	119
<b>34</b>	<b>number-theory</b>	<b>119</b>
34.1	ceiln . . . . .	119
34.2	digitsb . . . . .	119
34.3	floorn . . . . .	119
34.4	iseven . . . . .	120
34.5	multichoosek . . . . .	120
34.6	nchoosek_man . . . . .	120
34.7	pythagorean_triple . . . . .	120
34.8	roundn . . . . .	120
<b>35</b>	<b>numerical-methods</b>	<b>120</b>
35.1	advect_analytic . . . . .	120
35.2	advection_kernel . . . . .	120
<b>36</b>	<b>numerical-methods/differentiation</b>	<b>121</b>
36.1	derivative1 . . . . .	121
36.2	derivative2 . . . . .	121
<b>37</b>	<b>numerical-methods/finite-difference</b>	<b>121</b>

37.1	cdiff . . . . .	121
37.2	cdiffb . . . . .	121
37.3	central_difference . . . . .	121
37.4	cmean . . . . .	121
37.5	cmean2 . . . . .	121
37.6	derivative_matrix_1_1d . . . . .	122
37.7	derivative_matrix_2_1d . . . . .	122
37.8	derivative_matrix_2d . . . . .	122
37.9	derivative_matrix_curvilinear . . . . .	122
37.10	derivative_matrix_curvilinear_2 . . . . .	122
37.11	difference_kernel . . . . .	122
37.12	diffusion_matrix_2d_anisotropic . . . . .	122
37.13	diffusion_matrix_2d_anisotropic2 . . . . .	123
37.14	directional_neighbour . . . . .	123
37.15	distmat . . . . .	123
37.16	downwind_difference . . . . .	123
37.17	gradpde2d . . . . .	123
37.18	laplacian . . . . .	123
37.19	laplacian_fdm . . . . .	123
37.20	lrmean . . . . .	123
<b>38 numerical-methods/finite-difference/master</b>		<b>124</b>
38.1	fdm_adaptive_grid . . . . .	124
38.2	fdm_adaptive_refinement_old . . . . .	124
38.3	fdm_assemble_d1_2d . . . . .	124
38.4	fdm_assemble_d2_2d . . . . .	124
38.5	fdm_confinement . . . . .	124
38.6	fdm_d_vargrid . . . . .	124
38.7	fdm_h_unstructured . . . . .	124
38.8	fdm_hydrogen_vargrid . . . . .	124
38.9	fdm_mark_unstructured_2d . . . . .	124
38.10	fdm_plot . . . . .	125
38.11	fdm_plot_series . . . . .	125
38.12	fdm_refine_2d . . . . .	125
38.13	fdm_refine_3d . . . . .	125
38.14	fdm_refine_unstructured_2d . . . . .	125
38.15	fdm_schroedinger_2d . . . . .	125
38.16	fdm_schroedinger_3d . . . . .	125
38.17	relocate . . . . .	125
<b>39 numerical-methods/finite-difference</b>		<b>125</b>
39.1	mid . . . . .	125
39.2	pwmid . . . . .	126
39.3	ratio . . . . .	126

39.4	steplength . . . . .	126
39.5	swapoddeven . . . . .	126
39.6	test_derivative_matrix_2d . . . . .	126
39.7	test_derivative_matrix_curvilinear . . . . .	126
39.8	test_difference_kernel . . . . .	126
39.9	upwind_difference . . . . .	126
<b>40</b>	<b>numerical-methods/finite-element</b>	<b>127</b>
40.1	Mesh_2d.java . . . . .	127
40.2	Tree_2d.java . . . . .	127
40.3	assemble_1d_dphi_dphi . . . . .	127
40.4	assemble_1d_phi_phi . . . . .	127
40.5	assemble_2d_dphi_dphi.java . . . . .	127
40.6	assemble_2d_phi_phi.java . . . . .	127
40.7	assemble_3d_dphi_dphi.java . . . . .	127
40.8	assemble_3d_phi_phi.java . . . . .	127
40.9	boundary_1d . . . . .	127
40.10	boundary_2d . . . . .	128
40.11	boundary_3d . . . . .	128
40.12	check_area_2d . . . . .	128
40.13	circmesh . . . . .	128
40.14	cropradius . . . . .	128
40.15	display_2d . . . . .	128
40.16	display_3d . . . . .	128
40.17	distort . . . . .	128
40.18	err_2d . . . . .	128
40.19	estimate_err_2d_3 . . . . .	128
40.20	example_1d . . . . .	129
40.21	example_2d . . . . .	129
40.22	explode . . . . .	129
40.23	fem_2d . . . . .	129
40.24	fem_2d_heuristic_mesh . . . . .	129
40.25	fem_get_2d_radial . . . . .	129
40.26	fem_interpolation . . . . .	129
40.27	fem_plot_1d . . . . .	129
40.28	fem_plot_1d_series . . . . .	129
40.29	fem_plot_2d . . . . .	129
40.30	fem_plot_2d_series . . . . .	130
40.31	fem_plot_3d . . . . .	130
40.32	fem_plot_3d_series . . . . .	130
40.33	fem_plot_confine_series . . . . .	130
40.34	fem_radial . . . . .	130
40.35	flip_2d . . . . .	130
40.36	get_mesh_arrays . . . . .	130

40.37	hashkey . . . . .	130
<b>41</b>	<b>numerical-methods/finite-element/int</b>	<b>130</b>
41.1	int_1d.equal . . . . .	130
41.2	int_1d.equal.exp . . . . .	131
41.3	int_1d.gauss . . . . .	131
41.4	int_1d.gauss_1 . . . . .	131
41.5	int_1d.gauss_2 . . . . .	131
41.6	int_1d.gauss_3 . . . . .	131
41.7	int_1d.gauss_4 . . . . .	131
41.8	int_1d.gauss_5 . . . . .	131
41.9	int_1d.gauss_6 . . . . .	131
41.10	int_1d.gauss.lobatto . . . . .	131
41.11	int_1d.gauss.n . . . . .	132
41.12	int_1d.nc_2 . . . . .	132
41.13	int_1d.nc_3 . . . . .	132
41.14	int_1d.nc_4 . . . . .	132
41.15	int_1d.nc_5 . . . . .	132
41.16	int_1d.nc_6 . . . . .	132
41.17	int_1d.nc_7 . . . . .	132
41.18	int_1d.nc_7.hardy . . . . .	132
41.19	int_2d.gauss_1 . . . . .	132
41.20	int_2d.gauss_12 . . . . .	132
41.21	int_2d.gauss_13 . . . . .	133
41.22	int_2d.gauss_16 . . . . .	133
41.23	int_2d.gauss_19 . . . . .	133
41.24	int_2d.gauss_25 . . . . .	133
41.25	int_2d.gauss_3 . . . . .	133
41.26	int_2d.gauss_33 . . . . .	133
41.27	int_2d.gauss_4 . . . . .	133
41.28	int_2d.gauss_6 . . . . .	133
41.29	int_2d.gauss_7 . . . . .	133
41.30	int_2d.gauss_9 . . . . .	133
41.31	int_2d.nc_10 . . . . .	134
41.32	int_2d.nc_15 . . . . .	134
41.33	int_2d.nc_21 . . . . .	134
41.34	int_2d.nc_3 . . . . .	134
41.35	int_2d.nc_6 . . . . .	134
41.36	int_3d.gauss_1 . . . . .	134
41.37	int_3d.gauss_11 . . . . .	134
41.38	int_3d.gauss_14 . . . . .	134
41.39	int_3d.gauss_15 . . . . .	134
41.40	int_3d.gauss_24 . . . . .	134
41.41	int_3d.gauss_4 . . . . .	135

41.42	int_3d_gauss_45 . . . . .	135
41.43	int_3d_gauss_5 . . . . .	135
41.44	int_3d_nc_11 . . . . .	135
41.45	int_3d_nc_4 . . . . .	135
41.46	int_3d_nc_6 . . . . .	135
41.47	int_3d_nc_8 . . . . .	135
<b>42</b>	<b>numerical-methods/finite-element</b>	<b>135</b>
42.1	interpolation_matrix . . . . .	135
42.2	mark . . . . .	135
42.3	mark_1d . . . . .	136
42.4	mesh_1d_uniform . . . . .	136
42.5	mesh_3d_uniform . . . . .	136
42.6	mesh_interpolate . . . . .	136
42.7	neighbour_1d . . . . .	136
42.8	old . . . . .	136
42.9	pdeeig_1d . . . . .	136
42.10	pdeeig_2d . . . . .	136
42.11	pdeeig_3d . . . . .	136
42.12	polynomial_derivative_1d . . . . .	136
42.13	potential_const . . . . .	137
42.14	potential_coulomb . . . . .	137
42.15	potential_harmonic_oscillator . . . . .	137
42.16	project_circle . . . . .	137
42.17	project_rectangle . . . . .	137
42.18	promote_1d_2_3 . . . . .	137
42.19	promote_1d_2_4 . . . . .	137
42.20	promote_1d_2_5 . . . . .	137
42.21	promote_1d_2_6 . . . . .	137
42.22	quadrilaterate . . . . .	137
42.23	recalculate_regularity_2d . . . . .	138
42.24	refine_1d . . . . .	138
42.25	refine_2d_21 . . . . .	138
42.26	refine_2d_structural . . . . .	138
42.27	regularity_1d . . . . .	138
42.28	regularity_2d . . . . .	138
42.29	regularity_3d . . . . .	138
42.30	relocate_2d . . . . .	138
42.31	test_circmesh . . . . .	138
42.32	test_hermite . . . . .	139
42.33	tri_assign_points . . . . .	139
42.34	triangulation_uniform . . . . .	139
42.35	vander_1d . . . . .	139
42.36	vanderd_1d . . . . .	139



42.37	vanderi_1d . . . . .	139
<b>43</b>	<b>numerical-methods/finite-volume/@Advection</b>	<b>139</b>
43.1	Advection . . . . .	139
43.2	dot_advection . . . . .	139
<b>44</b>	<b>numerical-methods/finite-volume/@Burgers</b>	<b>140</b>
44.1	burgers_split . . . . .	140
44.2	dot_burgers_fdm . . . . .	140
44.3	dot_burgers_fft . . . . .	140
<b>45</b>	<b>numerical-methods/finite-volume/@Finite_Volume</b>	<b>140</b>
45.1	Finite_Volume . . . . .	140
45.2	apply_bc . . . . .	140
45.3	solve . . . . .	140
45.4	step_split_strang . . . . .	141
45.5	step_unsplit . . . . .	141
<b>46</b>	<b>numerical-methods/finite-volume/@Flux_Limiter</b>	<b>141</b>
46.1	Flux_Limiter . . . . .	141
46.2	beam_warming . . . . .	141
46.3	fromm . . . . .	141
46.4	lax_wendroff . . . . .	142
46.5	minmod . . . . .	142
46.6	monotized_central . . . . .	142
46.7	muscl . . . . .	142
46.8	superbee . . . . .	142
46.9	upwind . . . . .	142
46.10	vanLeer . . . . .	142
<b>47</b>	<b>numerical-methods/finite-volume/@KDV</b>	<b>143</b>
47.1	dot_kdv_fdm . . . . .	143
47.2	dot_kdv_fft . . . . .	143
47.3	kdv_split . . . . .	143
<b>48</b>	<b>numerical-methods/finite-volume/@Reconstruct_Average_Evolve</b>	<b>143</b>
48.1	Reconstruct_Average_Evolve . . . . .	143
48.2	advect_highres . . . . .	143
48.3	advect_lowress . . . . .	144
<b>49</b>	<b>numerical-methods/finite-volume</b>	<b>144</b>
49.1	Godunov . . . . .	144
49.2	Lax_Friedrich . . . . .	144
49.3	Measure . . . . .	144
49.4	Roe . . . . .	144

49.5	fv_swe . . . . .	144
49.6	staggered_euler . . . . .	145
49.7	staggered_grid . . . . .	145
<b>50</b>	<b>numerical-methods</b>	<b>145</b>
50.1	grid2quad . . . . .	145
<b>51</b>	<b>numerical-methods/integration</b>	<b>145</b>
51.1	cumintL . . . . .	145
51.2	cumintR . . . . .	145
51.3	cumint_trapezoidal . . . . .	145
51.4	int_1d_gauss_laguerre . . . . .	145
51.5	int_trapezoidal . . . . .	146
<b>52</b>	<b>numerical-methods/interpolation/@Kriging</b>	<b>146</b>
52.1	Kriging . . . . .	146
52.2	estimate_semivariance . . . . .	146
52.3	interpolate_ . . . . .	146
<b>53</b>	<b>numerical-methods/interpolation/@RegularizedInterpolator1</b>	<b>146</b>
53.1	RegularizedInterpolator1 . . . . .	146
53.2	init . . . . .	147
<b>54</b>	<b>numerical-methods/interpolation/@RegularizedInterpolator2</b>	<b>147</b>
54.1	RegularizedInterpolator2 . . . . .	147
54.2	init . . . . .	147
<b>55</b>	<b>numerical-methods/interpolation/@RegularizedInterpolator3</b>	<b>147</b>
55.1	RegularizedInterpolator3 . . . . .	147
55.2	init . . . . .	147
<b>56</b>	<b>numerical-methods/interpolation</b>	<b>147</b>
56.1	IDW . . . . .	147
56.2	IPoly . . . . .	147
56.3	IRBM . . . . .	148
56.4	ISparse . . . . .	148
56.5	Inn . . . . .	148
56.6	Interpolator . . . . .	148
56.7	fixnan . . . . .	148
56.8	idw1 . . . . .	148
56.9	idw2 . . . . .	148
56.10	inner2outer . . . . .	149
56.11	inner2outer2 . . . . .	149
56.12	interp1_circular . . . . .	149
56.13	interp1_limited . . . . .	149

56.14	interp1_man . . . . .	149
56.15	interp1_piecewise_linear . . . . .	149
56.16	interp1_save . . . . .	149
56.17	interp1_slope . . . . .	150
56.18	interp1_smooth . . . . .	150
56.19	interp1_unique . . . . .	150
56.20	interp2_man . . . . .	150
56.21	interp_angle . . . . .	150
56.22	interp_fourier . . . . .	150
56.23	interp_fourier_batch . . . . .	150
56.24	interp_sn . . . . .	151
56.25	interp_sn2 . . . . .	151
56.26	interp_sn3 . . . . .	151
56.27	interp_sn_ . . . . .	151
56.28	limit_by_distance_1d . . . . .	151
56.29	resample1 . . . . .	151
56.30	resample_d_min . . . . .	151
56.31	resample_vector . . . . .	152
56.32	test_interp1_limited . . . . .	152
<b>57</b>	<b>numerical-methods</b>	<b>152</b>
57.1	inverse_complex . . . . .	152
57.2	maccormack_step . . . . .	152
57.3	minmod . . . . .	152
<b>58</b>	<b>numerical-methods/multigrid</b>	<b>152</b>
58.1	mg_interpolate . . . . .	152
58.2	mg_restrict . . . . .	152
<b>59</b>	<b>numerical-methods/ode/@BVPS_Characteristic</b>	<b>152</b>
59.1	BVPS_Characteristic . . . . .	152
59.2	assemble1_A . . . . .	153
59.3	assemble1_A_Q . . . . .	153
59.4	assemble2_A . . . . .	153
59.5	assemble_AA . . . . .	153
59.6	assemble_AAA . . . . .	153
59.7	assemble_Ic . . . . .	153
59.8	bvp1c . . . . .	153
59.9	check_arguments . . . . .	153
59.10	couple_junctions . . . . .	154
59.11	derivative . . . . .	154
59.12	init . . . . .	154
59.13	inner2outer_bvp2c . . . . .	154
59.14	reconstruct . . . . .	154

59.15	resample . . . . .	154
59.16	solve . . . . .	154
59.17	test_assemble1_A . . . . .	155
59.18	test_assemble2_A . . . . .	155
<b>60</b>	<b>numerical-methods/ode/@Time_Stepper</b>	<b>155</b>
60.1	Time_Stepper . . . . .	155
60.2	solve . . . . .	155
<b>61</b>	<b>numerical-methods/ode</b>	<b>155</b>
61.1	bvp2fdm . . . . .	155
61.2	bvp2wavetrain . . . . .	156
61.3	bvp2wavetwopass . . . . .	156
61.4	ivp_euler_forward . . . . .	156
61.5	ivp_euler_forward2 . . . . .	156
61.6	ivprk2 . . . . .	156
61.7	ode2_matrix . . . . .	156
61.8	ode2characteristic . . . . .	156
61.9	step_trapezoidal . . . . .	157
61.10	test_bvp2 . . . . .	157
<b>62</b>	<b>numerical-methods/optimisation</b>	<b>157</b>
62.1	aitken_iteration . . . . .	157
62.2	anderson_iteration . . . . .	157
62.3	armijo_stopping_criterion . . . . .	157
62.4	astar . . . . .	157
62.5	binsearch . . . . .	157
62.6	bisection . . . . .	157
62.7	box1 . . . . .	158
62.8	box2 . . . . .	158
62.9	cauchy . . . . .	158
62.10	cauchy2 . . . . .	158
62.11	directional_derivative . . . . .	158
62.12	dud . . . . .	158
62.13	extreme3 . . . . .	158
62.14	extreme_quadratic . . . . .	159
62.15	ftest . . . . .	159
62.16	fzero_bisect . . . . .	159
62.17	fzero_newton . . . . .	159
62.18	grad . . . . .	159
62.19	hessian . . . . .	159
62.20	hessian_from_gradient . . . . .	160
62.21	hessian_projected . . . . .	160
62.22	line_search . . . . .	160

62.23	line_search2	160
62.24	line_search_polynomial	160
62.25	line_search_polynomial2	161
62.26	line_search_quadratic	161
62.27	line_search_quadratic2	161
62.28	line_search_wolfe	161
62.29	ls_bgfs	161
62.30	ls_broyden	162
62.31	ls_generalized_secant	162
62.32	nlcg	162
62.33	nlls	162
62.34	picard	162
62.35	poly_extrema	163
62.36	quadratic_function	163
62.37	quadratic_programming	163
62.38	quadratic_step	163
62.39	rosenbrock	163
62.40	sqr_heron	163
62.41	test_directional_derivative	163
62.42	test_dud	163
62.43	test_fzero_newton	163
62.44	test_line_search_quadratic2	164
62.45	test_ls_generalized_secant	164
62.46	test_nlcg_6_order	164
62.47	test_nlls	164
<b>63</b>	<b>numerical-methods/pde</b>	<b>164</b>
63.1	laplacian2d_fundamental_solution	164
<b>64</b>	<b>numerical-methods/piecewise-polynomials</b>	<b>164</b>
64.1	Hermite1	164
64.2	hp2_fit	164
64.3	hp2_predict	165
64.4	hp_predict	165
64.5	hp_regress	165
64.6	lp_count	165
64.7	lp_predict	165
64.8	lp_regress	165
64.9	lp_regress_	165
<b>65</b>	<b>numerical-methods</b>	<b>165</b>
65.1	step_advect_euler_explicit	165
65.2	step_advection_diffusion_euler_implicit	166
65.3	step_advection_diffusion_trapezoidal	166

