

Manual for Package: mathematics

Revision 2:13M

Karl Kästner

August 26, 2020

Contents

1	calendar	33
1.1	days_per_month	33
1.2	isnight	33
2	mathematics	33
2.1	cast_byte_to_integer	33
3	complex-analysis	33
3.1	complex_exp_product_im_im	33
3.2	complex_exp_product_im_re	33
3.3	complex_exp_product_re_im	34
3.4	complex_exp_product_re_re	34
3.5	croots	34
3.6	root_complex	35
3.7	test_imroots	35
4	derivation	35
4.1	derive_acfar1	35
4.2	derive_ar2param	35
4.3	derive_arc_length	35
4.4	derive_fourier_power	35
4.5	derive_fourier_power_exp	35
4.6	derive_laplacian_curvilinear	36
4.7	derive_laplacian_fourier_piecewise_linear	36
4.8	derive_logtripdf	36
4.9	derive_smooth1d_parametric	36
5	derivation/master	36
5.1	derive_bc_one_sided	36
5.2	derive_convergence	36

5.3	derive_error_fdm	36
5.4	derive_fdm_poly	36
5.5	derive_fdm_power	36
5.6	derive_fdm_taylor	37
5.7	derive_fdm_vargrid	37
5.8	derive_fem_2d_mass	37
5.9	derive_fem_error_2d	37
5.10	derive_fem_error_3d	37
5.11	derive_fem_sym_2d	37
5.12	derive_grid_constants	37
5.13	derive_interpolation	37
5.14	derive_laplacian	37
5.15	derive_limit	37
5.16	derive_nc_1d	38
5.17	derive_nc_1d_	38
5.18	derive_nc_2d	38
5.19	derive_nonuniform_symmetric	38
5.20	derive_richardson	38
5.21	derive_sum	38
5.22	nn	38
5.23	test_derive	38
5.24	test_derive_fdm_poly	38
5.25	test_filter	38
5.26	test_vargrid	39
6	derivation	39
6.1	simplify_atan	39
7	mathematics	39
7.1	entropy	39
7.2	exp10	39
7.3	filter_twosided	39
8	finance	39
8.1	derive_skewrnd_walsh_paramter	39
8.2	gbm_cdf	39
8.3	gbm_fit	40
8.4	gbm_fit_old	40
8.5	gbm_inv	40
8.6	gbm_mean	40
8.7	gbm_median	40
8.8	gbm_pdf	40
8.9	gbm_simulate	40
8.10	gbm_skewness	40

8.11	gbm_std	40
8.12	gbm_transform_time_step	40
8.13	put_price_black_scholes	41
8.14	skewgbm_simulate	41
8.15	skewrnd_walsh	41
9	finance/test	41
9.1	test_gbm	41
9.2	test_gbm_pdf	41
9.3	test_skewrnd_walsh	41
10	fourier/@STFT	41
10.1	STFT	41
10.2	itransform	42
10.3	stft_	42
10.4	stftmat	42
10.5	transform	42
11	fourier	42
11.1	amplitude_from_peak	42
11.2	dftmtx_man	43
11.3	example_fourier_window	43
11.4	fft_derivative	43
11.5	fft_man	43
11.6	fftsmooth	44
11.7	fix_fourier	44
11.8	fourier_axis	44
11.9	fourier_cesaro_correction	45
11.10	fourier_coefficient_piecewise_linear	45
11.11	fourier_coefficient_piecewise_linear_1	45
11.12	fourier_coefficient_ramp3	45
11.13	fourier_coefficient_ramp_pulse	46
11.14	fourier_coefficient_ramp_step	46
11.15	fourier_coefficient_square_pulse	46
11.16	fourier_cubic_interaction_coefficients	46
11.17	fourier_derivative	46
11.18	fourier_expand	46
11.19	fourier_fit	46
11.20	fourier_interpolate	46
11.21	fourier_matrix	47
11.22	fourier_matrix2	47
11.23	fourier_matrix3	47
11.24	fourier_matrix_exp	47
11.25	fourier_multiplicative_interaction_coefficients	47

11.26	fourier_power	47
11.27	fourier_power_exp	48
11.28	fourier_predict	48
11.29	fourier_quadratic_interaction_coefficients	48
11.30	fourier_range	48
11.31	fourier_regress	48
11.32	fourier_resampled_fit	48
11.33	fourier_resampled_predict	49
11.34	fourier_signed_square	49
11.35	fourier_transform	49
11.36	hyperbolic_fourier_box	49
11.37	idftmtx_man	49
11.38	laplace_2d_pwlinear	50
11.39	nanfft	50
11.40	peaks	50
11.41	roots_fourier	51
11.42	spectral_density	51
11.43	test_complex_exp_product	51
11.44	test_fourier_filter	51
11.45	test_idftmtx	51
12 mathematics		51
12.1	gaussfit_quantile	51
13 geometry/@Geometry		51
13.1	Geometry	51
13.2	arclength	52
13.3	arclength_old	52
13.4	arclength_old2	52
13.5	base_point	52
13.6	base_point_limited	52
13.7	centroid	52
13.8	cosa_min_max	52
13.9	cross2	53
13.10	curvature	53
13.11	ddot	53
13.12	distance	53
13.13	distance2	53
13.14	dot	53
13.15	edge_length	53
13.16	enclosed_angle	53
13.17	enclosing_triangle	54
13.18	hexagon	54
13.19	inPolygon	54

13.20	inTetra	54
13.21	inTetra2	54
13.22	inTriangle	54
13.23	intersect	54
13.24	lineintersect	54
13.25	lineintersect1	55
13.26	minimum_distance_lines	55
13.27	mittenpunkt	55
13.28	nagelpoint	55
13.29	onLine	55
13.30	orthocentre	55
13.31	plumb_line	55
13.32	poly_area	55
13.33	poly_edges	56
13.34	poly_set	56
13.35	poly_width	56
13.36	polyxpoly	56
13.37	project_to_curve	56
13.38	quad_isconvex	56
13.39	random_disk	56
13.40	random_simplex	57
13.41	sphere_volume	57
13.42	tetra_volume	57
13.43	tobarycentric	57
13.44	tobarycentric1	57
13.45	tobarycentric2	57
13.46	tobarycentric3	57
13.47	tri_angle	57
13.48	tri_area	58
13.49	tri_centroid	58
13.50	tri_distance_opposit_midpoint	58
13.51	tri_edge_length	58
13.52	tri_edge_midpoint	58
13.53	tri_excircle	58
13.54	tri_height	58
13.55	tri_incircle	58
13.56	tri_isacute	59
13.57	tri_isobtuse	59
13.58	tri_semiperimeter	59
13.59	tri_side_length	59
14	geometry	59
14.1	Polygon	59
14.2	bounding_box	59

14.3	curvature_1d	60
14.4	cvt	60
14.5	deg_to_frac	60
14.6	ellipse	60
14.7	ellipseX	60
14.8	ellipseY	60
14.9	first_intersect	60
14.10	golden_ratio	60
14.11	hypot3	61
14.12	meanangle	61
14.13	meanangle2	61
14.14	meanangle3	61
14.15	meanangle4	61
14.16	medianangle	61
14.17	medianangle2	61
14.18	pilim	62
14.19	streamline_radius_of_curvature	62
15 histogram/@Histogram		62
15.1	2x	62
15.2	Histogram	62
15.3	bimodes	62
15.4	cdf	62
15.5	cdfS	62
15.6	chi2test	62
15.7	cmoment	63
15.8	cmomentS	63
15.9	entropy	63
15.10	entropyS	63
15.11	iquantile	63
15.12	kstest	63
15.13	kurtosis	63
15.14	kurtosisS	63
15.15	mean	63
15.16	meanS	63
15.17	median	64
15.18	medianS	64
15.19	mode	64
15.20	modeS	64
15.21	moment	64
15.22	momentS	64
15.23	pdf	64
15.24	quantile	64
15.25	quantileS	64

15.26	setup	64
15.27	skewness	65
15.28	skewnessS	65
15.29	stairs	65
15.30	stairsS	65
15.31	std	65
15.32	stdS	65
15.33	var	65
15.34	varS	65
16	histogram	65
16.1	hist_man	65
16.2	histadapt	66
16.3	histconst	66
16.4	pdf_poly	66
16.5	plotcdf	66
16.6	test_histogram	66
17	linear-algebra	66
17.1	averaging_matrix_2	66
17.2	colnorm	66
17.3	condest_	66
18	linear-algebra/coordinate-transformation	67
18.1	barycentric2cartesian	67
18.2	barycentric2cartesian3	67
18.3	cartesian2barycentric	67
18.4	cartesian_to_unit_triangle_basis	67
18.5	ellipsoid2geoid	67
18.6	example_approximate_utm_conversion	67
18.7	latlon2utm	67
18.8	latlon2utm_simple	67
18.9	lowrance_mercator_to_wgs84	68
18.10	nmea2utm	68
18.11	sn2xy	68
18.12	unit_triangle_to_cartesian	68
18.13	utm2latlon	68
18.14	xy2nt	68
18.15	xy2sn	68
18.16	xy2sn_java	69
18.17	xy2sn_old	69
19	linear-algebra	69
19.1	det2x2	69

19.2	det3x3	69
19.3	det4x4	69
19.4	diag2x2	69
19.5	eig2x2	69
20	linear-algebra/eigenvalue	70
20.1	eig_bisection	70
20.2	eig_inverse	70
20.3	eig_inverse_iteration	70
20.4	eig_power_iteration	70
21	linear-algebra/eigenvalue/jacobi-davidson	70
21.1	afun_jdm	70
21.2	davidson	70
21.3	jacobi_davidson	70
21.4	jacobi_davidson_qr	70
21.5	jacobi_davidson_qz	70
21.6	jacobi_davidson_simple	71
21.7	jdqr	71
21.8	jdqr_sleijpen	74
21.9	jdqr_vorst	78
21.10	jdqz	80
21.11	mfunc_jdm	85
21.12	mgs	85
21.13	minres_	85
21.14	mv_jacobi_davidson	85
22	linear-algebra	85
22.1	first	85
22.2	gershgorin_circle	85
22.3	haussdorff	86
22.4	ieig2x2	86
22.5	inv2x2	86
22.6	inv3x3	86
22.7	inv4x4	86
23	linear-algebra/lanczos	86
23.1	arnoldi	86
23.2	arnoldi_new	86
23.3	eigs_lanczos_man	87
23.4	lanczos	87
23.5	lanczos_	87
23.6	lanczos_biorthogonal	87
23.7	lanczos_biorthogonal_improved	87

23.8	lanczos_ghep	87
23.9	mv_lanczos	87
23.10	reorthogonalise	87
23.11	test_lanczos	87
24	linear-algebra/linear-systems	88
24.1	gmres_man	88
24.2	minres_recycle	88
25	linear-algebra	88
25.1	lpmean	88
25.2	lpnorm	88
25.3	matvec3	88
25.4	max2d	88
25.5	mid	88
25.6	mpoweri	89
25.7	mtimes2x2	89
25.8	mtimes3x3	89
25.9	nannorm	89
25.10	nanshift	89
25.11	nl	89
25.12	normalise	89
25.13	normalize1	90
25.14	normrows	90
25.15	orth2	90
25.16	orth_man	90
25.17	orthogonalise	90
25.18	paddext	90
25.19	paddval1	90
25.20	paddval2	90
26	linear-algebra/polynomial	91
26.1	chebychev	91
26.2	piecewise_polynomial	91
26.3	roots1	91
26.4	roots2	91
26.5	roots2poly	91
26.6	roots3	91
26.7	roots4	91
26.8	roots_piecewise_linear	91
26.9	test_roots4	92
26.10	vanderi_1d	92
27	linear-algebra	92

