

Manual for Package: mathematics

Revision 29M

Karl Kästner

July 5, 2023

Contents

1	calendar	47
1.1	days_per_month	47
1.2	isnight	48
2	mathematics	48
2.1	cast_byte_to_integer	48
3	complex-analysis	48
3.1	complex_exp_product_im_im	48
3.2	complex_exp_product_im_re	48
3.3	complex_exp_product_re_im	49
3.4	complex_exp_product_re_re	49
3.5	croots	49
3.6	root_complex	50
3.7	test_imroots	50
4	derivation	50
4.1	derive_acfar1	50
4.2	derive_ar1_spectral_density	50
4.3	derive_ar2param	50
4.4	derive_arc_length	50
4.5	derive_fourier_power	50
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	51
4.12	derive_spectral_density_bandpass_initial_condition	51

5	derivation/master	51
5.1	derive_bc_one_sided	51
5.2	derive_convergence	51
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	52
5.9	derive_fem_error_2d	52
5.10	derive_fem_error_3d	52
5.11	derive_fem_sym_2d	52
5.12	derive_grid_constants	52
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	53
5.19	derive_nonuniform_symmetric	53
5.20	derive_richardson	53
5.21	derive_sum	53
5.22	nn	53
5.23	test_derive	54
5.24	test_derive_fdm_poly	54
5.25	test_filter	54
5.26	test_vargrid	54
6	derivation	54
6.1	simplify_atan	54
7	mathematics	54
7.1	entropy	54
8	finance	54
8.1	derive_skewrnd_walsh_paramter	54
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	55
8.7	gbm_cdf	55
8.8	gbm_fit	55
8.9	gbm_fit_old	55

8.10	gbm_geomean	55
8.11	gbm_geostd	55
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	56
8.17	gbm_moment2par_entire_series	56
8.18	gbm_pdf	56
8.19	gbm_simulate	56
8.20	gbm_skewness	56
8.21	gbm_std	56
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	57
9	finance/test	57
9.1	test_gbm	57
9.2	test_gbm_pdf	57
9.3	test_skewrnd_walsh	57
10	fourier/@STFT	58
10.1	STFT	58
10.2	itransform	58
10.3	stft_	58
10.4	stftmat	58
10.5	transform	58
11	fourier	58
11.1	amplitude_from_peak	59
11.2	caesaro_weight	59
11.3	dftmtx_man	59
11.4	example_fourier_window	59
11.5	fft2_cartesian2radial	59
11.6	fft_man	59
11.7	fft_rotate	60
11.8	fftsmooth	60
11.9	fix_fourier	60
11.10	fourier_2d_padd	61
11.11	fourier_2d_quadrants	61
11.12	fourier_axis	61
11.13	fourier_axis_2d	61

11.14	fourier_cesaro_correction	61
11.15	fourier_coefficient_piecewise_linear	61
11.16	fourier_coefficient_piecewise_linear_1	62
11.17	fourier_coefficient_ramp3	62
11.18	fourier_coefficient_ramp_pulse	62
11.19	fourier_coefficient_ramp_step	62
11.20	fourier_coefficient_square_pulse	62
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	63
11.25	fourier_derivative_matrix_2d	63
11.26	fourier_expand	63
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	64
11.33	fourier_matrix_exp	64
11.34	fourier_multiplicative_interaction_coefficients	65
11.35	fourier_power	65
11.36	fourier_power_exp	65
11.37	fourier_predict	65
11.38	fourier_quadratic_interaction_coefficients	65
11.39	fourier_random_phase_walk	65
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	66
11.45	fourier_transform	66
11.46	fourier_transform_fractional	67
11.47	fourier_truncate_negative_half_plane	67
11.48	hyperbolic_fourier_box	67
11.49	idftmtx_man	67
11.50	laplace_2d_pwlinear	67
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	69
11.61	var_fourier_power	69
12	mathematics	69
12.1	gaussfit_quantile	69
13	geometry/@Geometry	70
13.1	Geometry	70
13.2	arclength	70
13.3	arclength_old	70
13.4	arclength_old2	70
13.5	base_point	70
13.6	base_point_limited	70
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	71
13.13	distance2	71
13.14	dot	71
13.15	edge_length	72
13.16	enclosed_angle	72
13.17	enclosing_triangle	72
13.18	hexagon	72
13.19	inPolygon	72
13.20	inTetra	72
13.21	inTetra2	72
13.22	inTriangle	72
13.23	intersect	73
13.24	lineintersect	73
13.25	lineintersect1	73
13.26	minimum_distance_lines	73
13.27	mittenpunkt	73
13.28	nagelpoint	73
13.29	onLine	73
13.30	orthocentre	73
13.31	plumb_line	74
13.32	poly_area	74
13.33	poly_edges	74
13.34	poly_set	74
13.35	poly_width	74

13.36	polyxpoly	74
13.37	project_to_curve	74
13.38	quad_isconvex	75
13.39	random_disk	75
13.40	random_simplex	75
13.41	sphere_volume	75
13.42	tetra_volume	75
13.43	tobarycentric	75
13.44	tobarycentric1	75
13.45	tobarycentric2	75
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	76
13.51	tri_edge_length	76
13.52	tri_edge_midpoint	76
13.53	tri_excircle	76
13.54	tri_height	77
13.55	tri_incircle	77
13.56	tri_isacute	77
13.57	tri_isobtuse	77
13.58	tri_semiperimeter	77
13.59	tri_side_length	77
14	geometry	77
14.1	Polygon	77
14.2	bounding_box	78
14.3	curvature_1d	78
14.4	cvt	78
14.5	deg_to_frac	78
14.6	ellipse	78
14.7	ellipseX	78
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	79
14.14	meanangle3	79
14.15	meanangle4	79
14.16	medianangle	80
14.17	medianangle2	80
14.18	pilim	80

14.19	streamline_radius_of_curvature	80
15	histogram/@Histogram	80
15.1	2x	80
15.2	Histogram	80
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	81
15.10	entropyS	81
15.11	export_csv	81
15.12	iquantile	81
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	82
15.20	mode	82
15.21	modeS	82
15.22	moment	82
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	83
15.30	skewnessS	83
15.31	stairs	83
15.32	stairsS	83
15.33	std	84
15.34	stdS	84
15.35	var	84
15.36	varS	84
16	histogram	84
16.1	hist_man	84
16.2	histadapt	84
16.3	histconst	84

16.4	pdf_poly	84
16.5	plotcdf	84
16.6	test_histogram	85
17	mathematics	85
17.1	imrotmat	85
18	linear-algebra	85
18.1	averaging_matrix_2	85
18.2	colnorm	85
18.3	condest_	85
18.4	connectivity_matrix	85
19	linear-algebra/coordinate-transformation	85
19.1	barycentric2cartesian	85
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	86
19.8	latlon2utm_simple	86
19.9	lowrance_mercator_to_wgs84	86
19.10	nmea2utm	87
19.11	sn2xy	87
19.12	unit_triangle_to_cartesian	87
19.13	utm2latlon	87
19.14	xy2nt	87
19.15	xy2sn	87
19.16	xy2sn_java	87
19.17	xy2sn_old	88
20	linear-algebra	88
20.1	det2x2	88
20.2	det3x3	88
20.3	det4x4	88
20.4	diag2x2	88
20.5	down	88
20.6	eig2x2	88
21	linear-algebra/eigenvalue	89
21.1	eig_bisection	89
21.2	eig_inverse	89
21.3	eig_inverse_iteration	89

21.4	eig_power_iteration	89
22	linear-algebra/eigenvalue/jacobi-davidson	89
22.1	afun_jdm	89
22.2	davidson	89
22.3	jacobi_davidson	89
22.4	jacobi_davidson_qr	89
22.5	jacobi_davidson_qz	89
22.6	jacobi_davidson_simple	90
22.7	jdqr	90
22.8	jdqr_sleijpen	93
22.9	jdqr_vorst	97
22.10	jdqz	99
22.11	mfunc_jdm	104
22.12	mgs	104
22.13	minres_	104
22.14	mv_jacobi_davidson	104
23	linear-algebra	104
23.1	first	104
23.2	gershgorin_circle	104
23.3	haussdorff	105
23.4	ieig2x2	105
23.5	inv2x2	105
23.6	inv3x3	105
23.7	inv4x4	105
23.8	kernel2matrix	105
24	linear-algebra/lanczos	105
24.1	arnoldi	105
24.2	arnoldi_new	106
24.3	eigs_lanczos_man	106
24.4	lanczos	106
24.5	lanczos_	106
24.6	lanczos_biorthogonal	106
24.7	lanczos_biorthogonal_improved	106
24.8	lanczos_ghep	106
24.9	mv_lanczos	106
24.10	reorthogonalise	106
24.11	test_lanczos	106
25	linear-algebra	107
25.1	laplacian_eigenvalue	107
25.2	laplacian_eigenvector	107

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

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

31.23	plot_wavefunction	116
32	master/ported	116
32.1	assemble_2d_phi_phi	116
32.2	assemble_3d_dphi_dphi	116
32.3	assemble_3d_phi_phi	117
32.4	dV_2d_	117
32.5	derivative_2d	117
32.6	derivative_3d	117
32.7	element_neighbour_2d	117
32.8	prefetch_2d_	117
32.9	promote_2d_3_10	117
32.10	promote_2d_3_15	117
32.11	promote_2d_3_21	117
32.12	promote_2d_3_6	117
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	118
32.17	vander_3d	118
33	mathematics	118
33.1	monotoneous_indices	118
33.2	nearest_fractional_timestep	118
34	number-theory	118
34.1	ceiln	118
34.2	digitsb	119
34.3	floorn	119
34.4	iseven	119
34.5	multichoosek	119
34.6	nchoosek_man	119
34.7	pythagorean_triple	119
34.8	roundn	119
35	numerical-methods	120
35.1	advect_analytic	120
36	numerical-methods/differentiation	120
36.1	derivative1	120
36.2	derivative2	120
37	numerical-methods	120
37.1	diffuse_analytic	120

38	numerical-methods/finite-difference	120
38.1	cdiff	120
38.2	cdiffb	120
38.3	central_difference	121
38.4	cmean	121
38.5	cmean2	121
38.6	derivative_matrix_1_1d	121
38.7	derivative_matrix_2_1d	121
38.8	derivative_matrix_2d	121
38.9	derivative_matrix_curvilinear	121
38.10	derivative_matrix_curvilinear_2	121
38.11	difference_kernel	122
38.12	diffusion_matrix_2d_anisotropic	122
38.13	diffusion_matrix_2d_anisotropic2	122
38.14	directional_neighbour	122
38.15	distmat	122
38.16	downwind_difference	122
38.17	gradpde2d	122
38.18	laplacian	122
38.19	laplacian_fdm	123
38.20	lrmean	123
39	numerical-methods/finite-difference/master	123
39.1	fdm_adaptive_grid	123
39.2	fdm_adaptive_refinement_old	123
39.3	fdm_assemble_d1_2d	123
39.4	fdm_assemble_d2_2d	123
39.5	fdm_confinement	123
39.6	fdm_d_vargrid	123
39.7	fdm_h_unstructured	123
39.8	fdm_hydrogen_vargrid	124
39.9	fdm_mark_unstructured_2d	124
39.10	fdm_plot	124
39.11	fdm_plot_series	124
39.12	fdm_refine_2d	124
39.13	fdm_refine_3d	124
39.14	fdm_refine_unstructured_2d	124
39.15	fdm_schroedinger_2d	124
39.16	fdm_schroedinger_3d	124
39.17	relocate	124
40	numerical-methods/finite-difference	125
40.1	mid	125
40.2	pwmid	125

40.3	ratio	125
40.4	steplength	125
40.5	swapoddeven	125
40.6	test_derivative_matrix_2d	125
40.7	test_derivative_matrix_curvilinear	125
40.8	test_difference_kernel	125
40.9	upwind_difference	126
41	numerical-methods/finite-element	126
41.1	Mesh_2d.java	126
41.2	Tree_2d.java	126
41.3	assemble_1d_dphi_dphi	126
41.4	assemble_1d_phi_phi	126
41.5	assemble_2d_dphi_dphi_java	126
41.6	assemble_2d_phi_phi_java	126
41.7	assemble_3d_dphi_dphi_java	126
41.8	assemble_3d_phi_phi_java	126
41.9	boundary_1d	127
41.10	boundary_2d	127
41.11	boundary_3d	127
41.12	check_area_2d	127
41.13	circmesh	127
41.14	cropradius	127
41.15	display_2d	127
41.16	display_3d	127
41.17	distort	127
41.18	err_2d	127
41.19	estimate_err_2d_3	128
41.20	example_1d	128
41.21	example_2d	128
41.22	explode	128
41.23	fem_2d	128
41.24	fem_2d_heuristic_mesh	128
41.25	fem_get_2d_radial	128
41.26	fem_interpolation	128
41.27	fem_plot_1d	128
41.28	fem_plot_1d_series	128
41.29	fem_plot_2d	129
41.30	fem_plot_2d_series	129
41.31	fem_plot_3d	129
41.32	fem_plot_3d_series	129
41.33	fem_plot_confine_series	129
41.34	fem_radial	129
41.35	flip_2d	129

41.36	get_mesh_arrays	129
41.37	hashkey	129
42	numerical-methods/finite-element/int	130
42.1	int_1d_equal	130
42.2	int_1d_equal_exp	130
42.3	int_1d_gauss	130
42.4	int_1d_gauss_1	130
42.5	int_1d_gauss_2	130
42.6	int_1d_gauss_3	130
42.7	int_1d_gauss_4	130
42.8	int_1d_gauss_5	130
42.9	int_1d_gauss_6	131
42.10	int_1d_gauss_lobatto	131
42.11	int_1d_gauss_n	131
42.12	int_1d_nc_2	131
42.13	int_1d_nc_3	131
42.14	int_1d_nc_4	131
42.15	int_1d_nc_5	131
42.16	int_1d_nc_6	131
42.17	int_1d_nc_7	131
42.18	int_1d_nc_7_hardy	131
42.19	int_2d_gauss_1	132
42.20	int_2d_gauss_12	132
42.21	int_2d_gauss_13	132
42.22	int_2d_gauss_16	132
42.23	int_2d_gauss_19	132
42.24	int_2d_gauss_25	132
42.25	int_2d_gauss_3	132
42.26	int_2d_gauss_33	132
42.27	int_2d_gauss_4	132
42.28	int_2d_gauss_6	132
42.29	int_2d_gauss_7	133
42.30	int_2d_gauss_9	133
42.31	int_2d_nc_10	133
42.32	int_2d_nc_15	133
42.33	int_2d_nc_21	133
42.34	int_2d_nc_3	133
42.35	int_2d_nc_6	133
42.36	int_3d_gauss_1	133
42.37	int_3d_gauss_11	133
42.38	int_3d_gauss_14	133
42.39	int_3d_gauss_15	134
42.40	int_3d_gauss_24	134

42.41	int_3d_gauss_4	134
42.42	int_3d_gauss_45	134
42.43	int_3d_gauss_5	134
42.44	int_3d_nc_11	134
42.45	int_3d_nc_4	134
42.46	int_3d_nc_6	134
42.47	int_3d_nc_8	134

43 numerical-methods/finite-element 135

43.1	interpolation_matrix	135
43.2	mark	135
43.3	mark_1d	135
43.4	mesh_1d_uniform	135
43.5	mesh_3d_uniform	135
43.6	mesh_interpolate	135
43.7	neighbour_1d	135
43.8	old	135
43.9	pdeeig_1d	135
43.10	pdeeig_2d	136
43.11	pdeeig_3d	136
43.12	polynomial_derivative_1d	136
43.13	potential_const	136
43.14	potential_coulomb	136
43.15	potential_harmonic_oscillator	136
43.16	project_circle	136
43.17	project_rectangle	136
43.18	promote_1d_2_3	136
43.19	promote_1d_2_4	136
43.20	promote_1d_2_5	137
43.21	promote_1d_2_6	137
43.22	quadrilaterate	137
43.23	recalculate_regularity_2d	137
43.24	refine_1d	137
43.25	refine_2d_21	137
43.26	refine_2d_structural	137
43.27	regularity_1d	137
43.28	regularity_2d	137
43.29	regularity_3d	138
43.30	relocate_2d	138
43.31	test_circmesh	138
43.32	test_hermite	138
43.33	tri_assign_points	138
43.34	triangulation_uniform	138
43.35	vander_1d	138