65.4	step_diffuse_analytic . . . . .	166
65.5	step_diffuse_euler_explicit . . . . .	166
65.6	step_diffuse_euler_implicit . . . . .	166
65.7	step_diffuse_spectral . . . . .	166
65.8	step_diffuse_trapezoidal . . . . .	166
65.9	step_react_euler_explicit . . . . .	166
65.10	step_react_euler_implicit . . . . .	166
65.11	step_react_midpoint . . . . .	167
65.12	step_react_ralston . . . . .	167
65.13	step_react_ralston_exp . . . . .	167
65.14	step_react_ralston_exp_2 . . . . .	167
65.15	step_react_semi_analytic . . . . .	167
65.16	step_react_trapezoidal . . . . .	167
65.17	test_adams_bashforth . . . . .	167
<b>66</b>	<b>mathematics</b>	<b>167</b>
66.1	oversampleNZ . . . . .	167
<b>67</b>	<b>pdes</b>	<b>168</b>
67.1	heat_equation_fundamental_solution . . . . .	168
67.2	heat_equation_fundamental_std_to_time . . . . .	168
67.3	heat_equation_std . . . . .	168
67.4	heat_equation_width . . . . .	168
67.5	heat_equation_width_to_time . . . . .	168
<b>68</b>	<b>regression/@PolyOLS</b>	<b>168</b>
68.1	PolyOLS . . . . .	168
68.2	coefftest . . . . .	168
68.3	detrend . . . . .	168
68.4	fit . . . . .	169
68.5	fit_ . . . . .	169
68.6	predict . . . . .	169
68.7	predict_ . . . . .	169
68.8	slope . . . . .	169
<b>69</b>	<b>regression/@PowerLS</b>	<b>169</b>
69.1	PowerLS . . . . .	169
69.2	fit . . . . .	169
69.3	predict . . . . .	170
69.4	predict_ . . . . .	170
<b>70</b>	<b>regression/@Theil</b>	<b>170</b>
70.1	Theil . . . . .	170
70.2	detrend . . . . .	170

70.3	fit . . . . .	170
70.4	predict . . . . .	170
70.5	slope . . . . .	171
<b>71</b>	<b>regression</b>	<b>171</b>
71.1	Theil_Multivariate . . . . .	171
71.2	areg . . . . .	171
71.3	ginireg . . . . .	171
71.4	hesssimplereg . . . . .	171
71.5	lllin . . . . .	171
71.6	lsq_sparam . . . . .	172
71.7	polyfitd . . . . .	172
71.8	regression_method_of_moments . . . . .	172
71.9	robustlinreg . . . . .	172
71.10	theil2 . . . . .	172
71.11	theil_generalised . . . . .	172
71.12	total_least_squares . . . . .	173
71.13	weighted_median_regression . . . . .	173
<b>72</b>	<b>set-theory</b>	<b>173</b>
72.1	issubset . . . . .	173
<b>73</b>	<b>mathematics</b>	<b>173</b>
73.1	shuffle_index . . . . .	173
<b>74</b>	<b>signal-processing</b>	<b>173</b>
74.1	asymwin . . . . .	173
<b>75</b>	<b>signal-processing/autocorrelation</b>	<b>174</b>
75.1	acf_radial . . . . .	174
75.2	acfar1 . . . . .	174
75.3	acfar1_2 . . . . .	174
75.4	acfar2 . . . . .	174
75.5	acfar2_2 . . . . .	174
75.6	ar1_cutoff_frequency . . . . .	174
75.7	ar1_effective_sample_size . . . . .	174
75.8	ar1_mse_mu_single_sample . . . . .	175
75.9	ar1_mse_pop . . . . .	175
75.10	ar1_mse_range . . . . .	175
75.11	ar1_spectrum . . . . .	175
75.12	ar1_to_tikhonov . . . . .	175
75.13	ar1_var_factor . . . . .	175
75.14	ar1_var_factor_ . . . . .	175
75.15	ar1_var_range2 . . . . .	176

75.16	ar1delay . . . . .	176
75.17	ar1delay_old . . . . .	176
75.18	ar2_acf2c . . . . .	176
75.19	ar2conv . . . . .	176
75.20	ar2dof . . . . .	176
75.21	ar2param . . . . .	177
75.22	autocorr2 . . . . .	177
75.23	autocorr_angular . . . . .	177
75.24	autocorr_bandpass . . . . .	177
75.25	autocorr_decay_rate . . . . .	177
75.26	autocorr_effective_sample_size . . . . .	177
75.27	autocorr_fft . . . . .	177
75.28	autocorr_forest . . . . .	177
75.29	autocorr_genton . . . . .	178
75.30	autocorr_highpass . . . . .	178
75.31	autocorr_lowpass . . . . .	178
75.32	autocorr_periodic_additive_noise . . . . .	178
75.33	autocorr_periodic_windowed . . . . .	178
75.34	autocorr_radial . . . . .	178
75.35	autocorr_radial_hexagonal_pattern . . . . .	178
75.36	autocorrelation_max . . . . .	178
<b>76 signal-processing</b>		<b>178</b>
76.1	average_wave_shape . . . . .	178
76.2	bandpass . . . . .	179
76.3	bandpass_continuous_cdf . . . . .	179
76.4	bartlett . . . . .	179
76.5	bin1d . . . . .	179
76.6	bin2d . . . . .	179
76.7	binormrnd . . . . .	179
76.8	coherence . . . . .	179
76.9	conv1_man . . . . .	180
76.10	conv2_man . . . . .	180
76.11	conv2z . . . . .	180
76.12	conv30 . . . . .	180
76.13	conv_ . . . . .	180
76.14	conv_centered . . . . .	180
76.15	convz . . . . .	180
76.16	cosexpdelay . . . . .	180
76.17	csmooth . . . . .	181
76.18	daniell_window . . . . .	181
76.19	db2neper . . . . .	181
76.20	db2power . . . . .	181
76.21	derive_bandpass_continuous_scale . . . . .	181



76.22	derive_danielle_weight . . . . .	181
76.23	derive_limit_0_acfar . . . . .	181
76.24	detect_peak . . . . .	181
76.25	determine_phase_shift . . . . .	182
76.26	determine_phase_shift1 . . . . .	182
76.27	doublesum_ij . . . . .	182
76.28	effective_mask_size . . . . .	182
76.29	effective_sample_size_to_ar1 . . . . .	182
76.30	fcut2Lw_gausswin . . . . .	182
76.31	fcut_gausswin . . . . .	182
76.32	filt_hodges_lehman . . . . .	182
<b>77</b>	<b>signal-processing/filters</b>	<b>182</b>
77.1	circfilt2 . . . . .	182
77.2	filter1 . . . . .	183
77.3	filter2 . . . . .	183
77.4	filter_ . . . . .	183
77.5	filter_r_to_f0 . . . . .	183
77.6	filter_rho_to_f0 . . . . .	183
77.7	filter_twosided . . . . .	183
77.8	filteriir . . . . .	183
77.9	filterp . . . . .	184
77.10	filterp1 . . . . .	184
77.11	filterstd . . . . .	184
77.12	gaussfilt2 . . . . .	184
77.13	lowpass_discrete . . . . .	185
77.14	meanfilt2 . . . . .	185
77.15	medfilt1_man . . . . .	185
77.16	medfilt1_man2 . . . . .	185
77.17	medfilt1_padded . . . . .	185
77.18	medfilt1_reduced . . . . .	185
77.19	trifilt1 . . . . .	185
77.20	trifilt2 . . . . .	186
<b>78</b>	<b>signal-processing</b>	<b>186</b>
78.1	firls_man . . . . .	186
78.2	fit_spectral_density . . . . .	186
78.3	fit_spectral_density_2d . . . . .	186
78.4	fit_spectral_density_radial . . . . .	186
78.5	flattopwin . . . . .	186
78.6	frequency_response_boxcar . . . . .	186
78.7	freqz_boxcar . . . . .	186
78.8	gaussfilt1 . . . . .	187
78.9	hanchangewin . . . . .	187

78.10	hanchangewin2 . . . . .	187
78.11	hanwin . . . . .	187
78.12	hanwin_ . . . . .	187
78.13	high_pass_1d_simple . . . . .	187
78.14	kaiserwin . . . . .	187
78.15	kalman . . . . .	187
78.16	lanczoswin . . . . .	188
78.17	last . . . . .	188
78.18	maxfilt1 . . . . .	188
78.19	meanfilt1 . . . . .	188
78.20	mid_term_single_sample . . . . .	188
78.21	minfilt1 . . . . .	188
78.22	minmax . . . . .	188
78.23	mu2ar1 . . . . .	188
78.24	mysmooth . . . . .	189
78.25	nanautocorr . . . . .	189
78.26	nanmedfilt1 . . . . .	189
78.27	neper2db . . . . .	189
78.28	oscillator_noisy . . . . .	189
<b>79 signal-processing/passes</b>		<b>189</b>
79.1	bandpass1d . . . . .	189
79.2	bandpass1d_fft . . . . .	189
79.3	bandpass1d_implicit . . . . .	189
79.4	bandpass2 . . . . .	190
79.5	bandpass2d . . . . .	190
79.6	bandpass2d_convolution . . . . .	190
79.7	bandpass2d_fft . . . . .	190
79.8	bandpass2d_ideal . . . . .	190
79.9	bandpass2d_implicit . . . . .	190
79.10	bandpass2d_iso . . . . .	190
79.11	bandpass_arg . . . . .	190
79.12	bandpass_f0_to_rho . . . . .	190
79.13	bandpass_max . . . . .	191
79.14	bandpass_max2 . . . . .	191
79.15	highpass . . . . .	191
79.16	highpass1d_fft_cos . . . . .	191
79.17	highpass1d_implicit . . . . .	191
79.18	highpass2d_fft . . . . .	191
79.19	highpass2d_ideal . . . . .	191
79.20	highpass2d_implicit . . . . .	191
79.21	highpass_arg . . . . .	191
79.22	highpass_fc_to_rho . . . . .	192
79.23	lowpass . . . . .	192

79.24	lowpass1d_fft . . . . .	192
79.25	lowpass1d_implicit . . . . .	192
79.26	lowpass2 . . . . .	192
79.27	lowpass2d_anisotropic . . . . .	192
79.28	lowpass2d_convolution . . . . .	192
79.29	lowpass2d_fft . . . . .	192
79.30	lowpass2d_ideal . . . . .	193
79.31	lowpass2d_implicit . . . . .	193
79.32	lowpass_arg . . . . .	193
79.33	lowpass_fc_to_rho . . . . .	193
79.34	lowpass_iir . . . . .	193
79.35	lowpass_iir_symmetric . . . . .	193
79.36	lowpassfilter2 . . . . .	193
<b>80</b>	<b>signal-processing</b>	<b>193</b>
80.1	peaks_man . . . . .	193
<b>81</b>	<b>signal-processing/periodogram</b>	<b>194</b>
81.1	periodogram . . . . .	194
81.2	periodogram_2d . . . . .	194
81.3	periodogram_align . . . . .	194
81.4	periodogram_angular . . . . .	194
81.5	periodogram_bartlett . . . . .	194
81.6	periodogram_bootstrap . . . . .	194
81.7	periodogram_confidence_interval . . . . .	194
81.8	periodogram_filter . . . . .	195
81.9	periodogram_median . . . . .	195
81.10	periodogram_normalize . . . . .	195
81.11	periodogram_normalize_2d . . . . .	195
81.12	periodogram_p_value . . . . .	195
81.13	periodogram_qq . . . . .	195
81.14	periodogram_quantiles . . . . .	195
81.15	periodogram_radial . . . . .	195
81.16	periodogram_std . . . . .	196
81.17	periodogram_test_periodicity . . . . .	196
81.18	periodogram_test_periodicity_2d . . . . .	197
81.19	periodogram_test_stationarity . . . . .	198
81.20	periodogram_welsh . . . . .	199
<b>82</b>	<b>signal-processing</b>	<b>199</b>
82.1	polyfilt1 . . . . .	199
82.2	qmedfilt1 . . . . .	199
82.3	quadratfilt1 . . . . .	199
82.4	quadratwin . . . . .	199

82.5	randar1 . . . . .	199
82.6	randar1_dual . . . . .	199
82.7	randar2 . . . . .	199
82.8	randarp . . . . .	200
82.9	rectwin . . . . .	200
82.10	recursive_sum . . . . .	200
82.11	select_range . . . . .	200
82.12	smooth1d_parametric . . . . .	200
82.13	smooth2 . . . . .	200
82.14	smooth_man . . . . .	200
82.15	smooth_parametric . . . . .	200
82.16	smooth_parametric2 . . . . .	201
82.17	smooth_with_splines . . . . .	201
82.18	smoothfft . . . . .	201
<b>83</b>	<b>signal-processing/spectral-density</b>	<b>201</b>
83.1	hex_angular_pdf . . . . .	201
83.2	hex_angular_pdf_max . . . . .	201
83.3	hex_angular_pdf_max2par . . . . .	201
83.4	spectral_density_ar2 . . . . .	201
83.5	spectral_density_area . . . . .	201
83.6	spectral_density_estimate_2d . . . . .	201
83.7	spectral_density_flat . . . . .	202
83.8	spectral_density_forest . . . . .	202
83.9	spectral_density_gausswin . . . . .	202
83.10	spectral_density_lorentzian . . . . .	202
83.11	spectral_density_lorentzian_max . . . . .	202
83.12	spectral_density_lorentzian_max2par . . . . .	202
83.13	spectral_density_lorentzian_scale . . . . .	202
83.14	spectral_density_maximum_bias_corrected . . . . .	202
83.15	spectral_density_periodic_additive_noise . . . . .	202
83.16	spectral_density_rectwin . . . . .	203
83.17	spectral_density_wperiodic . . . . .	203
<b>84</b>	<b>signal-processing</b>	<b>203</b>
84.1	spectrogram . . . . .	203
84.2	sum_i_lag . . . . .	203
84.3	sum_ii . . . . .	203
84.4	sum_ii_ . . . . .	203
84.5	sum_ij . . . . .	203
84.6	sum_ij_ . . . . .	204
84.7	sum_ij_partial_ . . . . .	204
84.8	sum_multivar . . . . .	204
84.9	test_acfar1 . . . . .	204

84.10	tikhonov_to_ar1 . . . . .	204
84.11	trapwin . . . . .	204
84.12	triwin . . . . .	204
84.13	triwin2 . . . . .	204
84.14	tukeywin_man . . . . .	204
84.15	varar1 . . . . .	205
84.16	welch_spectrogram . . . . .	205
84.17	wfilt . . . . .	205
84.18	winbandpass . . . . .	205
<b>85</b>	<b>signal-processing/windows</b>	<b>205</b>
85.1	circwin . . . . .	205
85.2	danielle_window . . . . .	205
85.3	gausswin . . . . .	205
85.4	gausswin1 . . . . .	205
85.5	gausswin2 . . . . .	206
85.6	radial_window . . . . .	206
85.7	range_window . . . . .	206
85.8	rectwin_cutoff_frequency . . . . .	206
85.9	std_window . . . . .	206
85.10	window2d . . . . .	206
85.11	window_make_odd . . . . .	206
<b>86</b>	<b>signal-processing</b>	<b>206</b>
86.1	winfilt0 . . . . .	206
86.2	wlength . . . . .	207
86.3	wmeanfilt . . . . .	207
86.4	wmedfilt . . . . .	207
86.5	wordfilt . . . . .	207
86.6	wordfilt_edgeworth . . . . .	207
86.7	wrapphase . . . . .	207
86.8	xar1 . . . . .	207
86.9	xcorr_man . . . . .	207
<b>87</b>	<b>sorting</b>	<b>208</b>
87.1	sort2 . . . . .	208
87.2	sort2d . . . . .	208
<b>88</b>	<b>spatial-pattern-analysis/@Spatial_Pattern</b>	<b>208</b>
88.1	Spatial_Pattern . . . . .	208
88.2	analyze_grid . . . . .	208
88.3	analyze_transect . . . . .	208
88.4	clear_1d_properties . . . . .	208
88.5	clear_2d_properties . . . . .	208

88.6	fit_parametric_densities . . . . .	209
88.7	imread . . . . .	209
88.8	init . . . . .	209
88.9	plot . . . . .	209
88.10	plot_transect . . . . .	209
88.11	prepare_analysis . . . . .	209
88.12	report . . . . .	209
88.13	resample_functions . . . . .	209
88.14	tabulate . . . . .	210
<b>89</b>	<b>spatial-pattern-analysis/@Spatial_Pattern_Array</b>	<b>210</b>
89.1	Spatial_Pattern_Array . . . . .	210
89.2	analyze . . . . .	210
89.3	assign_regions . . . . .	210
89.4	export_shp . . . . .	210
89.5	fetch . . . . .	210
89.6	generate_filename . . . . .	210
89.7	quality_check . . . . .	210
<b>90</b>	<b>spatial-pattern-analysis</b>	<b>211</b>
90.1	approximate_ratio_distribution . . . . .	211
90.2	banded_pattern . . . . .	211
90.3	hexagonal_pattern . . . . .	211
90.4	patch_size_1d . . . . .	212
90.5	patch_size_2d . . . . .	212
90.6	pattern_isotropic_rotated . . . . .	212
90.7	reconstruct_isotropic_density . . . . .	212
90.8	separate_isotropic_from_anisotropic_density . . . . .	212
90.9	suppress_low_frequency_lobe . . . . .	212
<b>91</b>	<b>spatial-statistics</b>	<b>212</b>
91.1	cov_cell_averages_1d . . . . .	212
91.2	cov_cell_averages_2d . . . . .	213
<b>92</b>	<b>special-functions</b>	<b>213</b>
92.1	bessel_sphere . . . . .	213
92.2	besseliln_large_x . . . . .	213
92.3	beta_man . . . . .	213
92.4	betainc_man . . . . .	214
92.5	digamma_man . . . . .	214
92.6	exp10 . . . . .	214
92.7	hankel_sphere . . . . .	214
92.8	hermite . . . . .	214
92.9	laguerre_roots . . . . .	214

