

Manual for Package: mathematics

Revision 17M

Karl Kästner

December 22, 2021

Contents

1	calendar	36
1.1	days_per_month	36
1.2	isnight	36
2	mathematics	36
2.1	cast_byte_to_integer	36
3	complex-analysis	36
3.1	complex_exp_product_im_im	36
3.2	complex_exp_product_im_re	37
3.3	complex_exp_product_re_im	37
3.4	complex_exp_product_re_re	37
3.5	croots	38
3.6	root_complex	38
3.7	test_imroots	38
4	derivation	38
4.1	derive_acfar1	38
4.2	derive_ar1_spectral_density	38
4.3	derive_ar2param	38
4.4	derive_arc_length	39
4.5	derive_fourier_power	39
4.6	derive_fourier_power_exp	39
4.7	derive_laplacian_curvilinear	39
4.8	derive_laplacian_fourier_piecewise_linear	39
4.9	derive_logtripdf	39
4.10	derive_smooth1d_parametric	39
4.11	derive_spectral_density_bandpass_initial_condition	39
5	derivation/master	39

5.1	derive_bc_one_sided	39
5.2	derive_convergence	40
5.3	derive_error_fdm	40
5.4	derive_fdm_poly	40
5.5	derive_fdm_power	40
5.6	derive_fdm_taylor	40
5.7	derive_fdm_vargrid	40
5.8	derive_fem_2d_mass	40
5.9	derive_fem_error_2d	40
5.10	derive_fem_error_3d	40
5.11	derive_fem_sym_2d	40
5.12	derive_grid_constants	41
5.13	derive_interpolation	41
5.14	derive_laplacian	41
5.15	derive_limit	41
5.16	derive_nc_1d	41
5.17	derive_nc_1d_	41
5.18	derive_nc_2d	41
5.19	derive_nonuniform_symmetric	41
5.20	derive_richardson	41
5.21	derive_sum	41
5.22	nn	42
5.23	test_derive	42
5.24	test_derive_fdm_poly	42
5.25	test_filter	42
5.26	test_vargrid	42
6	derivation	42
6.1	simplify_atan	42
7	mathematics	42
7.1	entropy	42
8	finance	42
8.1	derive_skewrnd_walsh_paramter	42
8.2	gbm_cdf	43
8.3	gbm_fit	43
8.4	gbm_fit_old	43
8.5	gbm_inv	43
8.6	gbm_mean	43
8.7	gbm_median	43
8.8	gbm_pdf	43
8.9	gbm_simulate	43
8.10	gbm_skewness	43

8.11	gbm_std	43
8.12	gbm_transform_time_step	44
8.13	put_price_black_scholes	44
8.14	skewgbm_simulate	44
8.15	skewrnd_walsh	44
9	finance/test	44
9.1	test_gbm	44
9.2	test_gbm_pdf	44
9.3	test_skewrnd_walsh	44
10	fourier/@STFT	44
10.1	STFT	44
10.2	itransform	45
10.3	stft_	45
10.4	stftmat	45
10.5	transform	45
11	fourier	45
11.1	amplitude_from_peak	45
11.2	dftmtx_man	46
11.3	example_fourier_window	46
11.4	fft_derivative	46
11.5	fft_man	46
11.6	fftsmooth	47
11.7	fix_fourier	47
11.8	fourier_axis	47
11.9	fourier_cesaro_correction	48
11.10	fourier_coefficient_piecewise_linear	48
11.11	fourier_coefficient_piecewise_linear_1	48
11.12	fourier_coefficient_ramp3	48
11.13	fourier_coefficient_ramp_pulse	49
11.14	fourier_coefficient_ramp_step	49
11.15	fourier_coefficient_square_pulse	49
11.16	fourier_cubic_interaction_coefficients	49
11.17	fourier_derivative	49
11.18	fourier_derivative_matrix_1d	49
11.19	fourier_derivative_matrix_2d	49
11.20	fourier_expand	49
11.21	fourier_fit	50
11.22	fourier_interpolate	50
11.23	fourier_matrix	50
11.24	fourier_matrix2	50
11.25	fourier_matrix3	50

11.26	fourier_matrix_exp	50
11.27	fourier_multiplicative_interaction_coefficients	50
11.28	fourier_power	51
11.29	fourier_power_exp	51
11.30	fourier_predict	51
11.31	fourier_quadratic_interaction_coefficients	51
11.32	fourier_random_phase_walk	51
11.33	fourier_range	51
11.34	fourier_regress	52
11.35	fourier_resampled_fit	52
11.36	fourier_resampled_predict	52
11.37	fourier_signed_square	52
11.38	fourier_transform	52
11.39	fourier_transform_fractional	53
11.40	hyperbolic_fourier_box	53
11.41	idftmtx_man	53
11.42	laplace_2d_pwlinear	53
11.43	nanfft	53
11.44	peaks	54
11.45	roots_fourier	54
11.46	spectral_density	54
11.47	test_complex_exp_product	54
11.48	test_fourier_filter	54
11.49	test_idftmtx	55
 12 mathematics		55
12.1	gaussfit_quantile	55
 13 geometry/@Geometry		55
13.1	Geometry	55
13.2	arclength	55
13.3	arclength_old	55
13.4	arclength_old2	55
13.5	base_point	56
13.6	base_point_limited	56
13.7	centroid	56
13.8	cosa_min_max	56
13.9	cross2	56
13.10	curvature	56
13.11	ddot	56
13.12	distance	56
13.13	distance2	57
13.14	dot	57
13.15	edge_length	57

13.16	enclosed_angle	57
13.17	enclosing_triangle	57
13.18	hexagon	57
13.19	inPolygon	57
13.20	inTetra	58
13.21	inTetra2	58
13.22	inTriangle	58
13.23	intersect	58
13.24	lineintersect	58
13.25	lineintersect1	58
13.26	minimum_distance_lines	58
13.27	mittenpunkt	58
13.28	nagelpoint	59
13.29	onLine	59
13.30	orthocentre	59
13.31	plumb_line	59
13.32	poly_area	59
13.33	poly_edges	59
13.34	poly_set	59
13.35	poly_width	60
13.36	polyxpoly	60
13.37	project_to_curve	60
13.38	quad_isconvex	60
13.39	random_disk	60
13.40	random_simplex	60
13.41	sphere_volume	60
13.42	tetra_volume	61
13.43	tobarycentric	61
13.44	tobarycentric1	61
13.45	tobarycentric2	61
13.46	tobarycentric3	61
13.47	tri_angle	61
13.48	tri_area	61
13.49	tri_centroid	61
13.50	tri_distance_opposit_midpoint	62
13.51	tri_edge_length	62
13.52	tri_edge_midpoint	62
13.53	tri_excircle	62
13.54	tri_height	62
13.55	tri_incircle	62
13.56	tri_isacute	62
13.57	tri_isobtuse	62
13.58	tri_semiperimeter	63
13.59	tri_side_length	63

14	geometry	63
14.1	Polygon	63
14.2	bounding_box	63
14.3	curvature_1d	63
14.4	cvt	63
14.5	deg_to_frac	64
14.6	ellipse	64
14.7	ellipseX	64
14.8	ellipseY	64
14.9	first_intersect	64
14.10	golden_ratio	64
14.11	hypot3	64
14.12	meanangle	64
14.13	meanangle2	65
14.14	meanangle3	65
14.15	meanangle4	65
14.16	medianangle	65
14.17	medianangle2	65
14.18	pilim	65
14.19	streamline_radius_of_curvature	66
15	mathematics	66
15.1	hexagonal_pattern	66
16	histogram/@Histogram	66
16.1	2x	66
16.2	Histogram	66
16.3	bimodes	66
16.4	cdf	66
16.5	cdfS	66
16.6	chi2test	66
16.7	cmoment	67
16.8	cmomentS	67
16.9	entropy	67
16.10	entropyS	67
16.11	export_csv	67
16.12	iquantile	67
16.13	kstest	67
16.14	kurtosis	67
16.15	kurtosisS	67
16.16	mean	67
16.17	meanS	68
16.18	median	68
16.19	medianS	68

16.20	mode	68
16.21	modeS	68
16.22	moment	68
16.23	momentS	68
16.24	pdf	68
16.25	quantile	68
16.26	quantileS	68
16.27	resample	69
16.28	setup	69
16.29	skewness	69
16.30	skewnessS	69
16.31	stairs	69
16.32	stairsS	69
16.33	std	69
16.34	stdS	69
16.35	var	69
16.36	varS	69
17	histogram	70
17.1	hist_man	70
17.2	histadapt	70
17.3	histconst	70
17.4	pdf_poly	70
17.5	plotcdf	70
17.6	test_histogram	70
18	mathematics	70
18.1	imrotmat	70
19	linear-algebra	70
19.1	averaging_matrix_2	70
19.2	colnorm	71
19.3	condest_	71
19.4	connectivity_matrix	71
20	linear-algebra/coordinate-transformation	71
20.1	barycentric2cartesian	71
20.2	barycentric2cartesian3	71
20.3	cartesian2barycentric	71
20.4	cartesian_to_unit_triangle_basis	71
20.5	ellipsoid2geoid	71
20.6	example_approximate_utm_conversion	72
20.7	latlon2utm	72
20.8	latlon2utm_simple	72

20.9	lowrance_mercator_to_wgs84	72
20.10	nmea2utm	72
20.11	sn2xy	72
20.12	unit_triangle_to_cartesian	72
20.13	utm2latlon	72
20.14	xy2nt	73
20.15	xy2sn	73
20.16	xy2sn_java	73
20.17	xy2sn_old	73
21	linear-algebra	73
21.1	det2x2	73
21.2	det3x3	73
21.3	det4x4	74
21.4	diag2x2	74
21.5	down	74
21.6	eig2x2	74
22	linear-algebra/eigenvalue	74
22.1	eig_bisection	74
22.2	eig_inverse	74
22.3	eig_inverse_iteration	74
22.4	eig_power_iteration	74
23	linear-algebra/eigenvalue/jacobi-davidson	75
23.1	afun_jdm	75
23.2	davidson	75
23.3	jacobi_davidson	75
23.4	jacobi_davidson_qr	75
23.5	jacobi_davidson_qz	75
23.6	jacobi_davidson_simple	75
23.7	jdqr	75
23.8	jdqr_sleijpen	79
23.9	jdqr_vorst	82
23.10	jdqz	85
23.11	mfunc_jdm	89
23.12	mgs	90
23.13	minres_	90
23.14	mv_jacobi_davidson	90
24	linear-algebra	90
24.1	first	90
24.2	gershgorin_circle	90
24.3	hausdorff	90

24.4	ieig2x2	90
24.5	inv2x2	91
24.6	inv3x3	91
24.7	inv4x4	91
24.8	kernel2matrix	91
25	linear-algebra/lanczos	91
25.1	arnoldi	91
25.2	arnoldi_new	91
25.3	eigs_lanczos_man	91
25.4	lanczos	91
25.5	lanczos_	91
25.6	lanczos.biorthogonal	92
25.7	lanczos.biorthogonal_improved	92
25.8	lanczos_ghep	92
25.9	mv_lanczos	92
25.10	reorthogonalise	92
25.11	test_lanczos	92
26	linear-algebra	92
26.1	left	92
27	linear-algebra/linear-systems	92
27.1	gmres_man	92
27.2	minres_recycle	93
28	linear-algebra	93
28.1	lpmean	93
28.2	lpnorm	93
28.3	matvec3	93
28.4	max2d	93
28.5	mid	93
28.6	mpoweri	93
28.7	mtimes2x2	93
28.8	mtimes3x3	94
28.9	nannorm	94
28.10	nanshift	94
28.11	nl	94
28.12	normalise	94
28.13	normalize1	94
28.14	normrows	94
28.15	orth2	95
28.16	orth_man	95
28.17	orthogonalise	95

28.18	padd2	95
28.19	paddext	95
28.20	paddval1	95
28.21	paddval2	95
29	linear-algebra/polynomial	96
29.1	chebychev	96
29.2	piecewise_polynomial	96
29.3	roots1	96
29.4	roots2	96
29.5	roots2poly	96
29.6	roots3	96
29.7	roots4	96
29.8	roots_piecewise_linear	96
29.9	test_roots4	97
29.10	vanderi_1d	97
30	linear-algebra	97
30.1	randrot	97
30.2	right	97
30.3	rot2	97
30.4	rot2dir	97
30.5	rot3	97
30.6	rotR	97
30.7	rownorm	98
30.8	simmilarity_matrix	98
30.9	spnorm	98
30.10	spzeros	98
30.11	test_roots3	98
30.12	transform_minmax	98
30.13	transpose3	98
30.14	transposeall	98
30.15	up	98
30.16	vanderd_2d	99
31	logic	99
31.1	bitor_man	99
32	master/plot	99
32.1	attach_boundary_value	99
32.2	cartesian_polar	99
32.3	img_vargrid	99
32.4	plot_basis_functions	99
32.5	plot_convergence	99