43.36	vanderd_1d	138
43.37	vanderi_1d	138
44	numerical-methods/finite-volume/@Advection	139
44.1	Advection	139
44.2	dot_advection	139
45	numerical-methods/finite-volume/@Burgers	139
45.1	burgers_split	139
45.2	dot_burgers_fdm	139
45.3	dot_burgers_fft	139
46	numerical-methods/finite-volume/@Finite_Volume	139
46.1	Finite_Volume	139
46.2	apply_bc	140
46.3	solve	140
46.4	step_split_strang	140
46.5	step_unsplit	140
47	numerical-methods/finite-volume/@Flux_Limiter	140
47.1	Flux_Limiter	140
47.2	beam_warming	140
47.3	fromm	141
47.4	lax_wendroff	141
47.5	minmod	141
47.6	monotized_central	141
47.7	muscl	141
47.8	superbee	141
47.9	upwind	141
47.10	vanLeer	142
48	numerical-methods/finite-volume/@KDV	142
48.1	dot_kdv_fdm	142
48.2	dot_kdv_fft	142
48.3	kdv_split	142
49	numerical-methods/finite-volume/@Reconstruct_Average_Evolve	142
49.1	Reconstruct_Average_Evolve	142
49.2	advect_highres	143
49.3	advect_lowress	143
50	numerical-methods/finite-volume	143
50.1	Godunov	143
50.2	Lax_Friedrich	143
50.3	Measure	143

50.4	Roe	143
50.5	fv_swe	144
50.6	staggered_euler	144
50.7	staggered_grid	144
51	numerical-methods	144
51.1	grid2quad	144
52	numerical-methods/integration	144
52.1	cumintL	144
52.2	cumintR	144
52.3	cumint_trapezoidal	144
52.4	int_1d_gauss_laguerre	145
52.5	int_trapezoidal	145
53	numerical-methods/interpolation/@Kriging	145
53.1	Kriging	145
53.2	estimate_semivariance	145
53.3	interpolate_	145
54	numerical-methods/interpolation/@RegularizedInterpolator1	146
54.1	RegularizedInterpolator1	146
54.2	init	146
55	numerical-methods/interpolation/@RegularizedInterpolator2	146
55.1	RegularizedInterpolator2	146
55.2	init	146
56	numerical-methods/interpolation/@RegularizedInterpolator3	146
56.1	RegularizedInterpolator3	146
56.2	init	146
57	numerical-methods/interpolation	147
57.1	IDW	147
57.2	IPoly	147
57.3	IRBM	147
57.4	ISparse	147
57.5	Inn	147
57.6	Interpolator	147
57.7	fixnan	147
57.8	idw1	148
57.9	idw2	148
57.10	inner2outer	148
57.11	inner2outer2	148
57.12	interp1_circular	148

57.13	interp1_limited	148
57.14	interp1_man	148
57.15	interp1_piecewise_linear	149
57.16	interp1_save	149
57.17	interp1_slope	149
57.18	interp1_smooth	149
57.19	interp1_unique	149
57.20	interp2_man	149
57.21	interp_angle	149
57.22	interp_fourier	150
57.23	interp_fourier_batch	150
57.24	interp_sn	150
57.25	interp_sn2	150
57.26	interp_sn3	150
57.27	interp_sn_	150
57.28	limit_by_distance_1d	150
57.29	resample1	151
57.30	resample_d_min	151
57.31	resample_vector	151
57.32	test_interp1_limited	151
58	numerical-methods	151
58.1	inverse_complex	151
58.2	maccormack_step	151
58.3	minmod	151
59	numerical-methods/multigrid	151
59.1	mg_interpolate	151
59.2	mg_restrict	152
60	numerical-methods/ode/@BVPS_Characteristic	152
60.1	BVPS_Characteristic	152
60.2	assemble1_A	152
60.3	assemble1_A_Q	152
60.4	assemble2_A	152
60.5	assemble_AA	152
60.6	assemble_AAA	152
60.7	assemble_Ic	153
60.8	bvp1c	153
60.9	check_arguments	153
60.10	couple_junctions	153
60.11	derivative	153
60.12	init	153
60.13	inner2outer_bvp2c	153

60.14	reconstruct	153
60.15	resample	153
60.16	solve	154
60.17	test_assemble1_A	154
60.18	test_assemble2_A	154
61	numerical-methods/ode/@Time_Stepper	154
61.1	Time_Stepper	154
61.2	solve	154
62	numerical-methods/ode	154
62.1	bvp2fdm	154
62.2	bvp2wavetrain	155
62.3	bvp2wavetwopass	155
62.4	ivp_euler_forward	155
62.5	ivp_euler_forward2	155
62.6	ivprk2	155
62.7	ode2_matrix	156
62.8	ode2characteristic	156
62.9	step_trapezoidal	156
62.10	test_bvp2	156
63	numerical-methods/optimisation	156
63.1	aitken_iteration	156
63.2	anderson_iteration	156
63.3	armijo_stopping_criterion	156
63.4	astar	157
63.5	binsearch	157
63.6	bisection	157
63.7	box1	157
63.8	box2	157
63.9	cauchy	157
63.10	cauchy2	157
63.11	directional_derivative	158
63.12	dud	158
63.13	extreme3	158
63.14	extreme_quadratic	158
63.15	ftest	158
63.16	fzero_bisect	158
63.17	fzero_newton	159
63.18	grad	159
63.19	hessian	159
63.20	hessian_from_gradient	159
63.21	hessian_projected	159

63.22	line_search	159
63.23	line_search2	159
63.24	line_search_polynomial	160
63.25	line_search_polynomial2	160
63.26	line_search_quadratic	160
63.27	line_search_quadratic2	160
63.28	line_search_wolfe	161
63.29	ls_bgfs	161
63.30	ls_broyden	161
63.31	ls_generalized_secant	161
63.32	nlcg	161
63.33	nlls	162
63.34	picard	162
63.35	poly_extrema	162
63.36	quadratic_function	162
63.37	quadratic_programming	162
63.38	quadratic_step	162
63.39	rosenbrock	162
63.40	sqrt_heron	163
63.41	test_directional_derivative	163
63.42	test_dud	163
63.43	test_fzero_newton	163
63.44	test_line_search_quadratic2	163
63.45	test_ls_generalized_secant	163
63.46	test_nlcg_6_order	163
63.47	test_nlls	163
64 numerical-methods/pde		163
64.1	laplacian2d_fundamental_solution	163
65 numerical-methods/piecewise-polynomials		164
65.1	Hermite1	164
65.2	hp2_fit	164
65.3	hp2_predict	164
65.4	hp_predict	164
65.5	hp_regress	164
65.6	lp_count	164
65.7	lp_predict	165
65.8	lp_regress	165
65.9	lp_regress_	165
66 numerical-methods		165
66.1	test_adams_bashforth	165

67	mathematics	165
67.1	oversampleNZ	165
68	pdes	165
68.1	heat_equation_fundamental_solution	165
68.2	heat_equation_width	165
68.3	heat_equation_width_to_time	166
69	regression/@PolyOLS	166
69.1	PolyOLS	166
69.2	coefftest	166
69.3	detrend	166
69.4	fit	166
69.5	fit_	166
69.6	predict	166
69.7	predict_	166
69.8	slope	167
70	regression/@PowerLS	167
70.1	PowerLS	167
70.2	fit	167
70.3	predict	167
70.4	predict_	167
71	regression/@Theil	167
71.1	Theil	167
71.2	detrend	167
71.3	fit	168
71.4	predict	168
71.5	slope	168
72	regression	168
72.1	Theil_Multivariate	168
72.2	areg	168
72.3	ginireg	168
72.4	hesssimplereg	169
72.5	l1lin	169
72.6	lsq_sparam	169
72.7	polyfitd	169
72.8	regression_method_of_moments	169
72.9	robustlinreg	169
72.10	theil2	170
72.11	theil_generalised	170
72.12	total_least_squares	170

72.13	weighted_median_regression	170
73	set-theory	170
73.1	issubset	170
74	mathematics	170
74.1	shuffle_index	171
75	signal-processing	171
75.1	asymwin	171
76	signal-processing/autocorrelation	171
76.1	acf_radial	171
76.2	acfar1	171
76.3	acfar1_2	171
76.4	acfar2	171
76.5	acfar2_2	171
76.6	ar1_cutoff_frequency	172
76.7	ar1_effective_sample_size	172
76.8	ar1_mse_mu_single_sample	172
76.9	ar1_mse_pop	172
76.10	ar1_mse_range	172
76.11	ar1_spectrum	172
76.12	ar1_to_tikhonov	172
76.13	ar1_var_factor	173
76.14	ar1_var_factor_	173
76.15	ar1_var_range2	173
76.16	ar1delay	173
76.17	ar1delay_old	173
76.18	ar2_acf2c	173
76.19	ar2conv	174
76.20	ar2dof	174
76.21	ar2param	174
76.22	autocorr2	174
76.23	autocorr_angular	174
76.24	autocorr_bandpass	174
76.25	autocorr_decay_rate	174
76.26	autocorr_effective_sample_size	174
76.27	autocorr_fft	175
76.28	autocorr_forest	175
76.29	autocorr_genton	175
76.30	autocorr_highpass	175
76.31	autocorr_lowpass	175
76.32	autocorr_periodic_additive_noise	175

76.33	autocorr_periodic_windowed	175
76.34	autocorr_radial	175
76.35	autocorr_radial_hexagonal_pattern	175
76.36	autocorrelation_max	176
77	signal-processing	176
77.1	average_wave_shape	176
77.2	bandpass	176
77.3	bandpass_continuous_cdf	176
77.4	bartlett	176
77.5	bin1d	176
77.6	bin2d	177
77.7	binormrnd	177
77.8	coherence	177
77.9	conv1_man	177
77.10	conv2_man	177
77.11	conv2z	177
77.12	conv30	177
77.13	conv_	177
77.14	conv_centered	178
77.15	convz	178
77.16	cosexpdelay	178
77.17	csmooth	178
77.18	daniell_window	178
77.19	db2neper	178
77.20	db2power	178
77.21	derive_bandpass_continuous_scale	179
77.22	derive_danielle_weight	179
77.23	derive_limit_0_acfar	179
77.24	detect_peak	179
77.25	determine_phase_shift	179
77.26	determine_phase_shift1	179
77.27	doublesum_ij	179
77.28	effective_mask_size	179
77.29	effective_sample_size_to_ar1	179
77.30	fcut2Lw_gausswin	180
77.31	fcut_gausswin	180
77.32	filt_hodges_lehman	180
78	signal-processing/filters	180
78.1	circfilt2	180
78.2	filter1	180
78.3	filter2	180
78.4	filter_	180

78.5	filter_r_to_f0	180
78.6	filter_rho_to_f0	181
78.7	filter_twosided	181
78.8	filteriir	181
78.9	filterp	182
78.10	filterp1	182
78.11	filterstd	182
78.12	gaussfilt2	182
78.13	lowpass_discrete	182
78.14	meanfilt2	182
78.15	medfilt1_man	182
78.16	medfilt1_man2	182
78.17	medfilt1_padded	183
78.18	medfilt1_reduced	183
78.19	trifilt1	183
78.20	trifilt2	183
79 signal-processing		183
79.1	firls_man	183
79.2	fit_spectral_density	183
79.3	fit_spectral_density_2d	183
79.4	fit_spectral_density_radial	184
79.5	flattopwin	184
79.6	frequency_response_boxcar	184
79.7	freqz_boxcar	184
79.8	gaussfilt1	184
79.9	hanchangewin	184
79.10	hanchangewin2	184
79.11	hanwin	184
79.12	hanwin_	185
79.13	high_pass_1d_simple	185
79.14	kaiserwin	185
79.15	kalman	185
79.16	lanczoswin	185
79.17	last	185
79.18	maxfilt1	185
79.19	meanfilt1	185
79.20	mid_term_single_sample	186
79.21	minfilt1	186
79.22	minmax	186
79.23	mu2ar1	186
79.24	mysmooth	186
79.25	nanautocorr	186
79.26	nanmedfilt1	186

79.27	neper2db	186
79.28	oscillator_noisy	187
80	signal-processing/passes	187
80.1	bandpass1d	187
80.2	bandpass1d_fft	187
80.3	bandpass1d_implicit	187
80.4	bandpass2	187
80.5	bandpass2d	187
80.6	bandpass2d_convolution	187
80.7	bandpass2d_fft	187
80.8	bandpass2d_ideal	187
80.9	bandpass2d_implicit	188
80.10	bandpass2d_iso	188
80.11	bandpass_arg	188
80.12	bandpass_f0_to_rho	188
80.13	bandpass_max	188
80.14	bandpass_max2	188
80.15	highpass	188
80.16	highpass1d_fft_cos	188
80.17	highpass1d_implicit	189
80.18	highpass2d_fft	189
80.19	highpass2d_ideal	189
80.20	highpass2d_implicit	189
80.21	highpass_arg	189
80.22	highpass_fc_to_rho	189
80.23	lowpass	189
80.24	lowpass1d_fft	189
80.25	lowpass1d_implicit	189
80.26	lowpass2	190
80.27	lowpass2d_anisotropic	190
80.28	lowpass2d_convolution	190
80.29	lowpass2d_fft	190
80.30	lowpass2d_ideal	190
80.31	lowpass2d_implicit	190
80.32	lowpass_arg	190
80.33	lowpass_fc_to_rho	191
80.34	lowpass_iir	191
80.35	lowpass_iir_symmetric	191
80.36	lowpassfilter2	191
81	signal-processing	191
81.1	peaks_man	191

82	signal-processing/periodogram	191
82.1	periodogram	191
82.2	periodogram_2d	191
82.3	periodogram_align	191
82.4	periodogram_angular	192
82.5	periodogram_bartlett	192
82.6	periodogram_bootstrap	192
82.7	periodogram_confidence_interval	192
82.8	periodogram_filter	192
82.9	periodogram_median	192
82.10	periodogram_normalize	192
82.11	periodogram_normalize_2d	192
82.12	periodogram_p_value	193
82.13	periodogram_qq	193
82.14	periodogram_quantiles	193
82.15	periodogram_radial	193
82.16	periodogram_std	194
82.17	periodogram_test_periodicity	194
82.18	periodogram_test_periodicity_2d	195
82.19	periodogram_test_stationarity	195
82.20	periodogram_welsh	196
83	signal-processing	196
83.1	polyfilt1	196
83.2	qmedfilt1	196
83.3	quadratfilt1	196
83.4	quadratwin	196
83.5	randar1	196
83.6	randar1_dual	196
83.7	randar2	197
83.8	randarp	197
83.9	rectwin	197
83.10	recursive_sum	197
83.11	select_range	197
83.12	smooth1d_parametric	197
83.13	smooth2	197
83.14	smooth_man	197
83.15	smooth_parametric	198
83.16	smooth_parametric2	198
83.17	smooth_with_splines	198
83.18	smoothfft	198
84	signal-processing/spectral-density	198
84.1	hex_angular_pdf	198

84.2	hex_angular_pdf_max	198
84.3	hex_angular_pdf_max2par	198
84.4	spectral_density_ar2	198
84.5	spectral_density_area	199
84.6	spectral_density_estimate_2d	199
84.7	spectral_density_flat	199
84.8	spectral_density_forest	199
84.9	spectral_density_gausswin	199
84.10	spectral_density_lorentzian	199
84.11	spectral_density_lorentzian_max	199
84.12	spectral_density_lorentzian_max2par	199
84.13	spectral_density_lorentzian_scale	200
84.14	spectral_density_maximum_bias_corrected	200
84.15	spectral_density_periodic_additive_noise	200
84.16	spectral_density_rectwin	200
84.17	spectral_density_wperiodic	200
85	signal-processing	200
85.1	spectrogram	200
85.2	sum_i_lag	200
85.3	sum_ii	200
85.4	sum_ii_	201
85.5	sum_ij	201
85.6	sum_ij_	201
85.7	sum_ij_partial_	201
85.8	sum_multivar	201
85.9	test_acfar1	201
85.10	tikhonov_to_ar1	201
85.11	trapwin	201
85.12	triwin	202
85.13	triwin2	202
85.14	tukeywin_man	202
85.15	varar1	202
85.16	welch_spectrogram	202
85.17	wfilt	202
85.18	winbandpass	202
86	signal-processing/windows	202
86.1	circwin	202
86.2	danielle_window	203
86.3	gausswin	203
86.4	gausswin1	203
86.5	gausswin2	203
86.6	radial_window	203