92.10	lambertw_numeric . . . . .	214
92.11	legendre_man . . . . .	215
92.12	neumann_sphere . . . . .	215
<b>93</b>	<b>statistics</b>	<b>215</b>
93.1	atan_s2 . . . . .	215
93.2	binomial . . . . .	215
<b>94</b>	<b>statistics/circular</b>	<b>215</b>
94.1	circular_fmoment . . . . .	215
94.2	circular_fquantile . . . . .	215
94.3	circular_fstd . . . . .	215
94.4	circular_fvar . . . . .	216
<b>95</b>	<b>statistics</b>	<b>216</b>
95.1	coefficient_of_determination . . . . .	216
95.2	conditional_expectation_normal . . . . .	216
95.3	correlation_confidence_pearson . . . . .	216
<b>96</b>	<b>statistics/distributions</b>	<b>216</b>
96.1	PDF . . . . .	216
<b>97</b>	<b>statistics/distributions/anisotropic</b>	<b>216</b>
97.1	anisotropic_pattern . . . . .	216
97.2	anisotropic_pattern_acf . . . . .	216
97.3	anisotropic_pattern_pdf . . . . .	216
<b>98</b>	<b>statistics/distributions/beta</b>	<b>217</b>
98.1	beta_kurt . . . . .	217
98.2	beta_mean . . . . .	217
98.3	beta_moment2par . . . . .	217
98.4	beta_skew . . . . .	217
98.5	beta_std . . . . .	217
<b>99</b>	<b>statistics/distributions/bivariate-normal</b>	<b>217</b>
99.1	binorm_separation_coefficient . . . . .	217
99.2	binormcdf . . . . .	217
99.3	binormfit . . . . .	217
99.4	binormpdf . . . . .	218
<b>100</b>	<b>statistics/distributions/chi2</b>	<b>218</b>
100.1	chi2_kurt . . . . .	218
100.2	chi2_mean . . . . .	218
100.3	chi2_skew . . . . .	218
100.4	chi2_std . . . . .	218

<b>101</b>	<b>statistics/distributions/circular-normal</b>	<b>218</b>
101.1	wnormpdf . . . . .	218
<b>102</b>	<b>statistics/distributions/edgeworth</b>	<b>218</b>
102.1	edgeworth_cdf . . . . .	218
102.2	edgeworth_pdf . . . . .	219
<b>103</b>	<b>statistics/distributions/exp</b>	<b>219</b>
103.1	exppdf_max2par . . . . .	219
<b>104</b>	<b>statistics/distributions/fisher</b>	<b>219</b>
104.1	fisher_mean . . . . .	219
104.2	fisher_moment2par . . . . .	219
104.3	fisher_std . . . . .	219
<b>105</b>	<b>statistics/distributions/gamma</b>	<b>219</b>
105.1	gamma_mean . . . . .	219
105.2	gamma_mode . . . . .	219
105.3	gamma_mode2par . . . . .	220
105.4	gamma_moment2par . . . . .	220
105.5	gamma_std . . . . .	220
105.6	gamma_stirling . . . . .	220
105.7	gampdf_man . . . . .	220
105.8	generalized_gamma_mean . . . . .	220
<b>106</b>	<b>statistics/distributions/hotelling-t2</b>	<b>220</b>
106.1	t2cdf . . . . .	220
106.2	t2inv . . . . .	220
<b>107</b>	<b>statistics/distributions/kurt-normal</b>	<b>221</b>
107.1	kurtncdf . . . . .	221
107.2	kurtnpdf . . . . .	221
<b>108</b>	<b>statistics/distributions/log-triangular</b>	<b>221</b>
108.1	logtrialtcd . . . . .	221
108.2	logtrialtinv . . . . .	221
108.3	logtrialtmean . . . . .	221
108.4	logtrialtpdf . . . . .	221
108.5	logtrialtrnd . . . . .	222
108.6	logtricdf . . . . .	222
108.7	logtriinv . . . . .	222
108.8	logtrimean . . . . .	222
108.9	logtripdf . . . . .	222
108.10	logtrirnd . . . . .	222



<b>109</b>	<b>statistics/distributions/log-uniform</b>	<b>222</b>
109.1	logu_median . . . . .	222
109.2	logucdf . . . . .	222
109.3	logucm . . . . .	223
109.4	loguinv . . . . .	223
109.5	logumean . . . . .	223
109.6	logupdf . . . . .	223
109.7	logurnd . . . . .	223
109.8	loguvar . . . . .	223
<b>110</b>	<b>statistics/distributions/loglog</b>	<b>223</b>
110.1	loglogpdf . . . . .	223
<b>111</b>	<b>statistics/distributions/lognormal</b>	<b>224</b>
111.1	logn_corr . . . . .	224
111.2	logn_cov . . . . .	224
111.3	logn_mean . . . . .	224
111.4	logn_mode . . . . .	224
111.5	logn_mode2par . . . . .	224
111.6	logn_moment2par . . . . .	224
111.7	logn_moment2par_correlated . . . . .	224
111.8	logn_param2moment . . . . .	225
111.9	logn_skewness . . . . .	225
111.10	logn_std . . . . .	225
111.11	lognpdf_ . . . . .	225
111.12	lognpdf_entropy . . . . .	225
<b>112</b>	<b>statistics/distributions/logskew</b>	<b>225</b>
112.1	logskewcdf . . . . .	225
112.2	logskewpdf . . . . .	225
<b>113</b>	<b>statistics/distributions/mises</b>	<b>225</b>
113.1	mises_max2par . . . . .	225
113.2	mises_std . . . . .	226
113.3	mises_var . . . . .	226
113.4	misesn_max2par . . . . .	226
113.5	misesnpdf . . . . .	226
113.6	misespdf . . . . .	226
<b>114</b>	<b>statistics/distributions</b>	<b>226</b>
114.1	ncx2_moment2par . . . . .	226
<b>115</b>	<b>statistics/distributions/normal</b>	<b>226</b>
115.1	normpdf_entropy . . . . .	226
115.2	normpdf_mode . . . . .	226

115.3	normpdf_mode2par . . . . .	226
<b>116</b>	<b>statistics/distributions/passes</b>	<b>227</b>
116.1	bandpass1d_continuous_pdf . . . . .	227
116.2	bandpass1d_continuous_pdf_max . . . . .	227
116.3	bandpass1d_continuous_pdf_max2par . . . . .	227
116.4	bandpass1d_continuous_pdf_scale . . . . .	227
116.5	bandpass1d_discrete_pdf . . . . .	227
116.6	bandpass2d_discrete_pdf . . . . .	228
116.7	bandpass2d_pdf_exact . . . . .	228
116.8	bandpass2d_pdf_hankel . . . . .	228
116.9	bandpass2d_pdf_mode . . . . .	228
116.10	bandpass2d_pdf_mode2par . . . . .	228
116.11	bandpass2d_pdf_scale . . . . .	228
116.12	highpass1d_continuous_pdf . . . . .	228
116.13	highpass1d_discrete_cos_pdf . . . . .	229
116.14	highpass1d_discrete_pdf . . . . .	229
116.15	highpass2d_discrete_pdf . . . . .	229
116.16	highpass2d_pdf . . . . .	229
116.17	highpass2d_pdf_hankel . . . . .	229
116.18	lowpass1d_continuous_pdf . . . . .	230
116.19	lowpass1d_continuous_pdf_scale . . . . .	230
116.20	lowpass1d_discrete_pdf . . . . .	230
116.21	lowpass1d_one_sided_pdf . . . . .	230
116.22	lowpass2d_discrete_acf . . . . .	230
116.23	lowpass2d_discrete_pdf . . . . .	230
116.24	lowpass2d_pdf . . . . .	230
116.25	lowpass2d_pdf_hankel . . . . .	230
116.26	lowpass2d_pdf_series . . . . .	231
<b>117</b>	<b>statistics/distributions</b>	<b>231</b>
117.1	pdfsample . . . . .	231
<b>118</b>	<b>statistics/distributions/phase-drift</b>	<b>231</b>
118.1	phase_drift_acf . . . . .	231
118.2	phase_drift_acf_2d . . . . .	231
118.3	phase_drift_cdf . . . . .	231
118.4	phase_drift_inv . . . . .	231
118.5	phase_drift_parallel_acf . . . . .	231
118.6	phase_drift_parallel_pdf . . . . .	232
118.7	phase_drift_parallel_pdf_max . . . . .	232
118.8	phase_drift_parallel_pdf_max2par . . . . .	232
118.9	phase_drift_parallel_pdf_mode2par . . . . .	232
118.10	phase_drift_patch_size_distribution . . . . .	232

118.11	phase_drift_pdf . . . . .	232
118.12	phase_drift_pdf_2d . . . . .	232
118.13	phase_drift_pdf_mode . . . . .	232
118.14	phase_drift_pdf_mode2par . . . . .	232
118.15	phase_drift_pdf_reg2par . . . . .	233
118.16	phase_drift_pdf_scale . . . . .	233
<b>119</b>	<b>statistics/distributions/skew-normal</b>	<b>233</b>
119.1	skew_generalized_normal_fit . . . . .	233
119.2	skew_generalized_normpdf . . . . .	233
119.3	skewcdf . . . . .	233
119.4	skewparam_to_central_moments . . . . .	233
119.5	skewpdf . . . . .	233
119.6	skewpdf_entropy . . . . .	233
<b>120</b>	<b>statistics/distributions/triangular</b>	<b>234</b>
120.1	tricdf . . . . .	234
120.2	triinv . . . . .	234
120.3	trimediam . . . . .	234
120.4	tripdf . . . . .	234
120.5	trirnd . . . . .	234
<b>121</b>	<b>statistics/distributions/weibull</b>	<b>234</b>
121.1	wbl_std . . . . .	234
<b>122</b>	<b>statistics/distributions/wrapped-normal</b>	<b>234</b>
122.1	normpdf_wrapped . . . . .	234
122.2	normpdf_wrapped_mode . . . . .	235
122.3	normpdf_wrapped_mode2par . . . . .	235
<b>123</b>	<b>statistics</b>	<b>235</b>
123.1	error_propagation_fraction . . . . .	235
123.2	error_propagation_product . . . . .	235
123.3	example_standard_error_of_sample_quantiles . . . . .	235
123.4	f_var_finite . . . . .	235
123.5	gaussfit3 . . . . .	235
123.6	gaussfit_quantile . . . . .	235
123.7	geoserr . . . . .	235
123.8	geostd . . . . .	236
123.9	hodges_lehmann_correlation . . . . .	236
123.10	hodges_lehmann_dispersion . . . . .	236
<b>124</b>	<b>statistics/information-theory</b>	<b>236</b>
124.1	akaike_information_criterion . . . . .	236
124.2	bayesian_information_criterion . . . . .	237

<b>125</b>	<b>statistics</b>	<b>237</b>
125.1	jackknife_block . . . . .	237
125.2	kurtosis_bias_corrected . . . . .	237
125.3	limit . . . . .	237
125.4	logfactorial . . . . .	237
125.5	lognfit_quantile . . . . .	237
125.6	max_exprnd . . . . .	237
125.7	maxnnormals . . . . .	237
125.8	mean_angle . . . . .	238
125.9	mean_max_n . . . . .	238
125.10	mean_min_n . . . . .	238
125.11	midrange . . . . .	238
125.12	minavg . . . . .	238
125.13	mode_man . . . . .	238
<b>126</b>	<b>statistics/moment-statistics</b>	<b>238</b>
126.1	autocorr_man3 . . . . .	238
126.2	autocorr_man4 . . . . .	239
126.3	autocorr_man5 . . . . .	239
126.4	blockserr . . . . .	239
126.5	comoment . . . . .	239
126.6	corr_man . . . . .	239
126.7	cov_man . . . . .	240
126.8	dof . . . . .	240
126.9	edgeworth_quantile . . . . .	240
126.10	effective_sample_size . . . . .	240
126.11	f_correlation . . . . .	240
126.12	f_finite . . . . .	240
126.13	lmean . . . . .	240
126.14	lmoment . . . . .	241
126.15	maskmean . . . . .	241
126.16	masknanmean . . . . .	241
126.17	mean1 . . . . .	241
126.18	mean_man . . . . .	241
126.19	mse . . . . .	241
126.20	nanautocorr_man1 . . . . .	241
126.21	nanautocorr_man2 . . . . .	241
126.22	nanautocorr_man4 . . . . .	242
126.23	nancorr . . . . .	242
126.24	nancumsum . . . . .	242
126.25	nanlmean . . . . .	242
126.26	nanr2 . . . . .	242
126.27	nanrms . . . . .	242
126.28	nanrmse . . . . .	242

126.29	nanserr	243
126.30	nanwmean	243
126.31	nanwstd	243
126.32	nanwvar	243
126.33	nanxcorr	243
126.34	pearson	243
126.35	pearson_to_kendall	243
126.36	pool_samples	244
126.37	qmean	244
126.38	range_mean	244
126.39	rmse_	244
126.40	serr	244
126.41	serr1	244
126.42	test_qskew	244
126.43	test_qstd_qskew_optimal_p	244
126.44	wautocorr	245
126.45	wcorr	245
126.46	wcov	245
126.47	wdof	245
126.48	wkurt	245
126.49	wmean	245
126.50	wrms	246
126.51	wserr	246
126.52	wskew	246
126.53	wstd	246
126.54	wvar	246
<b>127statistics</b>		<b>246</b>
127.1	nangeomean	246
127.2	nangeostd	246
<b>128statistics/nonparametric-statistics</b>		<b>247</b>
128.1	kernel1d	247
128.2	kernel2d	247
<b>129statistics</b>		<b>247</b>
129.1	normalize_exponential_random_variable	247
129.2	normmoment	247
129.3	normpdf2	247
<b>130statistics/order-statistics</b>		<b>247</b>
130.1	hodges_lehmann_location	247
130.2	kendall	248
130.3	kendall_to_pearson	248

130.4	mad2sd . . . . .	248
130.5	madcorr . . . . .	248
130.6	median2_holder . . . . .	248
130.7	median_ci . . . . .	248
130.8	median_man . . . . .	248
130.9	mediani . . . . .	249
130.10	nanmadcorr . . . . .	249
130.11	nanwmedian . . . . .	249
130.12	nanwquantile . . . . .	249
130.13	oja_median . . . . .	249
130.14	qkurtosis . . . . .	249
130.15	qmoments . . . . .	250
130.16	qskew . . . . .	250
130.17	qskewq . . . . .	250
130.18	qstdq . . . . .	250
130.19	quantile1_optimisation . . . . .	250
130.20	quantile2_breckling . . . . .	250
130.21	quantile2_chaudhuri . . . . .	250
130.22	quantile2_projected . . . . .	251
130.23	quantile2_projected2 . . . . .	251
130.24	quantile_envelope . . . . .	251
130.25	quantile_regression_simple . . . . .	251
130.26	ranking . . . . .	251
130.27	spatial_median . . . . .	251
130.28	spatial_quantile . . . . .	251
130.29	spatial_quantile2 . . . . .	251
130.30	spatial_quantile3 . . . . .	252
130.31	spatial_rank . . . . .	252
130.32	spatial_sign . . . . .	252
130.33	spatial_signed_rank . . . . .	252
130.34	spearman . . . . .	252
130.35	spearman_rank . . . . .	252
130.36	spearman_to_pearson . . . . .	252
130.37	wmedian . . . . .	252
130.38	wquantile . . . . .	253
<b>131statistics</b>		<b>253</b>
131.1	qstd . . . . .	253
131.2	quantile_extrap . . . . .	253
131.3	quantile_sin . . . . .	253
<b>132statistics/random-number-generation</b>		<b>253</b>
132.1	laplacernd . . . . .	253
132.2	randc . . . . .	253

132.3	skewness2param . . . . .	253
132.4	skewpdf_central_moments . . . . .	253
132.5	skewrnd . . . . .	254
<b>133</b>	<b>statistics</b>	<b>254</b>
133.1	range . . . . .	254
133.2	resample_with_replacement . . . . .	254
<b>134</b>	<b>statistics/resampling-statistics/@Jackknife</b>	<b>254</b>
134.1	Jackknife . . . . .	254
134.2	estimated_STATIC . . . . .	255
134.3	matrix1_STATIC . . . . .	255
134.4	matrix2 . . . . .	255
<b>135</b>	<b>statistics/resampling-statistics</b>	<b>255</b>
135.1	block_jackknife . . . . .	255
135.2	jackknife_moments . . . . .	255
135.3	moving_block_jackknife . . . . .	255
135.4	randblockserr . . . . .	256
135.5	resample . . . . .	256
<b>136</b>	<b>statistics</b>	<b>256</b>
136.1	scale_quantile_sd . . . . .	256
136.2	sd_sample_quantiles . . . . .	256
136.3	spatialrnd . . . . .	257
136.4	trimmed_mean . . . . .	257
136.5	ttest2_man . . . . .	257
136.6	ttest_man . . . . .	257
136.7	ttest_paired . . . . .	257
136.8	uniformnpdf . . . . .	257
136.9	wgeomean . . . . .	257
136.10	wgeovar . . . . .	258
136.11	wharmean . . . . .	258
136.12	wharstd . . . . .	258
136.13	wharvar . . . . .	258
<b>137</b>	<b>stochastic</b>	<b>258</b>
137.1	brownian_drift_hitting_probability . . . . .	258
137.2	brownian_drift_hitting_probability2 . . . . .	258
137.3	brownian_field . . . . .	258
137.4	brownian_field_scaled . . . . .	259
137.5	brownian_motion_1d_acf . . . . .	259
137.6	brownian_motion_1d_cov . . . . .	259
137.7	brownian_motion_1d_fft . . . . .	259