32.6	plot_dof	99
32.7	plot_eigenbar	100
32.8	plot_error_estimation	100
32.9	plot_error_estimation_2	100
32.10	plot_error_fem	100
32.11	plot_fdm_kernel	100
32.12	plot_fdm_vs_fem	100
32.13	plot_fem_accuracy	100
32.14	plot_function_and_grid	100
32.15	plot_hat	100
32.16	plot_hydrogen_wf	100
32.17	plot_mesh	101
32.18	plot_mesh_2	101
32.19	plot_refine	101
32.20	plot_refine_3d	101
32.21	plot_runtime	101
32.22	plot_spectrum	101
32.23	plot_wavefunction	101
33	master/ported	101
33.1	assemble_2d_phi_phi	101
33.2	assemble_3d_dphi_dphi	101
33.3	assemble_3d_phi_phi	102
33.4	dV_2d_	102
33.5	derivative_2d	102
33.6	derivative_3d	102
33.7	element_neighbour_2d	102
33.8	prefetch_2d_	102
33.9	promote_2d_3_10	102
33.10	promote_2d_3_15	102
33.11	promote_2d_3_21	102
33.12	promote_2d_3_6	102
33.13	promote_3d_4_10	103
33.14	promote_3d_4_20	103
33.15	promote_3d_4_35	103
33.16	vander_2d	103
33.17	vander_3d	103
34	number-theory	103
34.1	ceiln	103
34.2	digitsb	103
34.3	floorn	103
34.4	iseven	104
34.5	multichoosek	104

34.6	nchoosek_man	104
34.7	pythagorean_triple	104
34.8	roundn	104
35	numerical-methods/differentiation	104
35.1	derivative1	104
35.2	derivative2	105
36	numerical-methods/finite-difference	105
36.1	cdiff	105
36.2	cdiffb	105
36.3	central_difference	105
36.4	cmean	105
36.5	cmean2	105
36.6	derivative_matrix_1_1d	105
36.7	derivative_matrix_2_1d	106
36.8	derivative_matrix_2d	106
36.9	derivative_matrix_curvilinear	106
36.10	derivative_matrix_curvilinear_2	106
36.11	difference_kernel	106
36.12	diffusion_matrix_2d_anisotropic	106
36.13	diffusion_matrix_2d_anisotropic2	106
36.14	distmat	106
36.15	downwind_difference	107
36.16	gradpde2d	107
36.17	laplacian	107
36.18	laplacian_fdm	107
36.19	lrmean	107
37	numerical-methods/finite-difference/master	107
37.1	fdm_adaptive_grid	107
37.2	fdm_adaptive_refinement_old	107
37.3	fdm_assemble_d1_2d	107
37.4	fdm_assemble_d2_2d	108
37.5	fdm_confinement	108
37.6	fdm_d_vargrid	108
37.7	fdm_h_unstructured	108
37.8	fdm_hydrogen_vargrid	108
37.9	fdm_mark_unstructured_2d	108
37.10	fdm_plot	108
37.11	fdm_plot_series	108
37.12	fdm_refine_2d	108
37.13	fdm_refine_3d	108
37.14	fdm_refine_unstructured_2d	109

37.15	fdm_schroedinger_2d	109
37.16	fdm_schroedinger_3d	109
37.17	relocate	109
38	numerical-methods/finite-difference	109
38.1	mid	109
38.2	pwmid	109
38.3	ratio	109
38.4	steplength	109
38.5	swapoddeven	110
38.6	test_derivative_matrix_2d	110
38.7	test_derivative_matrix_curvilinear	110
38.8	test_difference_kernel	110
38.9	upwind_difference	110
39	numerical-methods/finite-element	110
39.1	Mesh_2d.java	110
39.2	Tree_2d.java	110
39.3	assemble_1d_dphi_dphi	110
39.4	assemble_1d_phi_phi	110
39.5	assemble_2d_dphi_dphi.java	111
39.6	assemble_2d_phi_phi.java	111
39.7	assemble_3d_dphi_dphi.java	111
39.8	assemble_3d_phi_phi.java	111
39.9	boundary_1d	111
39.10	boundary_2d	111
39.11	boundary_3d	111
39.12	check_area_2d	111
39.13	circmesh	111
39.14	cropradius	111
39.15	display_2d	112
39.16	display_3d	112
39.17	distort	112
39.18	err_2d	112
39.19	estimate_err_2d_3	112
39.20	example_1d	112
39.21	example_2d	112
39.22	explode	112
39.23	fem_2d	112
39.24	fem_2d_heuristic_mesh	112
39.25	fem_get_2d_radial	113
39.26	fem_interpolation	113
39.27	fem_plot_1d	113
39.28	fem_plot_1d_series	113

39.29	fem_plot_2d	113
39.30	fem_plot_2d_series	113
39.31	fem_plot_3d	113
39.32	fem_plot_3d_series	113
39.33	fem_plot_confine_series	113
39.34	fem_radial	114
39.35	flip_2d	114
39.36	get_mesh_arrays	114
39.37	hashkey	114
40	numerical-methods/finite-element/int	114
40.1	int_1d_gauss	114
40.2	int_1d_gauss_1	114
40.3	int_1d_gauss_2	114
40.4	int_1d_gauss_3	114
40.5	int_1d_gauss_4	114
40.6	int_1d_gauss_5	115
40.7	int_1d_gauss_6	115
40.8	int_1d_gauss_lobatto	115
40.9	int_1d_gauss_n	115
40.10	int_1d_nc_2	115
40.11	int_1d_nc_3	115
40.12	int_1d_nc_4	115
40.13	int_1d_nc_5	115
40.14	int_1d_nc_6	115
40.15	int_1d_nc_7	115
40.16	int_1d_nc_7_hardy	116
40.17	int_2d_gauss_1	116
40.18	int_2d_gauss_12	116
40.19	int_2d_gauss_13	116
40.20	int_2d_gauss_16	116
40.21	int_2d_gauss_19	116
40.22	int_2d_gauss_25	116
40.23	int_2d_gauss_3	116
40.24	int_2d_gauss_33	116
40.25	int_2d_gauss_4	116
40.26	int_2d_gauss_6	117
40.27	int_2d_gauss_7	117
40.28	int_2d_gauss_9	117
40.29	int_2d_nc_10	117
40.30	int_2d_nc_15	117
40.31	int_2d_nc_21	117
40.32	int_2d_nc_3	117
40.33	int_2d_nc_6	117

40.34	int_3d_gauss_1	117
40.35	int_3d_gauss_11	117
40.36	int_3d_gauss_14	118
40.37	int_3d_gauss_15	118
40.38	int_3d_gauss_24	118
40.39	int_3d_gauss_4	118
40.40	int_3d_gauss_45	118
40.41	int_3d_gauss_5	118
40.42	int_3d_nc_11	118
40.43	int_3d_nc_4	118
40.44	int_3d_nc_6	118
40.45	int_3d_nc_8	118
41	numerical-methods/finite-element	119
41.1	interpolation_matrix	119
41.2	mark	119
41.3	mark_1d	119
41.4	mesh_1d_uniform	119
41.5	mesh_3d_uniform	119
41.6	mesh_interpolate	119
41.7	neighbour_1d	119
41.8	old	119
41.9	pdeeig_1d	119
41.10	pdeeig_2d	120
41.11	pdeeig_3d	120
41.12	polynomial_derivative_1d	120
41.13	potential_const	120
41.14	potential_coulomb	120
41.15	potential_harmonic_oscillator	120
41.16	project_circle	120
41.17	project_rectangle	120
41.18	promote_1d_2_3	120
41.19	promote_1d_2_4	120
41.20	promote_1d_2_5	121
41.21	promote_1d_2_6	121
41.22	quadrilaterate	121
41.23	recalculate_regularity_2d	121
41.24	refine_1d	121
41.25	refine_2d_21	121
41.26	refine_2d_structural	121
41.27	regularity_1d	121
41.28	regularity_2d	121
41.29	regularity_3d	122
41.30	relocate_2d	122

41.31	test_circmesh	122
41.32	test_hermite	122
41.33	tri_assign_points	122
41.34	triangulation_uniform	122
41.35	vander_1d	122
41.36	vanderd_1d	122
41.37	vanderi_1d	122
42	numerical-methods/finite-volume/@Advection	123
42.1	Advection	123
42.2	dot_advection	123
43	numerical-methods/finite-volume/@Burgers	123
43.1	burgers_split	123
43.2	dot_burgers_fdm	123
43.3	dot_burgers_fft	123
44	numerical-methods/finite-volume/@Finite_Volume	123
44.1	Finite_Volume	123
44.2	apply_bc	124
44.3	solve	124
44.4	step_split_strang	124
44.5	step_unsplit	124
45	numerical-methods/finite-volume/@Flux_Limiter	124
45.1	Flux_Limiter	124
45.2	beam_warming	124
45.3	fromm	125
45.4	lax_wendroff	125
45.5	minmod	125
45.6	monotized_central	125
45.7	muscl	125
45.8	superbee	125
45.9	upwind	125
45.10	vanLeer	126
46	numerical-methods/finite-volume/@KDV	126
46.1	dot_kdv_fdm	126
46.2	dot_kdv_fft	126
46.3	kdv_split	126
47	numerical-methods/finite-volume/@Reconstruct_Average_Evolve	126
47.1	Reconstruct_Average_Evolve	126
47.2	advect_highres	127
47.3	advect_lowress	127

48	numerical-methods/finite-volume	127
48.1	Godunov	127
48.2	Lax_Friedrich	127
48.3	Measure	127
48.4	Roe	127
48.5	fv_swe	128
48.6	staggered_euler	128
48.7	staggered_grid	128
49	numerical-methods	128
49.1	grid2quad	128
50	numerical-methods/integration	128
50.1	cumintL	128
50.2	cumintR	128
50.3	int_trapezoidal	128
51	numerical-methods/interpolation/@Kriging	129
51.1	Kriging	129
51.2	estimate_semivariance	129
51.3	interpolate_	129
52	numerical-methods/interpolation/@RegularizedInterpolator1	129
52.1	RegularizedInterpolator1	129
52.2	init	129
53	numerical-methods/interpolation/@RegularizedInterpolator2	130
53.1	RegularizedInterpolator2	130
53.2	init	130
54	numerical-methods/interpolation/@RegularizedInterpolator3	130
54.1	RegularizedInterpolator3	130
54.2	init	130
55	numerical-methods/interpolation	130
55.1	IDW	130
55.2	IPoly	130
55.3	IRBM	131
55.4	ISparse	131
55.5	Inn	131
55.6	Interpolator	131
55.7	fixnan	131
55.8	idw1	131
55.9	idw2	131
55.10	inner2outer	132

55.11	inner2outer2	132
55.12	interp1_circular	132
55.13	interp1_limited	132
55.14	interp1_man	132
55.15	interp1_piecewise_linear	132
55.16	interp1_save	132
55.17	interp1_slope	133
55.18	interp1_smooth	133
55.19	interp1_unique	133
55.20	interp2_man	133
55.21	interp_angle	133
55.22	interp_fourier	133
55.23	interp_fourier_batch	133
55.24	interp_sn	134
55.25	interp_sn2	134
55.26	interp_sn3	134
55.27	interp_sn_	134
55.28	limit_by_distance_1d	134
55.29	resample1	134
55.30	resample_d_min	134
55.31	resample_vector	135
55.32	test_interp1_limited	135
56 numerical-methods		135
56.1	inverse_complex	135
56.2	maccormack_step	135
56.3	minmod	135
57 numerical-methods/multigrid		135
57.1	mg_interpolate	135
57.2	mg_restrict	135
58 numerical-methods/ode/@BVPS_Characteristic		135
58.1	BVPS_Characteristic	135
58.2	assemble1_A	136
58.3	assemble1_A_Q	136
58.4	assemble2_A	136
58.5	assemble_AA	136
58.6	assemble_AAA	136
58.7	assemble_Ic	136
58.8	bvp1c	136
58.9	check_arguments	136
58.10	couple_junctions	137
58.11	derivative	137

58.12	init	137
58.13	inner2outer_bvp2c	137
58.14	reconstruct	137
58.15	resample	137
58.16	solve	137
58.17	test_assemble1_A	138
58.18	test_assemble2_A	138
59	numerical-methods/ode/@Time_Stepper	138
59.1	Time_Stepper	138
59.2	solve	138
60	numerical-methods/ode	138
60.1	bvp2fdm	138
60.2	bvp2wavetrain	139
60.3	bvp2wavetwopass	139
60.4	ivp_euler_forward	139
60.5	ivp_euler_forward2	139
60.6	ivprk2	139
60.7	ode2_matrix	139
60.8	ode2characteristic	139
60.9	step_trapezoidal	140
60.10	test_bvp2	140
61	numerical-methods/optimisation	140
61.1	aitken_iteration	140
61.2	anderson_iteration	140
61.3	armijo_stopping_criterion	140
61.4	astar	140
61.5	binsearch	140
61.6	bisection	140
61.7	box1	141
61.8	box2	141
61.9	cauchy	141
61.10	cauchy2	141
61.11	directional_derivative	141
61.12	dud	141
61.13	extreme3	141
61.14	extreme_quadratic	142
61.15	ftest	142
61.16	fzero_bisect	142
61.17	fzero_newton	142
61.18	grad	142
61.19	hessian	142

61.20	hessian_from_gradient	143
61.21	hessian_projected	143
61.22	line_search	143
61.23	line_search2	143
61.24	line_search_polynomial	143
61.25	line_search_polynomial2	144
61.26	line_search_quadratic	144
61.27	line_search_quadratic2	144
61.28	line_search_wolfe	144
61.29	ls_bgfs	144
61.30	ls_broyden	145
61.31	ls_generalized_secant	145
61.32	nlcg	145
61.33	nlls	145
61.34	picard	145
61.35	poly_extrema	146
61.36	quadratic_function	146
61.37	quadratic_programming	146
61.38	quadratic_step	146
61.39	rosenbrock	146
61.40	sqr_heron	146
61.41	test_directional_derivative	146
61.42	test_dud	146
61.43	test_fzero_newton	146
61.44	test_line_search_quadratic2	147
61.45	test_ls_generalized_secant	147
61.46	test_nlcg_6_order	147
61.47	test_nlls	147
62	numerical-methods/pde	147
62.1	laplacian2d_fundamental_solution	147
63	numerical-methods/piecewise-polynomials	147
63.1	Hermite1	147
63.2	hp2_fit	147
63.3	hp2_predict	148
63.4	hp_predict	148
63.5	hp_regress	148
63.6	lp_count	148
63.7	lp_predict	148
63.8	lp_regress	148
63.9	lp_regress_	148
64	numerical-methods	148