86.7	range_window	203
86.8	rectwin_cutoff_frequency	203
86.9	std_window	203
86.10	window2d	203
86.11	window_make_odd	204
87	signal-processing	204
87.1	winfilt0	204
87.2	wavelength	204
87.3	wmeanfilt	204
87.4	wmedfilt	204
87.5	wordfilt	204
87.6	wordfilt_edgeworth	204
87.7	wrapphase	205
87.8	xar1	205
87.9	xcorr_man	205
88	sorting	205
88.1	sort2	205
88.2	sort2d	205
89	spatial-pattern-analysis/@Spatial_Pattern	205
89.1	Spatial_Pattern	205
89.2	analyze_grid	205
89.3	analyze_transect	206
89.4	clear_1d_properties	206
89.5	clear_2d_properties	206
89.6	fit_parametric_densities	206
89.7	imread	206
89.8	plot	206
89.9	plot_transect	206
89.10	prepare_analysis	206
89.11	report	207
89.12	resample_functions	207
89.13	tabulate	207
90	spatial-pattern-analysis/@Spatial_Pattern_Array	207
90.1	Spatial_Pattern_Array	207
90.2	analyze	207
90.3	assign_regions	207
90.4	export_shp	207
90.5	fetch	207
90.6	generate_filename	208
90.7	quality_check	208

91 spatial-pattern-analysis	208
91.1 approximate_ratio_distribution	208
91.2 banded_pattern	208
91.3 hexagonal_pattern	209
91.4 patch_size_1d	209
91.5 patch_size_2d	209
91.6 pattern_isotropic_rotated	209
91.7 reconstruct_isotropic_density	209
91.8 separate_isotropic_from_anisotropic_density	209
91.9 suppress_low_frequency_lobe	209
92 spatial-statistics	210
92.1 cov_cell_averages_1d	210
92.2 cov_cell_averages_2d	210
93 special-functions	211
93.1 bessell_sphere	211
93.2 bessellln_large_x	211
93.3 beta_man	211
93.4 betainc_man	211
93.5 digamma_man	211
93.6 exp10	211
93.7 hankel_sphere	211
93.8 hermite	211
93.9 laguerre_roots	212
93.10 lambertw_numeric	212
93.11 legendre_man	212
93.12 neumann_sphere	212
94 statistics	212
94.1 atan_s2	212
94.2 binomial	212
95 statistics/circular	213
95.1 circular_fmoment	213
95.2 circular_fquantile	213
95.3 circular_fstd	213
95.4 circular_fvar	213
96 statistics	213
96.1 coefficient_of_determination	213
96.2 conditional_expectation_normal	213
96.3 correlation_confidence_pearson	213
97 statistics/distributions	213

97.1	PDF	213
98	statistics/distributions/anisotropic	214
98.1	anisotropic_pattern	214
98.2	anisotropic_pattern_acf	214
98.3	anisotropic_pattern_pdf	214
99	statistics/distributions/beta	214
99.1	beta_kurt	214
99.2	beta_mean	214
99.3	beta_moment2par	214
99.4	beta_skew	214
99.5	beta_std	214
100	statistics/distributions/bivariate-normal	215
100.1	binorm_separation_coefficient	215
100.2	binormcdf	215
100.3	binormfit	215
100.4	binormpdf	215
101	statistics/distributions/chi2	215
101.1	chi2_kurt	215
101.2	chi2_mean	215
101.3	chi2_skew	215
101.4	chi2_std	215
102	statistics/distributions/circular-normal	216
102.1	wnormpdf	216
103	statistics/distributions/edgeworth	216
103.1	edgeworth_cdf	216
103.2	edgeworth_pdf	216
104	statistics/distributions/exp	216
104.1	exppdf_max2par	216
105	statistics/distributions/fisher	216
105.1	fisher_mean	216
105.2	fisher_moment2par	216
105.3	fisher_std	217
106	statistics/distributions/gamma	217
106.1	gamma_mean	217
106.2	gamma_mode	217
106.3	gamma_mode2par	217

106.4	gamma_moment2par	217
106.5	gamma_std	217
106.6	gamma_stirling	217
106.7	gampdf_man	217
106.8	generalized_gamma_mean	217
107	statistics/distributions/hotelling-t2	218
107.1	t2cdf	218
107.2	t2inv	218
108	statistics/distributions/kurt-normal	218
108.1	kurtncdf	218
108.2	kurtnpdf	218
109	statistics/distributions/log-triangular	218
109.1	logtrialtcd	218
109.2	logtrialtinv	218
109.3	logtrialtmean	219
109.4	logtrialtpdf	219
109.5	logtrialtrnd	219
109.6	logtricdf	219
109.7	logtriinv	219
109.8	logtrimean	219
109.9	logtripdf	219
109.10	logtrirnd	220
110	statistics/distributions/log-uniform	220
110.1	logu_median	220
110.2	logucdf	220
110.3	logucm	220
110.4	loguinv	220
110.5	logumean	220
110.6	logupdf	220
110.7	logurnd	220
110.8	loguvar	221
111	statistics/distributions/loglog	221
111.1	loglogpdf	221
112	statistics/distributions/lognormal	221
112.1	logn_corr	221
112.2	logn_cov	221
112.3	logn_mean	221
112.4	logn_mode	221
112.5	logn_mode2par	221

112.6	logn_moment2par	222
112.7	logn_moment2par_correlated	222
112.8	logn_param2moment	222
112.9	logn_skewness	222
112.10	logn_std	222
112.11	lognpdf_	222
112.12	lognpdf_entropy	222
113	statistics/distributions/logskew	222
113.1	logskewcdf	222
113.2	logskewpdf	223
114	statistics/distributions/mises	223
114.1	mises_max2par	223
114.2	mises_std	223
114.3	mises_var	223
114.4	misesn_max2par	223
114.5	misesnpdf	223
114.6	misespdf	223
115	statistics/distributions	223
115.1	ncx2_moment2par	223
116	statistics/distributions/normal	224
116.1	normpdf_entropy	224
116.2	normpdf_mode	224
116.3	normpdf_mode2par	224
117	statistics/distributions/passes	224
117.1	bandpass1d_continuous_pdf	224
117.2	bandpass1d_continuous_pdf_max	224
117.3	bandpass1d_continuous_pdf_max2par	225
117.4	bandpass1d_continuous_pdf_scale	225
117.5	bandpass1d_discrete_pdf	225
117.6	bandpass2d_discrete_pdf	225
117.7	bandpass2d_pdf_exact	225
117.8	bandpass2d_pdf_hankel	225
117.9	bandpass2d_pdf_mode	225
117.10	bandpass2d_pdf_mode2par	225
117.11	bandpass2d_pdf_scale	226
117.12	highpass1d_continuous_pdf	226
117.13	highpass1d_discrete_cos_pdf	226
117.14	highpass1d_disrete_pdf	226
117.15	highpass2d_discrete_pdf	227

117.16	highpass2d_pdf	227
117.17	highpass2d_pdf_hankel	227
117.18	lowpass1d_continuous_pdf	227
117.19	lowpass1d_continuous_pdf_scale	227
117.20	lowpass1d_discrete_pdf	227
117.21	lowpass1d_one_sided_pdf	227
117.22	lowpass2d_discrete_acf	227
117.23	lowpass2d_discrete_pdf	227
117.24	lowpass2d_pdf	228
117.25	lowpass2d_pdf_hankel	228
117.26	lowpass2d_pdf_series	228
118	statistics/distributions	228
118.1	pdfsample	228
119	statistics/distributions/phase-drift	228
119.1	phase_drift_acf	228
119.2	phase_drift_acf_2d	228
119.3	phase_drift_cdf	229
119.4	phase_drift_inv	229
119.5	phase_drift_parallel_acf	229
119.6	phase_drift_parallel_pdf	229
119.7	phase_drift_parallel_pdf_max	229
119.8	phase_drift_parallel_pdf_max2par	229
119.9	phase_drift_parallel_pdf_mode2par	229
119.10	phase_drift_patch_size_distribution	229
119.11	phase_drift_pdf	229
119.12	phase_drift_pdf_2d	230
119.13	phase_drift_pdf_mode	230
119.14	phase_drift_pdf_mode2par	230
119.15	phase_drift_pdf_reg2par	230
119.16	phase_drift_pdf_scale	230
120	statistics/distributions/skew-normal	230
120.1	skew_generalized_normal_fit	230
120.2	skew_generalized_normpdf	230
120.3	skewcdf	230
120.4	skewparam_to_central_moments	231
120.5	skewpdf	231
120.6	skewpdf_entropy	231
121	statistics/distributions/triangular	231
121.1	triedf	231
121.2	triinv	231

121.3	trimedial	231
121.4	tripdf	231
121.5	trirnd	231
122	statistics/distributions/weibull	232
122.1	wbl_std	232
123	statistics/distributions/wrapped-normal	232
123.1	normpdf_wrapped	232
123.2	normpdf_wrapped_mode	232
123.3	normpdf_wrapped_mode2par	232
124	statistics	232
124.1	error_propagation_fraction	232
124.2	error_propagation_product	232
124.3	example_standard_error_of_sample_quantiles	232
124.4	f_var_finite	232
124.5	gaussfit3	233
124.6	gaussfit_quantile	233
124.7	geoserr	233
124.8	geostd	233
124.9	hodges_lehmann_correlation	233
124.10	hodges_lehmann_dispersion	233
125	statistics/information-theory	234
125.1	akaike_information_criterion	234
125.2	bayesian_information_criterion	234
126	statistics	234
126.1	jackknife_block	234
126.2	kurtosis_bias_corrected	234
126.3	limit	234
126.4	logfactorial	234
126.5	lognfit_quantile	234
126.6	max_exprnd	235
126.7	maxnormals	235
126.8	mean_angle	235
126.9	mean_max_n	235
126.10	mean_min_n	235
126.11	midrange	235
126.12	minavg	235
126.13	mode_man	235
127	statistics/moment-statistics	236
127.1	autocorr_man3	236

127.2	autocorr_man4	236
127.3	autocorr_man5	236
127.4	blockserr	236
127.5	comoment	236
127.6	corr_man	237
127.7	cov_man	237
127.8	dof	237
127.9	edgeworth_quantile	237
127.10	effective_sample_size	237
127.11	f_correlation	237
127.12	f_finite	238
127.13	lmean	238
127.14	lmoment	238
127.15	maskmean	238
127.16	masknanmean	238
127.17	mean1	238
127.18	mean_man	238
127.19	mse	238
127.20	nanautocorr_man1	239
127.21	nanautocorr_man2	239
127.22	nanautocorr_man4	239
127.23	nancorr	239
127.24	nancumsum	239
127.25	nanlmean	239
127.26	nanr2	239
127.27	nanrms	240
127.28	nanrmse	240
127.29	nanserr	240
127.30	nanwmean	240
127.31	nanwstd	240
127.32	nanwvar	240
127.33	nanxcorr	240
127.34	pearson	241
127.35	pearson_to_kendall	241
127.36	pool_samples	241
127.37	qmean	241
127.38	range_mean	241
127.39	rmse_	241
127.40	serr	241
127.41	serr1	241
127.42	test_qskew	242
127.43	test_qstd_qskew_optimal_p	242
127.44	wautocorr	242
127.45	wcorr	242

127.46	wcov	242
127.47	wdof	242
127.48	wkurt	242
127.49	wmean	243
127.50	wrms	243
127.51	wserr	243
127.52	wskew	243
127.53	wstd	243
127.54	wvar	243
128	statistics	244
128.1	nangeomean	244
128.2	nangeostd	244
129	statistics/nonparametric-statistics	244
129.1	kernel1d	244
129.2	kernel2d	244
130	statistics	244
130.1	normalize_exponential_random_variable	244
130.2	normmoment	244
130.3	normpdf2	245
131	statistics/order-statistics	245
131.1	hodges_lehmann_location	245
131.2	kendall	245
131.3	kendall_to_pearson	245
131.4	mad2sd	245
131.5	madcorr	245
131.6	median2_holder	246
131.7	median_ci	246
131.8	median_man	246
131.9	mediani	246
131.10	nanmadcorr	246
131.11	nanwmedian	246
131.12	nanwquantile	246
131.13	oja_median	247
131.14	qkurtosis	247
131.15	qmoments	247
131.16	qskew	247
131.17	qskewq	247
131.18	qstdq	248
131.19	quantile1_optimisation	248
131.20	quantile2_breckling	248

131.21	quantile2_chaudhuri	248
131.22	quantile2_projected	248
131.23	quantile2_projected2	248
131.24	quantile_envelope	248
131.25	quantile_regression_simple	248
131.26	ranking	249
131.27	spatial_median	249
131.28	spatial_quantile	249
131.29	spatial_quantile2	249
131.30	spatial_quantile3	249
131.31	spatial_rank	249
131.32	spatial_sign	249
131.33	spatial_signed_rank	249
131.34	spearman	250
131.35	spearman_rank	250
131.36	spearman_to_pearson	250
131.37	wmedian	250
131.38	wquantile	250
132	statistics	250
132.1	qstd	250
132.2	quantile_extrap	250
132.3	quantile_sin	250
133	statistics/random-number-generation	251
133.1	laplacernd	251
133.2	randc	251
133.3	skewness2param	251
133.4	skewpdf_central_moments	251
133.5	skewrnd	251
134	statistics	251
134.1	range	251
134.2	resample_with_replacement	251
135	statistics/resampling-statistics/@Jackknife	252
135.1	Jackknife	252
135.2	estimated_STATIC	252
135.3	matrix1_STATIC	252
135.4	matrix2	252
136	statistics/resampling-statistics	253
136.1	block_jackknife	253
136.2	jackknife_moments	253

136.3	moving_block_jackknife	253
136.4	randblockserr	253
136.5	resample	253
137	statistics	254
137.1	scale_quantile_sd	254
137.2	sd_sample_quantiles	254
137.3	spatialrnd	254
137.4	trimmed_mean	254
137.5	ttest2_man	254
137.6	ttest_man	254
137.7	ttest_paired	255
137.8	uniformnpdf	255
137.9	wgeomean	255
137.10	wgeovar	255
137.11	wharmean	255
137.12	wharstd	255
137.13	wharvar	255
138	stochastic	255
138.1	brownian_drift_hitting_probability	255
138.2	brownian_drift_hitting_probability2	256
138.3	brownian_field	256
138.4	brownian_field_scaled	256
138.5	brownian_motion_1d_acf	256
138.6	brownian_motion_1d_cov	256
138.7	brownian_motion_1d_fft	256
138.8	brownian_motion_1d_fourier	256
138.9	brownian_motion_1d_interleave	256
138.10	brownian_motion_1d_laplacian	257
138.11	brownian_motion_2d_cov	257
138.12	brownian_motion_2d_fft	257
138.13	brownian_motion_2d_fft_old	257
138.14	brownian_motion_2d_fourier	257
138.15	brownian_motion_2d_interleave	257
138.16	brownian_motion_2d_interleaving	257
138.17	brownian_motion_2d_kahunen	257
138.18	brownian_motion_2d_laplacian	257
138.19	brownian_motion_with_drift_hitting_probability	257
139	stochastic/geometric-ar1	258
139.1	geometric_ar1_2d_generate	258
139.2	geometric_ar1_2d_generate_1	258
139.3	geometric_ar1_2d_grid_cell_averaged_cov	258

139.4	geometric_ar1_2d_grid_cell_averaged_generate	258
139.5	geometric_ar1_2d_grid_cell_averaged_moment2par	258
139.6	geometric_ar1_2d_grid_cell_averaged_std	258
140	stochastic	258
140.1	ornstein_uhlenbeck_cov	258
140.2	ornstein_uhlenbeck_mean	259
140.3	ornstein_uhlenbeck_spectral_density	259
140.4	ornstein_uhlenbeck_std	259
141	mathematics	259
141.1	ternary_diagram	259
142	test/finance	259
142.1	test_gbb_mean	259
142.2	test_gbb_std	259
142.3	test_gbm_mean	259
142.4	test_gbm_mean_entire_series	259
142.5	test_gbm_moment2par	260
142.6	test_gbm_moment2par_entire_series	260
142.7	test_gbm_std	260
142.8	test_gbm_std_entire_series	260
143	test/fourier	260
143.1	test_fourier_freq2ind	260
144	test/master	260
144.1	dat_test_lanczos_3d_k_20_n_40	260
144.2	poisson2d_blk	260
144.3	qr_implicit_givens_2	260
144.4	spectral_derivative_2d	260
144.5	test_2d_eigensolver_hydrogen	261
144.6	test_2d_refine	261
144.7	test_3d_eigensolver_hydrogen	261
144.8	test_FEM	261
144.9	test_Mesh_3d	261
144.10	test_arnoldi	261
144.11	test_arpackc	261
144.12	test_assemble	261
144.13	test_assembly_performance	261
144.14	test_bc_one_sided	261
144.15	test_compare_solvers	262
144.16	test_complete	262
144.17	test_convergence	262