137.8	brownian_motion_1d_fourier . . . . .	259
137.9	brownian_motion_1d_interleave . . . . .	259
137.10	brownian_motion_1d_laplacian . . . . .	259
137.11	brownian_motion_2d_cov . . . . .	259
137.12	brownian_motion_2d_fft . . . . .	259
137.13	brownian_motion_2d_fft_old . . . . .	260
137.14	brownian_motion_2d_fourier . . . . .	260
137.15	brownian_motion_2d_interleave . . . . .	260
137.16	brownian_motion_2d_interleaving . . . . .	260
137.17	brownian_motion_2d_kahunen . . . . .	260
137.18	brownian_motion_2d_laplacian . . . . .	260
137.19	brownian_motion_with_drift_hitting_probability . . . . .	260
<b>138</b>	<b>stochastic/geometric-ar1</b>	<b>260</b>
138.1	geometric_ar1_2d_generate . . . . .	260
138.2	geometric_ar1_2d_generate_1 . . . . .	260
138.3	geometric_ar1_2d_grid_cell_averaged_cov . . . . .	261
138.4	geometric_ar1_2d_grid_cell_averaged_generate . . . . .	261
138.5	geometric_ar1_2d_grid_cell_averaged_moment2par . . . . .	261
138.6	geometric_ar1_2d_grid_cell_averaged_std . . . . .	261
<b>139</b>	<b>stochastic</b>	<b>261</b>
139.1	ornstein_uhlenbeck_cov . . . . .	261
139.2	ornstein_uhlenbeck_mean . . . . .	261
139.3	ornstein_uhlenbeck_spectral_density . . . . .	261
139.4	ornstein_uhlenbeck_std . . . . .	262
<b>140</b>	<b>mathematics</b>	<b>262</b>
140.1	ternary_diagram . . . . .	262
<b>141</b>	<b>test/finance</b>	<b>262</b>
141.1	test_gbb_mean . . . . .	262
141.2	test_gbb_std . . . . .	262
141.3	test_gbm_mean . . . . .	262
141.4	test_gbm_mean_entire_series . . . . .	262
141.5	test_gbm_moment2par . . . . .	262
141.6	test_gbm_moment2par_entire_series . . . . .	262
141.7	test_gbm_std . . . . .	263
141.8	test_gbm_std_entire_series . . . . .	263
<b>142</b>	<b>test/fourier</b>	<b>263</b>
142.1	test_fourier_freq2ind . . . . .	263
<b>143</b>	<b>test/master</b>	<b>263</b>
143.1	dat_test_lanczos_3d_k_20_n_40 . . . . .	263



143.2	poisson2d.blk . . . . .	263
143.3	qr_implicit_givens_2 . . . . .	263
143.4	spectral_derivative_2d . . . . .	263
143.5	test_2d_eigensolver_hydrogen . . . . .	263
143.6	test_2d_refine . . . . .	263
143.7	test_3d_eigensolver_hydrogen . . . . .	264
143.8	test_FEM . . . . .	264
143.9	test_Mesh_3d . . . . .	264
143.10	test_arnoldi . . . . .	264
143.11	test_arpackc . . . . .	264
143.12	test_assemble . . . . .	264
143.13	test_assembly_performance . . . . .	264
143.14	test_bc_one_sided . . . . .	264
143.15	test_compare_solvers . . . . .	264
143.16	test_complete . . . . .	264
143.17	test_convergence . . . . .	265
143.18	test_convergence_b . . . . .	265
143.19	test_df_2d . . . . .	265
143.20	test_eig_algs . . . . .	265
143.21	test_eig_inverse . . . . .	265
143.22	test_eigs_lanczos . . . . .	265
143.23	test_eigs_lanczos_1 . . . . .	265
143.24	test_eigs_lanczos_2 . . . . .	265
143.25	test_eigs_lanczos_performance . . . . .	265
143.26	test_fdm . . . . .	265
143.27	test_fdm_d_vargrid . . . . .	266
143.28	test_fdm_spectral . . . . .	266
143.29	test_fem . . . . .	266
143.30	test_fem_1d . . . . .	266
143.31	test_fem_1d_higher_order . . . . .	266
143.32	test_fem_2d_adaptive . . . . .	266
143.33	test_fem_2d_higher_order . . . . .	266
143.34	test_fem_3d_higher_order . . . . .	266
143.35	test_fem_3d_refine . . . . .	266
143.36	test_fem_b . . . . .	266
143.37	test_fem_derivative . . . . .	267
143.38	test_fem_quadrature . . . . .	267
143.39	test_final . . . . .	267
143.40	test_fix_substitution . . . . .	267
143.41	test_forward . . . . .	267
143.42	test_get_sparse_arrays . . . . .	267
143.43	test_harmonic_oscillator . . . . .	267
143.44	test_high_order_fdm_periodic_bc . . . . .	267
143.45	test_hydrogen_wf . . . . .	267

143.46	test_ichol . . . . .	267
143.47	test_interpolation . . . . .	268
143.48	test_inverse_problem . . . . .	268
143.49	test_it_vs_exact . . . . .	268
143.50	test_jama . . . . .	268
143.51	test_jd . . . . .	268
143.52	test_jdqz . . . . .	268
143.53	test_lanczos_2 . . . . .	268
143.54	test_lanczos_biorthogonal . . . . .	268
143.55	test_laplacian . . . . .	268
143.56	test_laplacian_non_uniform . . . . .	268
143.57	test_laplacian_simple . . . . .	269
143.58	test_mesh_2d_uniform . . . . .	269
143.59	test_mesh_2d_uniform_2 . . . . .	269
143.60	test_mesh_circle . . . . .	269
143.61	test_mesh_generation . . . . .	269
143.62	test_mesh_interpolate . . . . .	269
143.63	test_mg . . . . .	269
143.64	test_minres_recycle . . . . .	269
143.65	test_multigrid . . . . .	269
143.66	test_nc . . . . .	269
143.67	test_nonuniform_symmetric . . . . .	270
143.68	test_pde . . . . .	270
143.69	test_permutation . . . . .	270
143.70	test_poison_fem . . . . .	270
143.71	test_polar . . . . .	270
143.72	test_potential . . . . .	270
143.73	test_powers . . . . .	270
143.74	test_precondition . . . . .	270
143.75	test_project_rectangle . . . . .	270
143.76	test_qr . . . . .	270
143.77	test_quantum_well . . . . .	271
143.78	test_radial_adaptive . . . . .	271
143.79	test_radial_confinement . . . . .	271
143.80	test_radial_fixes . . . . .	271
143.81	test_refine_2d . . . . .	271
143.82	test_refine_2d_b . . . . .	271
143.83	test_refine_3d . . . . .	271
143.84	test_refine_structural . . . . .	271
143.85	test_regularisation . . . . .	271
143.86	test_round_off . . . . .	271
143.87	test_schrödinger_potentials . . . . .	272
143.88	test_uniform_mesh . . . . .	272
143.89	test_vargrid . . . . .	272

<b>144</b>	<b>test/numerical-methods/optimisation</b>	<b>272</b>
144.1	test_extreme3 . . . . .	272
<b>145</b>	<b>test/numerical-methods</b>	<b>272</b>
145.1	test_advection_kernel . . . . .	272
<b>146</b>	<b>test/signal-processing/autocorrelation</b>	<b>272</b>
146.1	test_acf . . . . .	272
146.2	test_acf_bias . . . . .	272
146.3	test_acfar1_2 . . . . .	272
146.4	test_acfar1_3 . . . . .	273
146.5	test_acfar1_4 . . . . .	273
146.6	test_acfar2 . . . . .	273
146.7	test_ar1_var_factor . . . . .	273
146.8	test_ar1_var_factor_2 . . . . .	273
146.9	test_ar1_var_mu_single_sample . . . . .	273
146.10	test_ar1_var_pop . . . . .	273
146.11	test_ar1_var_pop_1 . . . . .	273
146.12	test_ar1delay . . . . .	273
146.13	test_ar2 . . . . .	273
146.14	test_phase_drift_acf . . . . .	274
<b>147</b>	<b>test/signal-processing/passes</b>	<b>274</b>
147.1	test_bandpass2d . . . . .	274
147.2	test_bandpass2d_ideal . . . . .	274
147.3	test_lowpass1d_fft . . . . .	274
147.4	test_lowpass1d_implicit . . . . .	274
147.5	test_lowpass2d_anisotropic . . . . .	274
147.6	test_lowpass2d_fft . . . . .	274
147.7	test_lowpass2d_rho . . . . .	274
<b>148</b>	<b>test/signal-processing/periodogram</b>	<b>274</b>
148.1	test_periodicity_test_2d . . . . .	274
148.2	test_periodogram_bartlett_se . . . . .	275
148.3	test_periodogram_gauss . . . . .	275
148.4	test_periodogram_radial . . . . .	275
148.5	test_periodogram_test . . . . .	275
148.6	test_periodogram_test_periodicity_2d . . . . .	275
148.7	test_periogogram_significance . . . . .	275
<b>149</b>	<b>test/signal-processing/spectral-density</b>	<b>275</b>
149.1	test_phase_drift_parallel_pdf . . . . .	275
149.2	test_phase_drift_parallel_pdf_mode2par . . . . .	275
149.3	test_phase_drift_pdf . . . . .	275

149.4	test_phase_drift_pdf_2d . . . . .	276
149.5	test_phase_drift_pdf_mode . . . . .	276
149.6	test_phase_drift_pdf_mode2par . . . . .	276
149.7	test_phase_drift_pdf_scale . . . . .	276
149.8	test_spectral_density_2 . . . . .	276
149.9	test_spectral_density_bandpass_2d . . . . .	276
149.10	test_spectral_density_bandpass_2d_max2par . . . . .	276
149.11	test_spectral_density_bandpass_continuous . . . . .	276
149.12	test_spectral_density_bandpass_continuous_1 . . . . .	276
149.13	test_spectral_density_bandpass_maximum . . . . .	276
149.14	test_spectral_density_bandpass_scale . . . . .	277
149.15	test_spectral_density_bp . . . . .	277
149.16	test_spectral_density_bp_2d . . . . .	277
149.17	test_spectral_density_bp_approx . . . . .	277
149.18	test_spectral_density_flat . . . . .	277
149.19	test_spectral_density_hp_cos . . . . .	277
149.20	test_spectral_density_lorentzian_max . . . . .	277
149.21	test_spectral_density_lorentzian_scale . . . . .	277
149.22	test_spectral_density_lowpass . . . . .	277
149.23	test_spectral_density_lowpass_continuous . . . . .	277
149.24	test_spectral_density_lowpass_continuous_1 . . . . .	278
149.25	test_spectral_density_maximum_bias_corrected . . . . .	278
<b>150test/signal-processing</b>		<b>278</b>
150.1	test_autocorrelation_max . . . . .	278
150.2	test_cdf_bandpass_continuous . . . . .	278
150.3	test_fit_spectral_density . . . . .	278
150.4	test_phase_drift_cdf . . . . .	278
<b>151test/spatial-pattern-analysis</b>		<b>278</b>
151.1	test_approximate_ratio_distribution . . . . .	278
151.2	test_approximate_ratio_quantile . . . . .	278
151.3	test_separate_isotropic_density . . . . .	278
<b>152test/spatial-statistics</b>		<b>279</b>
152.1	test_cov_cell_averages_1d . . . . .	279
152.2	test_cov_cell_averages_2d . . . . .	279
<b>153test/statistics/distributions/anisotropic</b>		<b>279</b>
153.1	test_anisotropic_pattern . . . . .	279
153.2	test_anisotropic_pattern_pdf . . . . .	279
<b>154test/statistics/distributions/gamma</b>		<b>279</b>
154.1	test_generalized_gamma_mean . . . . .	279

<b>155</b>	<b>test/statistics/distributions/log-uniform</b>	<b>279</b>
155.1	test_logurnd . . . . .	279
<b>156</b>	<b>test/statistics/distributions/lognormal</b>	<b>279</b>
156.1	test_logn_cov . . . . .	279
<b>157</b>	<b>test/statistics/distributions/mises</b>	<b>280</b>
157.1	test_mises_std . . . . .	280
<b>158</b>	<b>test/statistics/distributions/passes</b>	<b>280</b>
158.1	test_bandpass2d_pdf . . . . .	280
158.2	test_bandpass2d_pdf_hankel . . . . .	280
158.3	test_bandpass2d_pdf_mode . . . . .	280
158.4	test_lowpass2d_pdf_hankel . . . . .	280
158.5	test_lowpass2d_pdf_series . . . . .	280
<b>159</b>	<b>test/statistics/distributions/skew-normal</b>	<b>280</b>
159.1	test_skew_generalized_normpdf . . . . .	280
<b>160</b>	<b>test/statistics/distributions</b>	<b>280</b>
160.1	test_normpdf_wrapped . . . . .	280
<b>161</b>	<b>test/statistics/distributions/weibull</b>	<b>281</b>
161.1	test_wbl_std . . . . .	281
<b>162</b>	<b>test/statistics/moment-statistics</b>	<b>281</b>
162.1	test_wmean . . . . .	281
<b>163</b>	<b>test/statistics</b>	<b>281</b>
163.1	test_fisher_moment2par . . . . .	281
163.2	test_gamma_mode . . . . .	281
163.3	test_normalize_exponential_random_variable . . . . .	281
<b>164</b>	<b>test/stochastic</b>	<b>281</b>
164.1	test_brownian_field . . . . .	281
164.2	test_brownian_field_scaled . . . . .	281
<b>165</b>	<b>test/stochastics</b>	<b>281</b>
165.1	test_brownian_surface . . . . .	281
<b>166</b>	<b>test</b>	<b>282</b>
166.1	test_S . . . . .	282
166.2	test_advect_analytic . . . . .	282
166.3	test_asympb . . . . .	282
166.4	test_bandwidth . . . . .	282
166.5	test_bartlett_angle . . . . .	282

166.6	test_bartlett_distribution . . . . .	282
166.7	test_bartlett_expansion . . . . .	282
166.8	test_beta . . . . .	282
166.9	test_betainc . . . . .	282
166.10	test_bivariate_covariance_term . . . . .	283
166.11	test_brownian_drift_hitting_probability . . . . .	283
166.12	test_brownian_drift_hitting_probability2 . . . . .	283
166.13	test_brownian_motion_1d . . . . .	283
166.14	test_brownian_motion_2d_cov . . . . .	283
166.15	test_brownian_motion_2d_fft . . . . .	283
166.16	test_brownian_noise_1d . . . . .	283
166.17	test_brownian_noise_2d . . . . .	283
166.18	test_brownian_noise_interleave . . . . .	283
166.19	test_coherence . . . . .	283
166.20	test_combined_spectral_density . . . . .	284
166.21	test_continuous_fourier_transform . . . . .	284
166.22	test_convexity . . . . .	284
166.23	test_d2 . . . . .	284
166.24	test_determine_phase_shift . . . . .	284
166.25	test_diffuse_analytic . . . . .	284
166.26	test_diffusion_matrix . . . . .	284
166.27	test_ellipse . . . . .	284
166.28	test_error_propagation_fraction . . . . .	284
166.29	test_f . . . . .	284
166.30	test_f2 . . . . .	285
166.31	test_fit_2d_spectral_density . . . . .	285
166.32	test_fourier . . . . .	285
166.33	test_fourier_derivative . . . . .	285
166.34	test_fourier_derivative_1 . . . . .	285
166.35	test_fourier_integral . . . . .	285
166.36	test_fourier_mask_covariance_matrix . . . . .	285
166.37	test_ft_bp . . . . .	285
166.38	test_gam . . . . .	285
166.39	test_gamma_distribution . . . . .	285
166.40	test_gampdf_man . . . . .	286
166.41	test_gaussfit3 . . . . .	286
166.42	test_gaussian_flat . . . . .	286
166.43	test_geoserr . . . . .	286
166.44	test_hexagonal_pattern . . . . .	286
166.45	test_iafrate . . . . .	286
166.46	test_implicit_ode . . . . .	286
166.47	test_imrotmat . . . . .	286
166.48	test_integration . . . . .	286
166.49	test_ivp . . . . .	286

166.50	test_jacobian . . . . .	287
166.51	test_lanczoswin . . . . .	287
166.52	test_laplacian_power . . . . .	287
166.53	test_lognfit_quantile . . . . .	287
166.54	test_ls_perpendicular_offset . . . . .	287
166.55	test_madcorr . . . . .	287
166.56	test_mask . . . . .	287
166.57	test_max_normal . . . . .	287
166.58	test_moments . . . . .	287
166.59	test_moments_fourier_power . . . . .	287
166.60	test_mtimes3x3 . . . . .	288
166.61	test_noisy_oscillator . . . . .	288
166.62	test_nonperiodic_pattern . . . . .	288
166.63	test_normaliztation . . . . .	288
166.64	test_ols . . . . .	288
166.65	test_parcorr . . . . .	288
166.66	test_positivity_preserving . . . . .	288
166.67	test_randar1 . . . . .	288
166.68	test_randar1_multivariate . . . . .	288
166.69	test_randar2 . . . . .	288
166.70	test_ratio_distributions . . . . .	289
166.71	test_sd_rectwin . . . . .	289
166.72	test_spatialrnd . . . . .	289
166.73	test_spectrum_additivity . . . . .	289
166.74	test_stationarity . . . . .	289
166.75	test_stationarity2 . . . . .	289
166.76	test_sum_ij . . . . .	289
166.77	test_sum_multivar . . . . .	289
166.78	test_trifilt1 . . . . .	289
166.79	test_wautocorr . . . . .	289
166.80	test_wavelet_transform . . . . .	290
166.81	test_whittle . . . . .	290
166.82	test_window . . . . .	290
166.83	test_wordfilt . . . . .	290
166.84	test_xar1_mid_term . . . . .	290
<b>167mathematics</b>		<b>290</b>
167.1	trapezoidal_fixed . . . . .	290
<b>168wavelet</b>		<b>290</b>
168.1	contiuous_wavelet_transform . . . . .	290
168.2	cwt_man . . . . .	291
168.3	cwt_man2 . . . . .	291
168.4	example_wavelets . . . . .	291

168.5	phasewrap . . . . .	291
168.6	test_cwt_man . . . . .	291
168.7	test_phasewrap . . . . .	291
168.8	test_wavelet . . . . .	291
168.9	test_wavelet2 . . . . .	291
168.10	test_wavelet_analysis . . . . .	291
168.11	test_wavelet_reconstruct . . . . .	292
168.12	test_wtc . . . . .	292
168.13	wavelet . . . . .	292
168.14	wavelet_reconstruct . . . . .	292
168.15	wavelet_transform . . . . .	292