27.1	randrot	92
27.2	right	92
27.3	rot2	92
27.4	rot2dir	92
27.5	rot3	92
27.6	rotR	92
27.7	rownorm	93
27.8	similarity_matrix	93
27.9	spnorm	93
27.10	spzeros	93
27.11	test_roots3	93
27.12	transform_minmax	93
27.13	transpose3	93
27.14	transposeall	93
28	logic	94
28.1	bitor_man	94
29	master/plot	94
29.1	attach_boundary_value	94
29.2	cartesian_polar	94
29.3	img_vargrid	94
29.4	plot_basis_functions	94
29.5	plot_convergence	94
29.6	plot_dof	94
29.7	plot_eigenbar	94
29.8	plot_error_estimation	95
29.9	plot_error_estimation_2	95
29.10	plot_error_fem	95
29.11	plot_fdm_kernel	95
29.12	plot_fdm_vs_fem	95
29.13	plot_fem_accuracy	95
29.14	plot_function_and_grid	95
29.15	plot_hat	95
29.16	plot_hydrogen_wf	95
29.17	plot_mesh	95
29.18	plot_mesh_2	96
29.19	plot_refine	96
29.20	plot_refine_3d	96
29.21	plot_runtime	96
29.22	plot_spectrum	96
29.23	plot_wavefunction	96
30	master/ported	96

30.1	assemble_2d_phi_phi	96
30.2	assemble_3d_dphi_dphi	96
30.3	assemble_3d_phi_phi	96
30.4	dV_2d_	97
30.5	derivative_2d	97
30.6	derivative_3d	97
30.7	element_neighbour_2d	97
30.8	prefetch_2d_	97
30.9	promote_2d_3_10	97
30.10	promote_2d_3_15	97
30.11	promote_2d_3_21	97
30.12	promote_2d_3_6	97
30.13	promote_3d_4_10	97
30.14	promote_3d_4_20	98
30.15	promote_3d_4_35	98
30.16	vander_2d	98
30.17	vander_3d	98
31	mathematics	98
31.1	myexp	98
32	number-theory	98
32.1	ceiln	98
32.2	digitsb	98
32.3	floorn	98
32.4	iseven	99
32.5	multichoosek	99
32.6	nchoosek_man	99
32.7	pythagorean_triple	99
32.8	roundn	99
33	numerical-methods/differentiation	99
33.1	derivative1	99
33.2	derivative2	100
34	numerical-methods/finite-difference	100
34.1	cdiff	100
34.2	cdiffb	100
34.3	central_difference	100
34.4	cmean	100
34.5	cmean2	100
34.6	derivative_matrix_1_1d	100
34.7	derivative_matrix_2_1d	101
34.8	derivative_matrix_2d	101

34.9	derivative_matrix_curvilinear	101
34.10	derivative_matrix_curvilinear_2	101
34.11	difference_kernel	101
34.12	distmat	101
34.13	downwind_difference	101
34.14	gradpde2d	102
34.15	laplacian	102
34.16	laplacian_fdm	102
34.17	left	102
34.18	lrmean	102
35	numerical-methods/finite-difference/master	102
35.1	fdm_adaptive_grid	102
35.2	fdm_adaptive_refinement_old	102
35.3	fdm_assemble_d1_2d	103
35.4	fdm_assemble_d2_2d	103
35.5	fdm_confinement	103
35.6	fdm_d_vargrid	103
35.7	fdm_h_unstructured	103
35.8	fdm_hydrogen_vargrid	103
35.9	fdm_mark_unstructured_2d	103
35.10	fdm_plot	103
35.11	fdm_plot_series	103
35.12	fdm_refine_2d	103
35.13	fdm_refine_3d	104
35.14	fdm_refine_unstructured_2d	104
35.15	fdm_schroedinger_2d	104
35.16	fdm_schroedinger_3d	104
35.17	relocate	104
36	numerical-methods/finite-difference	104
36.1	mid	104
36.2	pwmid	104
36.3	ratio	104
36.4	steplength	105
36.5	swapoddeven	105
36.6	test_derivative_matrix_2d	105
36.7	test_derivative_matrix_curvilinear	105
36.8	test_difference_kernel	105
36.9	upwind_difference	105
37	numerical-methods/finite-element	105
37.1	Mesh_2d.java	105
37.2	Tree_2d.java	105

37.3	assemble_1d_dphi_dphi	105
37.4	assemble_1d_phi_phi	106
37.5	assemble_2d_dphi_dphi_java	106
37.6	assemble_2d_phi_phi_java	106
37.7	assemble_3d_dphi_dphi_java	106
37.8	assemble_3d_phi_phi_java	106
37.9	boundary_1d	106
37.10	boundary_2d	106
37.11	boundary_3d	106
37.12	check_area_2d	106
37.13	circmesh	106
37.14	cropradius	107
37.15	display_2d	107
37.16	display_3d	107
37.17	distort	107
37.18	err_2d	107
37.19	estimate_err_2d_3	107
37.20	example_1d	107
37.21	example_2d	107
37.22	explode	107
37.23	fem_2d	107
37.24	fem_2d_heuristic_mesh	108
37.25	fem_get_2d_radial	108
37.26	fem_interpolation	108
37.27	fem_plot_1d	108
37.28	fem_plot_1d_series	108
37.29	fem_plot_2d	108
37.30	fem_plot_2d_series	108
37.31	fem_plot_3d	108
37.32	fem_plot_3d_series	108
37.33	fem_plot_confine_series	108
37.34	fem_radial	109
37.35	flip_2d	109
37.36	get_mesh_arrays	109
37.37	hashkey	109
38	numerical-methods/finite-element/int	109
38.1	int_1d_gauss	109
38.2	int_1d_gauss_1	109
38.3	int_1d_gauss_2	109
38.4	int_1d_gauss_3	109
38.5	int_1d_gauss_4	109
38.6	int_1d_gauss_5	110
38.7	int_1d_gauss_6	110

38.8	int_1d_gauss_lobatto	110
38.9	int_1d_gauss_n	110
38.10	int_1d_nc_2	110
38.11	int_1d_nc_3	110
38.12	int_1d_nc_4	110
38.13	int_1d_nc_5	110
38.14	int_1d_nc_6	110
38.15	int_1d_nc_7	110
38.16	int_1d_nc_7_hardy	111
38.17	int_2d_gauss_1	111
38.18	int_2d_gauss_12	111
38.19	int_2d_gauss_13	111
38.20	int_2d_gauss_16	111
38.21	int_2d_gauss_19	111
38.22	int_2d_gauss_25	111
38.23	int_2d_gauss_3	111
38.24	int_2d_gauss_33	111
38.25	int_2d_gauss_4	111
38.26	int_2d_gauss_6	112
38.27	int_2d_gauss_7	112
38.28	int_2d_gauss_9	112
38.29	int_2d_nc_10	112
38.30	int_2d_nc_15	112
38.31	int_2d_nc_21	112
38.32	int_2d_nc_3	112
38.33	int_2d_nc_6	112
38.34	int_3d_gauss_1	112
38.35	int_3d_gauss_11	112
38.36	int_3d_gauss_14	113
38.37	int_3d_gauss_15	113
38.38	int_3d_gauss_24	113
38.39	int_3d_gauss_4	113
38.40	int_3d_gauss_45	113
38.41	int_3d_gauss_5	113
38.42	int_3d_nc_11	113
38.43	int_3d_nc_4	113
38.44	int_3d_nc_6	113
38.45	int_3d_nc_8	113
39 numerical-methods/finite-element		114
39.1	interpolation_matrix	114
39.2	mark	114
39.3	mark_1d	114
39.4	mesh_1d_uniform	114

39.5	mesh_3d_uniform	114
39.6	mesh_interpolate	114
39.7	neighbour_1d	114
39.8	old	114
39.9	pdeeig_1d	114
39.10	pdeeig_2d	115
39.11	pdeeig_3d	115
39.12	polynomial_derivative_1d	115
39.13	potential_const	115
39.14	potential_coulomb	115
39.15	potential_harmonic_oscillator	115
39.16	project_circle	115
39.17	project_rectangle	115
39.18	promote_1d_2_3	115
39.19	promote_1d_2_4	115
39.20	promote_1d_2_5	116
39.21	promote_1d_2_6	116
39.22	quadrilaterate	116
39.23	recalculate_regularity_2d	116
39.24	refine_1d	116
39.25	refine_2d_21	116
39.26	refine_2d_structural	116
39.27	regularity_1d	116
39.28	regularity_2d	116
39.29	regularity_3d	117
39.30	relocate_2d	117
39.31	test_circmesh	117
39.32	test_hermite	117
39.33	tri_assign_points	117
39.34	triangulation_uniform	117
39.35	vander_1d	117
39.36	vanderd_1d	117
39.37	vanderi_1d	117
40	numerical-methods/finite-volume/@Advection	118
40.1	Advection	118
40.2	dot_advection	118
41	numerical-methods/finite-volume/@Burgers	118
41.1	burgers_split	118
41.2	dot_burgers_fdm	118
41.3	dot_burgers_fft	118
42	numerical-methods/finite-volume/@Finite_Volume	118

42.1	Finite_Volume	118
42.2	apply_bc	119
42.3	solve	119
42.4	step_split_strang	119
42.5	step_unsplit	119
43	numerical-methods/finite-volume/@Flux_Limiter	119
43.1	Flux_Limiter	119
43.2	beam_warming	119
43.3	fromm	120
43.4	lax_wendroff	120
43.5	minmod	120
43.6	monotized_central	120
43.7	muscl	120
43.8	superbee	120
43.9	upwind	120
43.10	vanLeer	121
44	numerical-methods/finite-volume/@KDV	121
44.1	dot_kdv_fdm	121
44.2	dot_kdv_fft	121
44.3	kdv_split	121
45	numerical-methods/finite-volume/@Reconstruct_Average_Evolve	121
45.1	Reconstruct_Average_Evolve	121
45.2	advect_highres	122
45.3	advect_lowress	122
46	numerical-methods/finite-volume	122
46.1	Godunov	122
46.2	Lax_Friedrich	122
46.3	Measure	122
46.4	Roe	122
46.5	fv_swe	123
46.6	staggered_euler	123
46.7	staggered_grid	123
47	numerical-methods	123
47.1	grid2quad	123
48	numerical-methods/integration	123
48.1	cumintL	123
48.2	cumintR	123
48.3	int_trapezoidal	123

49	numerical-methods/interpolation/@Kriging	124
49.1	Kriging	124
49.2	estimate_semivariance	124
49.3	interpolate_	124
50	numerical-methods/interpolation/@RegularizedInterpolator1	124
50.1	RegularizedInterpolator1	124
50.2	init	124
51	numerical-methods/interpolation/@RegularizedInterpolator2	125
51.1	RegularizedInterpolator2	125
51.2	init	125
52	numerical-methods/interpolation/@RegularizedInterpolator3	125
52.1	RegularizedInterpolator3	125
52.2	init	125
53	numerical-methods/interpolation	125
53.1	IDW	125
53.2	IPoly	125
53.3	IRBM	126
53.4	ISparse	126
53.5	Inn	126
53.6	Interpolator	126
53.7	fixnan	126
53.8	idw1	126
53.9	idw2	126
53.10	inner2outer	127
53.11	inner2outer2	127
53.12	interp1_limited	127
53.13	interp1_man	127
53.14	interp1_piecewise_linear	127
53.15	interp1_save	127
53.16	interp1_slope	128
53.17	interp1_smooth	128
53.18	interp1_unique	128
53.19	interp2_man	128
53.20	interp_angle	128
53.21	interp_fourier	128
53.22	interp_fourier_batch	128
53.23	interp_sn	129
53.24	interp_sn2	129
53.25	interp_sn3	129
53.26	interp_sn_	129

53.27	limit_by_distance_1d	129
53.28	resample1	129
53.29	resample_d_min	129
53.30	resample_vector	130
53.31	test_interp1_limited	130
54	numerical-methods	130
54.1	inverse_complex	130
54.2	maccormack_step	130
55	numerical-methods/ode/@Time_Stepper	130
55.1	Time_Stepper	130
55.2	solve	130
56	numerical-methods/ode	130
56.1	bvp1c	130
56.2	bvp1c_assemble	130
56.3	bvp1c_assemble_Q	131
56.4	bvp2_check_arguments	131
56.5	bvp2c	131
56.6	bvp2c_assemble	131
56.7	bvp2c_derivative	131
56.8	bvp2c_resample	131
56.9	bvp2fdm	132
56.10	bvp2wavetrain	132
56.11	bvp2wavetwopass	132
56.12	ivp_euler_forward	132
56.13	ivprk2	132
56.14	ode2_matrix	133
56.15	ode2characteristic	133
56.16	step_trapezoidal	133
56.17	test_bvp2	133
57	numerical-methods/optimisation	133
57.1	armijo_stopping_criterion	133
57.2	astar	133
57.3	binsearch	133
57.4	bisection	134
57.5	box1	134
57.6	box2	134
57.7	cauchy	134
57.8	cauchy2	134
57.9	directional_derivative	134
57.10	dud	134