144.18	test_convergence_b	262
144.19	test_df_2d	262
144.20	test_eig_algs	262
144.21	test_eig_inverse	262
144.22	test_eigs_lanczos	262
144.23	test_eigs_lanczos_1	262
144.24	test_eigs_lanczos_2	262
144.25	test_eigs_lanczos_performance	263
144.26	test_fdm	263
144.27	test_fdm_d_vargrid	263
144.28	test_fdm_spectral	263
144.29	test_fem	263
144.30	test_fem_1d	263
144.31	test_fem_1d_higher_order	263
144.32	test_fem_2d_adaptive	263
144.33	test_fem_2d_higher_order	263
144.34	test_fem_3d_higher_order	263
144.35	test_fem_3d_refine	264
144.36	test_fem_b	264
144.37	test_fem_derivative	264
144.38	test_fem_quadrature	264
144.39	test_final	264
144.40	test_fix_substitution	264
144.41	test_forward	264
144.42	test_get_sparse_arrays	264
144.43	test_harmonic_oscillator	264
144.44	test_high_order_fdm_periodic_bc	264
144.45	test_hydrogen_wf	265
144.46	test_ichol	265
144.47	test_interpolation	265
144.48	test_inverse_problem	265
144.49	test_it_vs_exact	265
144.50	test_jama	265
144.51	test_jd	265
144.52	test_jdqz	265
144.53	test_lanczos_2	265
144.54	test_lanczos_biorthogonal	265
144.55	test_laplacian	266
144.56	test_laplacian_non_uniform	266
144.57	test_laplacian_simple	266
144.58	test_mesh_2d_uniform	266
144.59	test_mesh_2d_uniform_2	266
144.60	test_mesh_circle	266
144.61	test_mesh_generation	266

144.62	test_mesh_interpolate	266
144.63	test_mg	266
144.64	test_minres_recycle	266
144.65	test_multigrid	267
144.66	test_nc	267
144.67	test_nonuniform_symmetric	267
144.68	test_pde	267
144.69	test_permutation	267
144.70	test_poison_fem	267
144.71	test_polar	267
144.72	test_potential	267
144.73	test_powers	267
144.74	test_precondition	267
144.75	test_project_rectangle	268
144.76	test_qr	268
144.77	test_quantum_well	268
144.78	test_radial_adaptive	268
144.79	test_radial_confinement	268
144.80	test_radial_fixes	268
144.81	test_refine_2d	268
144.82	test_refine_2d_b	268
144.83	test_refine_3d	268
144.84	test_refine_structural	268
144.85	test_regularisation	269
144.86	test_round_off	269
144.87	test_schrödinger_potentials	269
144.88	test_uniform_mesh	269
144.89	test_vargrid	269
145test/numerical-methods/optimisation		269
145.1	test_extreme3	269
146test/signal-processing/autocorrelation		269
146.1	test_acf	269
146.2	test_acf_bias	269
146.3	test_acfar1_2	269
146.4	test_acfar1_3	270
146.5	test_acfar1_4	270
146.6	test_acfar2	270
146.7	test_ar1_var_factor	270
146.8	test_ar1_var_factor_2	270
146.9	test_ar1_var_mu_single_sample	270
146.10	test_ar1_var_pop	270
146.11	test_ar1_var_pop_1	270

146.12	test_ar1delay	270
146.13	test_ar2	270
146.14	test_phase_drift_acf	271
147	test/signal-processing/passes	271
147.1	test_bandpass2d	271
147.2	test_bandpass2d_ideal	271
147.3	test_lowpass1d_fft	271
147.4	test_lowpass1d_implicit	271
147.5	test_lowpass2d_anisotropic	271
147.6	test_lowpass2d_fft	271
147.7	test_lowpass2d_rho	271
148	test/signal-processing/periodogram	271
148.1	test_periodicity_test_2d	271
148.2	test_periodogram_bartlett_se	272
148.3	test_periodogram_gauss	272
148.4	test_periodogram_radial	272
148.5	test_periodogram_test	272
148.6	test_periodogram_test_periodicity_2d	272
148.7	test_periogogram_significance	272
149	test/signal-processing/spectral-density	272
149.1	test_phase_drift_parallel_pdf	272
149.2	test_phase_drift_parallel_pdf_mode2par	272
149.3	test_phase_drift_pdf	272
149.4	test_phase_drift_pdf_2d	273
149.5	test_phase_drift_pdf_mode	273
149.6	test_phase_drift_pdf_mode2par	273
149.7	test_phase_drift_pdf_scale	273
149.8	test_spectral_density_2	273
149.9	test_spectral_density_bandpass_2d	273
149.10	test_spectral_density_bandpass_2d_max2par	273
149.11	test_spectral_density_bandpass_continuous	273
149.12	test_spectral_density_bandpass_continuous_1	273
149.13	test_spectral_density_bandpass_maximum	273
149.14	test_spectral_density_bandpass_scale	274
149.15	test_spectral_density_bp	274
149.16	test_spectral_density_bp_2d	274
149.17	test_spectral_density_bp_approx	274
149.18	test_spectral_density_flat	274
149.19	test_spectral_density_hp_cos	274
149.20	test_spectral_density_lorentzian_max	274
149.21	test_spectral_density_lorentzian_scale	274

149.22	test_spectral_density_lowpass	274
149.23	test_spectral_density_lowpass_continuous	274
149.24	test_spectral_density_lowpass_continuous_1	275
149.25	test_spectral_density_maxiumum_bias_corrected	275
150	test/signal-processing	275
150.1	test_autocorrelation_max	275
150.2	test_cdf_bandpass_continuous	275
150.3	test_fit_spectral_density	275
150.4	test_phase_drift_cdf	275
151	test/spatial-pattern-analysis	275
151.1	test_approximate_ratio_distribution	275
151.2	test_approximate_ratio_quantile	275
151.3	test_separate_isotropic_density	275
152	test/spatial-statistics	276
152.1	test_cov_cell_averages_1d	276
152.2	test_cov_cell_averages_2d	276
153	test/statistics/distributions/anisotropic	276
153.1	test_anisotropic_pattern	276
153.2	test_anisotropic_pattern_pdf	276
154	test/statistics/distributions/gamma	276
154.1	test_generalized_gamma_mean	276
155	test/statistics/distributions/log-uniform	276
155.1	test_logurnd	276
156	test/statistics/distributions/lognormal	276
156.1	test_logn_cov	276
157	test/statistics/distributions/mises	277
157.1	test_mises_std	277
158	test/statistics/distributions/passes	277
158.1	test_bandpass2d_pdf	277
158.2	test_bandpass2d_pdf_hankel	277
158.3	test_bandpass2d_pdf_mode	277
158.4	test_lowpass2d_pdf_hankel	277
158.5	test_lowpass2d_pdf_series	277
159	test/statistics/distributions/skew-normal	277
159.1	test_skew_generalized_normpdf	277

160	test/statistics/distributions	277
160.1	test_normpdf_wrapped	277
161	test/statistics/distributions/weibull	278
161.1	test_wbl_std	278
162	test/statistics/moment-statistics	278
162.1	test_wmean	278
163	test/statistics	278
163.1	test_fisher_moment2par	278
163.2	test_gamma_mode	278
163.3	test_normalize_exponential_random_variable	278
164	test/stochastic	278
164.1	test_brownian_field	278
164.2	test_brownian_field_scaled	278
165	test/stochastics	278
165.1	test_brownian_surface	278
166	test	279
166.1	test_S	279
166.2	test_advect_analytic	279
166.3	test_asymbp	279
166.4	test_bandwidth	279
166.5	test_bartlett_angle	279
166.6	test_bartlett_distribution	279
166.7	test_bartlett_expansion	279
166.8	test_beta	279
166.9	test_betainc	279
166.10	test_bivariate_covariance_term	280
166.11	test_brownian_drift_hitting_probability	280
166.12	test_brownian_drift_hitting_probability2	280
166.13	test_brownian_motion_1d	280
166.14	test_brownian_motion_2d_cov	280
166.15	test_brownian_motion_2d_fft	280
166.16	test_brownian_noise_1d	280
166.17	test_brownian_noise_2d	280
166.18	test_brownian_noise_interleave	280
166.19	test_coherence	280
166.20	test_combined_spectral_density	281
166.21	test_continuous_fourier_transform	281
166.22	test_convexity	281
166.23	test_d2	281

166.24	test_determine_phase_shift	281
166.25	test_diffuse_analytic	281
166.26	test_diffusion_matrix	281
166.27	test_ellipse	281
166.28	test_error_propagation_fraction	281
166.29	test_f	281
166.30	test_f2	282
166.31	test_fit_2d_spectral_density	282
166.32	test_fourier	282
166.33	test_fourier_derivative	282
166.34	test_fourier_derivative_1	282
166.35	test_fourier_integral	282
166.36	test_fourier_mask_covariance_matrix	282
166.37	test_ft_bp	282
166.38	test_gam	282
166.39	test_gamma_distribution	282
166.40	test_gampdf_man	283
166.41	test_gaussfit3	283
166.42	test_gaussian_flat	283
166.43	test_geoserr	283
166.44	test_hexagonal_pattern	283
166.45	test_iafrate	283
166.46	test_implicit_ode	283
166.47	test_imrotmat	283
166.48	test_integration	283
166.49	test_ivp	283
166.50	test_jacobian	284
166.51	test_lanczoswin	284
166.52	test_laplacian_power	284
166.53	test_lognfit_quantile	284
166.54	test_ls_perpendicular_offset	284
166.55	test_madcorr	284
166.56	test_mask	284
166.57	test_max_normal	284
166.58	test_moments	284
166.59	test_moments_fourier_power	284
166.60	test_mtimes3x3	285
166.61	test_noisy_oscillator	285
166.62	test_nonperiodic_pattern	285
166.63	test_normalization	285
166.64	test_ols	285
166.65	test_parcorr	285
166.66	test_positivity_preserving	285
166.67	test_randar1	285

166.68	test_randar1_multivariate	285
166.69	test_randar2	285
166.70	test_ratio_distributions	286
166.71	test_sd_rectwin	286
166.72	test_spatialrnd	286
166.73	test_spectrum_additivity	286
166.74	test_stationarity	286
166.75	test_stationarity2	286
166.76	test_sum_ij	286
166.77	test_sum_multivar	286
166.78	test_trifilt1	286
166.79	test_wautocorr	286
166.80	test_wavelet_transform	287
166.81	test_whittle	287
166.82	test_window	287
166.83	test_wordfilt	287
166.84	test_xar1_mid_term	287
167	mathematics	287
167.1	trapezoidal_fixed	287
168	wavelet	287
168.1	contiuous_wavelet_transform	287
168.2	cwt_man	288
168.3	cwt_man2	288
168.4	example_wavelets	288
168.5	phasewrap	288
168.6	test_cwt_man	288
168.7	test_phasewrap	288
168.8	test_wavelet	288
168.9	test_wavelet2	288
168.10	test_wavelet_analysis	288
168.11	test_wavelet_reconstruct	289
168.12	test_wtc	289
168.13	wavelet	289
168.14	wavelet_reconstruct	289
168.15	wavelet_transform	289

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

$$\begin{aligned} \text{re}(c1 \exp(i o1 x)) * \text{re}(c2 \exp(i o2 x)) = \\ 1/2 * (\text{real}(c1 * c2 * \exp(i * (n1 + n2) * o * x)) \dots \\ + \text{real}(\text{conj}(c1) * c2 * \exp(i * (n2 - n1) * o * x))) \end{aligned}$$

the product has two frequency components

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

3.5 croots

nth-roots of a complex number

input:
 c : complex number

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

r : roots of the complex number

3.6 root_complex

root of a complex number

3.7 test_imroots

4 derivation

derivation of several functions by means of symbolic computation

4.1 derive_acfar1

4.2 derive_ar1_spectral_density

4.3 derive_ar2param

4.4 derive_arc_length

4.5 derive_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 2π

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 2π

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

output :
 ab : coefficients for frequency components

11.17 `fourier_coefficient_ramp3`

fourier series coefficient of a ramp

11.18 `fourier_coefficient_ramp_pulse`

fourier series coefficient of a ramp pules

11.19 `fourier_coefficient_ramp_step`

fourier coefficient of a ramp-step

11.20 `fourier_coefficient_square_pulse`

fourier series coefficients of a square pulse

11.21 `fourier_complete_negative_half_plane`

11.22 `fourier_cubic_interaction_coefficients`

11.23 `fourier_derivative`

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

input:

x : data, sampled in equal intervals

k : order of the derivative

dx : kth-derivative of x

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

11.24 `fourier_derivative_matrix_1d`

11.25 `fourier_derivative_matrix_2d`

11.26 `fourier_expand`

expand values of fourier series

11.27 `fourier_fit`

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

11.28 `fourier_freq2ind`

11.29 `fourier_interpolate`

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

11.30 `fourier_matrix`

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

11.31 `fourier_matrix2`

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

11.32 `fourier_matrix3`

transformation matrix for the continuous fourier transform
this is a matrix with $(2*n+1)$ 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 2ω ignored in input
frequencies higher than 3ω 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
`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 $Q|Q|$
 $Q|Q| = Q_a^2 y$ (8.5)
 $= |\cos a + \cos t| (\cos a + \cos t)$ (8.6)
 $= a_0 + a_1 \cos t + \dots + a_n \cos n t$ (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

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

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

11.51 mean_fourier_power

11.52 moments_fourier_power

11.53 nanfft

discrete fourier transform of a data series with gaps

11.54 peaks

peaks of the power spectrum of a discrete fourier transform

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

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

works best, when spectrum has been smoothened

input :

f : frequency

y : absolute value of fourier transform (power spectrum)

L : length in space or time of series

output :

a0 : amplitude

s0 : standard deviation (error?) of amplitude

w0 : width of peak

lambda = wave length (period?)

pdx : index of peak

f : frequency (if not given as input)

11.55 roots_fourier

zeros of continuous fourier series series

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

11.56 spectral_density

spectral density

11.57 std_fourier_power

11.58 test_complex_exp_product

11.59 test_fourier_filter

11.60 test_idftmtx

11.61 var_fourier_power

12 mathematics

mathematical functions of various kind

12.1 gaussfit_quantile

13 geometry/@Geometry

13.1 Geometry

13.2 arclength

arc length of a two dimensional curve

8th order accurate

does not require the segments length to vary smoothly

note: the curve can be considered parametric, e.g. $x = x(t)$, $y = y(t)$
and

and $t = t(s)$, but the error term contains derivatives of t ,
thus a non smooth t (strongly varying distance between points)
requires the scaling as done below

13.3 arclength_old

arc length of a two dimensional function

13.4 arclength_old2

arc length of a two dimensional function

13.5 base_point

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

13.6 base_point_limited

base point (Fusspunkt) of a point on a line

13.7 centroid

centroid of a polygone

13.8 cosa_min_max

13.9 cross2

cross product in two dimensions

13.10 curvature

curvature of a function in two dimensions

13.11 ddot

sum of squares of cos of inner angles of triangle

13.12 distance

equclidan distance between two points

13.13 distance2

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

13.14 dot

dot product

13.15 edge_length

edge length

13.16 enclosed_angle

angle enclosed between two lines

13.17 enclosing_triangle

smallest enclosing equilateral triangle with bottom side parallel to
X axis

13.18 hexagon

coordinates of a hexagon, scaled and rotated

13.19 inPolygon

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

13.20 inTetra

flag points contained in tetrahedron

13.21 inTetra2

flag points contained in tetrahedron

13.22 inTriangle

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

13.23 intersect

intersect between two lines

13.24 lineintersect

intersect of two lines

13.25 lineintersect1

intersect of two lines

13.26 minimum_distance_lines

minimum distance of two lines in three dimensions

13.27 mittenpunkt

mittenpunkt of a triangle

13.28 nagelpoint

nagelpoint of a triangle

13.29 onLine

13.30 orthocentre

orthocentre of triangle

13.31 `plumb_line`

13.32 `poly_area`

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

13.33 `poly_edges`

edges of a polygon

13.34 `poly_set`

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

13.35 `poly_width`

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

13.36 `polyxpoly`

intersections of two polygons

13.37 `project_to_curve`

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

13.38 quad_isconvex

13.39 random_disk

draw random points on the unit disk

13.40 random_simplex

random point inside of a triangle

13.41 sphere_volume

volume of a sphere
function v = sphere_volume(r)