64.1	test_adams_bashforth	148
65	regression/@PolyOLS	149
65.1	PolyOLS	149
65.2	coefftest	149
65.3	detrend	149
65.4	fit	149
65.5	fit_	149
65.6	predict	149
65.7	predict_	149
65.8	slope	149
66	regression/@PowerLS	150
66.1	PowerLS	150
66.2	fit	150
66.3	predict	150
66.4	predict_	150
67	regression/@Theil	150
67.1	Theil	150
67.2	detrend	150
67.3	fit	151
67.4	predict	151
67.5	slope	151
68	regression	151
68.1	Theil_Multivariate	151
68.2	areg	151
68.3	ginireg	151
68.4	hesssimplereg	152
68.5	lllin	152
68.6	lsq_sparam	152
68.7	polyfitd	152
68.8	regression_method_of_moments	152
68.9	robustlinreg	152
68.10	theil2	153
68.11	theil_generalised	153
68.12	total_least_squares	153
68.13	weighted_median_regression	153
69	set-theory	153
69.1	issubset	153
70	mathematics	153
70.1	shuffle_index	153

71	signal-processing	154
71.1	acf_bp	154
71.2	acf_decay_rate	154
71.3	acf_effective_sample_size	154
71.4	acf_genton	154
71.5	acf_hp	154
71.6	acf_lp	154
71.7	acf_periodic_additive_noise	154
71.8	acf_periodic_windowed	154
71.9	acf_radial	154
71.10	acfar1	155
71.11	acfar1_2	155
71.12	acfar2	155
71.13	acfar2_2	155
71.14	ar1_cutoff_frequency	155
71.15	ar1_effective_sample_size	155
71.16	ar1_mse_mu_single_sample	155
71.17	ar1_mse_pop	156
71.18	ar1_mse_range	156
71.19	ar1_spectrum	156
71.20	ar1_to_tikhonov	156
71.21	ar1_var_factor	156
71.22	ar1_var_factor_	156
71.23	ar1_var_range2	156
71.24	ar1delay	157
71.25	ar1delay_old	157
71.26	ar2_acf2c	157
71.27	ar2conv	157
71.28	ar2dof	157
71.29	ar2param	157
71.30	asymwin	157
71.31	autocorr2	158
71.32	autocorr_fft	158
71.33	bandpass	158
71.34	bandpass1d	158
71.35	bandpass1d_fft	158
71.36	bandpass1d_implicit	158
71.37	bandpass2	158
71.38	bandpass2d	158
71.39	bandpass2d_2	158
71.40	bandpass2d_fft	159
71.41	bandpass2d_implicit	159
71.42	bandpass2d_iso	159
71.43	bandpass_max	159

71.44	bandpass_max2	159
71.45	bartlett	159
71.46	bin1d	159
71.47	bin2d	160
71.48	binormrnd	160
71.49	coherence	160
71.50	conv1_man	160
71.51	conv2_man	160
71.52	conv2z	160
71.53	conv30	160
71.54	conv_	160
71.55	conv_centered	161
71.56	convz	161
71.57	cosexpdelay	161
71.58	csmooth	161
71.59	daniell_window	161
71.60	danielle_window	161
71.61	db2neper	161
71.62	db2power	162
71.63	derive_bandpass_normalization_and_zeros	162
71.64	derive_danielle_weight	162
71.65	derive_limit_0_acfar	162
71.66	detect_peak	162
71.67	digital_low_pass_filter	162
71.68	doublesum_ij	162
71.69	effective_sample_size_to_ar1	162
71.70	filt_hodges_lehman	163
71.71	filter1	163
71.72	filter2	163
71.73	filter_	163
71.74	filter_arg	163
71.75	filter_f0_to_rho	163
71.76	filter_r_to_f0	163
71.77	filter_rho_to_f0	163
71.78	filter_twosided	163
71.79	filteriir	164
71.80	filterp	164
71.81	filterp1	164
71.82	filterstd	165
71.83	firls_man	165
71.84	fit_spectral_density	165
71.85	flattopwin	165
71.86	frequency_response_boxcar	165
71.87	freqz_boxcar	165

71.88	gaussfilt1	165
71.89	hanchangewin	165
71.90	hanchangewin2	166
71.91	hanwin	166
71.92	hanwin_	166
71.93	high_pass_1d_simple	166
71.94	highpass	166
71.95	highpass1d_implicit	166
71.96	highpass2d_fft	166
71.97	highpass2d_implicit	166
71.98	jackknife_block	166
71.99	kaiserwin	167
71.100	kalman	167
71.101	lanczoswin	167
71.102	last	167
71.103	lowpass	167
71.104	lowpass1d_fft	167
71.105	lowpass1d_implicit	167
71.106	lowpass2	167
71.107	lowpass2d_2	167
71.108	lowpass2d_anisotropic	168
71.109	lowpass2d_fft	168
71.110	lowpass2d_implicit	168
71.111	lowpass_iir	168
71.112	lowpass_iir_symmetric	168
71.113	lowpassfilter2	168
71.114	maxfilt1	168
71.115	meanfilt1	168
71.116	medfilt1_man	168
71.117	medfilt1_man2	169
71.118	medfilt1_padded	169
71.119	medfilt1_reduced	169
71.120	mid_term_single_sample	169
71.121	minfilt1	169
71.122	mu2ar1	169
71.123	mysmooth	169
71.124	nanautocorr	169
71.125	nanmedfilt1	170
71.126	neper2db	170
71.127	peaks_man	170
71.128	periodogram	170
71.129	periodogram_annular	170
71.130	periodogram_bartlett	170
71.131	periodogram_bootstrap	170

71.132	periodogram_confidence_interval	170
71.133	periodogram_median	170
71.134	periodogram_p_value	171
71.135	periodogram_qq	171
71.136	periodogram_quantiles	171
71.137	periodogram_std	171
71.138	periodogram_test	171
71.139	polyfilt1	171
71.140	qmedfilt1	171
71.141	randar1	171
71.142	randar1_dual	172
71.143	randar2	172
71.144	randarp	172
71.145	range_window	172
71.146	rectwin	172
71.147	recursive_sum	172
71.148	select_range	172
71.149	smooth1d_parametric	172
71.150	smooth2	173
71.151	smooth_man	173
71.152	smooth_parametric	173
71.153	smooth_parametric2	173
71.154	smooth_with_splines	173
71.155	smoothfft	173
71.156	spectral_density_ar2	173
71.157	spectral_density_area	173
71.158	spectral_density_bandpass_continuous	173
71.159	spectral_density_bandpass_discrete	174
71.160	spectral_density_brownian_phase	174
71.161	spectral_density_flat	174
71.162	spectral_density_highpass	174
71.163	spectral_density_lorentzian	174
71.164	spectral_density_lowpass	174
71.165	spectral_density_radial	174
71.166	spectral_density_wperiodic	174
71.167	spectrogram	174
71.168	std_window	175
71.169	sum_i_lag	175
71.170	sum_ii	175
71.171	sum_ii_	175
71.172	sum_ij	175
71.173	sum_ij_	175
71.174	sum_ij_partial	175
71.175	sum_multivar	175

71.176	test_acfar1	176
71.177	test_acfar1_2	176
71.178	test_acfar1_3	176
71.179	test_acfar1_4	176
71.180	test_acfar2	176
71.181	test_ar1_var_factor	176
71.182	test_ar1_var_factor_2	176
71.183	test_ar1_var_mu_single_sample	176
71.184	test_ar1_var_pop	176
71.185	test_ar1_var_pop_1	176
71.186	test_ar1delay	177
71.187	test_bivariate_covariance_term	177
71.188	test_convexity	177
71.189	test_lanczoswin	177
71.190	test_madcorr	177
71.191	test_randar1	177
71.192	test_randar1_multivariate	177
71.193	test_randar2	177
71.194	test_sum_ij	177
71.195	test_sum_multivar	177
71.196	test_trifilt1	178
71.197	test_wautocorr	178
71.198	test_wavelet_transform	178
71.199	test_wordfilt	178
71.200	test_xar1_mid_term	178
71.201	tikhonov_to_ar1	178
71.202	trapwin	178
71.203	trifilt1	178
71.204	trifilt2	178
71.205	triwin	179
71.206	triwin2	179
71.207	varar1	179
71.208	welch_spectrogram	179
71.209	wfilt	179
71.210	winbandpass	179
71.211	window2d	179
71.212	window_make_odd	179
71.213	winfilt0	180
71.214	winlength	180
71.215	wmeanfilt	180
71.216	wmedfilt	180
71.217	wordfilt	180
71.218	wordfilt_edgeworth	180
71.219	wrapphase	180

71.220	xar1	180
71.221	xcorr_man	181
72	sorting	181
72.1	sort2	181
72.2	sort2d	181
73	special-functions	181
73.1	bessel_sphere	181
73.2	digamma_man	181
73.3	exp10	181
73.4	hankel_sphere	181
73.5	hermite	182
73.6	legendre_man	182
73.7	neumann_sphere	182
74	statistics	182
74.1	atan_s2	182
74.2	beta_kurt	182
74.3	beta_mean	182
74.4	beta_mode_to_parameter	182
74.5	beta_skew	183
74.6	beta_std	183
74.7	chi2_kurt	183
74.8	chi2_mean	183
74.9	chi2_skew	183
74.10	chi2_std	183
74.11	coefficient_of_determination	183
74.12	conditional_expectation_normal	183
74.13	correlation_confidence_pearson	183
75	statistics/distributions	184
75.1	PDF	184
75.2	binorm_separation_coefficient	184
75.3	binormcdf	184
75.4	binormfit	184
75.5	binormpdf	184
75.6	edgeworth_cdf	184
75.7	edgeworth_pdf	184
75.8	gam_moment2param	185
75.9	logn_mean	185
75.10	logn_mode	185
75.11	logn_moment2param	185
75.12	logn_param2moment	185

75.13	logn_std	185
75.14	lognpdf_	185
75.15	pdfsample	185
75.16	t2cdf	186
75.17	t2inv	186
76	statistics	186
76.1	example_standard_error_of_sample_quantiles	186
76.2	f_mean	186
76.3	f_std	186
76.4	f_var_finite	186
76.5	gam_mean	186
76.6	gam_std	186
76.7	gamma_mode_to_parameter	187
76.8	gaussfit3	187
76.9	gaussfit_quantile	187
76.10	geoserr	187
76.11	geostd	187
76.12	hodges_lehmann_correlation	187
76.13	hodges_lehmann_dispersion	187
77	statistics/information-theory	188
77.1	akaike_information_criterion	188
77.2	bayesian_information_criterion	188
78	statistics	188
78.1	kurtncdf	188
78.2	kurtnpdf	188
78.3	kurtosis_bias_corrected	188
78.4	limit	189
78.5	logfactorial	189
78.6	loglogpdf	189
78.7	lognfit_quantile	189
78.8	logskewcdf	189
78.9	logskewpdf	189
79	statistics/logu	189
79.1	lambertw_numeric	189
79.2	logtrialtcd	189
79.3	logtrialtinv	190
79.4	logtrialtmean	190
79.5	logtrialtpdf	190
79.6	logtrialtrnd	190
79.7	logtricdf	190

79.8	logtriinv	190
79.9	logtrimean	191
79.10	logtripdf	191
79.11	logtrirnd	191
79.12	logucdf	191
79.13	logucm	191
79.14	loguinv	191
79.15	logumean	191
79.16	logupdf	191
79.17	logurnd	192
79.18	loguvar	192
79.19	medlogu	192
79.20	test_logurnd	192
79.21	tricdf	192
79.22	triinv	192
79.23	trimediam	192
79.24	tripdf	192
79.25	trirnd	193
80 statistics		193
80.1	max_exprnd	193
80.2	maxnnormals	193
80.3	mean_generalized_gampdf	193
80.4	midrange	193
80.5	minavg	193
80.6	mode_man	193
81 statistics/moment-statistics		194
81.1	autocorr_man3	194
81.2	autocorr_man4	194
81.3	autocorr_man5	194
81.4	blockserr	194
81.5	comoment	194
81.6	corr_man	195
81.7	cov_man	195
81.8	dof	195
81.9	edgeworth_quantile	195
81.10	effective_sample_size	195
81.11	f_correlation	195
81.12	f_finite	196
81.13	lmean	196
81.14	lmoment	196
81.15	maskmean	196
81.16	masknanmean	196