## 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

$$\begin{aligned} \text{re}(c1 \exp(i\omega_1 x)) * \text{re}(c2 \exp(i\omega_2 x)) = \\ \frac{1}{2} * ( \text{real}(c1 * c2 * \exp(i * (\omega_1 + \omega_2) * x)) \dots \\ + \text{real}(\text{conj}(c1) * c2 * \exp(i * (\omega_2 - \omega_1) * x)) ) \end{aligned}$$

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_fourier_power`

4.6 `derive_fourier_power_exp`

4.7 `derive_laplacian_curvilinear`

4.8 `derive_laplacian_fourier_piecewise_linear`

4.9 `derive_logtripdf`

4.10 `derive_phase_drift_inv`

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 gbb\_geostd\_entire\_series

### 8.3 gbb\_mean

### 8.4 gbb\_simulate

### 8.5 gbb\_std

8.6 `gbm_bridge`

8.7 `gbm_cdf`

8.8 `gbm_fit`

8.9 `gbm_fit_old`

8.10 `gbm_geomean`

8.11 `gbm_geostd`

8.12 `gbm_inv`

8.13 `gbm_mean`

8.14 `gbm_mean_entire_series`

8.15 `gbm_median`



8.16 `gbm_moment2par`

8.17 `gbm_moment2par_entire_series`

8.18 `gbm_pdf`

8.19 `gbm_simulate`

8.20 `gbm_skewness`

8.21 `gbm_std`

8.22 `gbm_std_entire_series`

8.23 `gbm_transform_time_step`

8.24 `put_price_black_scholes`

8.25 `skewgbm_simulate`

## 8.26 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  $T_i$  should be set to at least  $2 \cdot \max(T)$ , 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`

frequency axis of the 2d fourier transform as computed by Matlab  
function `[fx, fy, fr, ft, Tx, Ty, mask, N] = fourier_axis_2d(L,n)`

### 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-continous 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  
 $t_i$

#### 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  $(2*n+1)$  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

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

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

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

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  
 $\text{range}(y) = \max(y) - \min(y)$

### 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\_series\_signed\_square

coefficients of the Fourier series of  $Q|Q|$

$$Q|Q| = Q_a^2 y \quad (8.5)$$

$$= |\cos a + \cos t| (\cos a + \cos t) \quad (8.6)$$

$$= a_0 + a_1 \cos t + \dots + a_n \cos n t \quad (8.7)$$

$\cos a$  is midrange

$\cos t$  is tidal variation

c.f Dronkers 1964, eq. 8.10

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

euclidian 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

hypothenuse in 3D

## 14.12 meanangle

weighted mean of angles

### 14.13 meanangle2

mean angle

### 14.14 meanangle3

mean angle

### 14.15 meanangle4

mean angle

### 14.16 medianangle

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

### 14.17 medianangle2

median angle

input  
alpha : x\*m, [rad] angle

output  
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 rotate 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**   **inrotmat**

## **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 deflation\_matrix

### 20.2 det2x2

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



## 20.3 det3x3

determinant of stacked 3x3 matrices

## 20.4 det4x4

determinant of stacked 4x4 matrices

## 20.5 diag2x2

diagonal of stacked 2x2 matrices

## 20.6 down

## 20.7 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
%    $V*V=eye(j)$ ,  $Qschur'*V=0$ ,
%    $W=A*V$ ,  $M=W'*W$ 
%
% 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'

```

## 22.9 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/snm]];
% 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/snm]];
% 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**   `hausdorff`

hausdorff 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**   `ieig2x2`

reconstruct matrix from eigenvalue decomposition

### 23.5 inv2x2

2x2 inverse of stacked matrices

### 23.6 inv3x3

### 23.7 inv4x4

inverse of stacked 4x4 matrices

### 23.8 kernel2matrix

## 24 linear-algebra/lanczos

### 24.1 arnoldi

### 24.2 arnoldi\_new

### 24.3 eigs\_lanczos\_man

### 24.4 lanczos

### 24.5 lanczos\_

**24.6**   `lanczos_biorthogonal`

**24.7**   `lanczos_biorthogonal_improved`

**24.8**   `lanczos_ghep`

**24.9**   `mv_lanczos`

**24.10**   `reorthogonalise`

**24.11**   `test_lanczos`

**25**   `linear-algebra`

**25.1**   `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

c.f. Dronkers 1964, eq. 8.15, p. 300  
chebycheff polynomials  
function c = chebychev(x,n)

## 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

### 35.2 advection\_kernel



## **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/finite-difference**

### **37.1 cdiff**

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

### **37.2 cdiffb**

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

### **37.3 central\_difference**

### **37.4 cmean**

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

### **37.5 cmean2**

### 37.6 derivative\_matrix\_1\_1d

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

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

### 37.7 derivative\_matrix\_2\_1d

finite derivative matrix of second derivative in one dimension

### 37.8 derivative\_matrix\_2d

finite difference derivative matrix in two dimensions

### 37.9 derivative\_matrix\_curvilinear

derivative matrix on a curvilinear grid

### 37.10 derivative\_matrix\_curvilinear\_2

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

### 37.11 difference\_kernel

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

### 37.12 diffusion\_matrix\_2d\_anisotropic

### 37.13 diffusion\_matrix\_2d\_anisotropic2

### 37.14 directional\_neighbour

### 37.15 distmat

distance matrix for a 2 dimensional rectangular matrix

### 37.16 downwind\_difference

### 37.17 gradpde2d

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

### 37.18 laplacian

### 37.19 laplacian\_fdm

finite difference matrix of the laplacian  
BC

### 37.20 lrmean

mean of the left and right element

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

38.1 `fdm_adaptive_grid`

38.2 `fdm_adaptive_refinement_old`

38.3 `fdm_assemble_d1_2d`

38.4 `fdm_assemble_d2_2d`

38.5 `fdm_confinement`

38.6 `fdm_d_vargrid`

38.7 `fdm_h_unstructured`

38.8 `fdm_hydrogen_vargrid`

38.9 `fdm_mark_unstructured_2d`

**38.10** `fdm_plot`

**38.11** `fdm_plot_series`

**38.12** `fdm_refine_2d`

**38.13** `fdm_refine_3d`

**38.14** `fdm_refine_unstructured_2d`

**38.15** `fdm_schroedinger_2d`

**38.16** `fdm_schroedinger_3d`

**38.17** `relocate`

## **39** `numerical-methods/finite-difference`

**39.1** `mid`

`mid` point between neighbouring vector elements

## 39.2 pwmid

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

## 39.3 ratio

ratio of two subsequent values

## 39.4 steplength

step length of a vector if it were equispaced

## 39.5 swapoddeven

swap odd and even elements in a vector

## 39.6 test\_derivative\_matrix\_2d

## 39.7 test\_derivative\_matrix\_curvilinear

## 39.8 test\_difference\_kernel

## 39.9 upwind\_difference

## 40 numerical-methods/finite-element

40.1 Mesh\_2d.java

40.2 Tree\_2d.java

40.3 assemble\_1d\_dphi\_dphi

40.4 assemble\_1d\_phi\_phi

40.5 assemble\_2d\_dphi\_dphi.java

40.6 assemble\_2d\_phi\_phi.java

40.7 assemble\_3d\_dphi\_dphi.java

40.8 assemble\_3d\_phi\_phi.java

40.9 boundary\_1d

40.10 boundary\_2d

40.11 boundary\_3d

40.12 check\_area\_2d

40.13 circmesh

40.14 cropradius

40.15 display\_2d

40.16 display\_3d

40.17 distort

40.18 err\_2d

40.19 estimate\_err\_2d\_3



40.20 `example_1d`

40.21 `example_2d`

40.22 `explode`

40.23 `fem_2d`

40.24 `fem_2d_heuristic_mesh`

40.25 `fem_get_2d_radial`

40.26 `fem_interpolation`

40.27 `fem_plot_1d`

40.28 `fem_plot_1d_series`

40.29 `fem_plot_2d`

40.30 fem\_plot\_2d\_series

40.31 fem\_plot\_3d

40.32 fem\_plot\_3d\_series

40.33 fem\_plot\_confine\_series

40.34 fem\_radial

adaptive grid  
constant grid

40.35 flip\_2d

40.36 get\_mesh\_arrays

40.37 hashkey

41 numerical-methods/finite-element/int

41.1 int\_1d\_equal

## 41.2 int\_1d\_equal\_exp

## 41.3 int\_1d\_gauss

## 41.4 int\_1d\_gauss\_1

```
w : weights
    2/(1-xi^2)(P'_n(xi))^2
b : baricentric coordinates
    ith-root of legendre polynomial of order n
second order, midpoint rule
function [w, b, flag] = int_1d_gauss_1()
```

## 41.5 int\_1d\_gauss\_2

## 41.6 int\_1d\_gauss\_3

## 41.7 int\_1d\_gauss\_4

## 41.8 int\_1d\_gauss\_5

## 41.9 int\_1d\_gauss\_6

## 41.10 int\_1d\_gauss\_lobatto

41.11 int\_1d\_gauss\_n

41.12 int\_1d\_nc\_2

41.13 int\_1d\_nc\_3

41.14 int\_1d\_nc\_4

41.15 int\_1d\_nc\_5

41.16 int\_1d\_nc\_6

41.17 int\_1d\_nc\_7

41.18 int\_1d\_nc\_7\_hardy

41.19 int\_2d\_gauss\_1

41.20 int\_2d\_gauss\_12

41.21 int\_2d\_gauss\_13

41.22 int\_2d\_gauss\_16

41.23 int\_2d\_gauss\_19

41.24 int\_2d\_gauss\_25

41.25 int\_2d\_gauss\_3

41.26 int\_2d\_gauss\_33

41.27 int\_2d\_gauss\_4

41.28 int\_2d\_gauss\_6

41.29 int\_2d\_gauss\_7

41.30 int\_2d\_gauss\_9

41.31 int\_2d\_nc\_10

41.32 int\_2d\_nc\_15

41.33 int\_2d\_nc\_21

41.34 int\_2d\_nc\_3

41.35 int\_2d\_nc\_6

41.36 int\_3d\_gauss\_1

41.37 int\_3d\_gauss\_11

41.38 int\_3d\_gauss\_14

41.39 int\_3d\_gauss\_15

41.40 int\_3d\_gauss\_24

41.41 int\_3d\_gauss\_4

41.42 int\_3d\_gauss\_45

41.43 int\_3d\_gauss\_5

41.44 int\_3d\_nc\_11

41.45 int\_3d\_nc\_4

41.46 int\_3d\_nc\_6

41.47 int\_3d\_nc\_8

## 42 numerical-methods/finite-element

42.1 interpolation\_matrix

42.2 mark

42.3 mark\_1d

42.4 mesh\_1d\_uniform

42.5 mesh\_3d\_uniform

42.6 mesh\_interpolate

42.7 neighbour\_1d

42.8 old

42.9 pdeeig\_1d

42.10 pdeeig\_2d

42.11 pdeeig\_3d

42.12 polynomial\_derivative\_1d



42.13 potential\_const

42.14 potential\_coulomb

42.15 potential\_harmonic\_oscillator

42.16 project\_circle

42.17 project\_rectangle

42.18 promote\_1d\_2\_3

42.19 promote\_1d\_2\_4

42.20 promote\_1d\_2\_5

42.21 promote\_1d\_2\_6

42.22 quadrilaterate

42.23 recalculate\_regularity\_2d

42.24 refine\_1d

42.25 refine\_2d\_21

42.26 refine\_2d\_structural

42.27 regularity\_1d

42.28 regularity\_2d

42.29 regularity\_3d

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

42.30 relocate\_2d

42.31 test\_circmesh

42.32 test\_hermite

42.33 tri\_assign\_points

42.34 triangulation\_uniform

42.35 vander\_1d

van der Monde matrix

42.36 vanderd\_1d

42.37 vanderi\_1d

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

### 43.1 Advection

FVM treatment of the Advection equation

### 43.2 dot\_advection

advection equation

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

### 44.1 burgers\_split

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

### 44.2 dot\_burgers\_fdm

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

### 44.3 dot\_burgers\_fft

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

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

### 45.1 Finite\_Volume

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

### 45.2 apply\_bc

apply boundary conditions

### 45.3 solve

solve the the PDE by successively stepping in time  
 this is a trivial implmentation with constant step length  
 severity of diffusive error depends on  $dt/dx$ -ratio  
 stability depends on wave height

```
printf('Progress %2.1f%% %2.1fs\n',100*(t-Ti
(1))/(Ti(2)-Ti(1)),t_real);
```

#### 45.4 step\_split\_strang

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

#### 45.5 step\_unsplit

step in time, without splitting the inhomogeneous term

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

#### 46.1 Flux\_Limiter

class of flux limiters

#### 46.2 beam\_warming

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

#### 46.3 fromm

fromme limiter  
 low res

## 46.4 lax\_wendroff

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

## 46.5 minmod

min-mod schock limiter

## 46.6 monotized\_central

monotonized central flux limiter

## 46.7 muscl

muscl flux limiter

## 46.8 superbee

superbee limiter

## 46.9 upwind

godunov scheme  
godunov, first order accurate

## 46.10 vanLeer

van Leer limiter

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

### 47.1 dot\_kdv\_fdm

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

### 47.2 dot\_kdv\_fft

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

### 47.3 kdv\_split

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

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

### 48.1 Reconstruct\_Average\_Evolve

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

```
McCronack Scheme
```

```
err = O(dt^2) + O(dx^2), except as discontinuities
```

```
error:
```

```
h_xxx(3:end-2) = 1/dx^3*( -0.5*h(1:end-4) + h(2:end-3) - h(4:
    end-1) + 0.5*h(5:end) );
th = -1/6*dx^2*qh_.*(1 - (qh_*dt/dx).^2).*h_xxx;
```

### 48.2 advect\_highres

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

### 48.3 advect\_lowress

single time step  
low resolution

## 49 numerical-methods/finite-volume

### 49.1 Godunov

Godunov, upwind method for systems of pdes

### 49.2 Lax Friedrich

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

### 49.3 Measure

### 49.4 Roe

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

The roe solver guarantess:

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

### 49.5 fv\_swe

wrapper for solving SWE



## 49.6 staggered\_euler

forward euler method with staggered grid

## 49.7 staggered\_grid

staggered grid approximation to the SWE

# 50 numerical-methods

## 50.1 grid2quad

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

# 51 numerical-methods/integration

## 51.1 cumintL

cumulative integral from left to right

## 51.2 cumintR

cumulative integral from right to left

## 51.3 cumint\_trapezoidal

integrate y along x with the trapezoidal rule

## 51.4 int\_1d\_gauss\_laguerre

### 51.5 `int_trapezoidal`

integrate y along x with the trapezoidal rule

## 52 `numerical-methods/interpolation/@Kriging`

### 52.1 `Kriging`

class for Kriging interpolation

### 52.2 `estimate_semivariance`

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

### 52.3 `interpolate_`

interpolate with Kriging 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

## 53 `numerical-methods/interpolation/@RegularizedInterpolator`

### 53.1 `RegularizedInterpolator1`

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

## 53.2 init

initialize the interpolator with a set of sampling points

## 54 numerical-methods/interpolation/@RegularizedInterpolator2

### 54.1 RegularizedInterpolator2

class for regularized interpolation on an unstructures mesh (  
interpolation)

## 54.2 init

initialize the interpolator with a set of point samples

## 55 numerical-methods/interpolation/@RegularizedInterpolator3

### 55.1 RegularizedInterpolator3

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

## 55.2 init

initialize the interpolator with a set of sampling points

## 56 numerical-methods/interpolation

### 56.1 IDW

spatial averaging by inverse distance weighting

### 56.2 IPoly

polynomial interpolation class

### 56.3 IRBM

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

### 56.4 ISparse

```
sparse interpolation class
```

### 56.5 Inn

```
nearest neighbour interpolation
```

### 56.6 Interpolator

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

### 56.7 fixnan

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

### 56.8 idw1

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

### 56.9 idw2

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

### 56.10 inner2outer

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

### 56.11 inner2outer2

interpolate from element (segment) centres to edge points

### 56.12 interp1\_circular

### 56.13 interp1\_limited

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

### 56.14 interp1\_man

interpolate

### 56.15 interp1\_piecewise\_linear

### 56.16 interp1\_save

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

### **56.17 interp1\_slope**

quadratic interpolation returning value and derivative(s)

### **56.18 interp1\_smooth**

### **56.19 interp1\_unique**

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

### **56.20 interp2\_man**

nearest neighbour interpolation in two dimensions

### **56.21 interp\_angle**

interpolate an angle

### **56.22 interp\_fourier**

interpolation by the fourier method

### **56.23 interp\_fourier\_batch**

batch interpolation by the fourier interpolation

## 56.24 interp\_sn

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

## 56.25 interp\_sn2

interpolation in streamwise coordinates

## 56.26 interp\_sn3

## 56.27 interp\_sn\_

## 56.28 limit\_by\_distance\_1d

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

## 56.29 resample1

interpolation along a parametric curve with variable step width

## 56.30 resample\_d\_min

resample a function

### 56.31 `resample_vector`

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

### 56.32 `test_interp1_limited`

## 57 `numerical-methods`

### 57.1 `inverse_complex`

### 57.2 `maccormack_step`

### 57.3 `minmod`

## 58 `numerical-methods/multigrid`

### 58.1 `mg_interpolate`

### 58.2 `mg_restrict`

## 59 `numerical-methods/ode/@BVPS_Characteristic`

### 59.1 `BVPS_Characteristic`

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



## 59.2 assemble1\_A

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

## 59.3 assemble1\_A\_Q

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

## 59.4 assemble2\_A

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

## 59.5 assemble\_AA

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

## 59.6 assemble\_AAA

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

## 59.7 assemble\_Ic

## 59.8 bvp1c

## 59.9 check\_arguments

## 59.10 couple\_junctions

## 59.11 derivative

## 59.12 init

## 59.13 inner2outer\_bvp2c

## 59.14 reconstruct

## 59.15 resample

## 59.16 solve

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

odefun provides ode coefficients c:

$$\begin{aligned} c(x,1) y''(x) + c(x,2) y'(x) + c(x,3) y &= c(x,4) \\ c_1 y'' + c_2 y' + c_3 y + c_4 &= c_4 \end{aligned}$$

subject to the boundary conditions

bcfun provides v and p and optionally q, so that:

$$b_1 y + b_2 y' = f$$

$$\begin{aligned} q(x,1)*(p(x,1) y_l(x) + p(x,2) y_l'(x) \\ + q(x,2)*(p(x,1) y_r(x) + p(x,2) y_r'(x) &= v(x) \end{aligned}$$

where q weighs the waves travelling from left to right and right to left (default [1 1])

59.17 test\_assemble1\_A

59.18 test\_assemble2\_A

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

60.1 Time\_Stepper

60.2 solve

## 61 numerical-methods/ode

61.1 bvp2fdm

solve system of non-linear second order odes (in more than one variable)

as boundary value problems by the finite difference method

odefun provides ode coefficients c:

$$c(x,1) y''(x) + c(x,2) y'(x) + c(x,3) y = c(x,4)$$
$$c_1 y'' + c_2 y' + c_3 y + c_4 = 0$$

subject to the boundary conditions

bcfun provides v and p and optionally q, so that:

$$b_1 y + b_2 y' = f$$
$$q(x,1)*(p(x,1) y_l(x) + p(x,2) y_l'(x) + q(x,2)*(p(x,1) y_r(x) + p(x,2) y_r'(x) = v(x)$$

where q weighs the waves travelling from left to right and right to left (default [1 1])

## 61.2 bvp2wavetrain

solve second order boundary value problem by repeated integration

## 61.3 bvp2wavetwopass

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

## 61.4 ivp\_euler\_forward

solve initial value problem by the euler forward method

## 61.5 ivp\_euler\_forward2

## 61.6 ivprk2

solve initial value problem by the two step runge kutta method

## 61.7 ode2\_matrix

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

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

## 61.8 ode2characteristic

second order odes  
transmitted and reflected wave

## **61.9 step\_trapezoidal**

single trapezoidal step

## **61.10 test\_bvp2**

# **62 numerical-methods/optimisation**

## **62.1 aitken\_iteration**

## **62.2 anderson\_iteration**

## **62.3 armijo\_stopping\_criterion**

armijo stopping criterion for optimizations

## **62.4 astar**

astar path finding algorithm

## **62.5 binsearch**

binary search on a line

## **62.6 bisection**

bisection

## 62.7 box1

test objective function for optimisation routines

## 62.8 box2

## 62.9 cauchy

## 62.10 cauchy2

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

## 62.11 directional\_derivative

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

## 62.12 dud

optimization by the dud algorithm

## 62.13 extreme3

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

## 62.14 extreme\_quadratic

## 62.15 ftest

## 62.16 fzero\_bisect

## 62.17 fzero\_newton

## 62.18 grad

numerical gradient

## 62.19 hessian

numerical hessian

## 62.20 hessian\_from\_gradient

numerical hessian from gradient

## 62.21 hessian\_projected

numerical hessian projected to one dimension

## 62.22 line\_search

bisection routine

## 62.23 line\_search2

bisection method

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

## 62.24 line\_search\_polynomial

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



## 62.25 line\_search\_polynomial2

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

## 62.26 line\_search\_quadratic

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

## 62.27 line\_search\_quadratic2