13.42 tetra_volume

volume of a tetrahedron

13.43 tobarycentric

cartesian to barycentric coordinates

13.44 tobarycentric1

cartesian to barycentric coordinates

13.45 tobarycentric2

cartesian to barycentric coordinates

13.46 tobarycentric3

cartesian to barycentric coordinates

13.47 tri_angle

cos of angles of a triangle

13.48 tri_area

angle of a triangle

13.49 tri_centroid

centroid of a triangle

13.50 tri_distance_opposit_midpoint

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

13.51 tri_edge_length

edge length of a triangle

13.52 tri_edge_midpoint

mid point of a triangle

13.53 tri_excircle

excircle of a triangle

13.54 tri_height

height of a triangle

13.55 tri_incircle

incircle of a triangle

13.56 tri_isacute

flag acute triangles

13.57 tri_isobtuse

flag obtuse triangles

13.58 tri_semiperimeter

semiperimeter of a triangle

13.59 tri_side_length

edge length of triangle

14 geometry

14.1 Polygon

Simple 2D polygon class

Polygon properties:

x - x coordinates of polygon

y - y coordinates of polygon

nnodes - number of nodes in the polygon

Polygon methods:

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

14.2 bounding_box

bounding box of X

14.3 curvature_1d

curvature of a sampled parametric curve in two dimensions

14.4 cvt

centroidal voronoi tessellation

14.5 deg_to_frac

degree, minutes and seconds to fractions

14.6 ellipse

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

14.7 ellipseX

x-coordinates of y-coordinates of an ellipse

14.8 ellipseY

14.9 first_intersect

get first intersection between lines in A and B

14.10 golden_ratio

golden ratio

14.11 hypot3

hypothenuse in 3D

14.12 meanangle

weighted mean of angles

14.13 meanangle2

mean angle

14.14 meanangle3

mean angle

14.15 meanangle4

mean angle

14.16 medianangle

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

14.17 medianangle2

median angle

input
alpha : x*m, [rad] angle

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

14.18 pilim

limit to +- pi

14.19 streamline_radius_of_curvature

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

15 histogram/@Histogram

15.1 2x

15.2 Histogram

15.3 bimodes

15.4 cdf

15.5 cdfS

15.6 chi2test

15.7 cmoment

15.8 cmomentS

15.9 entropy

15.10 entropyS

15.11 export_csv

15.12 iquantile

15.13 kstest

15.14 kurtosis

15.15 kurtosisS

15.16 mean

15.17 meanS

15.18 median

15.19 medianS

15.20 mode

15.21 modeS

15.22 moment

15.23 momentS

15.24 pdf

15.25 quantile

15.26 quantileS

15.27 resample

15.28 setup

15.29 skewness

15.30 skewnessS

15.31 stairs

15.32 stairsS

15.33 std

15.34 stdS

15.35 var

15.36 varS

16 histogram

16.1 hist_man

16.2 histadapt

16.3 histconst

16.4 pdf_poly

16.5 plotcdf

16.6 test_histogram

17 mathematics

mathematical functions of various kind

17.1 imrotmat

18 linear-algebra

18.1 averaging_matrix_2

18.2 colnorm

norms of columns

18.3 condest_

estimation of the condition number

18.4 connectivity_matrix

19 linear-algebra/coordinate-transformation

19.1 barycentric2cartesian

barycentric to cartesian coordinates

19.2 barycentric2cartesian3

convert barycentric to cartesian coordinates

19.3 cartesian2barycentric

cartesian to barycentric coordinates

19.4 cartesian_to_unit_triangle_basis

transform coordinates into unit triangle

19.5 ellipsoid2geoid

19.6 example_approximate_utm_conversion

19.7 latlon2utm

transform latitude and longitude to WGS84 UTM

19.8 latlon2utm_simple

19.9 lowrance_mercator_to_wgs84

convert lowrance coordinates to wgs84

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

19.10 nmea2utm

convert nmea messages to utm coordinates

19.11 sn2xy

convert sn to xy coordinates

19.12 unit_triangle_to_cartesian

transform coordinates in unit triangle to cartesian coordinates

19.13 utm2latlon

convert wgs84 utm to latitude and longitude

19.14 xy2nt

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

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

19.15 xy2sn

convert cartesian to streamwise coordiantes

19.16 xy2sn.java

use java port for speed up

19.17 xy2sn_old

transform points from cartesian into streamwise coordinates

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

20 linear-algebra

20.1 det2x2

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

20.2 det3x3

determinant of stacked 3x3 matrices

20.3 det4x4

determinant of stacked 4x4 matrices

20.4 diag2x2

diagonal of stacked 2x2 matrices

20.5 down

20.6 eig2x2

eigenvalues of stacked 2x2 matrices

21 linear-algebra/eigenvalue

21.1 eig_bisection

21.2 eig_inverse

21.3 eig_inverse_iteration

21.4 eig_power_iteration

22 linear-algebra/eigenvalue/jacobi-davidson

22.1 afun_jdm

22.2 davidson

22.3 jacobi_davidson

22.4 jacobi_davidson_qr

22.5 jacobi_davidson_qz

22.6 jacobi_davidson_simple

22.7 jdqr

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

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

```

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

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

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

```

```

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
W=V*Q; V=V(:,1:j)/R; E=E/R; R=eye(j); M=Q(1:j,:)' /R;
W=V*H; V(:,j+1)=[];R=R'*R; M=H(1:j,:)' ;
%===== ARNOLDI (for initializing spaces)
=====
%===== END ARNOLDI
=====

% not accurate enough M=Rw'\(M/Rv);
%===== COMPUTE SORTED JORDAN FORM
=====

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

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

```

```

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

```

22.8 jdqr sleipen

```

% 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'

```


22.9 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
%    $V*V=eye(j)$ ,  $Qschur'*V=0$ ,
%    $W=A*V$ ,  $M=W'*W$ 
%
% 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
=====
% accepted separation between eigenvalues:
% no preconditioning
% solve left preconditioned system
% compute vectors and matrices for skew projection
% precondition and project r
% solve preconditioned system
% no preconditioning
% solve two-sided expl. preconditioned system
% compute vectors and matrices for skew projection
% precondition and project r
% solve preconditioned system
% "unprecondition" solution
%%% u(:,j+1)=Atilde*u(:,j)
%%% r(:,j+1)=Atilde*r(:,j)
%----- compute schur form -----
A*Q=Q*S, Q'*Q=eye(size(A));
% transform real schur form to complex schur form
%----- find order eigenvalues -----
%----- reorder schur form -----
%----- compute qz form -----
%----- sort eigenvalues -----
%----- sort qz form -----
% i>j, move ith eigenvalue to position j
% determine dimension
% defaults

```

22.10 jdqz

```

% Read/set parameters
% Return if eigenvalueproblem is trivial
% Initialize target, test space and interaction matrices
% V=RepGS(Qschur,V); [AV,BV]=MV(V); %%% more stability??
% W=RepGS(Zschur,eval(testspace)); %%% dangerous if sigma~lambda
% Solve the preconditioned correction equation

```

```

% Expand the subspaces and the interaction matrices
% Check for stagnation
% Solve projected eigenproblem
% Compute approximate eigenpair and residual
%=== an alternative, but less stable way of computing z =====
% display history
% save history
% check convergence
% EXPAND Schur form
% Expand preconditioned Schur matrix MinvZ=M\Zschur
% check for conjugate pair
% To detect whether another eigenpair is accurate enough
% restart if dim(V)> jmax
% Initialize target, test space and interaction matrices
% additional stabilisation. May not be needed
% V=RepGS(Zschur,V); [AV,BV]=MV(V);
% end add. stab.
% Solve the preconditioned correction equation
% expand the subspaces and the interaction matrices
% Check for stagnation
% compute approximate eigenpair
% Compute approximate eigenpair and residual
% display history
% save history
% check convergence
% expand Schur form
% ZastQ=Z'*Q0
% the final Qschur
% check for conjugate pair
% t perp Zschur, t in span(Q0,imag(q))
% To detect whether another eigenpair is accurate enough
% restart if dim(V)> jmax
%===== END JDQZ
=====
%=====
%===== PREPROCESSING
=====
%=====
%===== ARNOLDI (for initial spaces)
=====
%% then precondition=I and target = 0: apply Arnoldi with A
%===== END ARNOLDI
=====
%=====
%===== POSTPROCESSING
=====

```

```

%=====

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

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

%=====

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

% solve preconditioned system
%=====

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

% [At,Bt]=MV(x); At=theta(2)*At-theta(1)*Bt;
% xtol=norm(r-At+Z*(Z'*At))/norm(r);
%===== Iterative methods
%=====

% 0 step of bicgstab eq. 1 step of bicgstab
% Then x is a multiple of b
% HIST=[0,1];
% explicit preconditioning
% compute norm in l-space
% HIST=[HIST;[nmv,rnorm/snorm]];
% sufficient accuracy. No need to update r,u
% implicit preconditioning
% collect the updates for x in l-space
% but, do the orth to Z implicitly
% compute norm in l-space
% HIST=[HIST;[nmv,rnorm/snorm]];
% sufficient accuracy. No need to update r,u
% Do the orth to Z explicitly
% In exact arithmetic not needed, but
% appears to be more stable.

```

```

% plot(HIST(:,1),log10(HIST(:,2)+eps),'*'), drawnow
% 0 step of gmres eq. 1 step of gmres
% Then x is a multiple of b
%=====

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

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

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

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

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

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

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

```

```

%=====

%=====

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

%===== INITIALIZATION
=====
%=====

%=====

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

```

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

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

22.11 mfunc_jdm

22.12 mgs

22.13 minres_

22.14 mv_jacobi_davidson

23 linear-algebra

23.1 first

23.2 gershgorin_circle

range of eigenvalues determined by the gershgorin circle theorem

23.3 haussdorff

haussdorf dimension

box counting: count rectangles passed through by line (covered by polygon)

Koch snow flake 3:4 -> 1.2619

Kantor set 2:3, (4:9) -> 0.6309

quadrat 4:2, 9:3, 16:4 -> 2

23.4 eig2x2

reconstruct matrix from eigenvalue decomposition

23.5 inv2x2

2x2 inverse of stacked matrices

23.6 inv3x3

23.7 inv4x4

inverse of stacked 4x4 matrices

23.8 kernel2matrix

24 linear-algebra/lanczos

24.1 arnoldi

24.2 `arnoldi_new`

24.3 `eigs_lanczos_man`

24.4 `lanczos`

24.5 `lanczos_`

24.6 `lanczos_biorthogonal`

24.7 `lanczos_biorthogonal_improved`

24.8 `lanczos_ghep`

24.9 `mv_lanczos`

24.10 `reorthogonalise`

24.11 `test_lanczos`

25 linear-algebra

25.1 laplacian_eigenvalue

25.2 laplacian_eigenvector

25.3 laplacian_power

25.4 least_squares_perpendicular_offset

25.5 left

left element of vector, leftmost column is extrapolated

26 linear-algebra/linear-systems

26.1 gmres_man

break on convergence

26.2 minres_recycle

27 linear-algebra

27.1 lpmean

mean of pth-power of a

27.2 `lpnorm`

norm of lth-power of a

27.3 `matvec3`

matrix-vector product of stacked matrices and vectors

27.4 `max2d`

maximum value and i-j index for matrix

27.5 `mid`

mid point between neighbouring vector elements

27.6 `mpoweri`

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

27.7 `mtimes2x2`

27.8 `mtimes3x3`

product of stacked 3x3 matrices

27.9 `nannorm`

norm of a vector, skips nan-values

27.10 nanshift

shift vector, but set out of range values to NaN

27.11 nl

number rows (lines) of a matrix

analogue to unix nl command

27.12 normalise

normalise a vector or the columns of a matrix

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

27.13 normalize1

normalize columns in x to [-1,1]

27.14 normrows

27.15 orth2

make matrix A orthogonal to B

27.16 orth_man

orthogonalize the columns of A

27.17 orthogonalise

make x orthogonal to Y

27.18 padd2

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

27.19 paddext

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

27.20 paddval1

padd values at end of x

27.21 paddval2

padd values to x

28 linear-algebra/polynomial

28.1 chebychev

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

36 numerical-methods/differentiation

36.1 derivative1

first derivative on variable mesh
second order accurate

36.2 derivative2

second derivative on a variable mesh

37 numerical-methods

37.1 diffuse_analytic

38 numerical-methods/finite-difference

38.1 cdiff

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

38.2 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

38.3 central_difference

38.4 cmean

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

38.5 cmean2

38.6 derivative_matrix_1_1d

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

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

38.7 derivative_matrix_2_1d

finite derivative matrix of second derivative in one dimension

38.8 derivative_matrix_2d

finite difference derivative matrix in two dimensions

38.9 derivative_matrix_curvilinear

derivative matrix on a curvilinear grid

38.10 derivative_matrix_curvilinear_2

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

38.11 difference_kernel

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

38.12 diffusion_matrix_2d_anisotropic

38.13 diffusion_matrix_2d_anisotropic2

38.14 directional_neighbour

38.15 distmat

distance matrix for a 2 dimensional rectangular matrix

38.16 downwind_difference

38.17 gradpde2d

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

38.18 laplacian

38.19 laplacian_fdm

finite difference matrix of the laplacian
BC

38.20 lrmean

mean of the left and right element

39 numerical-methods/finite-difference/master

39.1 fdm_adaptive_grid

39.2 fdm_adaptive_refinement_old

39.3 fdm_assemble_d1_2d

39.4 fdm_assemble_d2_2d

39.5 fdm_confinement

39.6 fdm_d_vargrid

39.7 fdm_h_unstructured

39.8 `fdm_hydrogen_vargrid`

39.9 `fdm_mark_unstructured_2d`

39.10 `fdm_plot`

39.11 `fdm_plot_series`

39.12 `fdm_refine_2d`

39.13 `fdm_refine_3d`

39.14 `fdm_refine_unstructured_2d`

39.15 `fdm_schroedinger_2d`

39.16 `fdm_schroedinger_3d`

39.17 `relocate`

40 numerical-methods/finite-difference

40.1 mid

mid point between neighbouring vector elements

40.2 pwmid

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

40.3 ratio

ratio of two subsequent values

40.4 steplength

step length of a vector if it were equispaced

40.5 swapoddeven

swap odd and even elements in a vector

40.6 test_derivative_matrix_2d

40.7 test_derivative_matrix_curvilinear

40.8 test_difference_kernel

40.9 upwind_difference

41 numerical-methods/finite-element

41.1 Mesh_2d.java

41.2 Tree_2d.java

41.3 assemble_1d_dphi_dphi

41.4 assemble_1d_phi_phi

41.5 assemble_2d_dphi_dphi.java

41.6 assemble_2d_phi_phi.java

41.7 assemble_3d_dphi_dphi.java

41.8 assemble_3d_phi_phi.java

41.9 boundary_1d

41.10 boundary_2d

41.11 boundary_3d

41.12 check_area_2d

41.13 circmesh

41.14 cropradius

41.15 display_2d

41.16 display_3d

41.17 distort

41.18 err_2d

41.19 `estimate_err_2d_3`

41.20 `example_1d`

41.21 `example_2d`

41.22 `explode`

41.23 `fem_2d`

41.24 `fem_2d_heuristic_mesh`

41.25 `fem_get_2d_radial`

41.26 `fem_interpolation`

41.27 `fem_plot_1d`

41.28 `fem_plot_1d_series`

41.29 fem_plot_2d

41.30 fem_plot_2d_series

41.31 fem_plot_3d

41.32 fem_plot_3d_series

41.33 fem_plot_confine_series

41.34 fem_radial

adaptive grid
constant grid

41.35 flip_2d

41.36 get_mesh_arrays

41.37 hashkey

42 numerical-methods/finite-element/int

42.1 int_1d_equal

42.2 int_1d_equal_exp

42.3 int_1d_gauss

42.4 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()
```