81.17	mean1	196
81.18	mean_man	196
81.19	mse	196
81.20	nanautocorr_man1	197
81.21	nanautocorr_man2	197
81.22	nanautocorr_man4	197
81.23	nancorr	197
81.24	nancumsum	197
81.25	nanlmean	197
81.26	nanr2	197
81.27	nanrms	198
81.28	nanrmse	198
81.29	nanserr	198
81.30	nanwmean	198
81.31	nanwstd	198
81.32	nanwvar	198
81.33	nanxcorr	198
81.34	pearson	199
81.35	pearson_to_kendall	199
81.36	pool_samples	199
81.37	qmean	199
81.38	range_mean	199
81.39	rmse_	199
81.40	serr	199
81.41	serr1	199
81.42	test_qskew	200
81.43	test_qstd_qskew_optimal_p	200
81.44	wautocorr	200
81.45	wcorr	200
81.46	wcov	200
81.47	wdof	200
81.48	wkurt	200
81.49	wmean	201
81.50	wrms	201
81.51	wserr	201
81.52	wskew	201
81.53	wstd	201
81.54	wvar	201
82	statistics	202
82.1	nangeomean	202
82.2	nangeostd	202
83	statistics/nonparametric-statistics	202

83.1	kernel1d	202
83.2	kernel2d	202
84	statistics	202
84.1	normmoment	202
84.2	normpdf2	202
85	statistics/order-statistics	203
85.1	hodges_lehmann_location	203
85.2	kendall	203
85.3	kendall_to_pearson	203
85.4	mad2sd	203
85.5	madcorr	203
85.6	median2_holder	203
85.7	median_ci	204
85.8	median_man	204
85.9	mediani	204
85.10	nanmadcorr	204
85.11	nanwmedian	204
85.12	nanwquantile	204
85.13	oja_median	204
85.14	qkurtosis	205
85.15	qmoments	205
85.16	qskew	205
85.17	qskewq	205
85.18	qstdq	205
85.19	quantile1_optimisation	205
85.20	quantile2_breckling	206
85.21	quantile2_chaudhuri	206
85.22	quantile2_projected	206
85.23	quantile2_projected2	206
85.24	quantile_envelope	206
85.25	quantile_regression_simple	206
85.26	ranking	206
85.27	spatial_median	206
85.28	spatial_quantile	207
85.29	spatial_quantile2	207
85.30	spatial_quantile3	207
85.31	spatial_rank	207
85.32	spatial_sign	207
85.33	spatial_signed_rank	207
85.34	spearman	207
85.35	spearman_rank	207
85.36	spearman_to_pearson	208

85.37	wmedian	208
85.38	wquantile	208
86	statistics	208
86.1	qstd	208
86.2	quantile_extrap	208
86.3	quantile_sin	208
87	statistics/random-number-generation	208
87.1	laplacernd	208
87.2	randc	208
87.3	skewness2param	209
87.4	skewpdf_central_moments	209
87.5	skewrnd	209
87.6	skewrnd2	209
88	statistics	209
88.1	range	209
88.2	resample_with_replacement	209
89	statistics/resampling-statistics/@Jackknife	209
89.1	Jackknife	209
89.2	estimated_STATIC	210
89.3	matrix1_STATIC	210
89.4	matrix2	210
90	statistics/resampling-statistics	210
90.1	block_jackknife	210
90.2	jackknife_moments	210
90.3	moving_block_jackknife	211
90.4	randblockserr	211
90.5	resample	211
91	statistics	211
91.1	scale_quantile_sd	211
91.2	sd_sample_quantiles	212
91.3	skew_generalized_normal_fit	212
91.4	skew_generalized_normpdf	212
91.5	skewcdf	212
91.6	skewparam_to_central_moments	212
91.7	skewpdf	212
91.8	spatialrnd	212
91.9	test_mean_generalized_gampdf	212
91.10	test_skew_generalized_normpdf	213
91.11	trimmed_mean	213

91.12	ttest2_man	213
91.13	ttest_man	213
91.14	ttest_paired	213
91.15	wgeomean	213
91.16	wgeovar	213
91.17	wharmean	214
91.18	wharstd	214
91.19	wharvar	214
91.20	wnormpdf	214
92 mathematics		214
92.1	ternary_diagram	214
93 test/master		214
93.1	dat_test_lanczos_3d_k_20_n_40	214
93.2	poisson2d.blk	214
93.3	qr_implicit_givens_2	214
93.4	spectral_derivative_2d	215
93.5	test_2d_eigensolver_hydrogen	215
93.6	test_2d_refine	215
93.7	test_3d_eigensolver_hydrogen	215
93.8	test_FEM	215
93.9	test_Mesh_3d	215
93.10	test_arnoldi	215
93.11	test_arpackc	215
93.12	test_assemble	215
93.13	test_assembly_performance	215
93.14	test_bc_one_sided	216
93.15	test_compare_solvers	216
93.16	test_complete	216
93.17	test_convergence	216
93.18	test_convergence_b	216
93.19	test_df_2d	216
93.20	test_eig_algs	216
93.21	test_eig_inverse	216
93.22	test_eigs_lanczos	216
93.23	test_eigs_lanczos_1	216
93.24	test_eigs_lanczos_2	217
93.25	test_eigs_lanczos_performance	217
93.26	test_fdm	217
93.27	test_fdm_d_vargrid	217
93.28	test_fdm_spectral	217
93.29	test_fem	217
93.30	test_fem_1d	217

93.31	test_fem_1d_higher_order	217
93.32	test_fem_2d_adaptive	217
93.33	test_fem_2d_higher_order	217
93.34	test_fem_3d_higher_order	218
93.35	test_fem_3d_refine	218
93.36	test_fem_b	218
93.37	test_fem_derivative	218
93.38	test_fem_quadrature	218
93.39	test_final	218
93.40	test_fix_substitution	218
93.41	test_forward	218
93.42	test_get_sparse_arrays	218
93.43	test_harmonic_oscillator	218
93.44	test_high_order_fdm_periodic_bc	219
93.45	test_hydrogen_wf	219
93.46	test_ichol	219
93.47	test_interpolation	219
93.48	test_inverse_problem	219
93.49	test_it_vs_exact	219
93.50	test_jama	219
93.51	test_jd	219
93.52	test_jdqz	219
93.53	test_lanczos_2	219
93.54	test_lanczos_biorthogonal	220
93.55	test_laplacian	220
93.56	test_laplacian_non_uniform	220
93.57	test_laplacian_simple	220
93.58	test_mesh_2d_uniform	220
93.59	test_mesh_2d_uniform_2	220
93.60	test_mesh_circle	220
93.61	test_mesh_generation	220
93.62	test_mesh_interpolate	220
93.63	test_mg	220
93.64	test_minres_recycle	221
93.65	test_multigrid	221
93.66	test_nc	221
93.67	test_nonuniform_symmetric	221
93.68	test_pde	221
93.69	test_permutation	221
93.70	test_poison_fem	221
93.71	test_polar	221
93.72	test_potential	221
93.73	test_powers	221
93.74	test_precondition	222

93.75	test_project_rectangle	222
93.76	test_qr	222
93.77	test_quantum_well	222
93.78	test_radial_adaptive	222
93.79	test_radial_confinement	222
93.80	test_radial_fixes	222
93.81	test_refine_2d	222
93.82	test_refine_2d_b	222
93.83	test_refine_3d	222
93.84	test_refine_structural	223
93.85	test_regularisation	223
93.86	test_round_off	223
93.87	test_schrödinger_potentials	223
93.88	test_uniform_mesh	223
93.89	test_vargrid	223
94	test	223
94.1	test_bandpass2d	223
94.2	test_gaussfit3	223
94.3	test_geoserr	223
94.4	test_hexagonal_pattern	224
94.5	test_lognfit_quantile	224
94.6	test_lowpass1d_implicit	224
94.7	test_lowpass2d_anisotropic	224
94.8	test_lowpass2d_fft	224
94.9	test_max_normal	224
94.10	test_mtimes3x3	224
95	wavelet	224
95.1	contiuuous_wavelet_transform	224
95.2	cwt_man	225
95.3	example_wavelets	225
95.4	phasewrap	225
95.5	test_cwt_man	225
95.6	test_phasewrap	225
95.7	test_wavelet	225
95.8	test_wavelet2	225
95.9	test_wavelet_analysis	225
95.10	test_wavelet_reconstruct	225
95.11	test_wtc	226
95.12	wavelet	226
95.13	wavelet_reconstruct	226
95.14	wavelet_transform	226

1 calendar

1.1 days_per_month

1.2 isnight

2 mathematics

mathematical functions of various kind

2.1 cast_byte_to_integer

cast byte to integer

3 complex-analysis

operations on complex numbers

3.1 complex_exp_product_im_im

product of the imaginary part of two complex exponentials

the product has two frequency components

input :

c : complex amplitudes

o : frequencies

output :

cp : amplitude of the product

op : frequencies of the product

3.2 complex_exp_product_im_re

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

the product has two frequency components

```

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

```

3.3 complex_exp_product_re_im

the product has two frequency components

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

```

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

```

3.4 complex_exp_product_re_re

product of the real part of two complex exponentials

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

the product has two frequency components

```

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

```

3.5 croots

nth-roots of a complex number

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

r : roots of the complex number

3.6 root_complex

root of a complex number

3.7 test_imroots

4 derivation

derivation of several functions by means of symbolic computation

4.1 derive_acfar1

4.2 derive_ar1_spectral_density

4.3 derive_ar2param

4.4 derive_arc_length

4.5 derive_fourier_power

4.6 `derive_fourier_power_exp`

4.7 `derive_laplacian_curvilinear`

4.8 `derive_laplacian_fourier_piecewise_linear`

4.9 `derive_logtripdf`

4.10 `derive_smooth1d_parametric`

4.11 `derive_spectral_density_bandpass_initial_condition`

5 `derivation/master`

5.1 `derive_bc_one_sided`

5.2 `derive_convergence`

5.3 `derive_error_fdm`

5.4 `derive_fdm_poly`

5.5 `derive_fdm_power`

5.6 `derive_fdm_taylor`

5.7 `derive_fdm_vargrid`

5.8 `derive_fem_2d_mass`

5.9 `derive_fem_error_2d`

5.10 `derive_fem_error_3d`

5.11 `derive_fem_sym_2d`

5.12 `derive_grid_constants`

5.13 `derive_interpolation`

5.14 `derive_laplacian`

5.15 `derive_limit`

5.16 `derive_nc_1d`

5.17 `derive_nc_1d_`

5.18 `derive_nc_2d`

5.19 `derive_nonuniform_symmetric`

%

5.20 `derive_richardson`

5.21 `derive_sum`

5.22 `nn`

5.23 `test_derive`

5.24 test_derive_fdm_poly

5.25 test_filter

5.26 test_vargrid

6 derivation

derivation of several functions by means of symbolic computation

6.1 simplify_atan

symbolic simplification of the arcus tangent

7 mathematics

mathematical functions of various kind

7.1 entropy

8 finance

8.1 derive_skewrnd_walsh_paramter

8.2 gbm_cdf

8.3 `gbm_fit`

8.4 `gbm_fit_old`

8.5 `gbm_inv`

8.6 `gbm_mean`

8.7 `gbm_median`

8.8 `gbm_pdf`

8.9 `gbm_simulate`

8.10 `gbm_skewness`

8.11 `gbm_std`

8.12 `gbm_transform_time_step`

8.13 put_price_black_scholes

8.14 skewgbm_simulate

8.15 skewrnd_walsh

9 finance/test

9.1 test_gbm

9.2 test_gbm_pdf

9.3 test_skewrnd_walsh

10 fourier/@STFT

10.1 STFT

class for short time fourier transform

Note: the interval T_i should be set to at least $2 \cdot \max(T)$, as
otherwise coefficients

tend to oscillate in the presence of noise

Note: for convenience, the independent variable is labeled as time
(t),

but the independent variable is arbitrary, so it works
likewise in space

10.2 itransform

inverse of the short time fourier transform

10.3 stft_

static wrapper for STFT

10.4 stftmat

transformation matrix for the short time fourier transform

10.5 transform

short time fourier transform

11 fourier

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

11.1 amplitude_from_peak

amplitude and standard deviation of the amplitude of a frequency
component

represented by a peak in the fourier domain

input :

h : peak height

w : peak width at half height

output:

a : amplitude in real space

s : standard deviation of the frequency (!)