57.11	extreme3	135
57.12	extreme_quadratic	135
57.13	ftest	135
57.14	fzero_bisect	135
57.15	fzero_newton	135
57.16	grad	135
57.17	hessian	136
57.18	hessian_from_gradient	136
57.19	hessian_projected	136
57.20	line_search	136
57.21	line_search2	136
57.22	line_search_polynomial	136
57.23	line_search_polynomial2	137
57.24	line_search_quadratic	137
57.25	line_search_quadratic2	137
57.26	line_search_wolfe	137
57.27	ls_bgfs	137
57.28	ls_broyden	138
57.29	ls_generalized_secant	138
57.30	nlcg	138
57.31	nlls	138
57.32	picard	138
57.33	poly_extrema	139
57.34	quadratic_function	139
57.35	quadratic_programming	139
57.36	quadratic_step	139
57.37	rosenbrock	139
57.38	sqrt_heron	139
57.39	test_directional_derivative	139
57.40	test_dud	139
57.41	test_fzero_newton	139
57.42	test_line_search_quadratic2	140
57.43	test_ls_generalized_secant	140
57.44	test_nlcg_6_order	140
57.45	test_nlls	140
58	numerical-methods/pde	140
58.1	laplacian2d_fundamental_solution	140
59	numerical-methods/piecewise-polynomials	140
59.1	Hermite1	140
59.2	hp2_fit	140
59.3	hp2_predict	141
59.4	hp_predict	141

59.5	hp_regress	141
59.6	lp_count	141
59.7	lp_predict	141
59.8	lp_regress	141
59.9	lp_regress_	141
60	numerical-methods	141
60.1	test_adams_bashforth	141
61	regression/@PolyOLS	142
61.1	PolyOLS	142
61.2	coefftest	142
61.3	detrend	142
61.4	fit	142
61.5	fit_	142
61.6	predict	142
61.7	predict_	142
61.8	slope	142
62	regression/@PowerLS	143
62.1	PowerLS	143
62.2	fit	143
62.3	predict	143
62.4	predict_	143
63	regression/@Theil	143
63.1	Theil	143
63.2	detrend	143
63.3	fit	144
63.4	predict	144
63.5	slope	144
64	regression	144
64.1	Theil_Multivariate	144
64.2	areg	144
64.3	ginireg	144
64.4	hesssimplereg	145
64.5	lllin	145
64.6	lsq_sparam	145
64.7	polyfitd	145
64.8	regression_method_of_moments	145
64.9	robustlinreg	145
64.10	theil2	146
64.11	theil_generalised	146

64.12	total_least_squares	146
64.13	weighted_median_regression	146
65	set-theory	146
65.1	issubset	146
66	signal-processing	146
66.1	acf_effective_sample_size	146
66.2	acf_genton	147
66.3	acfar1	147
66.4	acfar1_2	147
66.5	acfar2	147
66.6	acfar2_2	147
66.7	ar1_cutoff_frequency	147
66.8	ar1_effective_sample_size	147
66.9	ar1_mse_mu_single_sample	148
66.10	ar1_mse_pop	148
66.11	ar1_mse_range	148
66.12	ar1_spectrum	148
66.13	ar1_to_tikhonov	148
66.14	ar1_var_factor	148
66.15	ar1_var_factor_	148
66.16	ar1_var_range2	149
66.17	ar1delay	149
66.18	ar1delay_old	149
66.19	ar2conv	149
66.20	ar2dof	149
66.21	ar2param	149
66.22	asymwin	150
66.23	autocorr_fft	150
66.24	bandpass	150
66.25	bandpass2	150
66.26	bartlett	150
66.27	bartlett_spectrogram	150
66.28	bin1d	150
66.29	bin2d	151
66.30	binormrnd	151
66.31	conv1_man	151
66.32	conv2_man	151
66.33	conv2z	151
66.34	conv30	151
66.35	conv_	151
66.36	conv_centered	152
66.37	convz	152

66.38	cosexpdelay	152
66.39	csmooth	152
66.40	daniell_window	152
66.41	danielle_window	152
66.42	db2neper	152
66.43	db2power	153
66.44	derive_danielle_weight	153
66.45	derive_limit_0_acfar	153
66.46	detect_peak	153
66.47	digital_low_pass_filter	153
66.48	doublesum_ij	153
66.49	effective_sample_size_to_ar1	153
66.50	filt_hodges_lehman	153
66.51	filter1	154
66.52	filter2	154
66.53	filter_	154
66.54	filteriir	154
66.55	filterp	155
66.56	filterp1	155
66.57	filterstd	155
66.58	firls_man	155
66.59	flattopwin	155
66.60	frequency_response_boxcar	155
66.61	freqz_boxcar	155
66.62	gaussfilt1	156
66.63	hanchangewin	156
66.64	hanchangewin2	156
66.65	hanwin	156
66.66	hanwin_	156
66.67	highpass	156
66.68	kaiserwin	156
66.69	kalman	156
66.70	lanczoswin	157
66.71	last	157
66.72	lowpass	157
66.73	lowpass2	157
66.74	lowpass_iir	157
66.75	lowpass_iir_symmetric	157
66.76	lowpassfilter2	157
66.77	maxfilt1	157
66.78	meanfilt1	158
66.79	medfilt1_man	158
66.80	medfilt1_man2	158
66.81	medfilt1_padded	158

66.82	medfilt1_reduced	158
66.83	mid_term_single_sample	158
66.84	minfilt1	158
66.85	mu2ar1	158
66.86	mysmooth	159
66.87	nanautocorr	159
66.88	nanmedfilt1	159
66.89	neper2db	159
66.90	peaks_man	159
66.91	polyfilt1	159
66.92	qmedfilt1	159
66.93	randar1	159
66.94	randar1_dual	160
66.95	randar2	160
66.96	randarp	160
66.97	range_window	160
66.98	rectwin	160
66.99	recursive_sum	160
66.100	select_range	160
66.101	smooth1d_parametric	160
66.102	smooth2	161
66.103	smooth_man	161
66.104	smooth_parametric	161
66.105	smooth_parametric2	161
66.106	smooth_with_splines	161
66.107	smoothfft	161
66.108	spectrogram	161
66.109	std_window	161
66.110	sum_i_lag	162
66.111	sum_ii	162
66.112	sum_ii_	162
66.113	sum_ij	162
66.114	sum_ij_	162
66.115	sum_ij_partial_	162
66.116	sum_multivar	162
66.117	test_acfar1	162
66.118	test_acfar1_2	163
66.119	test_acfar1_3	163
66.120	test_acfar1_4	163
66.121	test_acfar2	163
66.122	test_ar1_var_factor	163
66.123	test_ar1_var_factor_2	163
66.124	test_ar1_var_mu_single_sample	163
66.125	test_ar1_var_pop	163

66.126	test_ar1_var_pop_1	163
66.127	test_ar1delay	163
66.128	test_bivariate_covariance_term	164
66.129	test_convexity	164
66.130	test_lanczoswin	164
66.131	test_madcorr	164
66.132	test_randar1	164
66.133	test_randar1_multivariate	164
66.134	test_randar2	164
66.135	test_sum_ij	164
66.136	test_sum_multivar	164
66.137	test_trifilt1	164
66.138	test_wautocorr	165
66.139	test_wavelet_transform	165
66.140	test_wordfilt	165
66.141	test_xar1_mid_term	165
66.142	tikhonov_to_ar1	165
66.143	trapwin	165
66.144	trifilt1	165
66.145	triwin	165
66.146	triwin2	165
66.147	varar1	166
66.148	welch_spectrogram	166
66.149	wfilt	166
66.150	winbandpass	166
66.151	window_make_odd	166
66.152	winfilt0	166
66.153	winlength	166
66.154	wmeanfilt	167
66.155	wmedfilt	167
66.156	wordfilt	167
66.157	wordfilt_edgeworth	167
66.158	xar1	167
66.159	xcorr_man	167
67	sorting	167
67.1	sort2	167
67.2	sort2d	167
68	special-functions	168
68.1	bessel_sphere	168
68.2	digamma_man	168
68.3	hankel_sphere	168
68.4	hermite	168

68.5	legendre_man	168
68.6	neumann_sphere	168
69	statistics	169
69.1	atan_s2	169
69.2	beta_mode_to_parameter	169
69.3	coefficient_of_determination	169
69.4	conditional_expectation_normal	169
69.5	correlation_confidence_pearson	169
70	statistics/distributions	169
70.1	PDF	169
70.2	binorm_separation_coefficient	169
70.3	binormcdf	170
70.4	binormfit	170
70.5	binormpdf	170
70.6	edgeworth_cdf	170
70.7	edgeworth_pdf	170
70.8	logn_mode2param	170
70.9	logn_param2mode	170
70.10	lognpdf	171
70.11	pdfsample	171
70.12	t2cdf	171
70.13	t2inv	171
71	statistics	171
71.1	example_standard_error_of_sample_quantiles	171
71.2	f_var_finite	171
71.3	gamma_mode_to_parameter	171
71.4	gaussfit3	171
71.5	gaussfit_quantile	172
71.6	geoserr	172
71.7	geostd	172
71.8	hodges_lehmann_correlation	172
71.9	hodges_lehmann_dispersion	172
72	statistics/information-theory	172
72.1	akaike_information_criterion	172
72.2	bayesian_information_criterion	173
73	statistics	173
73.1	kurtncdf	173
73.2	kurtnpdf	173
73.3	kurtosis_bias_corrected	173

73.4	limit	173
73.5	logfactorial	173
73.6	loglogpdf	173
73.7	lognfit_quantile	174
73.8	logskewcdf	174
73.9	logskewpdf	174
74	statistics/logu	174
74.1	lambertw_numeric	174
74.2	logtrialtcdf	174
74.3	logtrialtinv	174
74.4	logtrialtmean	175
74.5	logtrialtpdf	175
74.6	logtrialtrnd	175
74.7	logtricdf	175
74.8	logtriinv	175
74.9	logtrimean	175
74.10	logtripdf	175
74.11	logtrirnd	175
74.12	logucdf	176
74.13	logucm	176
74.14	loguinv	176
74.15	logumean	176
74.16	logupdf	176
74.17	logurnd	176
74.18	loguvar	176
74.19	medlogu	176
74.20	test_logurnd	177
74.21	tricdf	177
74.22	triinv	177
74.23	trimediam	177
74.24	tripdf	177
74.25	trirnd	177
75	statistics	177
75.1	max_exprnd	177
75.2	maxnnormals	177
75.3	mean_generalized_gampdf	178
75.4	midrange	178
75.5	minavg	178
75.6	mode_man	178
76	statistics/moment-statistics	178
76.1	autocorr_man3	178

76.2	autocorr_man4	178
76.3	autocorr_man5	178
76.4	blockserr	179
76.5	comoment	179
76.6	corr_man	179
76.7	cov_man	179
76.8	dof	179
76.9	edgeworth_quantile	180
76.10	effective_sample_size	180
76.11	f_correlation	180
76.12	f_finite	180
76.13	lmean	180
76.14	lmoment	180
76.15	maskmean	180
76.16	masknanmean	181
76.17	mean1	181
76.18	mean_man	181
76.19	mse	181
76.20	nanautocorr_man1	181
76.21	nanautocorr_man2	181
76.22	nanautocorr_man4	181
76.23	nancorr	182
76.24	nancumsum	182
76.25	nanlmean	182
76.26	nanr2	182
76.27	nanrms	182
76.28	nanrmse	182
76.29	nanserr	182
76.30	nanwmean	182
76.31	nanwstd	183
76.32	nanwvar	183
76.33	nanxcorr	183
76.34	pearson	183
76.35	pearson_to_kendall	183
76.36	pool_samples	183
76.37	qmean	183
76.38	range_mean	183
76.39	rmse_	184
76.40	serr	184
76.41	serr1	184
76.42	test_kskew	184
76.43	test_qstd_kskew_optimal_p	184
76.44	wautocorr	184
76.45	wcorr	184