42.5 int_1d_gauss_2

42.6 int_1d_gauss_3

42.7 int_1d_gauss_4

42.8 int_1d_gauss_5

42.9 `int_1d_gauss_6`

42.10 `int_1d_gauss_lobatto`

42.11 `int_1d_gauss_n`

42.12 `int_1d_nc_2`

42.13 `int_1d_nc_3`

42.14 `int_1d_nc_4`

42.15 `int_1d_nc_5`

42.16 `int_1d_nc_6`

42.17 `int_1d_nc_7`

42.18 `int_1d_nc_7_hardy`

42.19 int_2d_gauss_1

42.20 int_2d_gauss_12

42.21 int_2d_gauss_13

42.22 int_2d_gauss_16

42.23 int_2d_gauss_19

42.24 int_2d_gauss_25

42.25 int_2d_gauss_3

42.26 int_2d_gauss_33

42.27 int_2d_gauss_4

42.28 int_2d_gauss_6

42.29 int_2d_gauss_7

42.30 int_2d_gauss_9

42.31 int_2d_nc_10

42.32 int_2d_nc_15

42.33 int_2d_nc_21

42.34 int_2d_nc_3

42.35 int_2d_nc_6

42.36 int_3d_gauss_1

42.37 int_3d_gauss_11

42.38 int_3d_gauss_14

42.39 int_3d_gauss_15

42.40 int_3d_gauss_24

42.41 int_3d_gauss_4

42.42 int_3d_gauss_45

42.43 int_3d_gauss_5

42.44 int_3d_nc_11

42.45 int_3d_nc_4

42.46 int_3d_nc_6

42.47 int_3d_nc_8

43 numerical-methods/finite-element

43.1 interpolation_matrix

43.2 mark

43.3 mark_1d

43.4 mesh_1d_uniform

43.5 mesh_3d_uniform

43.6 mesh_interpolate

43.7 neighbour_1d

43.8 old

43.9 pdeeig_1d

43.10 pdeeig_2d

43.11 pdeeig_3d

43.12 polynomial_derivative_1d

43.13 potential_const

43.14 potential_coulomb

43.15 potential_harmonic_oscillator

43.16 project_circle

43.17 project_rectangle

43.18 promote_1d_2_3

43.19 promote_1d_2_4

43.20 promote_1d_2_5

43.21 promote_1d_2_6

43.22 quadrilaterate

43.23 recalculate_regularity_2d

43.24 refine_1d

43.25 refine_2d_21

43.26 refine_2d_structural

43.27 regularity_1d

43.28 regularity_2d

43.29 regularity_3d

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

43.30 relocate_2d

43.31 test_circmesh

43.32 test_hermite

43.33 tri_assign_points

43.34 triangulation_uniform

43.35 vander_1d

van der Monde matrix

43.36 vanderd_1d

43.37 vanderi_1d

44 numerical-methods/finite-volume/@Advection

44.1 Advection

FVM treatment of the Advection equation

44.2 dot_advection

advection equation

45 numerical-methods/finite-volume/@Burgers

45.1 burgers_split

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

45.2 dot_burgers_fdm

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

45.3 dot_burgers_fft

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

46 numerical-methods/finite-volume/@Finite_Volume

46.1 Finite_Volume

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

46.2 apply_bc

apply boundary conditions

46.3 solve

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

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

46.4 step_split_strang

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

46.5 step_unsplit

step in time, without splitting the inhomogeneous term

47 numerical-methods/finite-volume/@Flux_Limiter

47.1 Flux_Limiter

class of flux limiters

47.2 beam_warming

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

47.3 fromm

fromme limiter
low res

47.4 lax_wendroff

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

47.5 minmod

min-mod schock limiter

47.6 monotized_central

monotonized central flux limiter

47.7 muscl

muscl flux limiter

47.8 superbee

superbee limiter

47.9 upwind

godunov scheme
godunov, first order accurate

47.10 vanLeer

van Leer limiter

48 numerical-methods/finite-volume/@KDV

48.1 dot_kdv_fdm

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

48.2 dot_kdv_fft

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

48.3 kdv_split

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

49 numerical-methods/finite-volume/@Reconstruct_Average_E

49.1 Reconstruct_Average_Evolve

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

McCronack Scheme

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

error:

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

49.2 advect_highres

single time step for the reconstruct evolve algorithm

49.3 advect_lowres

single time step
low resolution

50 numerical-methods/finite-volume

50.1 Godunov

Godunov, upwind method for systems of pdes

50.2 Lax_Friedrich

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

50.3 Measure

50.4 Roe

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

The roe solver guarantess:

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

50.5 fv_swe

wrapper for solving SWE

50.6 staggered_euler

forward euler method with staggered grid

50.7 staggered_grid

staggered grid approximation to the SWE

51 numerical-methods

51.1 grid2quad

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

52 numerical-methods/integration

52.1 cumintL

cumulative integral from left to right

52.2 cumintR

cumulative integral from right to left

52.3 cumint_trapezoidal

integrate y along x with the trapezoidal rule

52.4 int_1d_gauss_laguerre

52.5 int_trapezoidal

integrate y along x with the trapezoidal rule

53 numerical-methods/interpolation/@Kriging

53.1 Kriging

class for Kriging interpolation

53.2 estimate_semivariance

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

53.3 interpolate_

interpolate with 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

54 numerical-methods/interpolation/@RegularizedInterpolator

54.1 RegularizedInterpolator1

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

54.2 init

initialize the interpolator with a set of sampling points

55 numerical-methods/interpolation/@RegularizedInterpolator

55.1 RegularizedInterpolator2

class for regularized interpolation on an unstructures mesh (
interpolation)

55.2 init

initialize the interpolator with a set of point samples

56 numerical-methods/interpolation/@RegularizedInterpolator

56.1 RegularizedInterpolator3

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

56.2 init

initialize the interpolator with a set of sampling points

57 numerical-methods/interpolation

57.1 IDW

spatial averaging by inverse distance weighting

57.2 IPoly

polynomial interpolation class

57.3 IRBM

interpolate by the radial basis function method

```
fprintf(1,'Progress IRBM: %d%%\n',round(100*  
    idx/size(Xi,1)));
```

57.4 ISparse

sparse interpolation class

57.5 Inn

nearest neighbour interpolation

57.6 Interpolator

interpolator super-class

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

57.7 fixnan

fill nan-values in vector with gaps

57.8 idw1

spatial average by inverse distance weighting

57.9 idw2

spatial average by inverse distance weighting

57.10 inner2outer

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

57.11 inner2outer2

interpolate from element (segment) centres to edge points

57.12 interp1_circular

57.13 interp1_limited

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

57.14 interp1_man

interpolate

57.15 `interp1_piecewise_linear`

57.16 `interp1_save`

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

57.17 `interp1_slope`

quadratic interpolation returning value and derivative(s)

57.18 `interp1_smooth`

57.19 `interp1_unique`

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

57.20 `interp2_man`

nearest neighbour interpolation in two dimensions

57.21 `interp_angle`

interpolate an angle

57.22 interp_fourier

interpolation by the fourier method

57.23 interp_fourier_batch

batch interpolation by the fourier interpolation

57.24 interp_sn

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

57.25 interp_sn2

interpolation in streamwise coordinates

57.26 interp_sn3

57.27 interp_sn_

57.28 limit_by_distance_1d

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

57.29 resample1

interpolation along a parametric curve with variable step width

57.30 resample_d_min

resample a function

57.31 resample_vector

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

57.32 test_interp1_limited

58 numerical-methods

58.1 inverse_complex

58.2 maccormack_step

58.3 minmod

59 numerical-methods/multigrid

59.1 mg_interpolate

59.2 mg_restrict

60 numerical-methods/ode/@BVPS_Characteristic

60.1 BVPS_Characteristic

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

60.2 assemble1_A

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

60.3 assemble1_A_Q

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

60.4 assemble2_A

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

60.5 assemble_AA

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

60.6 assemble_AAA

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

60.7 **assemble_Ic**

60.8 **bvp1c**

60.9 **check_arguments**

60.10 **couple_junctions**

60.11 **derivative**

60.12 **init**

60.13 **inner2outer_bvp2c**

60.14 **reconstruct**

60.15 **resample**

60.16 solve

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

odefun provides ode coefficients c:

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

subject to the boundary conditions

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

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

$$q(x,1)*(p(x,1) y_l(x) + p(x,2) y_l'(x) \\ + q(x,2)*(p(x,1) y_r(x) + p(x,2) y_r'(x) = v(x)$$

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

60.17 test_assemble1_A

60.18 test_assemble2_A

61 numerical-methods/ode/@Time_Stepper

61.1 Time_Stepper

61.2 solve

62 numerical-methods/ode

62.1 bvp2fdm

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

 odefun provides ode coefficients c:

$$c(x,1) y''(x) + c(x,2) y'(x) + c(x,3) y = c(x,4)$$

$$c_1 y'' + c_2 y' + c_3 y + c_4 = 0$$

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

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

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

62.2 bvp2wavetrain

solve second order boundary value problem by repeated integration

62.3 bvp2wavetwopass

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

62.4 ivp_euler_forward

solve initial value problem by the euler forward method

62.5 ivp_euler_forward2

62.6 ivprk2

solve initial value problem by the two step runge kutta method

62.7 ode2_matrix

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

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

62.8 ode2characteristic

second order odes
transmitted and reflected wave

62.9 step_trapezoidal

single trapezoidal step

62.10 test_bvp2

63 numerical-methods/optimisation

63.1 aitken_iteration

63.2 anderson_iteration

63.3 armijo_stopping_criterion

armijo stopping criterion for optimizations

63.4 astar

astar path finding algorithm

63.5 binsearch

binary search on a line

63.6 bisection

bisection

63.7 box1

test objective function for optimisation routines

63.8 box2

63.9 cauchy

63.10 cauchy2

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

63.11 directional_derivative

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

63.12 dud

optimization by the dud algorithm

63.13 extreme3

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

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

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

63.14 extreme_quadratic

63.15 ftest

63.16 fzero_bisect

63.17 `fzero_newton`

63.18 `grad`

numerical gradient

63.19 `hessian`

numerical hessian

63.20 `hessian_from_gradient`

numerical hessian from gradient

63.21 `hessian_projected`

numerical hessian projected to one dimension

63.22 `line_search`

bisection routine

63.23 `line_search2`

bisection method

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

63.24 line_search_polynomial

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

63.25 line_search_polynomial2

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

63.26 line_search_quadratic

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

63.27 line_search_quadratic2

```
quadratic line search
```


63.28 line_search_wolfe

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

63.29 ls_bgfs

least squares by the bgfs method

63.30 ls_broyden

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

63.31 ls_generalized_secant

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

63.32 nlcg

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

opt : struct options
fdx : gradient constraint

63.33 nlls

non-linear least squares

63.34 picard

picard iteration

63.35 poly_extrema

extrema of a polynomial

63.36 quadratic_function

evaluate quadratic function in higher dimensions

63.37 quadratic_programming

optimize by quadratic programming

63.38 quadratic_step

single step of the quadratic programming

63.39 rosenbrock

rosenbrock test function

63.40 `sqrt_heron`

Heron's method for the square root

63.41 `test_directional_derivative`

63.42 `test_dud`

63.43 `test_fzero_newton`

63.44 `test_line_search_quadratic2`

63.45 `test_ls_generalized_secant`

63.46 `test_nlcg_6_order`

63.47 `test_nlls`

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

64 numerical-methods/pde

64.1 `laplacian2d_fundamental_solution`

65 numerical-methods/piecewise-polynomials

65.1 Hermite1

hermite polynomial interpolation in 1d

65.2 hp2_fit

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

65.3 hp2_predict

prediction with pw hermite polynomial
c are values at support points

65.4 hp_predict

predict with piecewise hermite polynomial

65.5 hp_regress

fit piecewise hermite polynomial
coefficients are values and derivatives

65.6 lp_count

lagrangian basis for interpolation
count number of valid samples

65.7 lp_predict

lagrangian basis piecwie interpolation, predicator

65.8 lp_regress

65.9 lp_regress_

66 numerical-methods

66.1 test_adams_bashforth

67 mathematics

mathematical functions of various kind

67.1 oversampleNZ

68 pdes

68.1 heat_equation_fundamental_solution

68.2 heat_equation_width

68.3 heat_equation_width_to_time

69 regression/@PolyOLS

69.1 PolyOLS

class for polynomial least squares

69.2 coefftest

69.3 detrend

detrending by polynomial regression

69.4 fit

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

69.5 fit_

fit a polynomial function

69.6 predict

predict polynomial function values

69.7 predict_

69.8 slope

slope by linear regression

70 regression/@PowerLS

70.1 PowerLS

class for power law regression

70.2 fit

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

70.3 predict

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

70.4 predict_

71 regression/@Theil

71.1 Theil

Kendal-Theil-Sen robust regression

71.2 detrend

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

71.3 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

71.4 predict

predict values and confidence intervals with the Theil-Sen method

71.5 slope

fit the slope with the Theil-Sen method

72 regression

linear and non-linear regression

72.1 Theil_Multivariate

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

72.2 areg

regression using the pth-fraction of samples with smallest residual

72.3 ginireg

gini regression

72.4 hesssimplereg

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

72.5 llin

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

72.6 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)

72.7 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

72.8 regression_method_of_moments

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

72.9 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

72.10 theil2

Theil senn-estimator for two dimensions (glm)

72.11 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

72.12 total_least_squares

total least squares

72.13 weighted_median_regression

weighted median regression
c.f. Scholz, 1978

73 set-theory

73.1 issubset

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

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

74 mathematics

mathematical functions of various kind

74.1 shuffle_index

75 signal-processing

75.1 asymwin

creates asymmetrical filter windows
filter will always have negative weights

76 signal-processing/autocorrelation

76.1 acf_radial

76.2 acfar1

Autocorrelation function of the finite AR1 process

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

76.3 acfar1_2

autocorrelation of the ar1 process

76.4 acfar2

impulse response of the ar2 process

76.5 acfar2_2

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

76.6 ar1_cutoff_frequency

76.7 ar1_effective_sample_size

effective sample size correction for autocorrelated series

76.8 ar1_mse_mu_single_sample

standard error of a single sample of an ar1 correlated process

76.9 ar1_mse_pop

variance of the population mean of a single realisation around zero

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

76.10 ar1_mse_range

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

76.11 ar1_spectrum

spectrum of the ar1 process

76.12 ar1_to_tikhonov

convert ar1 correlation to tikhonovs lambda

76.13 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

76.14 ar1_var_factor_

variance of an autocorrelated finite process

76.15 ar1_var_range2

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

76.16 ar1delay

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

76.17 ar1delay_old

autocorrelation of the residual

76.18 ar2_acf2c

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

76.19 ar2conv

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

76.20 ar2dof

effective samples size for the ar2 process

76.21 ar2param

ar2 parameter estimation from first two terms of acf
acf = [1 a1 a2 ...]