11.2 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_derivative_matrix_1d`

11.19 `fourier_derivative_matrix_2d`

11.20 `fourier_expand`

expand values of fourier series

11.21 `fourier_fit`

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

11.22 `fourier_interpolate`

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

11.23 `fourier_matrix`

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

11.24 `fourier_matrix2`

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

11.25 `fourier_matrix3`

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

11.26 `fourier_matrix_exp`

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

11.27 `fourier_multiplicative_interaction_coefficients`

11.28 `fourier_power`

powers of a continuous fourier series in sin/cos form

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

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

11.29 `fourier_power_exp`

powers of the continuous fourier series

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

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

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

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

11.30 `fourier_predict`

expand a continuous fourier series at times t

11.31 `fourier_quadratic_interaction_coefficients`

11.32 `fourier_random_phase_walk`

11.33 `fourier_range`

approximate range of a continuous Fourier series with 2 components
 $\text{range}(y) = \max(y) - \min(y)$

11.34 `fourier_regress`

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

11.35 `fourier_resampled_fit`

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

11.36 `fourier_resampled_predict`

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

11.37 `fourier_signed_square`

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

11.38 `fourier_transform`

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

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

output :
 A : fourier matrix

```

p : fourier transformation of b
tt : TODO

```

11.39 fourier_transform_fractional

11.40 hyperbolic_fourier_box

11.41 idftmtx_man

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

11.42 laplace_2d_pwlinear

solution to the Laplacian in two dimensions for a finite
 rectangular domain
 with piecewise constant boundary conditions
 linear system with 4 unknowns per frequency component
 these are coefficients of s, c, sh, ch

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

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

11.43 nanfft

discrete fourier transform of a data series with gaps

11.44 peaks

peaks of the power spectrum of a discrete fourier transform

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

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

works best, when spectrum has been smoothened

input :

f : frequency

y : absolute value of fourier transform (power spectrum)

L : length in space or time of series

output :

a0 : amplitude

s0 : standard deviation (error?) of amplitude

w0 : width of peak

lambda = wave length (period?)

pdx : index of peak

f : frequency (if not given as input)

11.45 roots_fourier

zeros of continuous fourier series series

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

11.46 spectral_density

spectral density

11.47 test_complex_exp_product

11.48 test_fourier_filter

11.49 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

euclidan distance between two points

13.13 distance2

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

13.14 dot

dot product

13.15 edge_length

edge length

13.16 enclosed_angle

angle enclosed between two lines

13.17 enclosing_triangle

smallest enclosing equilateral triangle with bottom side paralle to
X axis

13.18 hexagon

coordinates of a hexagon, scaled and rotated

13.19 inPolygon

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

13.20 inTetra

flag points contained in tetrahedron

13.21 inTetra2

flag points contained in tetrahedron

13.22 inTriangle

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

13.23 intersect

intersect between two lines

13.24 lineintersect

intersect of two lines

13.25 lineintersect1

intersect of two lines

13.26 minimum_distance_lines

minimum distance of two lines in three dimensions

13.27 mittenpunkt

mittenpunkt of a triangle

13.28 nagelpoint

nagelpoint of a triangle

13.29 onLine

13.30 orthocentre

orthocentre of triangle

13.31 plumb_line

13.32 poly_area

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

13.33 poly_edges

edges of a polygon

13.34 poly_set

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

13.35 poly_width

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

13.36 polyxpoly

intersections of two polygons

13.37 project_to_curve

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

13.38 quad_isconvex

13.39 random_disk

draw random points on the unit disk

13.40 random_simplex

random point inside of a triangle

13.41 sphere_volume

volume of a sphere
function v = sphere_volume(r)

13.42 tetra_volume

volume of a tetrahedron

13.43 tobarycentric

cartesian to barycentric coordinates

13.44 tobarycentric1

cartesian to barycentric coordinates

13.45 tobarycentric2

cartesian to barycentric coordinates

13.46 tobarycentric3

cartesian to barycentric coordinates

13.47 tri_angle

cos of angles of a triangle

13.48 tri_area

angle of a triangle

13.49 tri_centroid

centroid of a triangle

13.50 tri_distance_opposit_midpoint

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

13.51 tri_edge_length

edge length of a triangle

13.52 tri_edge_midpoint

mid point of a triangle

13.53 tri_excircle

excircle of a triangle

13.54 tri_height

height of a triangle

13.55 tri_incircle

incircle of a triangle

13.56 tri_isacute

flag acute triangles

13.57 tri_isobtuse

flag obtuse triangles

13.58 tri_semiperimeter

semiperimeter of a triangle

13.59 tri_side_length

edge length of triangle

14 geometry

14.1 Polygon

Simple 2D polygon class

Polygon properties:

- x - x coordinates of polygon
- y - y coordinates of polygon
- nnodes - number of nodes in the polygon

Polygon methods:

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

14.2 bounding_box

bounding box of X

14.3 curvature_1d

curvature of a sampled parametric curve in two dimensions

14.4 cvt

centroidal voronoi tessellation

14.5 deg_to_frac

degree, minutes and seconds to fractions

14.6 ellipse

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

14.7 ellipseX

x-coordinates of y-coordinates of an ellipse

14.8 ellipseY

14.9 first_intersect

get first intersection between lines in A and B

14.10 golden_ratio

golden ratio

14.11 hypot3

hypotenuse in 3D

14.12 meanangle

weighted mean of angles

14.13 meanangle2

mean angle

14.14 meanangle3

mean angle

14.15 meanangle4

mean angle

14.16 medianangle

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

14.17 medianangle2

median angle

input
alpha : x*m, [rad] angle

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

14.18 pilim

limit to +- pi

14.19 streamline_radius_of_curvature

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

15 mathematics

mathematical functions of various kind

15.1 hexagonal_pattern

16 histogram/@Histogram

16.1 2x

16.2 Histogram

16.3 bimodes

16.4 cdf

16.5 cdfS

16.6 chi2test

16.7 `cmoment`

16.8 `cmomentS`

16.9 `entropy`

16.10 `entropyS`

16.11 `export_csv`

16.12 `iquantile`

16.13 `kstest`

16.14 `kurtosis`

16.15 `kurtosisS`

16.16 `mean`

16.17 meanS

16.18 median

16.19 medianS

16.20 mode

16.21 modeS

16.22 moment

16.23 momentS

16.24 pdf

16.25 quantile

16.26 quantileS

16.27 resample

16.28 setup

16.29 skewness

16.30 skewnessS

16.31 stairs

16.32 stairsS

16.33 std

16.34 stdS

16.35 var

16.36 varS

17 histogram

17.1 hist_man

17.2 histadapt

17.3 histconst

17.4 pdf_poly

17.5 plotcdf

17.6 test_histogram

18 mathematics

mathematical functions of various kind

18.1 imrotmat

19 linear-algebra

19.1 averaging_matrix_2

19.2 colnorm

norms of columns

19.3 condest_

estimation of the condition number

19.4 connectivity_matrix

20 linear-algebra/coordinate-transformation

20.1 barycentric2cartesian

barycentric to cartesian coordinates

20.2 barycentric2cartesian3

convert barycentric to cartesian coordinates

20.3 cartesian2barycentric

cartesian to barycentric coordinates

20.4 cartesian_to_unit_triangle_basis

transform coordinates into unit triangle

20.5 ellipsoid2geoid

20.6 example_approximate_utm_conversion

20.7 latlon2utm

transform latitude and longitude to WGS84 UTM

20.8 latlon2utm_simple

20.9 lowrance_mercator_to_wgs84

convert lowrance coordinates to wgs84

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

20.10 nmea2utm

convert nmea messages to utm coordinates

20.11 sn2xy

convert sn to xy coordinates

20.12 unit_triangle_to_cartesian

transform coordinates in unit triangle to cartesian coordinates

20.13 utm2latlon

convert wgs84 utm to latitude and longitude

20.14 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

20.15 xy2sn

convert cartesian to streamwise coordiantes

20.16 xy2sn.java

use java port for speed up

20.17 xy2sn_old

transform points from cartesian into streamwise coordinates

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

21 linear-algebra

21.1 det2x2

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

21.2 det3x3

determinant of stacked 3x3 matrices

21.3 det4x4

determinant of stacked 4x4 matrices

21.4 diag2x2

diagonal of stacked 2x2 matrices

21.5 down

21.6 eig2x2

eigenvalues of stacked 2x2 matrices

22 linear-algebra/eigenvalue

22.1 eig_bisection

22.2 eig_inverse

22.3 eig_inverse_iteration

22.4 eig_power_iteration

23 linear-algebra/eigenvalue/jacobi-davidson

23.1 afun_jdm

23.2 davidson

23.3 jacobi_davidson

23.4 jacobi_davidson_qr

23.5 jacobi_davidson_qz

23.6 jacobi_davidson_simple

23.7 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*M; V(:,j+1)=[];R=R'*R; M=H(1:j,:)'
%===== ARNOLDI (for initializing spaces)
=====
%===== END ARNOLDI
=====

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

% compute vectors and matrices for skew projection
% solve preconditioned system
% 0 step of bicgstab eq. 1 step of bicgstab

```

```

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

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

```

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

23.8 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/snorm]];
% sufficient accuracy. No need to update r,u
% Do the orth to Q explicitly
% In exact arithmetic not needed, but
% appears to be more stable.
% plot(HIST(:,1),log10(HIST(:,2)+eps),'*'), drawnow, pause
% 0 step of gmres eq. 1 step of gmres
% Then x is a multiple of b
%=====

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

```

```

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

```

23.9 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

```

23.10 jdqz

```

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

```

```

% Check for stagnation
% compute approximate eigenpair
% Compute approximate eigenpair and residual
% display history
% save history
% check convergence
% expand Schur form
% ZastQ=Z'*Q0
% the final Qschur
% check for conjugate pair
% t perp Zschur, t in span(Q0,imag(q))
% To detect whether another eigenpair is accurate enough
% restart if dim(V)> jmax
%===== END JDQZ
=====
%=====
%===== PREPROCESSING
=====
%=====
%===== ARNOLDI (for initial spaces)
=====
%% then precondition=I and target = 0: apply Arnoldi with A
%===== END ARNOLDI
=====
%=====
%===== POSTPROCESSING
=====
%=====
%===== SORT QZ DECOMPOSITION INTERACTION MATRICES
=====
%===== COMPUTE SORTED JORDAN FORM
=====
%===== END JORDAN FORM
=====
%===== OUTPUT
=====
%=====
%===== UPDATE PRECONDITIONED SCHUR VECTORS
=====
%=====
%=====

```