76.46	wcov	185
76.47	wdof	185
76.48	wkurt	185
76.49	wmean	185
76.50	wrms	185
76.51	wserr	185
76.52	wskew	185
76.53	wstd	186
76.54	wvar	186
77	statistics	186
77.1	nangeomean	186
77.2	nangeostd	186
78	statistics/nonparametric-statistics	186
78.1	kernel1d	186
78.2	kernel2d	186
79	statistics	187
79.1	normmoment	187
79.2	normpdf2	187
80	statistics/order-statistics	187
80.1	hodges_lehmann_location	187
80.2	kendall	187
80.3	kendall_to_pearson	187
80.4	mad2sd	187
80.5	madcorr	188
80.6	median2_holder	188
80.7	median_ci	188
80.8	median_man	188
80.9	mediani	188
80.10	nanmadcorr	188
80.11	nanwmedian	188
80.12	nanwquantile	189
80.13	oja_median	189
80.14	qkurtosis	189
80.15	qmoments	189
80.16	qskew	189
80.17	qskewq	190
80.18	qstdq	190
80.19	quantile1_optimisation	190
80.20	quantile2_breckling	190
80.21	quantile2_chaudhuri	190

80.22	quantile2_projected	190
80.23	quantile2_projected2	190
80.24	quantile_envelope	190
80.25	quantile_regression_simple	191
80.26	ranking	191
80.27	spatial_median	191
80.28	spatial_quantile	191
80.29	spatial_quantile2	191
80.30	spatial_quantile3	191
80.31	spatial_rank	191
80.32	spatial_sign	191
80.33	spatial_signed_rank	192
80.34	spearman	192
80.35	spearman_rank	192
80.36	spearman_to_pearson	192
80.37	wmedian	192
80.38	wquantile	192
81	statistics	192
81.1	qstd	192
81.2	quantile_extrap	192
82	statistics/random-number-generation	193
82.1	laplacernd	193
82.2	randc	193
82.3	skewness2param	193
82.4	skewpdf_central_moments	193
82.5	skewrnd	193
82.6	skewrnd2	193
83	statistics	193
83.1	range	193
83.2	resample_with_replacement	193
84	statistics/resampling-statistics/@Jackknife	194
84.1	Jackknife	194
84.2	estimated_STATIC	194
84.3	matrix1_STATIC	194
84.4	matrix2	194
85	statistics/resampling-statistics	195
85.1	block_jackknife	195
85.2	jackknife_moments	195
85.3	moving_block_jackknife	195

85.4	randblockserr	195
85.5	resample	195
86	statistics	196
86.1	scale_quantile_sd	196
86.2	sd_sample_quantiles	196
86.3	skewpdf	196
86.4	test_mean_generalized_gampdf	196
86.5	trimmed_mean	196
86.6	ttest2_man	196
86.7	ttest_man	197
86.8	ttest_paired	197
86.9	wgeomean	197
86.10	wgeovar	197
86.11	wharmean	197
86.12	wharstd	197
86.13	wharvar	197
87	mathematics	198
87.1	ternary_diagram	198
88	test/master	198
88.1	dat_test_lanczos_3d_k_20_n_40	198
88.2	poisson2d_blk	198
88.3	qr_implicit_givens_2	198
88.4	spectral_derivative_2d	198
88.5	test_2d_eigensolver_hydrogen	198
88.6	test_2d_refine	198
88.7	test_3d_eigensolver_hydrogen	198
88.8	test_FEM	199
88.9	test_Mesh_3d	199
88.10	test_arnoldi	199
88.11	test_arpackc	199
88.12	test_assemble	199
88.13	test_assembly_performance	199
88.14	test_bc_one_sided	199
88.15	test_compare_solvers	199
88.16	test_complete	199
88.17	test_convergence	199
88.18	test_convergence_b	200
88.19	test_df_2d	200
88.20	test_eig_algs	200
88.21	test_eig_inverse	200
88.22	test_eigs_lanczos	200

88.23	test_eigs_lanczos_1	200
88.24	test_eigs_lanczos_2	200
88.25	test_eigs_lanczos_performance	200
88.26	test_fdm	200
88.27	test_fdm_d_vargrid	200
88.28	test_fdm_spectral	201
88.29	test_fem	201
88.30	test_fem_1d	201
88.31	test_fem_1d_higher_order	201
88.32	test_fem_2d_adaptive	201
88.33	test_fem_2d_higher_order	201
88.34	test_fem_3d_higher_order	201
88.35	test_fem_3d_refine	201
88.36	test_fem_b	201
88.37	test_fem_derivative	201
88.38	test_fem_quadrature	202
88.39	test_final	202
88.40	test_fix_substitution	202
88.41	test_forward	202
88.42	test_get_sparse_arrays	202
88.43	test_harmonic_oscillator	202
88.44	test_high_order_fdm_periodic_bc	202
88.45	test_hydrogen_wf	202
88.46	test_ichol	202
88.47	test_interpolation	202
88.48	test_inverse_problem	203
88.49	test_it_vs_exact	203
88.50	test_jama	203
88.51	test_jd	203
88.52	test_jdqz	203
88.53	test_lanczos_2	203
88.54	test_lanczos_biorthogonal	203
88.55	test_laplacian	203
88.56	test_laplacian_non_uniform	203
88.57	test_laplacian_simple	203
88.58	test_mesh_2d_uniform	204
88.59	test_mesh_2d_uniform_2	204
88.60	test_mesh_circle	204
88.61	test_mesh_generation	204
88.62	test_mesh_interpolate	204
88.63	test_mg	204
88.64	test_minres_recycle	204
88.65	test_multigrid	204
88.66	test_nc	204

88.67	test_nonuniform_symmetric	204
88.68	test_pde	205
88.69	test_permutation	205
88.70	test_poisson_fem	205
88.71	test_polar	205
88.72	test_potential	205
88.73	test_powers	205
88.74	test_precondition	205
88.75	test_project_rectangle	205
88.76	test_qr	205
88.77	test_quantum_well	205
88.78	test_radial_adaptive	206
88.79	test_radial_confinement	206
88.80	test_radial_fixes	206
88.81	test_refine_2d	206
88.82	test_refine_2d_b	206
88.83	test_refine_3d	206
88.84	test_refine_structural	206
88.85	test_regularisation	206
88.86	test_round_off	206
88.87	test_schrödinger_potentials	206
88.88	test_uniform_mesh	207
88.89	test_vargrid	207
89	test	207
89.1	test_gaussfit3	207
89.2	test_geoserr	207
89.3	test_lognfit_quantile	207
89.4	test_max_normal	207
89.5	test_mtimes3x3	207
90	mathematics	207
90.1	vanderd_2d	207
91	wavelet	208
91.1	contiuous_wavelet_transform	208
91.2	cwt_man	208
91.3	example_wavelets	208
91.4	phasewrap	208
91.5	test_cwt_man	208
91.6	test_phasewrap	208
91.7	test_wavelet	208
91.8	test_wavelet2	208
91.9	test_wavelet_analysis	209

91.10	test_wavelet_reconstruct	209
91.11	test_wtc	209
91.12	wavelet	209
91.13	wavelet_reconstruct	209
91.14	wavelet_transform	209
92	mathematics	209
92.1	wrapphase	209

1 calendar

1.1 days_per_month

1.2 isnight

2 mathematics

mathematical functions of various kind

2.1 cast_byte_to_integer

cast byte to integer

3 complex-analysis

operations on complex numbers

3.1 complex_exp_product_im_im

product of the imaginary part of two complex exponentials

the product has two frequency components

input :

c : complex amplitudes
o : frequencies

output :

cp : amplitude of the product
op : frequencies of the product

3.2 complex_exp_product_im_re

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

the product has two frequency components

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

3.3 complex_exp_product_re_im

the product has two frequency components

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

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

3.4 complex_exp_product_re_re

product of the real part of two complex exponentials

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

the product has two frequency components

input :
 c : complex amplitudes
 o : frequencies
output :

cp : amplitude of the product
op : frequencies of the product

3.5 croots

nth-roots of a complex number

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

r : roots of the complex number

3.6 root_complex

root of a complex number

3.7 test_imroots

4 derivation

derivation of several functions by means of symbolic computation

4.1 derive_acfar1

4.2 derive_ar2param

4.3 derive_arc_length

4.4 `derive_fourier_power`

4.5 `derive_fourier_power_exp`

4.6 `derive_laplacian_curvilinear`

4.7 `derive_laplacian_fourier_piecewise_linear`

4.8 `derive_logtripdf`

4.9 `derive_smooth1d_parametric`

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

7.2 exp10

7.3 filter_twosided

8 finance

8.1 derive_skewrnd_walsh_paramter

8.2 `gbm_cdf`

8.3 `gbm_fit`

8.4 `gbm_fit_old`

8.5 `gbm_inv`

8.6 `gbm_mean`

8.7 `gbm_median`

8.8 `gbm_pdf`

8.9 `gbm_simulate`

8.10 `gbm_skewness`

8.11 `gbm_std`

8.12 `gbm_transform_time_step`

8.13 `put_price_black_scholes`

8.14 `skewgbm_simulate`

8.15 `skewrnd_walsh`

9 `finance/test`

9.1 `test_gbm`

9.2 `test_gbm_pdf`

9.3 `test_skewrnd_walsh`

10 `fourier/@STFT`

10.1 `STFT`

`class for short time fourier transform`

Note: the interval `Ti` should be set to at least `2*max(T)`, as
otherwise coefficients
tend to oscillate in the presence of noise

Note: for convenience, the independent variable is labeled as time
(t),
but the independent variable is arbitrary, so it works
likewise in space

10.2 itransform

inverse of the short time fourier transform

10.3 stft_

static wrapper for STFT

10.4 stftmat

transformation matrix for the short time fourier transform

10.5 transform

short time fourier transform

11 fourier

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

11.1 amplitude_from_peak

amplitude and standard deviation of the amplitude of a frequency
component

represented by a peak in the fourier domain

input :

h : peak height

w : peak width at half height

output:

a : amplitude in real space

s : standard deviation of the frequency (!)