76.22 autocorr2

76.23 autocorr_angular

76.24 autocorr_bandpass

76.25 autocorr_decay_rate

estimate exponential decay of the autocorrelation

76.26 autocorr_effective_sample_size

effective sample size from acf

76.27 autocorr_fft

estimate sample autocorrelation function

76.28 autocorr_forest

76.29 autocorr_genton

autocorrelation function

76.30 autocorr_highpass

76.31 autocorr_lowpass

76.32 autocorr_periodic_additive_noise

76.33 autocorr_periodic_windowed

76.34 autocorr_radial

76.35 autocorr_radial_hexagonal_pattern

76.36 autocorrelation_max

77 signal-processing

77.1 average_wave_shape

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

77.2 bandpass

bandpass filter

77.3 bandpass_continuous_cdf

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

77.5 bin1d

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

77.6 bin2d

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

77.7 binormrnd

generate two correlated normally distributed vectors

77.8 coherence

77.9 conv1_man

convolutions with padding

77.10 conv2_man

convolution in 2d

77.11 conv2z

77.12 conv30

convolve with rectangular window of lenght n
 circular boundaries

77.13 conv_

convolution of a with b

77.14 conv_centered

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

77.15 convz

77.16 cosexpdelay

77.17 csmooth

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

77.18 daniell_window

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

77.19 db2neper

convert decibel to neper

77.20 db2power

power ratio from db

77.21 `derive_bandpass_continuous_scale`

77.22 `derive_danielle_weight`

77.23 `derive_limit_0_acfar`

77.24 `detect_peak`

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

77.25 `determine_phase_shift`

77.26 `determine_phase_shift1`

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

77.27 `doublesum_ij`

double sum of r^i

77.28 `effective_mask_size`

77.29 `effective_sample_size_to_ar1`

convert effective sample size to ar1 correlation

77.30 `fcut2Lw_gausswin`

77.31 `fcut_gausswin`

77.32 `filt_hodges_lehman`

78 `signal-processing/filters`

78.1 `circfilt2`

`smooth` (filter) the 2D image `z` with a circular disk of radius `nf`
apply periodic boundary conditions

78.2 `filter1`

`filter` along one dimension

78.3 `filter2`

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

78.4 `filter_`

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

78.5 `filter_r_to_f0`

78.6 filter_rho_to_f0

78.7 filter_twosided

78.8 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

78.9 filterp

78.10 filterp1

fir filter with some fancy extras

78.11 filterstd

78.12 gaussfilt2

smooth (filter) the 2D image z with a gaussian window
apply periodic boundary conditions

78.13 lowpass_discrete

design coefficients of a low pass filter with specified cut of
frequency
and sampling period
analogue low pass with pole at $s = -\omega_c = 1/\tau = 1/RC$
 $H_a = \tau / (\tau + s) = 1 / (1 + \omega_c s)$

78.14 meanfilt2

filter with a rectangular window along both dimensions

78.15 medfilt1_man

moving median filter, supports columnwise operation

78.16 medfilt1_man2

moving median filter with special treatment of boundaries

78.17 `medfilt1_padded`

median filter with padding

78.18 `medfilt1_reduced`

median filter with padding

78.19 `trifilt1`

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

78.20 `trifilt2`

filter with a triangular window along both dimensions

79 signal-processing

79.1 `firls_man`

design finite impulse response filter by the least squares method

79.2 `fit_spectral_density`

fit spectral densities (probability distributions)

79.3 `fit_spectral_density_2d`

fit spectral densities

79.4 `fit_spectral_density_radial`

fit spectral densities

79.5 `flattopwin`

the flat top window

79.6 `frequency_response_boxcar`

frequency response of a boxcar filter

79.7 `freqz_boxcar`

frequency response of a boxcar filter

79.8 `gaussfilt1`

filter data series with a gaussian window, assumes periodic bc

79.9 `hanchangewin`

hanning window for change point detection

79.10 `hanchangewin2`

nanning window for chage point detection

79.11 `hanwin`

hanning filter window

79.12 hanwin_

hanning filter window

79.13 high_pass_1d_simple

79.14 kaiserwin

kaiser filter window

79.15 kalman

Kalman filter

79.16 lanczoswin

Lanczos window

79.17 last

lake tail, but for matrices

79.18 maxfilt1

79.19 meanfilt1

moving average filter with special treatment of the boundaries

79.20 mid_term_single_sample

variance of single sample, mid term

79.21 minfilt1

79.22 minmax

79.23 mu2ar1

error variance of the mean of the finite length ar1 process

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

79.24 mysmooth

79.25 nanautocorr

autocorrelation with nan-values

79.26 nanmedfilt1

medfilt1, skipping nans

79.27 neper2db

convert neper to db

79.28 oscillator_noisy

80 signal-processing/passes

80.1 bandpass1d

80.2 bandpass1d_fft

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

80.3 bandpass1d_implicit

80.4 bandpass2

bandpass filter

80.5 bandpass2d

80.6 bandpass2d_convolution

80.7 bandpass2d_fft

80.8 bandpass2d_ideal

80.9 bandpass2d_implicit

bandpass filter the surface x by solving the implicit relation:

80.10 bandpass2d_iso

80.11 bandpass_arg

determine correlation coefficient from frequency of mode for the
symmetric

80.12 bandpass_f0_to_rho

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

80.13 bandpass_max

80.14 bandpass_max2

80.15 highpass

high pass filter

80.16 highpass1d_fft_cos

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

80.17 `highpass1d_implicit`

80.18 `highpass2d_fft`

80.19 `highpass2d_ideal`

80.20 `highpass2d_implicit`

80.21 `highpass_arg`

80.22 `highpass_fc_to_rho`

80.23 `lowpass`

`low pass filter`

80.24 `lowpass1d_fft`

80.25 `lowpass1d_implicit`

80.26 lowpass2

design low pass filter with cutoff-frequency f1

80.27 lowpass2d_anisotropic

80.28 lowpass2d_convolution

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

80.29 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)

80.30 lowpass2d_ideal

lowpass filter the input x in the Frequency Domain

TODD no need to provide dx, follows from size of x
function [y,S,R,r]=lowpass2d_ideal(x,L,dx,varargin)

80.31 lowpass2d_implicit

function [y] = lowpass2d_implicit(x,rho,a,order,direct)

80.32 lowpass_arg

80.33 lowpass_fc_to_rho

80.34 lowpass_iir

iir-low pass

80.35 lowpass_iir_symmetric

two-sided iir low pass filter (for symmetry)

80.36 lowpassfilter2

low-pass filter of data

81 signal-processing

81.1 peaks_man

peaks of a periodogram

82 signal-processing/periodogram

82.1 periodogram

compute the normalized periodogram

82.2 periodogram_2d

compute the normalized periodogram in two dimensions

82.3 periodogram_align

82.4 periodogram_angular

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

82.5 periodogram_bartlett

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

82.6 periodogram_bootstrap

82.7 periodogram_confidence_interval

```
confidence interval for periodogram values
```

82.8 periodogram_filter

82.9 periodogram_median

82.10 periodogram_normalize

82.11 periodogram_normalize_2d

82.12 periodogram_p_value

82.13 periodogram_qq

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

82.14 periodogram_quantiles

quantiles of a periodogram

82.15 periodogram_radial

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

compute the radially averaged density

input:

S2d : 2-dimensional density or periodogram

L =[Lx,Ly] : domain length

output:

S_r.mu : radially averaged periodogram

S_r.normalized : normalized radially averaged periodogram

A : matrix operator s that $S_r = (A \cdot A')^{-1} A' \cdot S2d$

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 fr
 $k_r = 2\pi \cdot f_r$

radially averaged periodogram:
$$S_r(k_r) = \frac{1}{k_r} \int_0^{k_r} S2d(k_r,s) ds$$
$$= \frac{1}{(2\pi)} \int_0^{2\pi} S2d(k_r,\theta) d\theta$$
$$\sim \frac{1}{(2\pi)} \sum_{nt} S2d(k_r,\theta) * (2\pi/nt)$$
$$\sim \frac{1}{nt} \sum_{nt} S2d(k_r,\theta)$$

```

nt ~ k_r/df = k_r*L
normalization:
    S_r.normalize = S_r/int_0^inf S_r dfr
                  ~ S_r/(sum_0^nr S_r Delta fr)

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

when S is flattened into a vector, the isotropic part of the 2D
density can be recovered with:
S_iso    = (A*S_radial)
S_radial = A^-1 S_hat

```

82.16 periodogram_std

standard deviation of a periodogram

82.17 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

82.18 periodogram_test_periodicity_2d

test a periodogram for hidden periodic frequency components

```
[p,stat,ratio] = periodogram_test_periodicity_2d(b, nf, bmsk, fmsk,
ns)
```

input:

```
  b  (nx * ny): image to test for presence of hidden
periodicities,
      i.e. periodicities where the frequency is not known a
      priori
  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 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
  ns  : 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
```

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

82.19 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

82.20 periodogram_welsh

83 signal-processing

83.1 polyfilt1

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

83.2 qmedfilt1

medfilt1, after fitting a quadratic polynomial

83.3 quadratfilt1

83.4 quadratwin

83.5 randar1

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

83.6 randar1_dual

draw random variables of two correlated ar1 processes

83.7 randar2

generate ar2 process

83.8 randarp

randomly generate the instance of an ar-p process

83.9 rectwin

rectangular window

83.10 recursive_sum

83.11 select_range

83.12 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

83.13 smooth2

smooth vectos of X

83.14 smooth_man

83.15 smooth_parametric

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

83.16 smooth_parametric2

parametrically smooth the curve

83.17 smooth_with_splines

83.18 smoothfft

filter with fast fourier transform

84 signal-processing/spectral-density

84.1 hex_angular_pdf

84.2 hex_angular_pdf_max

84.3 hex_angular_pdf_max2par

84.4 spectral_density_ar2

84.5 `spectral_density_area`

integrate the spectral density over the positive half axis

84.6 `spectral_density_estimate_2d`

84.7 `spectral_density_flat`

flat spectral density of a random vector with iid elements

84.8 `spectral_density_forest`

84.9 `spectral_density_gausswin`

84.10 `spectral_density_lorentzian`

lorentzian spectral density

84.11 `spectral_density_lorentzian_max`

mode (maximum) of the lorentzian spectral density

84.12 `spectral_density_lorentzian_max2par`

transform maximum of the lorentzian spectral density to its
distribution parameters

84.13 spectral_density_lorentzian_scale

normalization scale of the lorentzian spectral density

84.14 spectral_density_maximum_bias_corrected

84.15 spectral_density_periodic_additive_noise

84.16 spectral_density_rectwin

84.17 spectral_density_wperiodic

85 signal-processing

85.1 spectrogram

spectrogram

85.2 sum_i_lag

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

85.3 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

85.4 `sum_ii_`

85.5 `sum_ij`

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

85.6 `sum_ij_`

85.7 `sum_ij_partial_`

85.8 `sum_multivar`

sum of matrix entries of bivariate ar1 process

85.9 `test_acfar1`

85.10 `tikhonov_to_ar1`

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

85.11 `trapwin`

trapezoidal filter window

85.12 triwin

triangular filter window

85.13 triwin2

triangular filter window

85.14 tukeywin_man

85.15 varar1

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

85.16 welch_spectrogram

welch spectrogram

85.17 wfilt

filter with window

85.18 winbandpass

filter with bandpass

86 signal-processing/windows

86.1 circwin

86.2 danielle_window

danielle fourier window

86.3 gausswin

86.4 gausswin1

86.5 gausswin2

86.6 radial_window

radial filter window in the 2d-frequency domain

86.7 range_window

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

86.8 rectwin_cutoff_frequency

86.9 std_window

moving block standard deviation

86.10 window2d

86.11 window_make_odd

87 signal-processing

87.1 winfilt0

filter with window

87.2 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

87.3 wmeanfilt

mean filter with window

87.4 wmedfilt

median filter with window

87.5 wordfilt

weighted order filter

87.6 wordfilt_edgeworth

weighed order filter

87.7 wrapphase

87.8 xar1

87.9 xcorr_man

cross correlation of two sampled ar1 processes

88 sorting

88.1 sort2

sort two numbers

88.2 sort2d

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

89 spatial-pattern-analysis/@Spatial_Pattern

89.1 Spatial_Pattern

class for analysis of remotely sensed and model generated
vegetation patterns

89.2 analyze_grid

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

89.3 analyze_transect

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

89.4 clear_1d_properties

89.5 clear_2d_properties

89.6 fit_parametric_densities

fit parametric spectral densities to the empirical density

89.7 imread

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

89.8 plot

plot the pattern or densities

89.9 plot_transect

plot 1D pattern

89.10 prepare_analysis

89.11 report

report statistics of analysis

89.12 resample_functions

resample empirical densities to a common grid

89.13 tabulate

summarize properties of multiple patterns in a single struct

90 spatial-pattern-analysis/@Spatial_Pattern_Array

90.1 Spatial_Pattern_Array

container class for Spatial_Pattern objects

90.2 analyze

analyze spatial patterns

90.3 assign_regions

90.4 export_shp

90.5 fetch

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

90.6 generate_filename

90.7 quality_check

91 spatial-pattern-analysis

91.1 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
```

91.2 banded_pattern

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

91.4 patch_size_1d

91.5 patch_size_2d

91.6 pattern_isotropic_rotated

91.7 reconstruct_isotropic_density

91.8 separate_isotropic_from_anisotropic_density

91.9 suppress_low_frequency_lobe

92 spatial-statistics

92.1 cov_cell_averages_1d

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

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

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

integrals approximated by Gauss' method

92.2 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 intervalles,
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

93 special-functions

93.1 `bessel_sphere`

spherical Bessel function of the first kind

93.2 `besseliln_large_x`

93.3 `beta_man`

93.4 `betainc_man`

93.5 `digamma_man`

93.6 `exp10`

93.7 `hankel_sphere`

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

93.8 `hermite`

probabilistic's hermite polynomial by recurrence relation

input :
n : order
x : value

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

93.9 laguerre_roots

93.10 lambertw_numeric

lambert-w function

93.11 legendre_man

legendre polynomials

93.12 neumann_sphere

spherical Neumann function
Bessel function of the second kind

94 statistics

94.1 atan_s2

stadard deviation of the arcus tangens by means of taylor expansion

94.2 binomial

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

95 statistics/circular

95.1 circular_fmoment

95.2 circular_fquantile

95.3 circular_fstd

95.4 circular_fvar

96 statistics

96.1 coefficient_of_determination

96.2 conditional_expectation_normal

96.3 correlation_confidence_pearson

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

97 statistics/distributions

97.1 PDF

class for quasi-distributions from a set of sampling points

98 statistics/distributions/anisotropic

98.1 anisotropic_pattern

98.2 anisotropic_pattern_acf

98.3 anisotropic_pattern_pdf

99 statistics/distributions/beta

99.1 beta_kurt

99.2 beta_mean

99.3 beta_moment2par

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

99.4 beta_skew

99.5 beta_std

100 statistics/distributions/bivariate-normal

100.1 binorm_separation_coefficient

separation coefficient of a bimodal normal distribution

100.2 binormcdf

bio-modal gaussian distribution

100.3 binormfit

fit sum of to normal distribution to a histogram

100.4 binormpdf

101 statistics/distributions/chi2

101.1 chi2_kurt

101.2 chi2_mean

101.3 chi2_skew

101.4 chi2_std

102 statistics/distributions/circular-normal

102.1 wnormpdf

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

103 statistics/distributions/edgeworth

103.1 edgeworth_cdf

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

103.2 edgeworth_pdf

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

104 statistics/distributions/exp

104.1 exppdf_max2par

105 statistics/distributions/fisher

105.1 fisher_mean

105.2 fisher_moment2par

105.3 fisher_std

106 statistics/distributions/gamma

106.1 gamma_mean

106.2 gamma_mode

106.3 gamma_mode2par

106.4 gamma_moment2par

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

106.5 gamma_std

106.6 gamma_stirling

106.7 gampdf_man

106.8 generalized_gamma_mean

107 statistics/distributions/hotelling-t2

107.1 t2cdf

Hotelling's T-squared cumulative distribution

107.2 t2inv

inverse of Hotelling's T-squared cumulative distribution

108 statistics/distributions/kurt-normal

108.1 kurtncdf

108.2 kurtnpdf

109 statistics/distributions/log-triangular

109.1 logtrialtcdf

pdf of a logarithmic triangular distribution

109.2 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 log²(b))/((log(a)
- log(b)) W((a^{-1/(log(a) - log(b))}) (b<sup>-log(c)/log(a) - 1/
log(a)</sup>) c^{-log(a)/(log(a) - log(b))}) (-d F log²(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)))

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

109.3 logtrialtmean

mean of the logarithmic triangular distribution

109.4 logtrialtpdf

density of the logarithmic triangular distribution

109.5 logtrialtrnd

109.6 logtricdf

cumulative distribution of the logarithmic triangular distribution

109.7 logtriinv

inverse of the logarithmic triangular distribution

109.8 logtrimean

mean of the logarithmic triangular distribution

109.9 logtripdf

probability density of the logarithmic triangular distribution

109.10 logtirnd

110 statistics/distributions/log-uniform

110.1 logu_median

median of the log-uniform distribution

110.2 logucdf

probability density of the logarithmic uniform distribution

110.3 logucm

central moments of the log-uniform distribution

110.4 loguinv

inverse of the log-uniform distribution

110.5 logumean

mean of the log-uniform distribution

110.6 logupdf

pdf of the log uniform distribution

110.7 logurnd

random numbers following a log-uniform distribution

110.8 loguvar

variance of the log-uniform distribution

111 statistics/distributions/loglog

111.1 loglogpdf

112 statistics/distributions/lognormal

112.1 logn_corr

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

112.2 logn_cov

covariance of two log-normally distributed random variables,
 $\text{cov}(ea, eb) = \text{cov}(\exp(\mu_a + s_a z_a), \exp(\mu_b + s_b z_b))$
where z_a, z_b are standard normal distributed and correlated

112.3 logn_mean

112.4 logn_mode

mode (maximum) of the log-normal density

112.5 logn_mode2par

112.6 logn_moment2par

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

112.7 logn_moment2par_correlated

112.8 logn_param2moment

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

112.9 logn_skewness

112.10 logn_std

112.11 lognpdf_

log normal distribution called by modes rather than parameters

112.12 lognpdf_entropy

113 statistics/distributions/logskew

113.1 logskewcdf

113.2 logskewpdf

114 statistics/distributions/mises

114.1 mises_max2par

114.2 mises_std

114.3 mises_var

114.4 misesn_max2par

114.5 misesnpdf

114.6 misespdf

115 statistics/distributions

115.1 ncx2_moment2par

116 statistics/distributions/normal

116.1 normpdf_entropy

116.2 normpdf_mode

116.3 normpdf_mode2par

117 statistics/distributions/passes

117.1 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

117.2 bandpass1d_continuous_pdf_max

maximum of the bandpass spectral density

117.3 bandpass1d_continuous_pdf_max2par

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

117.4 bandpass1d_continuous_pdf_scale

normaliztation scale of the spatial bandpass density

117.5 bandpass1d_discrete_pdf

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

117.6 bandpass2d_discrete_pdf

117.7 bandpass2d_pdf_exact

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

117.8 bandpass2d_pdf_hankel

117.9 bandpass2d_pdf_mode

117.10 bandpass2d_pdf_mode2par

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

117.11 bandpass2d_pdf_scale

117.12 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 -> 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
```