```

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

% solve preconditioned system
%=====

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

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

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

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

```

```

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

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

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

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

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

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

%=====

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

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

```



```

%=====

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

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

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

%=====

%=====

%=====

```

23.11 mfunc_jdm

23.12 mgs

23.13 minres_

23.14 mv_jacobi_davidson

24 linear-algebra

24.1 first

24.2 gershgorin_circle

range of eigenvalues determined by the gershgorin circle theorem

24.3 haussdorff

haussdorf dimension

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

Koch snow flake 3:4 -> 1.2619

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

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

24.4 ieig2x2

reconstruct matrix from eigenvalue decomposition

24.5 inv2x2

2x2 inverse of stacked matrices

24.6 inv3x3

24.7 inv4x4

inverse of stacked 4x4 matrices

24.8 kernel2matrix

25 linear-algebra/lanczos

25.1 arnoldi

25.2 arnoldi_new

25.3 eigs_lanczos_man

25.4 lanczos

25.5 lanczos_

25.6 lanczos_biorthogonal

25.7 lanczos_biorthogonal_improved

25.8 lanczos_ghep

25.9 mv_lanczos

25.10 reorthogonalise

25.11 test_lanczos

26 linear-algebra

26.1 left

left element of vector, leftmost column is extrapolated

27 linear-algebra/linear-systems

27.1 gmres_man

break on convergence

27.2 minres_recycle

28 linear-algebra

28.1 lpmean

mean of pth-power of a

28.2 lpnorm

norm of lth-power of a

28.3 matvec3

matrix-vector product of stacked matrices and vectors

28.4 max2d

maximum value and i-j index for matrix

28.5 mid

mid point between neighbouring vector elements

28.6 mpoweri

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

28.7 mtimes2x2

28.8 `mtimes3x3`

product of stacked 3x3 matrices

28.9 `nannorm`

norm of a vector, skips nan-values

28.10 `nanshift`

shift vector, but set out of range values to NaN

28.11 `nl`

number rows (lines) of a matrix

analogue to unix nl command

28.12 `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)

28.13 `normalize1`

normalize columns in x to [-1,1]

28.14 `normrows`

28.15 orth2

make matrix A orthogonal to B

28.16 orth_man

orthogonalize the columns of A

28.17 orthogonalise

make x orthogonal to Y

28.18 padd2

28.19 paddext

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

28.20 paddval1

padd values at end of x

28.21 paddval2

padd values to x

29 linear-algebra/polynomial

29.1 chebychev

chebycheff polynomials

29.2 piecewise_polynomial

evaluate piecewise polynomial

29.3 roots1

roots of linear functions

29.4 roots2

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

29.5 roots2poly

29.6 roots3

29.7 roots4

29.8 roots_piecewise_linear

29.9 test_roots4

29.10 vanderi_1d

vandermonde matrix of an integral

30 linear-algebra

30.1 randrot

random rotation matrix

30.2 right

get right column by shifting columns to left
extrapolate rightmost column

30.3 rot2

rotation matrix from angle

30.4 rot2dir

rotation matrix from direction vector

30.5 rot3

30.6 rotR

30.7 rownorm

30.8 simmilarity_matrix

30.9 spnorm

frobenius norm

30.10 spzeros

allocate a sparze matrix of zeros

30.11 test_roots3

30.12 transform_minmax

30.13 transpose3

transpose stacked 3x3 matrices

30.14 transposeall

30.15 up

30.16 vanderd_2d

31 logic

bitwise operations on integers

31.1 bitor_man

bitwise OR of the numbers of the columns of A

input:
A (positive integer)

32 master/plot

32.1 attach_boundary_value

32.2 cartesian_polar

32.3 img_vargrid

32.4 plot_basis_functions

32.5 plot_convergence

32.6 plot_dof

32.7 `plot_eigenbar`

32.8 `plot_error_estimation`

32.9 `plot_error_estimation_2`

32.10 `plot_error_fem`

32.11 `plot_fdm_kernel`

32.12 `plot_fdm_vs_fem`

32.13 `plot_fem_accuracy`

32.14 `plot_function_and_grid`

32.15 `plot_hat`

32.16 `plot_hydrogen_wf`

32.17 `plot_mesh`

32.18 `plot_mesh_2`

32.19 `plot_refine`

32.20 `plot_refine_3d`

32.21 `plot_runtime`

32.22 `plot_spectrum`

32.23 `plot_wavefunction`

33 `master/ported`

33.1 `assemble_2d_phi_phi`

33.2 `assemble_3d_dphi_dphi`

33.3 assemble_3d_phi_phi

33.4 dV_2d_

33.5 derivative_2d

33.6 derivative_3d

33.7 element_neighbour_2d

33.8 prefetch_2d_

33.9 promote_2d_3_10

33.10 promote_2d_3_15

33.11 promote_2d_3_21

33.12 promote_2d_3_6

33.13 `promote_3d_4_10`

33.14 `promote_3d_4_20`

33.15 `promote_3d_4_35`

33.16 `vander_2d`

33.17 `vander_3d`

34 `number-theory`

34.1 `ceiln`

`floor to leading n-digits`

34.2 `digitsb`

`number of digits with respect to specified base`

34.3 `floorn`

`floor to n-digits`

34.4 iseven

true for even numbers in X

34.5 multichoosek

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

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

34.6 nchoosek_man

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

34.7 pythagorean_triple

pythagorean triple

34.8 roundn

round to n digits

35 numerical-methods/differentiation

35.1 derivative1

first derivative on variable mesh
second order accurate

35.2 derivative2

second derivative on a variable mesh

36 numerical-methods/finite-difference

36.1 cdiff

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

36.2 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

36.3 central_difference

36.4 cmean

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

36.5 cmean2

36.6 derivative_matrix_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)

36.7 derivative_matrix_2_1d

finite derivative matrix of second derivative in one dimension

36.8 derivative_matrix_2d

finite difference derivative matrix in two dimensions

36.9 derivative_matrix_curvilinear

derivative matrix on a curvilinear grid

36.10 derivative_matrix_curvilinear_2

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

36.11 difference_kernel

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

36.12 diffusion_matrix_2d_anisotropic

36.13 diffusion_matrix_2d_anisotropic2

36.14 distmat

distance matrix for a 2 dimensional rectangular matrix

36.15 downwind_difference

36.16 gradpde2d

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

36.17 laplacian

36.18 laplacian_fdm

finite difference matrix of the laplacian
BC

36.19 lrmean

mean of the left and right element

37 numerical-methods/finite-difference/master

37.1 fdm_adaptive_grid

37.2 fdm_adaptive_refinement_old

37.3 fdm_assemble_d1_2d

37.4 `fdm_assemble_d2_2d`

37.5 `fdm_confinement`

37.6 `fdm_d_vargrid`

37.7 `fdm_h_unstructured`

37.8 `fdm_hydrogen_vargrid`

37.9 `fdm_mark_unstructured_2d`

37.10 `fdm_plot`

37.11 `fdm_plot_series`

37.12 `fdm_refine_2d`

37.13 `fdm_refine_3d`

37.14 `fdm_refine_unstructured_2d`

37.15 `fdm_schroedinger_2d`

37.16 `fdm_schroedinger_3d`

37.17 `relocate`

38 numerical-methods/finite-difference

38.1 `mid`

`mid point between neighbouring vector elements`

38.2 `pwmid`

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

38.3 `ratio`

`ratio of two subsequent values`

38.4 `steplength`

`step length of a vector if it were equispaced`

38.5 swapoddeven

swap odd and even elements in a vector

38.6 test_derivative_matrix_2d

38.7 test_derivative_matrix_curvilinear

38.8 test_difference_kernel

38.9 upwind_difference

39 numerical-methods/finite-element

39.1 Mesh_2d.java

39.2 Tree_2d.java

39.3 assemble_1d_dphi_dphi

39.4 assemble_1d_phi_phi

39.5 `assemble_2d_dphi_dphi_java`

39.6 `assemble_2d_phi_phi_java`

39.7 `assemble_3d_dphi_dphi_java`

39.8 `assemble_3d_phi_phi_java`

39.9 `boundary_1d`

39.10 `boundary_2d`

39.11 `boundary_3d`

39.12 `check_area_2d`

39.13 `circmesh`

39.14 `cropradius`

39.15 `display_2d`

39.16 `display_3d`

39.17 `distort`

39.18 `err_2d`

39.19 `estimate_err_2d_3`

39.20 `example_1d`

39.21 `example_2d`

39.22 `explode`

39.23 `fem_2d`

39.24 `fem_2d_heuristic_mesh`

39.25 fem_get_2d_radial

39.26 fem_interpolation

39.27 fem_plot_1d

39.28 fem_plot_1d_series

39.29 fem_plot_2d

39.30 fem_plot_2d_series

39.31 fem_plot_3d

39.32 fem_plot_3d_series

39.33 fem_plot_confine_series

39.34 fem_radial

adaptive grid
constant grid

39.35 flip_2d

39.36 get_mesh_arrays

39.37 hashkey

40 numerical-methods/finite-element/int

40.1 int_1d_gauss

40.2 int_1d_gauss_1

40.3 int_1d_gauss_2

40.4 int_1d_gauss_3

40.5 int_1d_gauss_4

40.6 int_1d_gauss_5

40.7 int_1d_gauss_6

40.8 int_1d_gauss_lobatto

40.9 int_1d_gauss_n

40.10 int_1d_nc_2

40.11 int_1d_nc_3

40.12 int_1d_nc_4

40.13 int_1d_nc_5

40.14 int_1d_nc_6

40.15 int_1d_nc_7

40.16 int_1d_nc_7_hardy

40.17 int_2d_gauss_1

40.18 int_2d_gauss_12

40.19 int_2d_gauss_13

40.20 int_2d_gauss_16

40.21 int_2d_gauss_19

40.22 int_2d_gauss_25

40.23 int_2d_gauss_3

40.24 int_2d_gauss_33

40.25 int_2d_gauss_4

40.26 int_2d_gauss_6

40.27 int_2d_gauss_7

40.28 int_2d_gauss_9

40.29 int_2d_nc_10

40.30 int_2d_nc_15

40.31 int_2d_nc_21

40.32 int_2d_nc_3

40.33 int_2d_nc_6

40.34 int_3d_gauss_1

40.35 int_3d_gauss_11

40.36 int_3d_gauss_14

40.37 int_3d_gauss_15

40.38 int_3d_gauss_24

40.39 int_3d_gauss_4

40.40 int_3d_gauss_45

40.41 int_3d_gauss_5

40.42 int_3d_nc_11

40.43 int_3d_nc_4

40.44 int_3d_nc_6

40.45 int_3d_nc_8

41 numerical-methods/finite-element

41.1 interpolation_matrix

41.2 mark

41.3 mark_1d

41.4 mesh_1d_uniform

41.5 mesh_3d_uniform

41.6 mesh_interpolate

41.7 neighbour_1d

41.8 old

41.9 pdeeig_1d

41.10 pdeeig_2d

41.11 pdeeig_3d

41.12 polynomial_derivative_1d

41.13 potential_const

41.14 potential_coulomb

41.15 potential_harmonic_oscillator

41.16 project_circle

41.17 project_rectangle

41.18 promote_1d_2_3

41.19 promote_1d_2_4

41.20 promote_1d_2_5

41.21 promote_1d_2_6

41.22 quadrilaterate

41.23 recalculate_regularity_2d

41.24 refine_1d

41.25 refine_2d_21

41.26 refine_2d_structural

41.27 regularity_1d

41.28 regularity_2d

41.29 regularity_3d

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

41.30 relocate_2d

41.31 test_circmesh

41.32 test_hermite

41.33 tri_assign_points

41.34 triangulation_uniform

41.35 vander_1d

van der Monde matrix

41.36 vanderd_1d

41.37 vanderi_1d

42 numerical-methods/finite-volume/@Advection

42.1 Advection

FVM treatment of the Advection equation

42.2 dot_advection

advection equation

43 numerical-methods/finite-volume/@Burgers

43.1 burgers_split

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

43.2 dot_burgers_fdm

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

43.3 dot_burgers_fft

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

44 numerical-methods/finite-volume/@Finite_Volume

44.1 Finite_Volume

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

44.2 apply_bc

apply boundary conditions

44.3 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);

44.4 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

44.5 step_unsplit

step in time, without splitting the inhomogeneous term

45 numerical-methods/finite-volume/@Flux_Limiter

45.1 Flux_Limiter

class of flux limiters

45.2 beam_warming

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

45.3 fromm

fromme limiter
low res

45.4 lax_wendroff

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

45.5 minmod

min-mod schock limiter

45.6 monotized_central

monotonized central flux limiter

45.7 muscl

muscl flux limiter

45.8 superbee

superbee limiter

45.9 upwind

godunov scheme
godunov, first order accurate

45.10 vanLeer

van Leer limiter

46 numerical-methods/finite-volume/@KDV

46.1 dot_kdv_fdm

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

46.2 dot_kdv_fft

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

46.3 kdv_split

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

47 numerical-methods/finite-volume/@Reconstruct_Average_E

47.1 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;
```

47.2 advect_highres

single time step for the reconstruct evolve algorithm

47.3 advect_lowres

single time step
low resolution

48 numerical-methods/finite-volume

48.1 Godunov

Godunov, upwind method for systems of pdes

48.2 Lax_Friedrich

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

48.3 Measure

48.4 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

48.5 fv_swe

wrapper for solving SWE

48.6 staggered_euler

forward euler method with staggered grid

48.7 staggered_grid

staggered grid approximation to the SWE

49 numerical-methods

49.1 grid2quad

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

50 numerical-methods/integration

50.1 cumintL

cumulative integral from left to right

50.2 cumintR

cumulative integral from right to left

50.3 int_trapezoidal

integrate y along x with the trapezoidal rule

51 numerical-methods/interpolation/@Kriging

51.1 Kriging

```
class for Kriging interpolation
```

51.2 estimate_semivariance

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

51.3 interpolate_

```
interpolate with Kriging method

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

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

52 numerical-methods/interpolation/@RegularizedInterpolator1

52.1 RegularizedInterpolator1

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

52.2 init

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

53 numerical-methods/interpolation/@RegularizedInterpolator

53.1 RegularizedInterpolator2

class for regularized interpolation on an unstructures mesh (
interpolation)

53.2 init

initialize the interpolator with a set of point samples

54 numerical-methods/interpolation/@RegularizedInterpolator

54.1 RegularizedInterpolator3

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

54.2 init

initialize the interpolator with a set of sampling points

55 numerical-methods/interpolation

55.1 IDW

spatial averaging by inverse distance weighting

55.2 IPoly

polynomial interpolation class

55.3 IRBM

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

55.4 ISparse

```
sparse interpolation class
```

55.5 Inn

```
nearest neighbour interpolation
```

55.6 Interpolator

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

55.7 fixnan

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

55.8 idw1

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

55.9 idw2

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

55.10 inner2outer

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

55.11 inner2outer2

interpolate from element (segment) centres to edge points

55.12 interp1_circular

55.13 interp1_limited

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

55.14 interp1_man

interpolate

55.15 interp1_piecewise_linear

55.16 interp1_save

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

55.17 interp1_slope

quadratic interpolation returning value and derivative(s)

55.18 interp1_smooth

55.19 interp1_unique

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

55.20 interp2_man

nearest neighbour interpolation in two dimensions

55.21 interp_angle

interpolate an angle

55.22 interp_fourier

interpolation by the fourier method

55.23 interp_fourier_batch

batch interpolation by the fourier interpolation

55.24 interp_sn

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

55.25 interp_sn2

interpolation in streamwise coordinates

55.26 interp_sn3

55.27 interp_sn_

55.28 limit_by_distance_1d

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

55.29 resample1

interpolation along a parametric curve with variable step width

55.30 resample_d_min

resample a function

55.31 resample_vector

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

55.32 test_interp1_limited

56 numerical-methods

56.1 inverse_complex

56.2 maccormack_step

56.3 minmod

57 numerical-methods/multigrid

57.1 mg_interpolate

57.2 mg_restrict

58 numerical-methods/ode/@BVPS_Characteristic

58.1 BVPS_Characteristic

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

58.2 assemble1_A

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

58.3 assemble1_A_Q

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

58.4 assemble2_A

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

58.5 assemble_AA

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

58.6 assemble_AAA

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

58.7 assemble_Ic

58.8 bvp1c

58.9 check_arguments

58.10 couple_junctions

58.11 derivative

58.12 init

58.13 inner2outer_bvp2c

58.14 reconstruct

58.15 resample

58.16 solve

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

odefun provides ode coefficients c:

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

subject to the boundary conditions

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

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

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

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

58.17 test_assemble1_A

58.18 test_assemble2_A

59 numerical-methods/ode/@Time_Stepper

59.1 Time_Stepper

59.2 solve

60 numerical-methods/ode

60.1 bvp2fdm

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

as boundary value problems by the finite difference method

odefun provides ode coefficients c:

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

subject to the boundary conditions

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

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

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

60.2 bvp2wavetrain

solve second order boundary value problem by repeated integration

60.3 bvp2wavetwopass

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

60.4 ivp_euler_forward

solve initial value problem by the euler forward method

60.5 ivp_euler_forward2

60.6 ivprk2

solve initial value problem by the two step runge kutta method

60.7 ode2_matrix

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

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

60.8 ode2characteristic

second order odes
transmitted and reflected wave

60.9 step_trapezoidal

single trapezoidal step

60.10 test_bvp2

61 numerical-methods/optimisation

61.1 aitken_iteration

61.2 anderson_iteration

61.3 armijo_stopping_criterion

armijo stopping criterion for optimizations

61.4 astar

astar path finding algorithm

61.5 binsearch

binary search on a line

61.6 bisection

bisection

61.7 box1

test objective function for optimisation routines

61.8 box2

61.9 cauchy

61.10 cauchy2

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

61.11 directional_derivative

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

61.12 dud

optimization by the dud algorithm

61.13 extreme3

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

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

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

61.14 `extreme_quadratic`

61.15 `ftest`

61.16 `fzero_bisect`

61.17 `fzero_newton`

61.18 `grad`

numerical gradient

61.19 `hessian`

numerical hessian

61.20 hessian_from_gradient

numerical hessian from gradient

61.21 hessian_projected

numerical hessian projected to one dimension

61.22 line_search

bisection routine

61.23 line_search2

bisection method

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

61.24 line_search_polynomial

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

61.25 line_search_polynomial2

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

61.26 line_search_quadratic

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

61.27 line_search_quadratic2