```
quadratic line search
```

## 62.28 line\_search\_wolfe

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

## 62.29 ls\_bgfs

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

### 62.30 ls\_broyden

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

BGFS:  
Broyden 1965  
Fletcher 1970  
Goldfarb 1970  
Shanno 1970

### 62.31 ls\_generalized\_secant

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

### 62.32 nlcg

non-linear conjugate gradient  
input:  
x : nx1 start vectort  
opt : struct options  
fdx : gradient constraint

### 62.33 nlls

non-linear least squares

### 62.34 picard

picard iteration

### **62.35 poly\_extrema**

extrema of a polynomial

### **62.36 quadratic\_function**

evaluate quadratic function in higher dimensions

### **62.37 quadratic\_programming**

optimize by quadratic programming

### **62.38 quadratic\_step**

single step of the quadratic programming

### **62.39 rosenbrock**

rosenbrock test function

### **62.40 sqrt\_heron**

Heron's method for the square root

### **62.41 test\_directional\_derivative**

### **62.42 test\_dud**

### **62.43 test\_fzero\_newton**

62.44 test\_line\_search\_quadratic2

62.45 test\_ls\_generalized\_secant

62.46 test\_nlcg\_6\_order

62.47 test\_nlls

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

## 63 numerical-methods/pde

63.1 laplacian2d\_fundamental\_solution

## 64 numerical-methods/piecewise-polynomials

64.1 Hermite1

hermite polynomial interpolation in 1d

64.2 hp2\_fit

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

### 64.3 hp2\_predict

prediction with pw hermite polynomial  
c are values at support points

### 64.4 hp\_predict

predict with piecewise hermite polynomial

### 64.5 hp\_regress

fit piecewise hermite polynomial  
coefficients are values and derivatives

### 64.6 lp\_count

lagrangian basis for interpolation  
count number of valid samples

### 64.7 lp\_predict

lagrangian basis piecwie interpolation, predictor

### 64.8 lp\_regress

### 64.9 lp\_regress\_

## 65 numerical-methods

### 65.1 step\_advect\_euler\_explicit

65.2 `step_advection_diffusion_euler_implicit`

65.3 `step_advection_diffusion_trapezoidal`

65.4 `step_diffuse_analytic`

analytic solution to the heat equation  
the spectral solution is not positivity preserving as it results in  
spurious oscillations, this is avoided here, by integrating over  
    segments  
rather than sampling at gridpoints

65.5 `step_diffuse_euler_explicit`

65.6 `step_diffuse_euler_implicit`

65.7 `step_diffuse_spectral`

65.8 `step_diffuse_trapezoidal`

65.9 `step_react_euler_explicit`

65.10 `step_react_euler_implicit`

65.11 `step_react_midpoint`

65.12 `step_react_ralston`

65.13 `step_react_ralston_exp`

65.14 `step_react_ralston_exp_2`

65.15 `step_react_semi_analytic`

65.16 `step_react_trapezoidal`

65.17 `test_adams_bashforth`

## 66 mathematics

mathematical functions of various kind

66.1 `oversampleNZ`

## 67 pdes

### 67.1 heat\_equation\_fundamental\_solution

### 67.2 heat\_equation\_fundamental\_std\_to\_time

### 67.3 heat\_equation\_std

### 67.4 heat\_equation\_width

### 67.5 heat\_equation\_width\_to\_time

## 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  
 $y + \text{eps} = \text{alpha} + \text{beta} * x$

## 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

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

pause

### 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 autocorrerlation 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\_decay\_rate

estimate exponential decay of the autocorrelation

## 75.26 autocorr\_effective\_sample\_size

effective sample size from acf

## 75.27 autocorr\_fft

estimate sample autocorrelation function

## 75.28 autocorr\_forest

## 75.29 autocorr\_genton

autocorrelation function

## 75.30 autocorr\_highpass

## 75.31 autocorr\_lowpass

## 75.32 autocorr\_periodic\_additive\_noise

## 75.33 autocorr\_periodic\_windowed

## 75.34 autocorr\_radial

## 75.35 autocorr\_radial\_hexagonal\_pattern

## 75.36 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 walues 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 coherence

## 76.9 conv1\_man

convolutions with padding

## 76.10 conv2\_man

convolution in 2d

## 76.11 conv2z

## 76.12 conv30

convolve with rectangular window of length n  
circular boundaries

## 76.13 conv\_

convolution of a with b

## 76.14 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

## 76.15 convz

## 76.16 cosexpdelay

### 76.17 csmooth

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

### 76.18 daniell\_window

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

### 76.19 db2neper

convert decibel to neper

### 76.20 db2power

power ratio from db

### 76.21 derive\_bandpass\_continuous\_scale

### 76.22 derive\_danielle\_weight

### 76.23 derive\_limit\_0\_acfar

### 76.24 detect\_peak

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

## 76.25 `determine_phase_shift`

## 76.26 `determine_phase_shift1`

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

## 76.27 `doublesum_ij`

double sum of  $r^i$

## 76.28 `effective_mask_size`

## 76.29 `effective_sample_size_to_ar1`

convert effective sample size to ar1 correlation

## 76.30 `fcut2Lw_gausswin`

## 76.31 `fcut_gausswin`

## 76.32 `flt_hodges_lehman`

# 77 `signal-processing/filters`

## 77.1 `circfilt2`

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 aliasing (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  
analogue 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  
trfilt1 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 (probability distributions)

### 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

nanning window for chage 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

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

## **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\_convolution

## 79.7 bandpass2d\_fft

## 79.8 bandpass2d\_ideal

## 79.9 bandpass2d\_implicit

bandpass filter the surface  $x$  by solving the implicit relation:

## 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  $p$ th-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_anisotropic`

## 79.28 `lowpass2d_convolution`

this function is computationally inefficient and serves merely for  
illustration  
and tests

## 79.29 `lowpass2d_fft`

note : this function is for testing purposes only,  
directly multiply the ft of the signal with the ft of the  
filter  
to obtain the filtered signal in a single step  
function `y = lowpass2d_fft(x,rho,a,order)`



### 79.30 lowpass2d\_ideal

lowpass filter the input x in the Frequency Domain

TODO no need to provide dx, follows from size of x  
function [y,S,R,r]=lowpass2d\_ideal(x,L,dx,varargin)

### 79.31 lowpass2d\_implicit

function [y] = lowpass2d\_implicit(x,rho,a,order,direct)

### 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

```
input:
    Sxy : nxn
output:
    Sra : n/2*(pi*n/2)
    angle
function [Sa,angle,A] = periodogram_angular(Sxy,L,nf)
```

### 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_normalize`

**81.11**   `periodogram_normalize_2d`

**81.12**   `periodogram_p_value`

**81.13**   `periodogram_qq`

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

**81.14**   `periodogram_quantiles`

quantiles of a periodogram

**81.15**   `periodogram_radial`

`function [Sr,fri,se,count] = periodogram_radial(S2d,L)`

compute the radially averaged density

input:

S2d : 2-dimensional density or periodogram

$L = [L_x, L_y]$  : domain length

output:

$S_r.\mu$  : radially averaged periodogram  
 $S_r.normalized$  : normalized radially averaged periodogram  
 $A$  : matrix operator s that  $S_r = (A \cdot A')^{-1} A' \cdot S_{2d}$

$f_r$  : radial frequencies, at which radial periodogram is determined  
discretized in same interval as the 2d-density :  $f = 1/L$

Definitions:

radial wavenumber, identical to circumferences of circles  
centred at origin with radial frequency  $f_r$   
 $k_r = 2\pi \cdot f_r$

radially averaged periodogram:  

$$S_r(k_r) = \frac{1}{k_r} \int_0^{k_r} S_{2d}(k_r, s) ds$$

$$= \frac{1}{(2\pi)} \int_0^{2\pi} S_{2d}(k_r, \theta) d\theta$$

$$\sim \frac{1}{(2\pi)} \sum_{n=1}^{nt} S_{2d}(k_r, \theta_n) \cdot (2\pi/nt)$$

$$\sim \frac{1}{nt} \sum_{n=1}^{nt} S_{2d}(k_r, \theta_n)$$

$nt \sim k_r/df = k_r \cdot L$

normalization:

$$S_r.normalized = S_r / \int_0^\infty S_r df_r$$

$$\sim S_r / (\sum_{n=1}^{nr} S_r \Delta f_r)$$

note : the radially averaged "periodogram", is actually a density estimate,  
for radial frequencies  $f_r$  are not small

when  $S$  is flattened into a vector, the isotropic part of the 2D density can be recovered with:

$S_{iso} = (A \cdot S_{radial})$   
 $S_{radial} = A^{-1} S_{iso}$

## 81.16 periodogram\_std

standard deviation of a periodogram

## 81.17 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

```

## 81.18 periodogram\_test\_periodicity\_2d

test a periodogram for hidden periodic frequency components

```

[pn, stat, out] = periodogram_test_periodicity_2d(b, L, nf, bmsk,
    fmsk, ns, significance_level)

input:
    b (nx * ny): image to test for presence of hidden
        periodicities,
        i.e. periodicities where the frequency is not known a
            priori
    L : domain size in arbitrary units, default is n
        only effects scaling of complementary outout Shat and
            Sbar
        does not effect test as it cancels out in the tested
            ratio Shat/Sbar
    nf : nfr or [nfx, nfy]
        radius of circular disk (in number of bins) used for
            smoothing
        the periodogram to estimate the spectral density,
            or axes of ellipses for smoothing
        when b is not square a good choice is nfx/nfy ~ Lx/Ly
    bmsk : mask in real space selecting parts of the image to
        include in
        the analysis. default is the entire image

```

the mask can have non-integer values to feather the borders of the mask

fmsk : mask in frequency selecting frequencies to test for periodicity

default is all frequencies

note: when b is real, one half plane can always be excluded

because of symmetry. This slightly increases the significance

n\_mc : number of samples for the monte-carlo determination of the test statistics, mc is only used when parts of the image are masked

otherwise the analytic test statistic is used

significance\_level :

output :

pn : p-value of largest frequency component with largest ratio Shat/Sbar

when testing all frequency components selected by fmsk

stat.max.ratio : max ratio value of Shat/Sbar

stat.max.Shat : periodogram value of frequency component with max ratio

stat.max.Shat\_rel : spectral energy contained frequency component with max ratio

stat.max.fx : x-component of frequency at max ratio

stat.max.fy : y-component of frequency of max ratio

stat.intShat\_sig : spectral energy contained in all significant frequency bins

stat.p1 : p-value of all frequency components

stat.pn : p-value of all frequency components, corrected for multiple comparisons

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  $nf^2$

## 81.19 periodogram\_test\_stationarity

test a periodogram for stationarity

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

### 81.20 `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 hex\_angular\_pdf

## 83.2 hex\_angular\_pdf\_max

## 83.3 hex\_angular\_pdf\_max2par

## 83.4 spectral\_density\_ar2

## 83.5 spectral\_density\_area

integrate the spectral density over the positive half axis

## 83.6 spectral\_density\_estimate\_2d

### **83.7 spectral\_density\_flat**

flat spectral density of a random vector with iid elements

### **83.8 spectral\_density\_forest**

### **83.9 spectral\_density\_gausswin**

### **83.10 spectral\_density\_lorentzian**

lorentzian spectral density

### **83.11 spectral\_density\_lorentzian\_max**

mode (maximum) of the lorentzian spectral density

### **83.12 spectral\_density\_lorentzian\_max2par**

transform maximum of the lorentzian spectral density to its  
distribution parameters

### **83.13 spectral\_density\_lorentzian\_scale**

normalization scale of the lorentzian spectral density

### **83.14 spectral\_density\_maximum\_bias\_corrected**

### **83.15 spectral\_density\_periodic\_additive\_noise**

83.16 spectral\_density\_rectwin

83.17 spectral\_density\_wperiodic

## 84 signal-processing

### 84.1 spectrogram

spectrogram

### 84.2 sum\_i\_lag

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

### 84.3 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

### 84.4 sum\_ii\_

### 84.5 sum\_ij

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

**84.6** `sum_ij_`

**84.7** `sum_ij_partial_`

**84.8** `sum_multivar`

sum of matrix entries of bivariate ar1 process

**84.9** `test_acfar1`

**84.10** `tikhonov_to_ar1`

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

**84.11** `trapwin`

trapezoidal filter window

**84.12** `triwin`

triangular filter window

**84.13** `triwin2`

triangular filter window

**84.14** `tukeywin_man`

### 84.15 `varar1`

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

### 84.16 `welch_spectrogram`

welch spectrogram

### 84.17 `wfilt`

filter with window

### 84.18 `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

class for analysis of remotely sensed and model generated  
vegetation patterns

### 88.2 analyze\_grid

analyze a 2D spatial pattern, estimate regularity and test for  
periodicity

### 88.3 analyze\_transect

analyze 1D transect through a spatial pattern,  
either remotely sensed or model generated

### 88.4 clear\_1d\_properties

### 88.5 clear\_2d\_properties



## 88.6 `fit_parametric_densities`

fit parametric spectral densities to the empirical density

## 88.7 `imread`

read an image file containing a pattern, mask and geospatial data

## 88.8 `init`

## 88.9 `plot`

plot the pattern or densities

## 88.10 `plot_transect`

plot 1D pattern

## 88.11 `prepare_analysis`

## 88.12 `report`

report statistics of analysis

## 88.13 `resample_functions`

resample empirical densities to a common grid

## 88.14 tabulate

summarize properties of multiple patterns in a single struct

# 89 spatial-pattern-analysis/@Spatial\_Pattern\_Array

## 89.1 Spatial\_Pattern\_Array

container class for Spatial\_Pattern objects

## 89.2 analyze

analyze spatial patterns

## 89.3 assign\_regions

## 89.4 export\_shp

## 89.5 fetch

determine the sampling interval for fetching images from the Google  
satellite  
server and later processing

## 89.6 generate\_filename

## 89.7 quality\_check

## 90 spatial-pattern-analysis

### 90.1 approximate\_ratio\_distribution

```
input :
bmsk : region selected for periodicity test (smoothes the
       periodogram)
nf    : radius of smoothing window (in bins) for estimating the
       spectral density
nsample : number of repetitions to estimate the ratio distribution
         recommended at

output:
pr     : probabilities for quantiles
qr1    : quantiles of the distribution for bin m
qrn    : quantiles of the distribution for the maximum of bins
         selected by fmsk
ratio  : ratios for each frequency bin and iteration (only for
         last block, for testing)

input:
bmsk   : mask region pattern/interest in the real domain
nf     : smoothing window radius in the frequency domain for
         density estimation
ns     : number of samples for the monte-carlo simulation
fmsk   : mask frequencies of interest
mdx    : selection of an a-priori known frequency bin

note the following complications:
- problem 1 : ratio locally differs near fr=0, fx,fy=fmax and
              fx,fy=fmax/2
- fits of the fisher or beta distribution are highly unstable
```

### 90.2 banded\_pattern

### 90.3 hexagonal\_pattern

```
function [z, x, y, xx, yy, xe, ye] = hexagonal_pattern(fc,n,L,
    angle0_rad,scale,sbm,p,q)

spot pattern of unit amplitude
output : z : pattern
        x : x-coordinate
        y : y-coordinate
```