11.2 dftmtx_man

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

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

output :
F : fourier matrix

11.3 example_fourier_window

11.4 fft_derivative

derivative by 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

11.5 fft_man

fast fourier transform for complex input data

input:
F : data in real space

output :

F : fourier transformation of F

11.6 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

11.7 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

11.8 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

11.9 fourier_cesaro_correction

11.10 fourier_coefficient_piecewise_linear

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

11.11 fourier_coefficient_piecewise_linear_1

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

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

output :
ab : coefficients for frequency components

11.12 fourier_coefficient_ramp3

fourier series coefficient of a ramp

11.13 `fourier_coefficient_ramp_pulse`

fourier series coefficient of a ramp pules

11.14 `fourier_coefficient_ramp_step`

fourier coefficient of a ramp-step

11.15 `fourier_coefficient_square_pulse`

fourier series coefficients of a square pulse

11.16 `fourier_cubic_interaction_coefficients`

11.17 `fourier_derivative`

coefficients of the derivative of a fourier series
not of discrete fourier transform (fft)

11.18 `fourier_expand`

expand values of fourier series

11.19 `fourier_fit`

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

11.20 `fourier_interpolate`

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

11.21 `fourier_matrix`

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

11.22 `fourier_matrix2`

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

11.23 `fourier_matrix3`

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

11.24 `fourier_matrix_exp`

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

11.25 `fourier_multiplicative_interaction_coefficients`

11.26 `fourier_power`

powers of a continuous fourier series in sin/cos form

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

frequencies higher than 2ω ignored in input
frequencies higher than 3ω not computed

11.27 `fourier_power_exp`

powers of the continuous fourier series

$$a^p = (u_r + u_1 \sin(\omega t) + u_2 \sin(\omega t + \phi))^p$$

phase of first component assumed 0

higher orders than 2 ignored input

higher order than 3 not computed in output

$$y = a_0 + \sum (a_j \sin(j\omega t) + b_j \cos(j\omega t))$$

$$= \text{Real}(\sum_{i=0}^{\infty} c_i \exp(i\omega t)), \quad c_i = a_i + b_i$$

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

11.28 `fourier_predict`

expand a continuous fourier series at times t

11.29 `fourier_quadratic_interaction_coefficients`

11.30 `fourier_range`

approximate range of a continuous Fourier series with 2 components

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

11.31 `fourier_regress`

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

11.32 `fourier_resampled_fit`

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

11.33 `fourier_resampled_predict`

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

11.34 `fourier_signed_square`

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

11.35 `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

11.36 `hyperbolic_fourier_box`

11.37 `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)

11.38 laplace_2d_pwlinear

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

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

11.39 nanfft

discrete fourier transform of a data series with gaps

11.40 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*y_peak, p = 0.5

works best, when spectrum has been smoothened

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

output :

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

11.41 roots_fourier

zeros of continuous fourier series series

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

11.42 spectral_density

spectral density

11.43 test_complex_exp_product

11.44 test_fourier_filter

11.45 test_idftmtx

12 mathematics

mathematical functions of various kind

12.1 gaussfit_quantile

13 geometry/@Geometry

13.1 Geometry

13.2 arclength

arc length of a two dimensional curve

8th order accurate

does not require the segments length to vary smoothly

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

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

13.3 arclength_old

arc length of a two dimensional function

13.4 arclength_old2

arc length of a two dimensional function

13.5 base_point

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

13.6 base_point_limited

base point (Fusspunkt) of a point on a line

13.7 centroid

centroid of a polygone

13.8 cosa_min_max

13.9 cross2

cross product in two dimensions

13.10 curvature

curvature of a function in two dimensions

13.11 ddot

sum of squares of cos of inner angles of triangle

13.12 distance

euclidian distance between two points

13.13 distance2

euclidean distance between two points
this function requires a and b 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

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`

n-points on an ellipse

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 rotate to streamwise coordinates to $R = 1/dv/ds * u$

15 histogram/@Histogram

15.1 2x

15.2 Histogram

15.3 bimodes

15.4 cdf

15.5 cdfS

15.6 chi2test

15.7 `cmoment`

15.8 `cmomentS`

15.9 `entropy`

15.10 `entropyS`

15.11 `iquantile`

15.12 `kstest`

15.13 `kurtosis`

15.14 `kurtosisS`

15.15 `mean`

15.16 `meanS`

15.17 median

15.18 medianS

15.19 mode

15.20 modeS

15.21 moment

15.22 momentS

15.23 pdf

15.24 quantile

15.25 quantileS

15.26 setup

15.27 skewness

15.28 skewnessS

15.29 stairs

15.30 stairsS

15.31 std

15.32 stdS

15.33 var

15.34 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 linear-algebra

17.1 averaging_matrix_2

17.2 colnorm

norms of columns

17.3 condest_

estimation of the condition number

18 linear-algebra/coordinate-transformation

18.1 barycentric2cartesian

barycentric to cartesian coordinates

18.2 barycentric2cartesian3

convert barycentric to cartesian coordinates

18.3 cartesian2barycentric

cartesian to barycentric coordinates

18.4 cartesian_to_unit_triangle_basis

transform coodinates into unit triangle

18.5 ellipsoid2geoid

18.6 example_approximate_utm_conversion

18.7 latlon2utm

transform latitude and longitude to WGS84 UTM

18.8 latlon2utm_simple

18.9 lowrance_mercator_to_wgs84

convert lowrance coordinates to wgs84

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

18.10 nmea2utm

convert nmea messages to utm coordinates

18.11 sn2xy

convert sn to xy coordinates

18.12 unit_triangle_to_cartesian

transform coordinates in unit triangle to cartesian coordinates

18.13 utm2latlon

convert wgs84 utm to latitude and longitude

18.14 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

18.15 xy2sn

convert cartesian to streamwise coordiantes

18.16 xy2sn_java

use java port for speed up

18.17 xy2sn_old

transform points from cartesian into streamwise coordinates

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

19 linear-algebra

19.1 det2x2

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

19.2 det3x3

determinant of stacked 3x3 matrices

19.3 det4x4

determinant of stacked 4x4 matrices

19.4 diag2x2

diagonal of stacked 2x2 matrices

19.5 eig2x2

eigenvalues of stacked 2x2 matrices

20 linear-algebra/eigenvalue

20.1 eig_bisection

20.2 eig_inverse

20.3 eig_inverse_iteration

20.4 eig_power_iteration

21 linear-algebra/eigenvalue/jacobi-davidson

21.1 afun_jdm

21.2 davidson

21.3 jacobi_davidson

21.4 jacobi_davidson_qr

21.5 jacobi_davidson_qz

21.6 jacobi_davidson_simple

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

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

```

```

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

```

21.8 jdqr_sleijpen

```

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

```

```

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

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

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

```

```

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

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
W=V*Q; V=V(:,1:j)/R; E=E/R; R=eye(j); M=Q(1:j,:)' /R;
W=V*H; V(:,j+1)=[];R=R'*R; M=H(1:j,:)' ;
%===== ARNOLDI (for initializing spaces)
=====
%===== END ARNOLDI
=====
% not accurate enough M=Rw'\(M/Rv);
%===== COMPUTE SORTED JORDAN FORM
=====
% compute vectors and matrices for skew projection
% solve preconditioned system
% 0 step of bicgstab eq. 1 step of bicgstab
% Then x is a multiple of b
% HIST=[0,1];
% explicit preconditioning
% compute norm in l-space
% HIST=[HIST;[nmv,rnorm/snorm]];

```

```

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

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

```

21.9 jdqr_vorst

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

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

```

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

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
W=V*Q; V=V(:,1:j)/R; E=E/R; R=eye(j); M=Q(1:j,:)' /R;
W=V*H; V(:,j+1)=[];R=R'*R; M=H(1:j,:)' ;
%===== ARNOLDI (for initializing spaces)
=====
%===== END ARNOLDI
=====
% not accurate enough M=Rw'\(M/Rv);
%===== COMPUTE SORTED JORDAN FORM
=====
% accepted separation between eigenvalues:
% no preconditioning
% solve left preconditioned system
% compute vectors and matrices for skew projection
% precondition and project r
% solve preconditioned system
% no preconditioning
% solve two-sided expl. preconditioned system
% compute vectors and matrices for skew projection
% precondition and project r
% solve preconditioned system
% "unprecondition" solution
%%% u(:,j+1)=Atilde*u(:,j)
%%% r(:,j+1)=Atilde*r(:,j)
%----- compute schur form -----
A*Q=Q*S, Q'*Q=eye(size(A));
% transform real schur form to complex schur form
%----- find order eigenvalues -----
%----- reorder schur form -----
%----- compute qz form -----
%----- sort eigenvalues -----
%----- sort qz form -----
% i>j, move ith eigenvalue to position j
% determine dimension
% defaults

```

21.10 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
%=====

%=====

%=====

%=====
```

21.11 mfunc_jdm

21.12 mgs

21.13 minres_

21.14 mv_jacobi_davidson

22 linear-algebra

22.1 first

22.2 gershgorin_circle

range of eigenvalues determined by the gershgorin circle theorem

22.3 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

22.4 eig2x2

reconstruct matrix from eigenvalue decomposition

22.5 inv2x2

2x2 inverse of stacked matrices

22.6 inv3x3

22.7 inv4x4

inverse of stacked 4x4 matrices

23 linear-algebra/lanczos

23.1 arnoldi

23.2 arnoldi_new

23.3 `eigs_lanczos_man`

23.4 `lanczos`

23.5 `lanczos_`

23.6 `lanczos_biorthogonal`

23.7 `lanczos_biorthogonal_improved`

23.8 `lanczos_ghep`

23.9 `mv_lanczos`

23.10 `reorthogonalise`

23.11 `test_lanczos`

24 linear-algebra/linear-systems

24.1 gmres_man

break on convergence

24.2 minres_recycle

25 linear-algebra

25.1 lpmean

mean of pth-power of a

25.2 lpnorm

norm of lth-power of a

25.3 matvec3

matrix-vector product of stacked matrices and vectors

25.4 max2d

maximum value and i-j index for matrix

25.5 mid

mid point between neighbouring vector elements

25.6 mpoweri

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

25.7 mtimes2x2

25.8 mtimes3x3

product of stacked 3x3 matrices

25.9 nannorm

norm of a vector, skips nan-values

25.10 nanshift

shift vector, but set out of range values to NaN

25.11 nl

number rows (lines) of a matrix

analogue to unix nl command

25.12 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)

25.13 `normalize1`

normalize columns in x to [-1,1]

25.14 `normrows`

25.15 `orth2`

make matrix A orthogonal to B

25.16 `orth_man`

orthogonalize the columns of A

25.17 `orthogonalise`

make x orthogonal to Y

25.18 `paddext`

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

25.19 `paddval1`

padd values at end of x

25.20 `paddval2`

padd values to x

26 linear-algebra/polynomial

26.1 chebychev

chebycheff polynomials

26.2 piecewise_polynomial

evaluate piecewise polynomial

26.3 roots1

roots of linear functions

26.4 roots2

roots of quadratic function
 $c_1 x^2 + c_2 x + c_3 = 0$

26.5 roots2poly

26.6 roots3

26.7 roots4

26.8 roots_piecewise_linear

26.9 test_roots4

26.10 vanderi_1d

vandermonde matrix of an integral

27 linear-algebra

27.1 randrot

random rotation matrix

27.2 right

get right column by shifting columns to left
extrapolate rightmost column

27.3 rot2

rotation matrix from angle

27.4 rot2dir

rotation matrix from direction vector

27.5 rot3

27.6 rotR

27.7 rownorm

27.8 simmilarity_matrix

27.9 spnorm

frobenius norm

27.10 spzeros

allocate a sparze matrix of zeros

27.11 test_roots3

27.12 transform_minmax

27.13 transpose3

transpose stacked 3x3 matrices

27.14 transposeall

28 logic

bitwise operations on integers

28.1 bitor_man

bitwise OR of the numbers of the columns of A

input:
A (positive integer)

29 master/plot

29.1 attach_boundary_value

29.2 cartesian_polar

29.3 img_vargrid

29.4 plot_basis_functions

29.5 plot_convergence

29.6 plot_dof

29.7 plot_eigenbar

29.8 `plot_error_estimation`

29.9 `plot_error_estimation_2`

29.10 `plot_error_fem`

29.11 `plot_fdm_kernel`

29.12 `plot_fdm_vs_fem`

29.13 `plot_fem_accuracy`

29.14 `plot_function_and_grid`

29.15 `plot_hat`

29.16 `plot_hydrogen_wf`

29.17 `plot_mesh`

29.18 `plot_mesh_2`

29.19 `plot_refine`

29.20 `plot_refine_3d`

29.21 `plot_runtime`

29.22 `plot_spectrum`

29.23 `plot_wavefunction`

30 `master/ported`

30.1 `assemble_2d_phi_phi`

30.2 `assemble_3d_dphi_dphi`

30.3 `assemble_3d_phi_phi`

30.4 `dV_2d_`

30.5 `derivative_2d`

30.6 `derivative_3d`

30.7 `element_neighbour_2d`

30.8 `prefetch_2d_`

30.9 `promote_2d_3_10`

30.10 `promote_2d_3_15`

30.11 `promote_2d_3_21`

30.12 `promote_2d_3_6`

30.13 `promote_3d_4_10`

30.14 `promote_3d_4_20`

30.15 `promote_3d_4_35`

30.16 `vander_2d`

30.17 `vander_3d`

31 `mathematics`

mathematical functions of various kind

31.1 `myexp`

32 `number-theory`

32.1 `ceiln`

floor to leading n-digits

32.2 `digitsb`

number of digits with respect to specified base

32.3 `floorn`

floor to n-digits

32.4 iseven

true for even numbers in X

32.5 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

32.6 nchoosek_man

vectorised binomial coefficient
 $b = N!/K!(N-K)!$

32.7 pythagorean_triple

pythagorean triple

32.8 roundn

round to n digits

33 numerical-methods/differentiation

33.1 derivative1

first derivative on variable mesh
second order accurate

33.2 derivative2

second derivative on a variable mesh

34 numerical-methods/finite-difference

34.1 cdiff

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

34.2 cdifb

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

34.3 central_difference

34.4 cmean

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

34.5 cmean2

34.6 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)