```
quadratic line search
```

61.28 line_search_wolfe

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

61.29 ls_bgfs

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


61.30 ls_broyden

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

BGFS:
Broyden 1965
Fletcher 1970
Goldfarb 1970
Shanno 1970

61.31 ls_generalized_secant

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

61.32 nlcg

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

61.33 nlls

non-linear least squares

61.34 picard

picard iteration

61.35 poly_extrema

extrema of a polynomial

61.36 quadratic_function

evaluate quadratic function in higher dimensions

61.37 quadratic_programming

optimize by quadratic programming

61.38 quadratic_step

single step of the quadratic programming

61.39 rosenbrock

rosenbrock test function

61.40 sqrt_heron

Heron's method for the square root

61.41 test_directional_derivative

61.42 test_dud

61.43 test_fzero_newton

61.44 test_line_search_quadratic2

61.45 test_ls_generalized_secant

61.46 test_nlcg_6_order

61.47 test_nlls

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

62 numerical-methods/pde

62.1 laplacian2d_fundamental_solution

63 numerical-methods/piecewise-polynomials

63.1 Hermite1

hermite polynomial interpolation in 1d

63.2 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)
```

63.3 hp2_predict

prediction with pw hermite polynomial
c are values at support points

63.4 hp_predict

predict with piecewise hermite polynomial

63.5 hp_regress

fit piecewise hermite polynomial
coefficients are values and derivatives

63.6 lp_count

lagrangian basis for interpolation
count number of valid samples

63.7 lp_predict

lagrangian basis piecwie interpolation, predicor

63.8 lp_regress

63.9 lp_regress_

64 numerical-methods

64.1 test_adams_bashforth

65 regression/@PolyOLS

65.1 PolyOLS

class for polynomial least squares

65.2 coefftest

65.3 detrend

detrending by polynomial regression

65.4 fit

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

65.5 fit_

fit a polynomial function

65.6 predict

predict polynomial function values

65.7 predict_

65.8 slope

slope by linear regression

66 regression/@PowerLS

66.1 PowerLS

class for power law regression

66.2 fit

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

66.3 predict

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

66.4 predict_

67 regression/@Theil

67.1 Theil

Kendal-Theil-Sen robust regression

67.2 detrend

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

67.3 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

67.4 predict

predict values and confidence intervals with the Theil-Sen method

67.5 slope

fit the slope with the Theil-Sen method

68 regression

linear and non-linear regression

68.1 Theil_Multivariate

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

68.2 areg

regression using the pth-fraction of samples with smallest residual

68.3 ginireg

gini regression

68.4 hesssimplereg

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

68.5 l1lin

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

68.6 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)

68.7 polyfitd

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

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

68.8 regression_method_of_moments

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

68.9 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

68.10 theil2

Theil senn-estimator for two dimensions (glm)