Note :  $z_{\text{gap}} = 1 - z_{\text{spot}}$

#### 90.4 patch\_size\_1d

#### 90.5 patch\_size\_2d

#### 90.6 pattern\_isotropic\_rotated

#### 90.7 reconstruct\_isotropic\_density

#### 90.8 separate\_isotropic\_from\_anisotropic\_density

#### 90.9 suppress\_low\_frequency\_lobe

### 91 spatial-statistics

#### 91.1 cov\_cell\_averages\_1d

determine covariance between grid cell averaged values of a  
stationary  
stochastic process on an equispaced grid

$$\begin{aligned}\text{cov}(e_{ij}, e_{kl}) &= E[ (e_{ij} - \mu) * (e_{kl} - \mu) ] \\ &= 1/dx^2 E[ \int (e(x1) - \mu) dx1 \int (e(x2) - \mu) dx2 ] \\ &= 1/dx^2 E[ \int \int (e(x1) - \mu) (e(x2) - \mu) dx1 dx2 \\ &\quad ) ] \\ &= 1/dx^2 \int \int \text{cov}(x2-x1) dx1 dx2\end{aligned}$$

$$f_{ij} = \int_{x_i - dx/2}^{x_i + dx/2} f(x) dx$$

integrals approximated by Gauss' method

## 91.2 cov\_cell\_averages\_2d

determine covariance between grid cell averaged values of a stationary stochastic process on an equispaced grid

$$\begin{aligned} \text{cov}(e_{ij}, e_{kl}) &= E[(e_{ij} - \mu)(e_{kl} - \mu)] \\ &= 1/dx^2 1/dy^2 E[\int \int (e(x_1, y_1) - \mu) dx_1 dy_1 \int \int (e(x_2, y_2) - \mu) dx_2 dy_2] \\ &= 1/dx^2 1/dy^2 E[\int \int \int \int (e(x_1, y_1) - \mu)(e(x_2, y_2) - \mu) dx_1 dy_1 dx_2 dy_2] \\ &= 1/dx^2 1/dy^2 \int \int \int \int \text{cov}(x_2 - x_1, y_2 - y_1) dx_1 dy_1 dx_2 dy_2 \end{aligned}$$

$$f_{ij} = \int_{x_i - dx/2}^{x_i + dx/2} \int_{y_i - dy/2}^{y_i + dy/2} f(x, y) dx dy$$

integrals approximated by equal spaced mid-point intervals, this allows to reduce the double-integral along each dimension into a single integral and hence to reduce the computational effort from  $m^4$  to  $m^2$

## 92 special-functions

### 92.1 bessell\_sphere

spherical Bessel function of the first kind

### 92.2 bessellln\_large\_x

### 92.3 beta\_man

## 92.4 betainc\_man

## 92.5 digamma\_man

## 92.6 exp10

## 92.7 hankel\_sphere

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

## 92.8 hermite

probabilistic's hermite polynomial by recurrence relation

input :  
n : order  
x : value

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

## 92.9 laguerre\_roots

## 92.10 lambertw\_numeric

lambert-w function

## 92.11 legendre\_man

legendre polynomials

## 92.12 neumann\_sphere

spherical Neumann function  
Bessel function of the second kind

## 93 statistics

### 93.1 atan\_s2

stadard deviation of the arcus tangens by means of taylor expansion

### 93.2 binomial

generalized binomial coefficient, working for non-integer arguments  
,  
in contrast to the matlab buildin function nchoosek

## 94 statistics/circular

### 94.1 circular\_fmoment

### 94.2 circular\_fquantile

### 94.3 circular\_fstd

#### 94.4 circular\_fvar

### 95 statistics

#### 95.1 coefficient\_of\_determination

#### 95.2 conditional\_expectation\_normal

#### 95.3 correlation\_confidence\_pearson

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

### 96 statistics/distributions

#### 96.1 PDF

class for quasi-distributions from a set of sampling points

### 97 statistics/distributions/anisotropic

#### 97.1 anisotropic\_pattern

#### 97.2 anisotropic\_pattern\_acf

#### 97.3 anisotropic\_pattern\_pdf



## **98 statistics/distributions/beta**

### **98.1 beta\_kurt**

### **98.2 beta\_mean**

### **98.3 beta\_moment2par**

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

### **98.4 beta\_skew**

### **98.5 beta\_std**

## **99 statistics/distributions/bivariate-normal**

### **99.1 binorm\_separation\_coefficient**

separation coefficient of a bimodal normal distribution

### **99.2 binormcdf**

bio-modal gaussian distribution

### **99.3 binormfit**

fit sum of to normal distribution to a histogram

## 99.4 binormpdf

# 100 statistics/distributions/chi2

## 100.1 chi2\_kurt

## 100.2 chi2\_mean

## 100.3 chi2\_skew

## 100.4 chi2\_std

# 101 statistics/distributions/circular-normal

## 101.1 wnormpdf

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

# 102 statistics/distributions/edgeworth

## 102.1 edgeworth\_cdf

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

## 102.2 edgeworth\_pdf

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

## 103 statistics/distributions/exp

### 103.1 exppdf\_max2par

## 104 statistics/distributions/fisher

### 104.1 fisher\_mean

### 104.2 fisher\_moment2par

### 104.3 fisher\_std

## 105 statistics/distributions/gamma

### 105.1 gamma\_mean

### 105.2 gamma\_mode

### 105.3 `gamma_mode2par`

### 105.4 `gamma_moment2par`

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

### 105.5 `gamma_std`

### 105.6 `gamma_stirling`

### 105.7 `gampdf_man`

### 105.8 `generalized_gamma_mean`

## 106 `statistics/distributions/hotelling-t2`

### 106.1 `t2cdf`

Hotelling's T-squared cumulative distribution

### 106.2 `t2inv`

inverse of Hotelling's T-squared cumulative distribution

## 107 statistics/distributions/kurt-normal

### 107.1 kurtncdf

### 107.2 kurtnpdf

## 108 statistics/distributions/log-triangular

### 108.1 logtrialtcdf

pdf of a logarithmic triangular distribution

### 108.2 logtrialtinv

inverse of the logarithmic triangular distribution

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

### 108.3 logtrialtmean

mean of the logarithmic triangular distribution

### 108.4 logtrialtpdf

density of the logarithmic triangular distribution

### 108.5 logtrialrnd

### 108.6 logtricdf

cumulative distribution of the logarithmic triangular distribution

### 108.7 logtriinv

inverse of the logarithmic triangular distribution

### 108.8 logtrimean

mean of the logarithmic triangular distribution

### 108.9 logtripdf

probability density of the logarithmic triangular distribution

### 108.10 logtirnd

## 109 statistics/distributions/log-uniform

### 109.1 logu\_median

median of the log-uniform distribution

### 109.2 logucdf

probability density of the logarithmic uniform distribution

### 109.3 logucm

central moments of the log-uniform distribution

### 109.4 loguinv

inverse of the log-uniform distribution

### 109.5 logumean

mean of the log-uniform distribution

### 109.6 logupdf

pdf of the log uniform distribution

### 109.7 logurnd

random numbers following a log-uniform distribution

### 109.8 loguvar

variance of the log-uniform distribution

## 110 statistics/distributions/loglog

### 110.1 loglogpdf

## 111 statistics/distributions/lognormal

### 111.1 logn\_corr

```
function corr_eaeb = logn_corr(lr,lmu_a,lmu_b,lsd_a,lsd_b)
```

correlation of two log-normal random variables, where the log of  
the variables  
is correlated with r

### 111.2 logn\_cov

```
covariance of two log-normally distributed random variables,  
cov(ea,eb) = cov(exp(mua + sa*za),exp(mub + sb*zb))  
where za, zb are standard normal distributed and correlated
```

### 111.3 logn\_mean

### 111.4 logn\_mode

mode (maximum) of the log-normal density

### 111.5 logn\_mode2par

### 111.6 logn\_moment2par

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

### 111.7 logn\_moment2par\_correlated



### 111.8 logn\_param2moment

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

### 111.9 logn\_skewness

### 111.10 logn\_std

### 111.11 lognpdf\_

log normal distribution called by modes rather than parameters

### 111.12 lognpdf\_entropy

## 112 statistics/distributions/logskew

### 112.1 logskewcdf

### 112.2 logskewpdf

## 113 statistics/distributions/mises

### 113.1 mises\_max2par

113.2 `mises_std`

113.3 `mises_var`

113.4 `misesn_max2par`

113.5 `misesnpdf`

113.6 `misespdf`

114 `statistics/distributions`

114.1 `ncx2_moment2par`

115 `statistics/distributions/normal`

115.1 `normpdf_entropy`

115.2 `normpdf_mode`

115.3 `normpdf_mode2par`

## 116 statistics/distributions/passes

### 116.1 bandpass1d\_continuous\_pdf

```
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
```

### 116.2 bandpass1d\_continuous\_pdf\_max

maximum of the bandpass spectral density

### 116.3 bandpass1d\_continuous\_pdf\_max2par

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

### 116.4 bandpass1d\_continuous\_pdf\_scale

normalization scale of the spatial bandpass density

### 116.5 bandpass1d\_discrete\_pdf

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

### 116.6 bandpass2d\_discrete\_pdf

### 116.7 bandpass2d\_pdf\_exact

```
function Sb = bandpass2d_pdf(fr,a,order)
not normalized, max (S) = 1;
```

### 116.8 bandpass2d\_pdf\_hankel

### 116.9 bandpass2d\_pdf\_mode

### 116.10 bandpass2d\_pdf\_mode2par

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

### 116.11 bandpass2d\_pdf\_scale

### 116.12 highpass1d\_continuous\_pdf

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

output :

S\_bp : spectral density of the highpass filter in continuous space  
limit case of the discrete highpass 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

### 116.13 highpass1d\_discrete\_cos\_pdf

cosine shaped spectral density of a highpass filter

### 116.14 highpass1d\_discrete\_pdf

spectral density of the pth-order high-pass

Note that there are two alternative definitions

$S_{hp} = S_{h1}^p = (1 - S_{l1})^p$  (recursive highpass-filtering)

or  $S_{hp} = (1 - S_{lp}^p)$  (1 - recursive lowpass-filtering)

here, recursive highpass filtering is represented

$$S_h = |F * A_h|^2$$

$$S_h^{(1/2p)} = F A_h = F(I - A_l)$$

$$= F(I - A_l)$$

$$= F(I - F^{-1} S_l^{1/2} F)$$

$$= (F F^{-1} - S_l^{1/2}) F$$

$$= (I - S_l^{(1/2)}) F$$

function [S\_h,S\_h1] = spectral\_density\_hp(f,r0,fmax,order,varargin)

### 116.15 highpass2d\_discrete\_pdf

### 116.16 highpass2d\_pdf

### 116.17 highpass2d\_pdf\_hankel

116.18 lowpass1d\_continuous\_pdf

116.19 lowpass1d\_continuous\_pdf\_scale

116.20 lowpass1d\_discrete\_pdf

116.21 lowpass1d\_one\_sided\_pdf

116.22 lowpass2d\_discrete\_acf

truncated, not wrapped at the end

116.23 lowpass2d\_discrete\_pdf

116.24 lowpass2d\_pdf

116.25 lowpass2d\_pdf\_hankel

spectral density of the two-dimensional lowpass filter with  
autocorrelation

$r = \exp(-a \sqrt{x^2 + y^2})$

efficiently estimated with gauss-laguerre integration and 1D-FFT:

for a radially symmetric function, the radial density is  
 $S_r(r) = S_{2d}(r,0) = S_{2d}(0,r)$

with density  $S_{2d}$  and autocorrelation  $R_{2d}$   
 $S_{2d} = F_{2d}^{-1}(R_{2d})$   
 by the slicing theorem:  
 $S_{2d}(x,0) = F_{1d}^{-1}(\int R_{2d}(x,y) dy)$

## 116.26 lowpass2d\_pdf\_series

## 117 statistics/distributions

### 117.1 pdfsample

pdf from sample distribution  
 Note: better use kernel density estimates

## 118 statistics/distributions/phase-drift

### 118.1 phase\_drift\_acf

### 118.2 phase\_drift\_acf\_2d

### 118.3 phase\_drift\_cdf

### 118.4 phase\_drift\_inv

### 118.5 phase\_drift\_parallel\_acf

118.6 `phase_drift_parallel_pdf`

118.7 `phase_drift_parallel_pdf_max`

118.8 `phase_drift_parallel_pdf_max2par`

118.9 `phase_drift_parallel_pdf_mode2par`

118.10 `phase_drift_patch_size_distribution`

118.11 `phase_drift_pdf`

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

118.12 `phase_drift_pdf_2d`

118.13 `phase_drift_pdf_mode`

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

118.14 `phase_drift_pdf_mode2par`

transform mode to parameters of the brownian phase spectral density



118.15 `phase_drift_pdf_reg2par`

118.16 `phase_drift_pdf_scale`

normalization scale of the brownian phase spectral density

119 `statistics/distributions/skew-normal`

119.1 `skew_generalized_normal_fit`

119.2 `skew_generalized_normpdf`

119.3 `skewcdf`

119.4 `skewparam_to_central_moments`

119.5 `skewpdf`

skew-normal distribution  
c.f. Azzalini 1985

119.6 `skewpdf_entropy`

## **120    statistics/distributions/triangular**

### **120.1    tricdf**

cumulative distribution of the log-triangular distribution

### **120.2    triinv**

inverse of the triangular distribution

### **120.3    trimediam**

median of the triangular distribution

### **120.4    tripdf**

probability density of the triangular distribution

### **120.5    trirnd**

random numbers of the triangular distribution

## **121    statistics/distributions/weibull**

### **121.1    wbl\_std**

## **122    statistics/distributions/wrapped-normal**

### **122.1    normpdf\_wrapped**

122.2 `normpdf_wrapped_mode`

122.3 `normpdf_wrapped_mode2par`

## 123 `statistics`

123.1 `error_propagation_fraction`

123.2 `error_propagation_product`

123.3 `example_standard_error_of_sample_quantiles`

123.4 `f_var_finite`

reduction of variance when sampling from a finite population  
without replacement

123.5 `gaussfit3`

123.6 `gaussfit_quantile`

123.7 `geoserr`

## 123.8 geostd

## 123.9 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

## 123.10 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

## 124 statistics/information-theory

### 124.1 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

## 124.2 bayesian\_information\_criterion

bayesian information criterion

## 125 statistics

### 125.1 jackknife\_block

### 125.2 kurtosis\_bias\_corrected

bias corrected kurtosis

### 125.3 limit

limit a by lower and upper bound

### 125.4 logfactorial

approximate log of the factorial

### 125.5 lognfit\_quantile

### 125.6 max\_exprnd

### 125.7 maxnnormals

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

125.8 mean\_angle

125.9 mean\_max\_n

125.10 mean\_min\_n

125.11 midrange

mid range of columns of X

125.12 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

125.13 mode\_man

## 126 statistics/moment-statistics

126.1 autocorr\_man3

autocorrelation of the columns of X

## 126.2 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

## 126.3 autocorr\_man5

autocorrelation of the columns of  $X$

## 126.4 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

## 126.5 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

$\mu$  :  $n \times 1$  mean vector

$C$  :  $n \times n$  covariance matrix

$k$  :  $n \times 1$  powers of variables in moments

## 126.6 corr\_man

correlation of two vectors

### 126.7 cov\_man

covariance matrix of two vectors

### 126.8 dof

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

### 126.9 edgeworth\_quantile

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

### 126.10 effective\_sample\_size

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

### 126.11 f\_correlation

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

### 126.12 f\_finite

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

### 126.13 lmean

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



#### **126.14 lmoment**

l-moment of vector x

#### **126.15 maskmean**

mean of the masked values of X

#### **126.16 masknanmean**

#### **126.17 mean1**

mean of x

#### **126.18 mean\_man**

mean and standard error of X

#### **126.19 mse**

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

#### **126.20 nanautocorr\_man1**

autocorrelation of a vector with nan-values

#### **126.21 nanautocorr\_man2**

autocorrelation of a vector with nan-values

## 126.22 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

## 126.23 nancorr

(co)-correlation matrix when samples a NaN

## 126.24 nancumsum

cumulative sum, setting nan values to zero

## 126.25 nanlmean

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

## 126.26 nanr2

coefficient of determination when samples are invalid

## 126.27 nanrms

root mean square value when sample contains nan-values

## 126.28 nanrmse

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

### 126.29 nanserr

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

### 126.30 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)

### 126.31 nanwstd

weighed standard deviation

### 126.32 nanwvar

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

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

### 126.33 nanxcorr

### 126.34 pearson

pearson correlation coefficient

### 126.35 pearson\_to\_kendall

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

### 126.36 pool\_samples

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

### 126.37 qmean

trimmed mean

### 126.38 range\_mean

### 126.39 rmse\_

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

### 126.40 serr

standard error of the mean of a set of uncorrelated samples

### 126.41 serr1

### 126.42 test\_qskew

### 126.43 test\_qstd\_qskew\_optimal\_p

## 126.44 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

## 126.45 wcorr

correlation of two vectors when samples are weighted

## 126.46 wcov

covariance of two vectors when samples are weighted

## 126.47 wdof

effective degrees of freedom for weighted samples

## 126.48 wkurt

kurtosis with weighted samples

## 126.49 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)

## 126.50 wrms

weighted root mean square

## 126.51 wserr

weighted root mean square error

## 126.52 wskew

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

## 126.53 wstd

weighed standard deviation

## 126.54 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

## 127 statistics

### 127.1 nangeomean

### 127.2 nangeostd

geometric standard deviation ignoring nan-values

## 128 statistics/nonparametric-statistics

### 128.1 kernel1d

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

### 128.2 kernel2d

kernel density estimate in two dimensions

## 129 statistics

### 129.1 normalize\_exponential\_random\_variable

### 129.2 normmoment

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

### 129.3 normpdf2

pdf of the bivariate normal distribution

## 130 statistics/order-statistics

### 130.1 hodges\_lehmann\_location

hodges lehman location estimator

Asymptotic rms efficiency of location estimte:

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

## 130.2 kendall

kendall correlation coefficient

## 130.3 kendall\_to\_pearson

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

c.f. Kruskal, 1958, p. 823

## 130.4 mad2sd

transform median absolute deviation to standard deviation  
for normal distributed values

## 130.5 madcorr

proxy correlation by median absolute deviation

## 130.6 median2\_holder

## 130.7 median\_ci

median and its confidence intervals under assumption of normality  
 $se\_me = \sqrt{1/2 \pi} \cdot 1.25331 \cdot sd/\sqrt{n}$