34.7 derivative_matrix_2_1d

finite derivative matrix of second derivative in one dimension

34.8 derivative_matrix_2d

finite difference derivative matrix in two dimensions

34.9 derivative_matrix_curvilinear

derivative matrix on a curvilinear grid

34.10 derivative_matrix_curvilinear_2

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

34.11 difference_kernel

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

34.12 distmat

distance matrix for a 2 dimensional rectangular matrix

34.13 downwind_difference

34.14 gradpde2d

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

34.15 laplacian

34.16 laplacian_fdm

finite difference matrix of the laplacian
BC

34.17 left

left element of vector, leftmost column is extrapolated

34.18 lrmean

mean of the left and right element

35 numerical-methods/finite-difference/master

35.1 fdm_adaptive_grid

35.2 fdm_adaptive_refinement_old

35.3 `fdm_assemble_d1_2d`

35.4 `fdm_assemble_d2_2d`

35.5 `fdm_confinement`

35.6 `fdm_d_vargrid`

35.7 `fdm_h_unstructured`

35.8 `fdm_hydrogen_vargrid`

35.9 `fdm_mark_unstructured_2d`

35.10 `fdm_plot`

35.11 `fdm_plot_series`

35.12 `fdm_refine_2d`

35.13 `fdm_refine_3d`

35.14 `fdm_refine_unstructured_2d`

35.15 `fdm_schroedinger_2d`

35.16 `fdm_schroedinger_3d`

35.17 `relocate`

36 `numerical-methods/finite-difference`

36.1 `mid`

`mid point between neighbouring vector elements`

36.2 `pwmid`

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

36.3 `ratio`

`ratio of two subsequent values`

36.4 steplength

step length of a vector if it were equispaced

36.5 swapoddeven

swap odd and even elements in a vector

36.6 test_derivative_matrix_2d

36.7 test_derivative_matrix_curvilinear

36.8 test_difference_kernel

36.9 upwind_difference

37 numerical-methods/finite-element

37.1 Mesh_2d.java

37.2 Tree_2d.java

37.3 assemble_1d_dphi_dphi

37.4 assemble_1d_phi_phi

37.5 assemble_2d_dphi_dphi_java

37.6 assemble_2d_phi_phi_java

37.7 assemble_3d_dphi_dphi_java

37.8 assemble_3d_phi_phi_java

37.9 boundary_1d

37.10 boundary_2d

37.11 boundary_3d

37.12 check_area_2d

37.13 circmesh

37.14 `cropradius`

37.15 `display_2d`

37.16 `display_3d`

37.17 `distort`

37.18 `err_2d`

37.19 `estimate_err_2d_3`

37.20 `example_1d`

37.21 `example_2d`

37.22 `explode`

37.23 `fem_2d`

37.24 `fem_2d_heuristic_mesh`

37.25 `fem_get_2d_radial`

37.26 `fem_interpolation`

37.27 `fem_plot_1d`

37.28 `fem_plot_1d_series`

37.29 `fem_plot_2d`

37.30 `fem_plot_2d_series`

37.31 `fem_plot_3d`

37.32 `fem_plot_3d_series`

37.33 `fem_plot_confine_series`

37.34 fem_radial

adaptive grid
constant grid

37.35 flip_2d

37.36 get_mesh_arrays

37.37 hashkey

38 numerical-methods/finite-element/int

38.1 int_1d_gauss

38.2 int_1d_gauss_1

38.3 int_1d_gauss_2

38.4 int_1d_gauss_3

38.5 int_1d_gauss_4

38.6 int_1d_gauss_5

38.7 int_1d_gauss_6

38.8 int_1d_gauss_lobatto

38.9 int_1d_gauss_n

38.10 int_1d_nc_2

38.11 int_1d_nc_3

38.12 int_1d_nc_4

38.13 int_1d_nc_5

38.14 int_1d_nc_6

38.15 int_1d_nc_7

38.16 int_1d_nc_7_hardy

38.17 int_2d_gauss_1

38.18 int_2d_gauss_12

38.19 int_2d_gauss_13

38.20 int_2d_gauss_16

38.21 int_2d_gauss_19

38.22 int_2d_gauss_25

38.23 int_2d_gauss_3

38.24 int_2d_gauss_33

38.25 int_2d_gauss_4

38.26 int_2d_gauss_6

38.27 int_2d_gauss_7

38.28 int_2d_gauss_9

38.29 int_2d_nc_10

38.30 int_2d_nc_15

38.31 int_2d_nc_21

38.32 int_2d_nc_3

38.33 int_2d_nc_6

38.34 int_3d_gauss_1

38.35 int_3d_gauss_11

38.36 int_3d_gauss_14

38.37 int_3d_gauss_15

38.38 int_3d_gauss_24

38.39 int_3d_gauss_4

38.40 int_3d_gauss_45

38.41 int_3d_gauss_5

38.42 int_3d_nc_11

38.43 int_3d_nc_4

38.44 int_3d_nc_6

38.45 int_3d_nc_8

39 numerical-methods/finite-element

39.1 interpolation_matrix

39.2 mark

39.3 mark_1d

39.4 mesh_1d_uniform

39.5 mesh_3d_uniform

39.6 mesh_interpolate

39.7 neighbour_1d

39.8 old

39.9 pdeeig_1d

39.10 pde eig_2d

39.11 pde eig_3d

39.12 polynomial_derivative_1d

39.13 potential_const

39.14 potential_coulomb

39.15 potential_harmonic_oscillator

39.16 project_circle

39.17 project_rectangle

39.18 promote_1d_2_3

39.19 promote_1d_2_4

39.20 promote_1d_2_5

39.21 promote_1d_2_6

39.22 quadrilaterate

39.23 recalculate_regularity_2d

39.24 refine_1d

39.25 refine_2d_21

39.26 refine_2d_structural

39.27 regularity_1d

39.28 regularity_2d

39.29 regularity_3d

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

39.30 relocate_2d

39.31 test_circmesh

39.32 test_hermite

39.33 tri_assign_points

39.34 triangulation_uniform

39.35 vander_1d

van der Monde matrix

39.36 vanderd_1d

39.37 vanderi_1d

40 numerical-methods/finite-volume/@Advection

40.1 Advection

FVM treatment of the Advection equation

40.2 dot_advection

advection equation

41 numerical-methods/finite-volume/@Burgers

41.1 burgers_split

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

41.2 dot_burgers_fdm

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

41.3 dot_burgers_fft

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

42 numerical-methods/finite-volume/@Finite_Volume

42.1 Finite_Volume

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

42.2 apply_bc

apply boundary conditions

42.3 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);
```

42.4 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

42.5 step_unsplit

step in time, without splitting the inhomogeneous term

43 numerical-methods/finite-volume/@Flux_Limiter

43.1 Flux_Limiter

class of flux limiters

43.2 beam_warming

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

43.3 fromm

fromme limiter
low res

43.4 lax_wendroff

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

43.5 minmod

min-mod schock limiter

43.6 monotized_central

monotonized central flux limiter

43.7 muscl

muscl flux limiter

43.8 superbee

superbee limiter

43.9 upwind

godunov scheme
godunov, first order accurate

43.10 vanLeer

van Leer limiter

44 numerical-methods/finite-volume/@KDV

44.1 dot_kdv_fdm

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

44.2 dot_kdv_fft

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

44.3 kdv_split

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

45 numerical-methods/finite-volume/@Reconstruct_Average_Evolve

45.1 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;
```

45.2 advect_highres

single time step for the reconstruct evolve algorithm

45.3 advect_lowres

single time step
low resolution

46 numerical-methods/finite-volume

46.1 Godunov

Godunov, upwind method for systems of pdes

46.2 Lax_Friedrich

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

46.3 Measure

46.4 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

46.5 fv_swe

wrapper for solving SWE

46.6 staggered_euler

forward euler method with staggered grid

46.7 staggered_grid

staggered grid approximation to the SWE

47 numerical-methods

47.1 grid2quad

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

48 numerical-methods/integration

48.1 cumintL

cumulative integral from left to right

48.2 cumintR

cumulative integral from right to left

48.3 int_trapezoidal

integrate y along x with the trapezoidal rule

49 numerical-methods/interpolation/@Kriging

49.1 Kriging

```
class for Kriging interpolation
```

49.2 estimate_semivariance

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

49.3 interpolate_

```
interpolate with Krieking method

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

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

50 numerical-methods/interpolation/@RegularizedInterpolator

50.1 RegularizedInterpolator1

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

50.2 init

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

51 numerical-methods/interpolation/@RegularizedInterpolator2

51.1 RegularizedInterpolator2

class for regularized interpolation on an unstructures mesh (
interpolation)

51.2 init

initialize the interpolator with a set of point samples

52 numerical-methods/interpolation/@RegularizedInterpolator3

52.1 RegularizedInterpolator3

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

52.2 init

initialize the interpolator with a set of sampling points

53 numerical-methods/interpolation

53.1 IDW

spatial averaging by inverse distance weighting

53.2 IPoly

polynomial interpolation class

53.3 IRBM

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

53.4 ISparse

```
sparse interpolation class
```

53.5 Inn

```
nearest neighbour interpolation
```

53.6 Interpolator

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

53.7 fixnan

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

53.8 idw1

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

53.9 idw2

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

53.10 inner2outer

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

53.11 inner2outer2

interpolate from element (segment) centres to edge points

53.12 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

53.13 interp1_man

interpolate

53.14 interp1_piecewise_linear

53.15 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

53.16 `interp1_slope`

quadratic interpolation returning value and derivative(s)

53.17 `interp1_smooth`

53.18 `interp1_unique`

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

53.19 `interp2_man`

nearest neighbour interpolation in two dimensions

53.20 `interp_angle`

interpolate an angle

53.21 `interp_fourier`

interpolation by the fourier method

53.22 `interp_fourier_batch`

batch interpolation by the fourier interpolation

53.23 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_);

53.24 interp_sn2

interpolation in streamwise coordinates

53.25 interp_sn3

53.26 interp_sn_

53.27 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
 $ratio^{-1} = \exp(a*1)$

53.28 resample1

interpolation along a parametric curve with variable step width

53.29 resample_d_min

resample a function

53.30 `resample_vector`

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

53.31 `test_interp1_limited`

54 `numerical-methods`

54.1 `inverse_complex`

54.2 `maccormack_step`

55 `numerical-methods/ode/@Time_Stepper`

55.1 `Time_Stepper`

55.2 `solve`

56 `numerical-methods/ode`

56.1 `bvp1c`

56.2 `bvp1c_assemble`

56.3 bvp1c_assemble_Q

56.4 bvp2_check_arguments

56.5 bvp2c

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])

56.6 bvp2c_assemble

56.7 bvp2c_derivative

56.8 bvp2c_resample

56.9 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])

56.10 bvp2wavetrain

solve second order boundary value problem by repeated integration

56.11 bvp2wavetwopass

two pass solution for the linearised wave equation

solve first for the wave number k, and then for y

56.12 ivp_euler_forward

solve initial value problem by the euler forward method

56.13 ivprk2

solve initial value problem by the two step runge kutta method

56.14 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]
```