68.11 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

68.12 total_least_squares

total least squares

68.13 weighted_median_regression

weighted median regression
c.f. Scholz, 1978

69 set-theory

69.1 issubset

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

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

70 mathematics

mathematical functions of various kind

70.1 shuffle_index

71 signal-processing

71.1 acf_bp

71.2 acf_decay_rate

71.3 acf_effective_sample_size

effective sample size from acf

71.4 acf_genton

autocorrelation function

71.5 acf_hp

71.6 acf_lp

71.7 acf_periodic_additive_noise

71.8 acf_periodic_windowed

71.9 acf_radial

71.10 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
```

71.11 acfar1_2

autocorrelation of the ar1 process

71.12 acfar2

impulse response of the ar2 process

71.13 acfar2_2

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

71.14 ar1_cutoff_frequency

71.15 ar1_effective_sample_size

effective sample size correction for autocorrelated series

71.16 ar1_mse_mu_single_sample

standard error of a single sample of an ar1 correlated process

71.17 ar1_mse_pop

variance of the population mean of a single realisation around zero

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

71.18 ar1_mse_range

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

71.19 ar1_spectrum

spectrum of the ar1 process

71.20 ar1_to_tikhonov

convert ar1 correlation to tikhonovs lambda

71.21 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

71.22 ar1_var_factor_

variance of an autocorrelated finite process

71.23 ar1_var_range2

variance of sub sample starting at the end of the series

from the finite length first order autocorrelated process

$$s2 = 1/m^2 \sum_i^m \sum_j^m \rho^{|i-j|}$$

71.24 ar1delay

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

71.25 ar1delay_old

autocorrelation of the residual

71.26 ar2_acf2c

71.27 ar2conv

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

71.28 ar2dof

effective samples size for the ar2 process

71.29 ar2param

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

71.30 asymwin

creates asymmetrical filter windows
filter will always have negative weights

71.31 autocorr2

71.32 autocorr_fft

autocorrelation function

71.33 bandpass

bandpass filter

71.34 bandpass1d

71.35 bandpass1d_fft

71.36 bandpass1d_implicit

71.37 bandpass2

bandpass filter

71.38 bandpass2d

71.39 bandpass2d_2

71.40 bandpass2d_fft

71.41 bandpass2d_implicit

71.42 bandpass2d_iso

71.43 bandpass_max

71.44 bandpass_max2

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

71.46 bin1d

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

71.47 bin2d

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

71.48 binormrnd

generate two correlated normally distributed vectors

71.49 coherence

71.50 conv1_man

convolutions with padding

71.51 conv2_man

convolution in 2d

71.52 conv2z

71.53 conv30

convolve with rectangular window of lenght n
circular boundaries

71.54 conv_

convolution of a with b

71.55 conv_centered

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

71.56 convz

71.57 cosexpdelay

71.58 csmooth

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

71.59 daniell_window

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

71.60 danielle_window

danielle fourier window

71.61 db2neper

convert decibel to neper

71.62 db2power

power ratio from db

71.63 derive_bandpass_normalization_and_zeros

71.64 derive_danielle_weight

71.65 derive_limit_0_acfar

71.66 detect_peak

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

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

71.68 doublesum_ij

double sum of r^i

71.69 effective_sample_size_to_ar1

convert effective sample size to ar1 correlation

71.70 `flt_hodges_lehman`

71.71 `filter1`

`filter` along one dimension

71.72 `filter2`

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

71.73 `filter_`

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

71.74 `filter_arg`

71.75 `filter_f0_to_rho`

71.76 `filter_r_to_f0`

71.77 `filter_rho_to_f0`

71.78 `filter_twosided`

71.79 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

71.80 filterp

71.81 filterp1

fir filter with some fancy extras

71.82 filterstd

71.83 firls_man

design finite impulse response filter by the least squares method

71.84 fit_spectral_density

71.85 flattopwin

the flat top window

71.86 frequency_response_boxcar

frequency response of a boxcar filter

71.87 freqz_boxcar

frequency response of a boxcar filter

71.88 gaussfilt1

filter data series with a gaussian window

71.89 hanchangewin

hanning window for change point detection

71.90 hanchangewin2

nanning window for chage point detection

71.91 hanwin

hanning filter window

71.92 hanwin_

hanning filter window

71.93 high_pass_1d_simple

71.94 highpass

high pass filter

71.95 highpass1d_implicit

71.96 highpass2d_fft

71.97 highpass2d_implicit

71.98 jackknife_block

71.99 kaiserwin

kaiser filter window

71.100 kalman

Kalman filter

71.101 lanczoswin

Lanczos window

71.102 last

lake tail, but for matrices

71.103 lowpass

low pass filter

71.104 lowpass1d_fft

71.105 lowpass1d_implicit

71.106 lowpass2

design low pass filter with cutoff-frequency f1

71.107 lowpass2d_2

71.108 `lowpass2d_anisotropic`

71.109 `lowpass2d_fft`

71.110 `lowpass2d_implicit`

71.111 `lowpass_iir`

`iir-low pass`

71.112 `lowpass_iir_symmetric`

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

71.113 `lowpassfilter2`

`low-pass filter of data`

71.114 `maxfilt1`

71.115 `meanfilt1`

`moving average filter with special treatment of the boundaries`

71.116 `medfilt1_man`

`moving median filter, supports columnwise operation`

71.117 medfilt1_man2

moving median filter with special treatment of boundaries

71.118 medfilt1_padded

median filter with padding

71.119 medfilt1_reduced

median filter with padding

71.120 mid_term_single_sample

variance of single sample, mid term

71.121 minfilt1

71.122 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 $\rho \rightarrow 1$

71.123 mysmooth

71.124 nanautocorr

autocorrelation with nan-values

71.125 `nanmedfilt1`

`medfilt1`, skipping nans

71.126 `neper2db`

convert neper to db

71.127 `peaks_man`

peaks of a periodogram

71.128 `periodogram`

71.129 `periodogram_annular`

71.130 `periodogram_bartlett`

TODO welsh sliding window

71.131 `periodogram_bootstrap`

71.132 `periodogram_confidence_interval`

71.133 `periodogram_median`

71.134 `periodogram_p_value`

71.135 `periodogram_qq`

71.136 `periodogram_quantiles`

71.137 `periodogram_std`

71.138 `periodogram_test`

71.139 `polyfilt1`

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

71.140 `qmedfilt1`

`medfilt1`, after fitting a quadratic polynomial

71.141 `randar1`

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

71.142 randar1_dual

draw random variables of two correlated ar1 processes

71.143 randar2

generate ar2 process

71.144 randarp

randomly generate the instance of an ar-p process

71.145 range_window

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

71.146 rectwin

rectangular window

71.147 recursive_sum

71.148 select_range

71.149 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

71.150 smooth2

smooth vectos of X

71.151 smooth_man

71.152 smooth_parametric

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

71.153 smooth_parametric2

parametrically smooth the curve

71.154 smooth_with_splines

71.155 smoothfft

filter with fast fourier transform

71.156 spectral_density_ar2

71.157 spectral_density_area

71.158 spectral_density_bandpass_continuous

71.159 spectral_density_bandpass_discrete

71.160 spectral_density_brownian_phase

71.161 spectral_density_flat

71.162 spectral_density_highpass

71.163 spectral_density_lorentzian

71.164 spectral_density_lowpass

71.165 spectral_density_radial

71.166 spectral_density_wperiodic

71.167 spectrogram

spectrogram

71.168 std_window

moving block standard deviation

71.169 sum_i_lag

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

71.170 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

71.171 sum_ii_

71.172 sum_ij

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

71.173 sum_ij_

71.174 sum_ij_partial_

71.175 sum_multivar

sum of matrix entries of bivariate ar1 process

71.176 test_acfar1

71.177 test_acfar1_2

71.178 test_acfar1_3

71.179 test_acfar1_4

71.180 test_acfar2

71.181 test_ar1_var_factor

71.182 test_ar1_var_factor_2

71.183 test_ar1_var_mu_single_sample

71.184 test_ar1_var_pop

71.185 test_ar1_var_pop_1

71.186 test_ar1delay

71.187 test_bivariate_covariance_term

71.188 test_convexity

71.189 test_lanczoswin

71.190 test_madcorr

71.191 test_randar1

71.192 test_randar1_multivariate

71.193 test_randar2

71.194 test_sum_ij

71.195 test_sum_multivar

71.196 test_trifilt1

71.197 test_wautocorr

71.198 test_wavelet_transform

71.199 test_wordfilt

71.200 test_xar1_mid_term

71.201 tikhonov_to_ar1

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

71.202 trapwin

trapezoidal filter window

71.203 trifilt1

filter with triangular window

71.204 trifilt2

71.205 triwin

triangular filter window

71.206 triwin2

triangular filter window

71.207 varar1

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

71.208 welch_spectrogram

welch spectrogram

71.209 wfilt

filter with window

71.210 winbandpass

filter with bandpass

71.211 window2d

71.212 window_make_odd

71.213 winfilt0

filter with window

71.214 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

71.215 wmeanfilt

mean filter with window

71.216 wmedfilt

median filter with window

71.217 wordfilt

weighted order filter

71.218 wordfilt_edgeworth

weighed order filter

71.219 wrapphase

71.220 xar1

71.221 xcorr_man

cross correlation of two sampled ar1 processes

72 sorting

72.1 sort2

sort two numbers

72.2 sort2d

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

73 special-functions

73.1 bessell_sphere

spherical Bessel function of the first kind

73.2 digamma_man

73.3 exp10

73.4 hankel_sphere

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

73.5 hermite

probabilistic's hermite polynomial by recurrence relation

input :
n : order
x : value

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

73.6 legendre_man

legendre polynomials

73.7 neumann_sphere

spherical Neumann function
Bessel function of the second kind

74 statistics

74.1 atan_s2

stadard deviation of the arcus tangens by means of taylor expansion

74.2 beta_kurt

74.3 beta_mean

74.4 beta_mode_to_parameter

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

74.5 `beta_skew`

74.6 `beta_std`

74.7 `chi2_kurt`

74.8 `chi2_mean`

74.9 `chi2_skew`

74.10 `chi2_std`

74.11 `coefficient_of_determination`

74.12 `conditional_expectation_normal`

74.13 `correlation_confidence_pearson`

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

75 statistics/distributions

75.1 PDF

class for quasi-distributions from a set of sampling points

75.2 binorm_separation_coefficient

separation coefficient of a bimodal normal distribution

75.3 binormcdf

bio-modal gaussian distribution

75.4 binormfit

fit sum of to normal distribution to a histogram

75.5 binormpdf

75.6 edgeworth_cdf

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

75.7 edgeworth_pdf

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

75.8 gam_moment2param

75.9 logn_mean

75.10 logn_mode

75.11 logn_moment2param

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

75.12 logn_param2moment

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

75.13 logn_std

75.14 lognpdf_

log normal distribution called by modes rather than parameters

75.15 pdfsample

pdf from sample distribution
Note: better use kernel density estimates

75.16 t2cdf

Hotelling's T-squared cumulative distribution

75.17 t2inv

inverse of Hotelling's T-squared cumulative distribution

76 statistics

76.1 example_standard_error_of_sample_quantiles

76.2 f_mean

76.3 f_std

76.4 f_var_finite

reduction of variance when sampling from a finite population
without replacement

76.5 gam_mean

76.6 gam_std

76.7 gamma_mode_to_parameter

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

76.8 gaussfit3

76.9 gaussfit_quantile

76.10 geoserr

76.11 geostd

76.12 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

76.13 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

77 statistics/information-theory

77.1 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

77.2 bayesian_information_criterion

bayesian information criterion

78 statistics

78.1 kurtncdf

78.2 kurtnpdf

78.3 kurtosis_bias_corrected

bias corrected kurtosis

78.4 limit

limit a by lower and upper bound

78.5 logfactorial

approximate log of the factorial

78.6 loglogpdf

78.7 lognfit_quantile

78.8 logskewcdf

78.9 logskewpdf

79 statistics/logu

79.1 lambertw_numeric

lambert-w function

79.2 logtrialtcdf

pdf of a logarithmic triangular distribution

79.3 logtrialtinv

inverse of the logarithmic triangular distribution

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

79.4 logtrialtmean

mean of the logarithmic triangular distribution

79.5 logtrialtpdf

density of the logarithmic triangular distribution

79.6 logtrialtrnd

79.7 logtricdf

cumulative distribution of the logarithmic triangular distribution

79.8 logtriinv

invere of the logarithmic triangular distribution

79.9 logtrimean

mean of the logarithmic triangular distribution

79.10 logtripdf

probability density of the logarithmic triangular distribution

79.11 logtirnd

79.12 logucdf

probability density of the logarithmic uniform distribution

79.13 logucm

central moments of the log-uniform distribution

79.14 loguinv

inverse of the log-uniform distribution

79.15 logumean

mean of the log-uniform distribution

79.16 logupdf

pdf of the log uniform distribution

79.17 logurnd

random numbers following a log-uniform distribution

79.18 loguvar

variance of the log-uniform distribution

79.19 medlogu

median of the log-uniform distribution

79.20 test_logurnd

79.21 tricdf

cumulative distribution of the log-triangular distribution

79.22 triinv

inverse of the triangular distribution

79.23 trimedian

median of the triangular distribution

79.24 tripdf

probability density of the triangular distribution

79.25 trirnd

random numbers of the triangular distribution

80 statistics

80.1 max_exprnd

80.2 maxnnormals

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

80.3 mean_generalized_gampdf

80.4 midrange

mid range of columns of X

80.5 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

80.6 mode_man

81 statistics/moment-statistics

81.1 autocorr_man3

autocorrelation of the columns of X

81.2 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

81.3 autocorr_man5

autocorrelation of the columns of X

81.4 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

81.5 comoment

non-central higher order moments of the multivariate normal distribution

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

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

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

81.6 corr_man

correlation of two vectors

81.7 cov_man

covariance matrix of two vectors

81.8 dof

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

81.9 edgeworth_quantile

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

81.10 effective_sample_size

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

81.11 f_correlation

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

81.12 `f_finite`

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

81.13 `lmean`

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

81.14 `lmoment`

l-moment of vector `x`

81.15 `maskmean`

mean of the masked values of `X`

81.16 `masknanmean`

81.17 `mean1`

mean of `x`

81.18 `mean_man`

mean and standard error of `X`

81.19 `mse`

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

81.20 nanautocorr_man1

autocorrelation of a vector with nan-values

81.21 nanautocorr_man2

autocorrelation of a vector with nan-values

81.22 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

81.23 nancorr

(co)-correlation matrix when samples a NaN

81.24 nancumsum

cumulative sum, setting nan values to zero

81.25 nanlmean

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

81.26 nanr2

coefficient of determination when samples are invalid

81.27 nanrms

root mean square value when sample contains nan-values

81.28 nanrmse

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

81.29 nanserr

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

81.30 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)

81.31 nanwstd

weighed standard deviation

81.32 nanwvar

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

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

81.33 nanxcorr

81.34 pearson

pearson correlation coefficient

81.35 pearson_to_kendall

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

81.36 pool_samples

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

81.37 qmean

trimmed mean

81.38 range_mean

81.39 rmse_

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

81.40 serr

standard error of the mean of a set of uncorrelated samples

81.41 serr1

81.42 test_kskew

81.43 test_qstd_kskew_optimal_p

81.44 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

81.45 wcorr

correlation of two vectors when samples are weighted

81.46 wcov

covariance of two vectors when samples are weighted

81.47 wdof

effective degrees of freedom for weighted samples

81.48 wkurt

kurtosis with weighted samples

81.49 wmean

weighted mean

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

varargin can be dim

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

81.50 wrms

weighted root mean square

81.51 wserr

weighted root mean square error

81.52 wskew

skewness of a weighted set of samples

81.53 wstd

weighed standard deviation

81.54 wvar

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

variance of the weighted sample mean of samples with same mean (but not necessarily same variance)

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

s2_mu : error of mean, s2_mu : sd of prediction

82 statistics

82.1 nangeomean

82.2 nangeostd

geometric standard deviation ignoring nan-values

83 statistics/nonparametric-statistics

83.1 kernel1d

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

83.2 kernel2d

kernel density estimate in two dimensions

84 statistics

84.1 normmmoment

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

84.2 normpdf2

pdf of the bivariate normal distribution

85 statistics/order-statistics

85.1 hodges_lehmann_location

hodges lehman location estimator

Asymptotic rms efficiency of location estimate:

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

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

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

85.2 kendall

kendall correlation coefficient

85.3 kendall_to_pearson

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

c.f. Kruska, 1985

85.4 mad2sd

transform median absolute deviation to standard deviation
for normal distributed values

85.5 madcorr

proxy correlation by median absolute deviation

85.6 median2_holder

85.7 median_ci

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

85.8 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`

85.9 mediani

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

85.10 nanmadcorr

proxy correlation by median absolute deviation

85.11 nanwmedian

weighted median, skips nan-values

85.12 nanwquantile

weighted quantile, skips nan values

85.13 oja_median

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

oja 1983, for extension to multivariate function, see chaudhri

85.14 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

85.15 qmoments

moments estimated from quantiles

85.16 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

85.17 qskewq

skewness estimated by quantiles

85.18 qstdq

proxy standard deviation determined by quantiles

85.19 quantile1_optimisation

85.20 quantile2_breckling

quantile regression

85.21 quantile2_chaudhuri

quantile regression

85.22 quantile2_projected

quantile in two dimensions

85.23 quantile2_projected2

spatial quantile for chosen direction

85.24 quantile_envelope

85.25 quantile_regression_simple

simple quantile regression

85.26 ranking

ranking for spearman statistics

85.27 spatial_median

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

85.28 spatial_quantile

spatial quantile

85.29 spatial_quantile2

spatial quantile

85.30 spatial_quantile3

spatial quantile

85.31 spatial_rank

unsigned rank

85.32 spatial_sign

spatial sign

85.33 spatial_signed_rank

signed rank

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

85.34 spearman

spearman's product moment coefficient

85.35 spearman_rank

85.36 spearman_to_pearson

conversion of spearman rank to person product moment correlation coefficient

85.37 wmedian

weighted median

85.38 wquantile

weighted quantile

86 statistics

86.1 qstd

86.2 quantile_extrap

86.3 quantile_sin

87 statistics/random-number-generation

87.1 laplacernd

random number of laplace distribution

87.2 randc

correlate to correlated standard normally distributed vectors

87.3 skewness2param

87.4 skewpdf_central_moments

87.5 skewrnd

random numbers of the skew normal distribution

87.6 skewrnd2

random numbers of the skew normal distribution

88 statistics

88.1 range

mid range

88.2 resample_with_replacement

89 statistics/resampling-statistics/@Jackknife

89.1 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 inferior to the d1
 jackknife

89.2 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

89.3 matrix1_STATIC

matrix of estimation for leaving out two samples at a time

89.4 matrix2

matrix of estimations for jackknife with two samples left out

90 statistics/resampling-statistics

90.1 block_jackknife

90.2 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

90.3 moving_block_jackknife

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

90.4 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

90.5 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

91 statistics

91.1 scale_quantile_sd

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

91.2 sd_sample_quantiles

91.3 skew_generalized_normal_fit

91.4 skew_generalized_normpdf

91.5 skewcdf

91.6 skewparam_to_central_moments

91.7 skewpdf

skew-normal distribution
c.f. Azzalini 1985

91.8 spatialrnd

91.9 test_mean_generalized_gampdf

91.10 test_skew_generalized_normpdf

91.11 trimmed_mean

trimmed mean

91.12 ttest2_man

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

91.13 ttest_man

two-sample t-test
unequal sample size
equal variance

91.14 ttest_paired

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

91.15 wgeomean

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

91.16 wgeovar

variance of the weighted geometric mean

91.17 wharmean

weighted harmonic mean

91.18 wharstd

91.19 wharvar

91.20 wnormpdf

92 mathematics

mathematical functions of various kind

92.1 ternary_diagram

93 test/master

93.1 dat_test_lanczos_3d_k_20_n_40

93.2 poisson2d_blk

93.3 qr_implicit_givens_2

93.4 spectral_derivative_2d

93.5 test_2d_eigensolver_hydrogen

93.6 test_2d_refine

93.7 test_3d_eigensolver_hydrogen

93.8 test_FEM

93.9 test_Mesh_3d

93.10 test_arnoldi

93.11 test_arpackc

93.12 test_assemble

93.13 test_assembly_performance

93.14 test_bc_one_sided

93.15 test_compare_solvers

93.16 test_complete

93.17 test_convergence

93.18 test_convergence_b

93.19 test_df_2d

93.20 test_eig_algs

93.21 test_eig_inverse

93.22 test_eigs_lanczos

93.23 test_eigs_lanczos_1

93.24 test_eigs_lanczos_2

93.25 test_eigs_lanczos_performance

93.26 test_fdm

93.27 test_fdm_d_vargrid

93.28 test_fdm_spectral

93.29 test_fem

93.30 test_fem_1d

93.31 test_fem_1d_higher_order

93.32 test_fem_2d_adaptive

93.33 test_fem_2d_higher_order

93.34 test_fem_3d_higher_order

93.35 test_fem_3d_refine

93.36 test_fem_b

93.37 test_fem_derivative

93.38 test_fem_quadrature

93.39 test_final

93.40 test_fix_substitution

93.41 test_forward

93.42 test_get_sparse_arrays

93.43 test_harmonic_oscillator

93.44 test_high_order_fdm_periodic_bc

93.45 test_hydrogen_wf

93.46 test_ichol

93.47 test_interpolation

93.48 test_inverse_problem

93.49 test_it_vs_exact

93.50 test_jama

93.51 test_jd

93.52 test_jdqz

93.53 test_lanczos_2

93.54 test_lanczos_biorthogonal

93.55 test_laplacian

93.56 test_laplacian_non_uniform

93.57 test_laplacian_simple

93.58 test_mesh_2d_uniform

93.59 test_mesh_2d_uniform_2

93.60 test_mesh_circle

93.61 test_mesh_generation

93.62 test_mesh_interpolate

93.63 test_mg

93.64 test_minres_recycle

93.65 test_multigrid

93.66 test_nc

93.67 test_nonuniform_symmetric

93.68 test_pde

93.69 test_permutation

93.70 test_poison_fem

93.71 test_polar

93.72 test_potential

93.73 test_powers

93.74 test_precondition

93.75 test_project_rectangle

93.76 test_qr

93.77 test_quantum_well

93.78 test_radial_adaptive

93.79 test_radial_confinement

93.80 test_radial_fixes

93.81 test_refine_2d

93.82 test_refine_2d_b

93.83 test_refine_3d

93.84 test_refine_structural

93.85 test_regularisation

93.86 test_round_off

93.87 test_schrödinger_potentials

93.88 test_uniform_mesh

93.89 test_vargrid

94 test

94.1 test_bandpass2d

94.2 test_gaussfit3

94.3 test_geoserr

94.4 test_hexagonal_pattern

94.5 test_lognfit_quantile

94.6 test_lowpass1d_implicit

94.7 test_lowpass2d_anisotropic

94.8 test_lowpass2d_fft

94.9 test_max_normal

94.10 test_mtimes3x3

95 wavelet

95.1 continuous_wavelet_transform

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

95.2 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

95.3 example_wavelets

95.4 phasewrap

wrap the phase to $\pm \pi$

95.5 test_cwt_man

95.6 test_phasewrap

95.7 test_wavelet

95.8 test_wavelet2

95.9 test_wavelet_analysis

95.10 test_wavelet_reconstruct

95.11 test_wtc

95.12 wavelet

wavelet windows

95.13 wavelet_reconstruct

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

95.14 wavelet_transform

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