## 130.8 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



### 130.9 mediani

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

### 130.10 nanmadcorr

proxy correlation by median absolute deviation

### 130.11 nanwmedian

weighted median, skips nan-values

### 130.12 nanwquantile

weighted quantile, skips nan values

### 130.13 oja\_median

two dimensional oja median

note: the multivariate median is not unique

oja 1983, for extension to multivariate function, see chaudhri

### 130.14 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

### 130.15 qmoments

moments estimated from quantiles

### 130.16 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

### 130.17 qskewq

skewness estimated by quantiles

### 130.18 qstdq

proxy standard deviation determined by quantiles

### 130.19 quantile1\_optimisation

### 130.20 quantile2\_breckling

quantile regression

### 130.21 quantile2\_chaudhuri

quantile regression

### **130.22 quantile2\_projected**

quantile in two dimensions

### **130.23 quantile2\_projected2**

spatial quantile for chosen direction

### **130.24 quantile\_envelope**

### **130.25 quantile\_regression\_simple**

simple quantile regression

### **130.26 ranking**

ranking for spearman statistics

### **130.27 spatial\_median**

c.f. Oja 2008

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

### **130.28 spatial\_quantile**

spatial quantile

### **130.29 spatial\_quantile2**

spatial quantile

### 130.30 spatial\_quantile3

spatial quantile

### 130.31 spatial\_rank

unsigned rank

### 130.32 spatial\_sign

spatial sign

### 130.33 spatial\_signed\_rank

signed rank

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

### 130.34 spearman

spearman's product moment coefficient

### 130.35 spearman\_rank

### 130.36 spearman\_to\_pearson

conversion of spearman rank to person product moment correlation  
coefficient

### 130.37 wmedian

weighted median

### 130.38 wquantile

weighted quantile

## 131 statistics

### 131.1 qstd

### 131.2 quantile\_extrap

### 131.3 quantile\_sin

## 132 statistics/random-number-generation

### 132.1 laplacernd

random number of laplace distribution

### 132.2 randc

correlate to correlated standard normally distributed vectors

### 132.3 skewness2param

### 132.4 skewpdf\_central\_moments

## 132.5 skewrnd

random numbers of the skew normal distribution

## 133 statistics

### 133.1 range

range and mid range of input

### 133.2 resample\_with\_replacement

## 134 statistics/resampling-statistics/@Jackknife

### 134.1 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

## 134.2 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

## 134.3 matrix1\_STATIC

matrix of estimation for leaving out two samples at a time

## 134.4 matrix2

matrix of estimations for jackknife with two samples left out

# 135 statistics/resampling-statistics

## 135.1 block\_jackknife

## 135.2 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

## 135.3 moving\_block\_jackknife

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

## 135.4 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

## 135.5 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

## 136 statistics

### 136.1 scale\_quantile\_sd

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

### 136.2 sd\_sample\_quantiles



### 136.3 spatialrnd

### 136.4 trimmed\_mean

trimmed mean

### 136.5 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

### 136.6 ttest\_man

two-sample t-test

unequal sample size

equal variance

### 136.7 ttest\_paired

paired t-test

unequal sample size

equal variance

more powerfull than unpaired test, as long as correlation between  
x1 and x2 > 0

### 136.8 uniformnpdf

### 136.9 wgeomean

weighted geometric mean

function mu = wgeomean(w,x)

### 136.10 wgeovar

variance of the weighted geometric mean

### 136.11 wharmean

weighted harmonic mean

### 136.12 wharstd

### 136.13 wharvar

## 137 stochastic

### 137.1 brownian\_drift\_hitting\_probability

### 137.2 brownian\_drift\_hitting\_probability2

### 137.3 brownian\_field

simulate Fractional Brownian field on unit disk, with Hurst  
parameter 'H';

Reference:

%%%

#### 137.4 brownian\_field\_scaled

generate a square (fractal brownian) field where the variance is  
ellyptic,  
i.e. increasing at different rates in both axes  
this is facilitated by cropping and stretching  
TODO can the kernel directly be adapted?

#### 137.5 brownian\_motion\_1d\_acf

#### 137.6 brownian\_motion\_1d\_cov

#### 137.7 brownian\_motion\_1d\_fft

#### 137.8 brownian\_motion\_1d\_fourier

#### 137.9 brownian\_motion\_1d\_interleave

#### 137.10 brownian\_motion\_1d\_laplacian

#### 137.11 brownian\_motion\_2d\_cov

#### 137.12 brownian\_motion\_2d\_fft

137.13 brownian\_motion\_2d\_fft\_old

137.14 brownian\_motion\_2d\_fourier

137.15 brownian\_motion\_2d\_interleave

137.16 brownian\_motion\_2d\_interleaving

137.17 brownian\_motion\_2d\_kahunen

137.18 brownian\_motion\_2d\_laplacian

137.19 brownian\_motion\_with\_drift\_hitting\_probability

138 stochastic/geometric-ar1

138.1 geometric\_ar1\_2d\_generate

138.2 geometric\_ar1\_2d\_generate\_1

realization of the spatial geometric ornstein (geometric ar1)  
process  
averaged over grid cells

### 138.3 geometric\_ar1\_2d\_grid\_cell\_averaged\_cov

### 138.4 geometric\_ar1\_2d\_grid\_cell\_averaged\_generate

simulate a grid cell averaged stochastic process  $\exp(z)$   
where  $z$  follows a geometric Ornstein-Uhlenbeck (AR1) process  
with mean  $\text{lmu}$ , standard deviation  $\text{sd}$  and stationary autocorrelation

```
val = exp(z)
mean(z) = lmu
std(z) ) ls
corr(z(0,0),z(x,y)) = exp(-sqrt(x^2 + y^2)/theta)
```

### 138.5 geometric\_ar1\_2d\_grid\_cell\_averaged\_moment2par

### 138.6 geometric\_ar1\_2d\_grid\_cell\_averaged\_std

mean of the values  $\text{val}$   
covariance function of the values

## 139 stochastic

### 139.1 ornstein\_uhlenbeck\_cov

### 139.2 ornstein\_uhlenbeck\_mean

### 139.3 ornstein\_uhlenbeck\_spectral\_density

139.4 `ornstein_uhlenbeck_std`

## 140 `mathematics`

mathematical functions of various kind

140.1 `ternary_diagram`

## 141 `test/finance`

141.1 `test_gbb_mean`

141.2 `test_gbb_std`

141.3 `test_gbm_mean`

141.4 `test_gbm_mean_entire_series`

141.5 `test_gbm_moment2par`

141.6 `test_gbm_moment2par_entire_series`

141.7 test\_gbm\_std

141.8 test\_gbm\_std\_entire\_series

142 test/fourier

142.1 test\_fourier\_freq2ind

143 test/master

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

143.2 poisson2d\_blk

143.3 qr\_implicit\_givens\_2

143.4 spectral\_derivative\_2d

143.5 test\_2d\_eigensolver\_hydrogen

143.6 test\_2d\_refine

143.7 test\_3d\_eigensolver\_hydrogen

143.8 test\_FEM

143.9 test\_Mesh\_3d

143.10 test\_arnoldi

143.11 test\_arpackc

143.12 test\_assemble

143.13 test\_assembly\_performance

143.14 test\_bc\_one\_sided

143.15 test\_compare\_solvers

143.16 test\_complete



143.17 test\_convergence

143.18 test\_convergence\_b

143.19 test\_df\_2d

143.20 test\_eig\_algs

143.21 test\_eig\_inverse

143.22 test\_eigs\_lanczos

143.23 test\_eigs\_lanczos\_1

143.24 test\_eigs\_lanczos\_2

143.25 test\_eigs\_lanczos\_performance

143.26 test\_fdm

143.27 test\_fdm\_d\_vargrid

143.28 test\_fdm\_spectral

143.29 test\_fem

143.30 test\_fem\_1d

143.31 test\_fem\_1d\_higher\_order

143.32 test\_fem\_2d\_adaptive

143.33 test\_fem\_2d\_higher\_order

143.34 test\_fem\_3d\_higher\_order

143.35 test\_fem\_3d\_refine

143.36 test\_fem\_b

143.37 test\_fem\_derivative

143.38 test\_fem\_quadrature

143.39 test\_final

143.40 test\_fix\_substitution

143.41 test\_forward

143.42 test\_get\_sparse\_arrays

143.43 test\_harmonic\_oscillator

143.44 test\_high\_order\_fdm\_periodic\_bc

143.45 test\_hydrogen\_wf

143.46 test\_ichol

143.47 test\_interpolation

143.48 test\_inverse\_problem

143.49 test\_it\_vs\_exact

143.50 test\_jama

143.51 test\_jd

143.52 test\_jdqz

143.53 test\_lanczos\_2

143.54 test\_lanczos\_biorthogonal

143.55 test\_laplacian

143.56 test\_laplacian\_non\_uniform

143.57 test\_laplacian\_simple

143.58 test\_mesh\_2d\_uniform

143.59 test\_mesh\_2d\_uniform\_2

143.60 test\_mesh\_circle

143.61 test\_mesh\_generation

143.62 test\_mesh\_interpolate

143.63 test\_mg

143.64 test\_minres\_recycle

143.65 test\_multigrid

143.66 test\_nc

143.67 test\_nonuniform\_symmetric

143.68 test\_pde

143.69 test\_permutation

143.70 test\_poison\_fem

143.71 test\_polar

143.72 test\_potential

143.73 test\_powers

143.74 test\_precondition

143.75 test\_project\_rectangle

143.76 test\_qr

143.77 test\_quantum\_well

143.78 test\_radial\_adaptive

143.79 test\_radial\_confinement

143.80 test\_radial\_fixes

143.81 test\_refine\_2d

143.82 test\_refine\_2d\_b

143.83 test\_refine\_3d

143.84 test\_refine\_structural

143.85 test\_regularisation

143.86 test\_round\_off

143.87 test\_schrödinger\_potentials

143.88 test\_uniform\_mesh

143.89 test\_vargrid

144 test/numerical-methods/optimisation

144.1 test\_extreme3

145 test/numerical-methods

145.1 test\_advection\_kernel

146 test/signal-processing/autocorrelation

146.1 test\_acf

146.2 test\_acf\_bias

146.3 test\_acfar1\_2



146.4 test\_acfar1\_3

146.5 test\_acfar1\_4

146.6 test\_acfar2

146.7 test\_ar1\_var\_factor

146.8 test\_ar1\_var\_factor\_2

146.9 test\_ar1\_var\_mu\_single\_sample

146.10 test\_ar1\_var\_pop

146.11 test\_ar1\_var\_pop\_1

146.12 test\_ar1delay

146.13 test\_ar2

146.14 test\_phase\_drift\_acf

## 147 test/signal-processing/passes

147.1 test\_bandpass2d

147.2 test\_bandpass2d\_ideal

147.3 test\_lowpass1d\_fft

147.4 test\_lowpass1d\_implicit

147.5 test\_lowpass2d\_anisotropic

147.6 test\_lowpass2d\_fft

147.7 test\_lowpass2d\_rho

## 148 test/signal-processing/periodogram

148.1 test\_periodicity\_test\_2d

148.2 test\_periodogram\_bartlett\_se

148.3 test\_periodogram\_gauss

148.4 test\_periodogram\_radial

148.5 test\_periodogram\_test

148.6 test\_periodogram\_test\_periodicity\_2d

148.7 test\_periogogram\_significance

149 test/signal-processing/spectral-density

149.1 test\_phase\_drift\_parallel\_pdf

149.2 test\_phase\_drift\_parallel\_pdf\_mode2par

149.3 test\_phase\_drift\_pdf

149.4 test\_phase\_drift\_pdf\_2d

149.5 test\_phase\_drift\_pdf\_mode

149.6 test\_phase\_drift\_pdf\_mode2par

149.7 test\_phase\_drift\_pdf\_scale

149.8 test\_spectral\_density\_2

149.9 test\_spectral\_density\_bandpass\_2d

149.10 test\_spectral\_density\_bandpass\_2d\_max2par

149.11 test\_spectral\_density\_bandpass\_continuous

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

149.12 test\_spectral\_density\_bandpass\_continuous\_1

149.13 test\_spectral\_density\_bandpass\_maximum

- 149.14 `test_spectral_density_bandpass_scale`
- 149.15 `test_spectral_density_bp`
- 149.16 `test_spectral_density_bp_2d`
- 149.17 `test_spectral_density_bp_approx`
- 149.18 `test_spectral_density_flat`
- 149.19 `test_spectral_density_hp_cos`
- 149.20 `test_spectral_density_lorentzian_max`
- 149.21 `test_spectral_density_lorentzian_scale`
- 149.22 `test_spectral_density_lowpass`
- 149.23 `test_spectral_density_lowpass_continuous`

149.24 test\_spectral\_density\_lowpass\_continuous\_1

149.25 test\_spectral\_density\_maxiumum\_bias\_corrected

## 150 test/signal-processing

150.1 test\_autocorrelation\_max

150.2 test\_cdf\_bandpass\_continuous

150.3 test\_fit\_spectral\_density

150.4 test\_phase\_drift\_cdf

## 151 test/spatial-pattern-analysis

151.1 test\_approximate\_ratio\_distribution

151.2 test\_approximate\_ratio\_quantile

151.3 test\_separate\_isotropic\_density

## **152 test/spatial-statistics**

**152.1 test\_cov\_cell\_averages\_1d**

**152.2 test\_cov\_cell\_averages\_2d**

## **153 test/statistics/distributions/anisotropic**

**153.1 test\_anisotropic\_pattern**

**153.2 test\_anisotropic\_pattern\_pdf**

## **154 test/statistics/distributions/gamma**

**154.1 test\_generalized\_gamma\_mean**

## **155 test/statistics/distributions/log-uniform**

**155.1 test\_logurnd**

## **156 test/statistics/distributions/lognormal**

**156.1 test\_logn\_cov**

## **157 test/statistics/distributions/mises**

### **157.1 test\_mises\_std**

## **158 test/statistics/distributions/passes**

### **158.1 test\_bandpass2d\_pdf**

### **158.2 test\_bandpass2d\_pdf\_hankel**

### **158.3 test\_bandpass2d\_pdf\_mode**

### **158.4 test\_lowpass2d\_pdf\_hankel**

### **158.5 test\_lowpass2d\_pdf\_series**

## **159 test/statistics/distributions/skew-normal**

### **159.1 test\_skew\_generalized\_normpdf**

## **160 test/statistics/distributions**

### **160.1 test\_normpdf\_wrapped**



**161**    **test/statistics/distributions/weibull**

161.1    test\_wbl\_std

**162**    **test/statistics/moment-statistics**

162.1    test\_wmean

**163**    **test/statistics**

163.1    test\_fisher\_moment2par

163.2    test\_gamma\_mode

163.3    test\_normalize\_exponential\_random\_variable

**164**    **test/stochastic**

164.1    test\_brownian\_field

164.2    test\_brownian\_field\_scaled

**165**    **test/stochastics**

165.1    test\_brownian\_surface

**166**    **test**

**166.1**    **test\_S**

**166.2**    **test\_advect\_analytic**

**166.3**    **test\_asymp**

**166.4**    **test\_bandwidth**

**166.5**    **test\_bartlett\_angle**

**166.6**    **test\_bartlett\_distribution**

**166.7**    **test\_bartlett\_expansion**

**166.8**    **test\_beta**

**166.9**    **test\_betainc**

166.10 test\_bivariate\_covariance\_term

166.11 test\_brownian\_drift\_hitting\_probability

166.12 test\_brownian\_drift\_hitting\_probability2

166.13 test\_brownian\_motion\_1d

166.14 test\_brownian\_motion\_2d\_cov

166.15 test\_brownian\_motion\_2d\_fft

166.16 test\_brownian\_noise\_1d

166.17 test\_brownian\_noise\_2d

166.18 test\_brownian\_noise\_interleave

166.19 test\_coherence

166.20 test\_combined\_spectral\_density

166.21 test\_continuous\_fourier\_transform

166.22 test\_convexity

166.23 test\_d2

166.24 test\_determine\_phase\_shift

166.25 test\_diffuse\_analytic

166.26 test\_diffusion\_matrix

166.27 test\_ellipse

166.28 test\_error\_propagation\_fraction

166.29 test\_f

166.30 test\_f2

166.31 test\_fit\_2d\_spectral\_density

166.32 test\_fourier

166.33 test\_fourier\_derivative

166.34 test\_fourier\_derivative\_1

166.35 test\_fourier\_integral

166.36 test\_fourier\_mask\_covariance\_matrix

166.37 test\_ft\_bp

166.38 test\_gam

166.39 test\_gamma\_distribution

166.40 test\_gampdf\_man

166.41 test\_gaussfit3

166.42 test\_gaussian\_flat

166.43 test\_geoserr

166.44 test\_hexagonal\_pattern

166.45 test\_iafrate

166.46 test\_implicit\_ode

166.47 test\_imrotmat

166.48 test\_integration

166.49 test\_ivp

166.50 test\_jacobian

166.51 test\_lanczoswin

166.52 test\_laplacian\_power

166.53 test\_lognfit\_quantile

166.54 test\_ls\_perpendicular\_offset

166.55 test\_madcorr

166.56 test\_mask

166.57 test\_max\_normal

166.58 test\_moments

166.59 test\_moments\_fourier\_power

166.60 test\_mtimes3x3

166.61 test\_noisy\_oscillator

166.62 test\_nonperiodic\_pattern

166.63 test\_normaliztation

166.64 test\_ols

166.65 test\_parcorr

166.66 test\_positivity\_preserving

166.67 test\_randar1

166.68 test\_randar1\_multivariate

166.69 test\_randar2



166.70 test\_ratio\_distributions

166.71 test\_sd\_rectwin

166.72 test\_spatialrnd

166.73 test\_spectrum\_additivity

166.74 test\_stationarity

166.75 test\_stationarity2

166.76 test\_sum\_ij

166.77 test\_sum\_multivar

166.78 test\_trifilt1

166.79 test\_wautocorr

166.80 test\_wavelet\_transform

166.81 test\_whittle

166.82 test\_window

166.83 test\_wordfilt

166.84 test\_xar1\_mid\_term

## 167 mathematics

mathematical functions of various kind

167.1 trapezoidal\_fixed

## 168 wavelet

168.1 continuous\_wavelet\_transform

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

## 168.2 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

## 168.3 cwt\_man2

## 168.4 example\_wavelets

## 168.5 phasewrap

wrap the phase to +/- pi

## 168.6 test\_cwt\_man

## 168.7 test\_phasewrap

## 168.8 test\_wavelet

## 168.9 test\_wavelet2

## 168.10 test\_wavelet\_analysis

168.11 test\_wavelet\_reconstruct

168.12 test\_wtc

168.13 wavelet

wavelet windows

168.14 wavelet\_reconstruct

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

168.15 wavelet\_transform

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