56.15 ode2characteristic

second order odes
transmitted and reflected wave

56.16 step_trapezoidal

single trapezoidal step

56.17 test_bvp2

57 numerical-methods/optimisation

57.1 armijo_stopping_criterion

armijo stopping criterion for optimizations

57.2 astar

astar path finding algorithm

57.3 binsearch

binary search on a line

57.4 bisection

bisection

57.5 box1

test objective function for optimisation routines

57.6 box2

57.7 cauchy

57.8 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

57.9 directional_derivative

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

57.10 dud

optimization by the dud algorithm

57.11 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

57.12 extreme_quadratic

57.13 ftest

57.14 fzero_bisect

57.15 fzero_newton

57.16 grad

numerical gradient

57.17 hessian

numerical hessian

57.18 hessian_from_gradient

numerical hessian from gradient

57.19 hessian_projected

numerical hessian projected to one dimension

57.20 line_search

bisection routine

57.21 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

57.22 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

57.23 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

57.24 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

57.25 line_search_quadratic2

quadratic line search

57.26 line_search_wolfe

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

57.27 ls_bgfs

least squares by the bgfs method

57.28 ls_broyden

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

BGFS:
Broyden 1965
Fletcher 1970
Goldfarb 1970
Shanno 1970

57.29 ls_generalized_secant

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

57.30 nlcg

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

57.31 nlls

non-linear least squares

57.32 picard

picard iteration

57.33 poly_extrema

extrema of a polynomial

57.34 quadratic_function

evaluate quadratic function in higher dimensions

57.35 quadratic_programming

optimize by quadratic programming

57.36 quadratic_step

single step of the quadratic programming

57.37 rosenbrock

rosenbrock test function

57.38 sqrt_heron

Heron's method for the square root

57.39 test_directional_derivative

57.40 test_dud

57.41 test_fzero_newton

57.42 test_line_search_quadratic2

57.43 test_ls_generalized_secant

57.44 test_nlcg_6_order

57.45 test_nlls

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

58 numerical-methods/pde

58.1 laplacian2d_fundamental_solution

59 numerical-methods/piecewise-polynomials

59.1 Hermite1

hermite polynomial interpolation in 1d

59.2 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)
```

59.3 hp2_predict

prediction with pw hermite polynomial
c are values at support points

59.4 hp_predict

predict with piecewise hermite polynomial

59.5 hp_regress

fit piecewise hermite polynomial
coefficients are values and derivatives

59.6 lp_count

lagrangian basis for interpolation
count number of valid samples

59.7 lp_predict

lagrangian basis piecwie interpolation, predicor

59.8 lp_regress

59.9 lp_regress_

60 numerical-methods

60.1 test_adams_bashforth

61 regression/@PolyOLS

61.1 PolyOLS

class for polynomial least squares

61.2 coefftest

61.3 detrend

detrending by polynomial regression

61.4 fit

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

61.5 fit_

fit a polynomial function

61.6 predict

predict polynomial function values

61.7 predict_

61.8 slope

slope by linear regression

62 regression/@PowerLS

62.1 PowerLS

class for power law regression

62.2 fit

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

62.3 predict

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

62.4 predict_

63 regression/@Theil

63.1 Theil

Kendal-Theil-Sen robust regression

63.2 detrend

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

63.3 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

63.4 predict

predict values and confidence intervals with the Theil-Sen method

63.5 slope

fit the slope with the Theil-Sen method

64 regression

linear and non-linear regression

64.1 Theil_Multivariate

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

64.2 areg

regression using the pth-fraction of samples with smallest residual

64.3 ginireg

gini regression

64.4 hesssimplereg

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

64.5 llin

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

64.6 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)

64.7 polyfitd

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

 x_0 must contain at least for conditioning as otherwise the
intercept
cannot be determined

64.8 regression_method_of_moments

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

64.9 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

64.10 theil2

Theil senn-estimator for two dimensions (glm)

64.11 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

64.12 total_least_squares

total least squares

64.13 weighted_median_regression

weighted median regression
c.f. Scholz, 1978

65 set-theory

65.1 issubset

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

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

66 signal-processing

66.1 acf_effective_sample_size

effective sample size from acf

66.2 acf_genton

autocorrelation function

66.3 acfar1

Autocorrelation function of the finite AR1 process

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

66.4 acfar1_2

autocorrelation of the ar1 process

66.5 acfar2

impulse response of the ar2 process

66.6 acfar2_2

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

66.7 ar1_cutoff_frequency

66.8 ar1_effective_sample_size

effective sample size correction for autocorrelated series

66.9 ar1_mse_mu_single_sample

standard error of a single sample of an ar1 correlated process

66.10 ar1_mse_pop

variance of the population mean of a single realisation around zero

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

66.11 ar1_mse_range

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

66.12 ar1_spectrum

spectrum of the ar1 process

66.13 ar1_to_tikhonov

convert ar1 correlation to tikhonovs lambda

66.14 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

66.15 ar1_var_factor_

variance of an autocorrelated finite process

66.16 ar1_var_range2

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

66.17 ar1delay

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

66.18 ar1delay_old

autocorrelation of the residual

66.19 ar2conv

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

66.20 ar2dof

effective samples size for the ar2 process

66.21 ar2param

ar2 parameter estimation from first two terms of acf
$$\text{acf} = [1 \ a1 \ a2 \ \dots]$$

66.22 asymwin

creates asymmetrical filter windows
filter will always have negative weights

66.23 autocorr_fft

autocorrelation function

66.24 bandpass

bandpass filter

66.25 bandpass2

bandpass filter

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

66.27 bartlett_spectrogram

bartlet spectrogramm
TODO sliding window

66.28 bin1d

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

66.29 bin2d

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

66.30 binormrnd

generate two correlated normally distributed vectors

66.31 conv1_man

convolutions with padding

66.32 conv2_man

convolution in 2d

66.33 conv2z

66.34 conv30

convolve with rectangular window of lenght n
circular boundaries

66.35 conv_

convolution of a with b

66.36 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

66.37 convz

66.38 cosexpdelay

66.39 csmooth

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

66.40 daniell_window

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

66.41 danielle_window

danielle fourier window

66.42 db2neper

convert decibel to neper

66.43 db2power

power ratio from db

66.44 derive_danielle_weight

66.45 derive_limit_0_acfar

66.46 detect_peak

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

66.47 digital_low_pass_filter

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

66.48 doublesum_ij

double sum of r^i

66.49 effective_sample_size_to_ar1

convert effective sample size to ar1 correlation

66.50 filt_hodges_lehman

66.51 filter1

filter along one dimension

66.52 filter2

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

66.53 filter_

invalidate values that exceed n-times the robust standard deviation

66.54 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

66.55 filterp

66.56 filterp1

fir filter with some fancy extras

66.57 filterstd

66.58 firls_man

design finite impulse response filter by the least squares method

66.59 flattopwin

the flat top window

66.60 frequency_response_boxcar

frequency response of a boxcar filter

66.61 freqz_boxcar

frequency response of a boxcar filter

66.62 gaussfilt1

filter data series with a gaussian window

66.63 hanchangewin

hanning window for change point detection

66.64 hanchangewin2

nanning window for chage point detection

66.65 hanwin

hanning filter window

66.66 hanwin_

hanning filter window

66.67 highpass

high pass filter

66.68 kaiserwin

kaiser filter window

66.69 kalman

Kalman filter

66.70 lanczoswin

Lanczos window

66.71 last

lake tail, but for matrices

66.72 lowpass

low pass filter

66.73 lowpass2

design low pass filter with cutoff-frequency f1

66.74 lowpass_iir

iir-low pass

66.75 lowpass_iir_symmetric

two-sided iir low pass filter (for symmetry)

66.76 lowpassfilter2

low-pass filter of data

66.77 maxfilt1

66.78 meanfilt1

moving average filter with special treatment of the boundaries

66.79 medfilt1_man

moving median filter, supports columnwise operation

66.80 medfilt1_man2

moving median filter with special treatment of boundaries

66.81 medfilt1_padded

median filter with padding

66.82 medfilt1_reduced

median filter with padding

66.83 mid_term_single_sample

variance of single sample, mid term

66.84 minfilt1

66.85 mu2ar1

error variance of the mean of the finite length ar1 process

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

66.86 mysmooth

66.87 nanautocorr

autocorrelation with nan-values

66.88 nanmedfilt1

medfilt1, skipping nans

66.89 neper2db

convert neper to db

66.90 peaks_man

peaks of a periodogram

66.91 polyfilt1

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

66.92 qmedfilt1

medfilt1, after fitting a quadratic polynomial

66.93 randar1

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

66.94 randar1_dual

draw random variables of two correlated ar1 processes

66.95 randar2

generate ar2 process

66.96 randarp

randomly generate the instance of an ar-p process

66.97 range_window

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

66.98 rectwin

rectangular window

66.99 recursive_sum

66.100 select_range

66.101 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

66.102 smooth2

smooth vectos of X

66.103 smooth_man

66.104 smooth_parametric

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

66.105 smooth_parametric2

parametrically smooth the curve

66.106 smooth_with_splines

66.107 smoothfft

filter with fast fourier transform

66.108 spectrogram

spectrogram

66.109 std_window

moving block standard deviation

66.110 sum_i_lag

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

66.111 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

66.112 sum_ii_

66.113 sum_ij

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

66.114 sum_ij_

66.115 sum_ij_partial_

66.116 sum_multivar

sum of matrix entries of bivariate ar1 process

66.117 test_acfar1

66.118 test_acfar1_2

66.119 test_acfar1_3

66.120 test_acfar1_4

66.121 test_acfar2

66.122 test_ar1_var_factor

66.123 test_ar1_var_factor_2

66.124 test_ar1_var_mu_single_sample

66.125 test_ar1_var_pop

66.126 test_ar1_var_pop_1

66.127 test_ar1delay

66.128 test_bivariate_covariance_term

66.129 test_convexity

66.130 test_lanczoswin

66.131 test_madcorr

66.132 test_randar1

66.133 test_randar1_multivariate

66.134 test_randar2

66.135 test_sum_ij

66.136 test_sum_multivar

66.137 test_trifilt1

66.138 test_wautocorr

66.139 test_wavelet_transform

66.140 test_wordfilt

66.141 test_xar1_mid_term

66.142 tikhonov_to_ar1

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

66.143 trapwin

trapezoidal filter window

66.144 trifilt1

filter with triangular window

66.145 triwin

triangular filter window

66.146 triwin2

triangular filter window

66.147 varar1

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

66.148 welch_spectrogram

welch spectrogram

66.149 wfilt

filter with window

66.150 winbandpass

filter with bandpass

66.151 window_make_odd

66.152 winfilt0

filter with window

66.153 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

66.154 wmeanfilt

mean filter with window

66.155 wmedfilt

median filter with window

66.156 wordfilt

weighted order filter

66.157 wordfilt_edgeworth

weighed order filter

66.158 xar1

66.159 xcorr_man

cross correlation of two sampled ar1 processes

67 sorting

67.1 sort2

sort two numbers

67.2 sort2d

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

68 special-functions

68.1 `bessel_sphere`

spherical Bessel function of the first kind

68.2 `digamma_man`

68.3 `hankel_sphere`

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

68.4 `hermite`

probabilistic's hermite polynomial by recurrence relation

input :
n : order
x : value

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

68.5 `legendre_man`

legendre polynomials

68.6 `neumann_sphere`

spherical Neumann function
Bessel function of the second kind

69 statistics

69.1 atan_s2

stadard deviation of the arcus tangens by means of taylor expansion

69.2 beta_mode_to_parameter

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

69.3 coefficient_of_determination

69.4 conditional_expectation_normal

69.5 correlation_confidence_pearson

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

70 statistics/distributions

70.1 PDF

class for quasi-distributions from a set of sampling points

70.2 binorm_separation_coefficient

separation coefficient of a bimodal normal distribution

70.3 `binormcdf`

bio-modal gaussian distribution

70.4 `binormfit`

fit sum of to normal distribution to a histogram

70.5 `binormpdf`

70.6 `edgeworth_cdf`

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