117.13 highpass1d_discrete_cos_pdf

cosine shaped spectral density of a highpass filter

117.14 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

$$S_h = |F \cdot A_h|^2$$

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

$$= F(I - A_l)$$

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

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

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

```
function [S_h,S_h1] = spectral_density_hp(f,r0,fmax,order,varargin)
```

117.15 `highpass2d_discrete_pdf`

117.16 `highpass2d_pdf`

117.17 `highpass2d_pdf_hankel`

117.18 `lowpass1d_continuous_pdf`

117.19 `lowpass1d_continuous_pdf_scale`

117.20 `lowpass1d_discrete_pdf`

117.21 `lowpass1d_one_sided_pdf`

117.22 `lowpass2d_discrete_acf`

truncated, not wrapped at the end

117.23 `lowpass2d_discrete_pdf`

117.24 lowpass2d_pdf

117.25 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:
 $S_{2d}(x,0) = F_{1d}^{-1}(\int R_{2d}(x,y) dy)$

117.26 lowpass2d_pdf_series

118 statistics/distributions

118.1 pdfsample

pdf from sample distribution
Note: better use kernel density estimates

119 statistics/distributions/phase-drift

119.1 phase_drift_acf

119.2 phase_drift_acf_2d

119.3 `phase_drift_cdf`

119.4 `phase_drift_inv`

119.5 `phase_drift_parallel_acf`

119.6 `phase_drift_parallel_pdf`

119.7 `phase_drift_parallel_pdf_max`

119.8 `phase_drift_parallel_pdf_max2par`

119.9 `phase_drift_parallel_pdf_mode2par`

119.10 `phase_drift_patch_size_distribution`

119.11 `phase_drift_pdf`

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

119.12 `phase_drift_pdf_2d`

119.13 `phase_drift_pdf_mode`

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

119.14 `phase_drift_pdf_mode2par`

transform mode to parameters of the brownian phase spectral density

119.15 `phase_drift_pdf_reg2par`

119.16 `phase_drift_pdf_scale`

normalization scale of the brownian phase spectral density

120 `statistics/distributions/skew-normal`

120.1 `skew_generalized_normal_fit`

120.2 `skew_generalized_normpdf`

120.3 `skewcdf`

120.4 skewparam_to_central_moments

120.5 skewpdf

skew-normal distribution
c.f. Azzalini 1985

120.6 skewpdf_entropy

121 statistics/distributions/triangular

121.1 tricdf

cumulative distribution of the log-triangular distribution

121.2 triinv

inverse of the triangular distribution

121.3 trimediam

median of the triangular distribution

121.4 tripdf

probability density of the triangular distribution

121.5 trirnd

random numbers of the triangular distribution

122 statistics/distributions/weibull

122.1 wbl_std

123 statistics/distributions/wrapped-normal

123.1 normpdf_wrapped

123.2 normpdf_wrapped_mode

123.3 normpdf_wrapped_mode2par

124 statistics

124.1 error_propagation_fraction

124.2 error_propagation_product

124.3 example_standard_error_of_sample_quantiles

124.4 f_var_finite

reduction of variance when sampling from a finite population
without replacement

124.5 gaussfit3

124.6 gaussfit_quantile

124.7 geoserr

124.8 geostd

124.9 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

124.10 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

125 statistics/information-theory

125.1 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

125.2 bayesian_information_criterion

bayesian information criterion

126 statistics

126.1 jackknife_block

126.2 kurtosis_bias_corrected

bias corrected kurtosis

126.3 limit

limit a by lower and upper bound

126.4 logfactorial

approximate log of the factorial

126.5 lognfit_quantile

126.6 max_exprnd

126.7 maxnnormals

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

126.8 mean_angle

126.9 mean_max_n

126.10 mean_min_n

126.11 midrange

mid range of columns of X

126.12 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

126.13 mode_man

127 statistics/moment-statistics

127.1 autocorr_man3

autocorrelation of the columns of X

127.2 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

127.3 autocorr_man5

autocorrelation of the columns of X

127.4 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

127.5 comoment

non-central higher order moments of the multivariate normal distribution

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

note : there seem to be some typos in the original paper, for x^4 c_{ii}^2 , the square seems to be missing

mu : nx1 mean vector
C : nxn covariance matrix
k : nx1 powers of variables in moments

127.6 corr_man

correlation of two vectors

127.7 cov_man

covariance matrix of two vectors

127.8 dof

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

127.9 edgeworth_quantile

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

127.10 effective_sample_size

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

127.11 f_correlation

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

127.12 `f_finite`

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

127.13 `lmean`

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

127.14 `lmoment`

l-moment of vector `x`

127.15 `maskmean`

mean of the masked values of `X`

127.16 `masknanmean`

127.17 `mean1`

mean of `x`

127.18 `mean_man`

mean and standard error of `X`

127.19 `mse`

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

127.20 nanautocorr_man1

autocorrelation of a vector with nan-values

127.21 nanautocorr_man2

autocorrelation of a vector with nan-values

127.22 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

127.23 nancorr

(co)-correlation matrix when samples a NaN

127.24 nancumsum

cumulative sum, setting nan values to zero

127.25 nanlmean

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

127.26 nanr2

coefficient of determination when samples are invalid

127.27 nanrms

root mean square value when sample contains nan-values

127.28 nanrmse

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

127.29 nanserr

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

127.30 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)

127.31 nanwstd

weighed standard deviation

127.32 nanwvar

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

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

127.33 nanxcorr

127.34 pearson

pearson correlation coefficient

127.35 pearson_to_kendall

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

127.36 pool_samples

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

127.37 qmean

trimmed mean

127.38 range_mean

127.39 rmse_

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

127.40 serr

standard error of the mean of a set of uncorrelated samples

127.41 serr1

127.42 test_kskew

127.43 test_qstd_kskew_optimal_p

127.44 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

127.45 wcorr

correlation of two vectors when samples are weighted

127.46 wcov

covariance of two vectors when samples are weighted

127.47 wdof

effective degrees of freedom for weighted samples

127.48 wkurt

kurtosis with weighted samples

127.49 wmean

weighted mean

$\min_x \sum w (x - \mu)^2 \Rightarrow \mu = \frac{\sum wx}{\sum w}$

varargin can be dim

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

127.50 wrms

weighted root mean square

127.51 wserr

weighted root mean square error

127.52 wskew

skewness of a weighted set of samples

function sk = wskew(w,x)

127.53 wstd

weighed standard deviation

127.54 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)

$s^2 = \frac{\sum w^2 (x - \frac{\sum wx}{\sum w})^2}{\sum w}$

s2_mu : error of mean, s2_mu : sd of prediction

128 statistics

128.1 nangeomean

128.2 nangeostd

geometric standard deviation ignoring nan-values

129 statistics/nonparametric-statistics

129.1 kernel1d

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

129.2 kernel2d

kernel density estimate in two dimensions

130 statistics

130.1 normalize_exponential_random_variable

130.2 normmmoment

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

130.3 normpdf2

pdf of the bivariate normal distribution

131 statistics/order-statistics

131.1 hodges_lehmann_location

hodges lehman location estimator

Asymptotic rms efficiency of location estimate:

mean: $1 s/\sqrt{n}$

hodges lehman: $\sqrt{\pi/3} * s \sim 1.0233 s/\sqrt{n}$

median: $\pi/2 s/\sqrt{n} \sim 1.25 s / \sqrt{n}$

131.2 kendall

kendall correlation coefficient

131.3 kendall_to_pearson

convert kendall rank correlation coefficient to the person product
moment

correlation coefficient

c.f. Kruskal, 1958, p. 823

131.4 mad2sd

transform median absolute deviation to standard deviation
for normal distributed values

131.5 madcorr

proxy correlation by median absolute deviation

131.6 median2_holder

131.7 median_ci

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

131.8 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`

131.9 mediani

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

131.10 nanmadcorr

proxy correlation by median absolute deviation

131.11 nanwmedian

weighted median, skips nan-values

131.12 nanwquantile

weighted quantile, skips nan values

131.13 oja_median

two dimensional oja median

note: the multivariate median is not unique

oja 1983, for extension to multivariate function, see chaudhri

131.14 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

131.15 qmoments

moments estimated from quantiles

131.16 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

131.17 qskewq

skewness estimated by quantiles

131.18 qstdq

proxy standard deviation determined by quantiles

131.19 quantile1_optimisation

131.20 quantile2_breckling

qunatile regression

131.21 quantile2_chaudhuri

quantile regression

131.22 quantile2_projected

quantile in two dimensions

131.23 quantile2_projected2

spatial qunatile for chosen direction

131.24 quantile_envelope

131.25 quantile_regression_simple

simple quantile regression

131.26 ranking

ranking for spearman statistics

131.27 spatial_median

c.f. Oja 2008

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

131.28 spatial_quantile

spatial quantile

131.29 spatial_quantile2

spatial quantile

131.30 spatial_quantile3

spatial quantile

131.31 spatial_rank

unsigned rank

131.32 spatial_sign

spatial sign

131.33 spatial_signed_rank

signed rank

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

131.34 spearman

spearman's product moment coefficient

131.35 spearman_rank

131.36 spearman_to_pearson

conversion of spearman rank to person product moment correlation coefficient

131.37 wmedian

weighted median

131.38 wquantile

weighted quantile

132 statistics

132.1 qstd

132.2 quantile_extrap

132.3 quantile_sin

133 statistics/random-number-generation

133.1 laplacernd

random number of laplace distribution

133.2 randc

correlate to correlated standard normally distributed vectors

133.3 skewness2param

133.4 skewpdf_central_moments

133.5 skewrnd

random numbers of the skew normal distribution

134 statistics

134.1 range

range and mid range of input

134.2 resample_with_replacement

135 statistics/resampling-statistics/@Jackknife

135.1 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

135.2 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

135.3 matrix1_STATIC

matrix of estimation for leaving out two samples at a time

135.4 matrix2

matrix of estimations for jacknive with two samples left out

136 statistics/resampling-statistics

136.1 block_jackknife

136.2 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

136.3 moving_block_jackknife

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

136.4 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

136.5 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

137 statistics

137.1 scale_quantile_sd

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

137.2 sd_sample_quantiles

137.3 spatialrnd

137.4 trimmed_mean

trimmed mean

137.5 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

137.6 ttest_man

two-sample t-test
unequal sample size
equal variance

137.7 `ttest_paired`

paired t-test
unequal sample size
equal variance
more powerfull than unpaired test, as long as correlation between
 x_1 and $x_2 > 0$

137.8 `uniformnpdf`

137.9 `wgeomean`

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

137.10 `wgeovar`

variance of the weighted geometric mean

137.11 `wharmean`

weighted harmonic mean

137.12 `wharstd`

137.13 `wharvar`

138 stochastic

138.1 `brownian_drift_hitting_probability`

138.2 brownian_drift_hitting_probability2

138.3 brownian_field

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

138.4 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?
```

138.5 brownian_motion_1d_acf

138.6 brownian_motion_1d_cov

138.7 brownian_motion_1d_fft

138.8 brownian_motion_1d_fourier

138.9 brownian_motion_1d_interleave

138.10 brownian_motion_1d_laplacian

138.11 brownian_motion_2d_cov

138.12 brownian_motion_2d_fft

138.13 brownian_motion_2d_fft_old

138.14 brownian_motion_2d_fourier

138.15 brownian_motion_2d_interleave

138.16 brownian_motion_2d_interleaving

138.17 brownian_motion_2d_kahunen

138.18 brownian_motion_2d_laplacian

138.19 brownian_motion_with_drift_hitting_probability

139 stochastic/geometric-ar1

139.1 geometric_ar1_2d_generate

139.2 geometric_ar1_2d_generate_1

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

139.3 geometric_ar1_2d_grid_cell_averaged_cov

139.4 geometric_ar1_2d_grid_cell_averaged_generate

simulate a grid cell averaged stochastic process $\exp(z)$
where z follows an Ornstein-Uhlenbeck (AR1) process
with mean lmu , standard deviation sd and stationary autocorrelation
 $\text{corr}(z(0,0), z(x,y)) = \exp(-\sqrt{x^2 + y^2}/\text{theta})$

139.5 geometric_ar1_2d_grid_cell_averaged_moment2par

139.6 geometric_ar1_2d_grid_cell_averaged_std

mean of the values val
covariance function of the values

140 stochastic

140.1 ornstein_uhlenbeck_cov

140.2 `ornstein_uhlenbeck_mean`

140.3 `ornstein_uhlenbeck_spectral_density`

140.4 `ornstein_uhlenbeck_std`

141 `mathematics`

mathematical functions of various kind

141.1 `ternary_diagram`

142 `test/finance`

142.1 `test_gbb_mean`

142.2 `test_gbb_std`

142.3 `test_gbm_mean`

142.4 `test_gbm_mean_entire_series`

142.5 test_gbm_moment2par

142.6 test_gbm_moment2par_entire_series

142.7 test_gbm_std

142.8 test_gbm_std_entire_series

143 test/fourier

143.1 test_fourier_freq2ind

144 test/master

144.1 dat_test_lanczos_3d_k_20_n_40

144.2 poisson2d_blk

144.3 qr_implicit_givens_2

144.4 spectral_derivative_2d

144.5 test_2d_eigensolver_hydrogen

144.6 test_2d_refine

144.7 test_3d_eigensolver_hydrogen

144.8 test_FEM

144.9 test_Mesh_3d

144.10 test_arnoldi

144.11 test_arpackc

144.12 test_assemble

144.13 test_assembly_performance

144.14 test_bc_one_sided

144.15 test_compare_solvers

144.16 test_complete

144.17 test_convergence

144.18 test_convergence_b

144.19 test_df_2d

144.20 test_eig_algs

144.21 test_eig_inverse

144.22 test_eigs_lanczos

144.23 test_eigs_lanczos_1

144.24 test_eigs_lanczos_2

144.25 test_eigs_lanczos_performance

144.26 test_fdm

144.27 test_fdm_d_vargrid

144.28 test_fdm_spectral

144.29 test_fem

144.30 test_fem_1d

144.31 test_fem_1d_higher_order

144.32 test_fem_2d_adaptive

144.33 test_fem_2d_higher_order

144.34 test_fem_3d_higher_order

144.35 test_fem_3d_refine

144.36 test_fem_b

144.37 test_fem_derivative

144.38 test_fem_quadrature

144.39 test_final

144.40 test_fix_substitution

144.41 test_forward

144.42 test_get_sparse_arrays

144.43 test_harmonic_oscillator

144.44 test_high_order_fdm_periodic_bc

144.45 test_hydrogen_wf

144.46 test_ichol

144.47 test_interpolation

144.48 test_inverse_problem

144.49 test_it_vs_exact

144.50 test_jama

144.51 test_jd

144.52 test_jdqz

144.53 test_lanczos_2

144.54 test_lanczos_biorthogonal

144.55 test_laplacian

144.56 test_laplacian_non_uniform

144.57 test_laplacian_simple

144.58 test_mesh_2d_uniform

144.59 test_mesh_2d_uniform_2

144.60 test_mesh_circle

144.61 test_mesh_generation

144.62 test_mesh_interpolate

144.63 test_mg

144.64 test_minres_recycle

144.65 test_multigrid

144.66 test_nc

144.67 test_nonuniform_symmetric

144.68 test_pde

144.69 test_permutation

144.70 test_poison_fem

144.71 test_polar

144.72 test_potential

144.73 test_powers

144.74 test_precondition

144.75 test_project_rectangle

144.76 test_qr

144.77 test_quantum_well

144.78 test_radial_adaptive

144.79 test_radial_confinement

144.80 test_radial_fixes

144.81 test_refine_2d

144.82 test_refine_2d_b

144.83 test_refine_3d

144.84 test_refine_structural

144.85 test_regularisation

144.86 test_round_off

144.87 test_schrödinger_potentials

144.88 test_uniform_mesh

144.89 test_vargrid

145 test/numerical-methods/optimisation

145.1 test_extreme3

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_asymbp

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

inverse 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 $1/f_0$