70.7 `edgeworth_pdf`

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

70.8 `logn_mode2param`

transform modes (μ, σ) to parameters of the log normal
distribution

70.9 `logn_param2mode`

transform parameters to mode (μ, σ) for the log normal
distribution

70.10 lognpdf_

log normal distribution called by modes rather than parameters

70.11 pdfsample

pdf from sample distribution

Note: better use kernel density estimates

70.12 t2cdf

Hotelling's T-squared cumulative distribution

70.13 t2inv

inverse of Hotelling's T-squared cumulative distribution

71 statistics

71.1 example_standard_error_of_sample_quantiles

71.2 f_var_finite

reduction of variance when sampling from a finite population
without replacement

71.3 gamma_mode_to_parameter

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

71.4 gaussfit3

71.5 gaussfit_quantile

71.6 geoserr

71.7 geostd

71.8 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

71.9 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

72 statistics/information-theory

72.1 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

72.2 bayesian_information_criterion

bayesian information criterion

73 statistics

73.1 kurtncdf

73.2 kurtnpdf

73.3 kurtosis_bias_corrected

bias corrected kurtosis

73.4 limit

limit a by lower and upper bound

73.5 logfactorial

approximate log of the factorial

73.6 loglogpdf

73.7 lognfit_quantile

73.8 logskewcdf

73.9 logskewpdf

74 statistics/logu

74.1 lambertw_numeric

lambert-w function

74.2 logtrialtcdf

pdf of a logarithmic triangular distribution

74.3 logtrialtinv

inverse of the logarithmic triangular distribution

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

74.4 logtrialtmean

mean of the logarithmic triangular distribution

74.5 logtrialtpdf

density of the logarithmic triangular distribution

74.6 logtrialtrnd

74.7 logtricdf

cumulative distribution of the logarithmic triangular distribution

74.8 logtriinv

inverse of the logarithmic triangular distribution

74.9 logtrimean

mean of the logarithmic triangular distribution

74.10 logtripdf

probability density of the logarithmic triangular distribution

74.11 logtrirnd

74.12 logucdf

probability density of the logarithmic uniform distribution

74.13 logucm

central moments of the log-uniform distribution

74.14 loguinv

inverse of the log-uniform distribution

74.15 logumean

mean of the log-uniform distribution

74.16 logupdf

pdf of the log uniform distribution

74.17 logurnd

random numbers following a log-uniform distribution

74.18 loguvar

variance of the log-uniform distribution

74.19 medlogu

median of the log-uniform distribution

74.20 test_logurnd

74.21 tricdf

cumulative distribution of the log-triangular distribution

74.22 triinv

inverse of the triangular distribution

74.23 trimedian

median of the triangular distribution

74.24 tripdf

probability density of the triangular distribution

74.25 trirnd

random numbers of the triangular distribution

75 statistics

75.1 max_exprnd

75.2 maxnnormals

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

75.3 mean_generalized_gampdf

75.4 midrange

mid range of columns of X

75.5 minavg

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

75.6 mode_man

76 statistics/moment-statistics

76.1 autocorr_man3

autocorrelation of the columns of X

76.2 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

76.3 autocorr_man5

autocorrellation of the columns of X

76.4 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

76.5 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
 μ : $n \times 1$ mean vector
 C : $n \times n$ covariance matrix
 k : $n \times 1$ powers of variables in moments

76.6 corr_man

correlation of two vectors

76.7 cov_man

covariance matrix of two vectors

76.8 dof

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

76.9 edgeworth_quantile

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

76.10 effective_sample_size

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

76.11 f_correlation

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

76.12 f_finite

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

76.13 lmean

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

76.14 lmoment

l-moment of vector x

76.15 maskmean

mean of the masked values of X

76.16 masknanmean

76.17 mean1

mean of x

76.18 mean_man

mean and standard error of X

76.19 mse

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

76.20 nanautocorr_man1

autocorrelation of a vector with nan-values

76.21 nanautocorr_man2

autocorrelation of a vector with nan-values

76.22 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

76.23 `nancorr`

(co)-correlation matrix when samples a NaN

76.24 `nancumsum`

cumulative sum, setting nan values to zero

76.25 `nanlmean`

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

76.26 `nanr2`

coefficient of determination when samples are invalid

76.27 `nanrms`

root mean square value when sample contains nan-values

76.28 `nanrmse`

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

76.29 `nanserr`

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

76.30 `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)

76.31 nanwstd

weighed standard deviation

76.32 nanwvar

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

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

76.33 nanxcorr

76.34 pearson

pearson correlation coefficient

76.35 pearson_to_kendall

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

76.36 pool_samples

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

76.37 qmean

trimmed mean

76.38 range_mean

76.39 `rmse_`

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

76.40 `serr`

standard error of the mean of a set of uncorrelated samples

76.41 `serr1`

76.42 `test_qskew`

76.43 `test_qstd_qskew_optimal_p`

76.44 `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

76.45 `wcorr`

correlation of two vectors when samples are weighted

76.46 wcov

covariance of two vectors when samples are weighted

76.47 wdof

effective degrees of freedom for weighted samples

76.48 wkurt

kurtosis with weighted samples

76.49 wmean

weighted mean

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

varargin can be dim

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

76.50 wrms

weighted root mean square error

76.51 wserr

weighted root mean square error

76.52 wskew

skewness of a weighted set of samples

76.53 wstd

weighed standard deviation

76.54 wvar

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

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

77 statistics

77.1 nangeomean

77.2 nangeostd

geometric standard deviation ignoring nan-values

78 statistics/nonparametric-statistics

78.1 kernel1d

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

78.2 kernel2d

kernel density estimate in two dimensions

79 statistics

79.1 normmoment

expected norm of \bar{x} , when values x in x are iid normal with μ and σ

79.2 normpdf2

pdf of the bivariate normal distribution

80 statistics/order-statistics

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

80.2 kendall

kendall correlation coefficient

80.3 kendall_to_pearson

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

c.f. Kruska, 1985

80.4 mad2sd

transform median absolute deviation to standard deviation
for normal distributed values

80.5 `madcorr`

proxy correlation by median absolute deviation

80.6 `median2_holder`

80.7 `median_ci`

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

80.8 `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`

80.9 `mediani`

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

80.10 `nanmadcorr`

proxy correlation by median absolute deviation

80.11 `nanwmedian`

weighted median, skips nan-values

80.12 nanwquantile

weighted quantile, skips nan values

80.13 oja_median

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

oja 1983, for extension to multivariate function, see chaudhri

80.14 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

80.15 qmoments

moments estimated from quantiles

80.16 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

80.17 qskewq

skewness estimated by quantiles

80.18 qstdq

proxy standard deviation determined by quantiles

80.19 quantile1_optimisation

80.20 quantile2_breckling

quantile regression

80.21 quantile2_chaudhuri

quantile regression

80.22 quantile2_projected

quantile in two dimensions

80.23 quantile2_projected2

spatial quantile for chosen direction

80.24 quantile_envelope

80.25 quantile_regression_simple

simple quantile regression

80.26 ranking

ranking for spearman statistics

80.27 spatial_median

c.f. Oja 2008

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

80.28 spatial_quantile

spatial quantile

80.29 spatial_quantile2

spatial quantile

80.30 spatial_quantile3

spatial quantile

80.31 spatial_rank

unsigned rank

80.32 spatial_sign

spatial sign

80.33 `spatial_signed_rank`

signed rank

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

80.34 `spearman`

spearman's product moment coefficient

80.35 `spearman_rank`

80.36 `spearman_to_pearson`

conversion of spearman rank to person product moment correlation
coefficient

80.37 `wmedian`

weighted median

80.38 `wquantile`

weighted quantile

81 `statistics`

81.1 `qstd`

81.2 `quantile_extrap`

82 statistics/random-number-generation

82.1 laplacernd

random number of laplace distribution

82.2 randc

correlate to correlated standard normally distributed vectors

82.3 skewness2param

82.4 skewpdf_central_moments

82.5 skewrnd

random numbers of the skew normal distribution

82.6 skewrnd2

random numbers of the skew normal distribution

83 statistics

83.1 range

mid range

83.2 resample_with_replacement

84 statistics/resampling-statistics/@Jackknife

84.1 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

84.2 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

84.3 matrix1_STATIC

matrix of estimation for leaving out two samples at a time

84.4 matrix2

matrix of estimations for jacknive with two samples left out

85 statistics/resampling-statistics

85.1 block_jackknife

85.2 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

85.3 moving_block_jackknife

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

85.4 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

85.5 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

86 statistics

86.1 scale_quantile_sd

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

86.2 sd_sample_quantiles

86.3 skewpdf

skew-normal distribution
c.f. Azzalini 1985

86.4 test_mean_generalized_gampdf

86.5 trimmed_mean

trimmed mean

86.6 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

86.7 ttest_man

two-sample t-test
unequal sample size
equal variance

86.8 ttest_paired

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

86.9 wgeomean

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

86.10 wgeovar

variance of the weighted geometric mean

86.11 wharmean

weighted harmonic mean

86.12 wharstd

86.13 wharvar

87 mathematics

mathematical functions of various kind

87.1 ternary_diagram

88 test/master

88.1 dat_test_lanczos_3d_k_20_n_40

88.2 poisson2d_blk

88.3 qr_implicit_givens_2

88.4 spectral_derivative_2d

88.5 test_2d_eigensolver_hydrogen

88.6 test_2d_refine

88.7 test_3d_eigensolver_hydrogen

88.8 test_FEM

88.9 test_Mesh_3d

88.10 test_arnoldi

88.11 test_arpackc

88.12 test_assemble

88.13 test_assembly_performance

88.14 test_bc_one_sided

88.15 test_compare_solvers

88.16 test_complete

88.17 test_convergence

88.18 `test_convergence_b`

88.19 `test_df_2d`

88.20 `test_eig_algs`

88.21 `test_eig_inverse`

88.22 `test_eigs_lanczos`

88.23 `test_eigs_lanczos_1`

88.24 `test_eigs_lanczos_2`

88.25 `test_eigs_lanczos_performance`

88.26 `test_fdm`

88.27 `test_fdm_d_vargrid`

88.28 test_fdm_spectral

88.29 test_fem

88.30 test_fem_1d

88.31 test_fem_1d_higher_order

88.32 test_fem_2d_adaptive

88.33 test_fem_2d_higher_order

88.34 test_fem_3d_higher_order

88.35 test_fem_3d_refine

88.36 test_fem_b

88.37 test_fem_derivative

88.38 test_fem_quadrature

88.39 test_final

88.40 test_fix_substitution

88.41 test_forward

88.42 test_get_sparse_arrays

88.43 test_harmonic_oscillator

88.44 test_high_order_fdm_periodic_bc

88.45 test_hydrogen_wf

88.46 test_ichol

88.47 test_interpolation

88.48 test_inverse_problem

88.49 test_it_vs_exact

88.50 test_jama

88.51 test_jd

88.52 test_jdqz

88.53 test_lanczos_2

88.54 test_lanczos_biorthogonal

88.55 test_laplacian

88.56 test_laplacian_non_uniform

88.57 test_laplacian_simple

88.58 test_mesh_2d_uniform

88.59 test_mesh_2d_uniform_2

88.60 test_mesh_circle

88.61 test_mesh_generation

88.62 test_mesh_interpolate

88.63 test_mg

88.64 test_minres_recycle

88.65 test_multigrid

88.66 test_nc

88.67 test_nonuniform_symmetric

88.68 test_pde

88.69 test_permutation

88.70 test_poison_fem

88.71 test_polar

88.72 test_potential

88.73 test_powers

88.74 test_precondition

88.75 test_project_rectangle

88.76 test_qr

88.77 test_quantum_well

88.78 test_radial_adaptive

88.79 test_radial_confinement

88.80 test_radial_fixes

88.81 test_refine_2d

88.82 test_refine_2d_b

88.83 test_refine_3d

88.84 test_refine_structural

88.85 test_regularisation

88.86 test_round_off

88.87 test_schrödinger_potentials

88.88 test_uniform_mesh

88.89 test_vargrid

89 test

89.1 test_gaussfit3

89.2 test_geoserr

89.3 test_lognfit_quantile

89.4 test_max_normal

89.5 test_mtimes3x3

90 mathematics

mathematical functions of various kind

90.1 vanderd_2d

91 wavelet

91.1 continuous_wavelet_transform

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

91.2 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

91.3 example_wavelets

91.4 phasewrap

wrap the phase to +/- pi

91.5 test_cwt_man

91.6 test_phasewrap

91.7 test_wavelet

91.8 test_wavelet2

91.9 test_wavelet_analysis

91.10 test_wavelet_reconstruct

91.11 test_wtc

91.12 wavelet

wavelet windows

91.13 wavelet_reconstruct

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

91.14 wavelet_transform

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

92 mathematics

mathematical functions of various kind

92.1 wrapphase