

Manual for Package: mathematics

Revision 16M

Karl Kästner

September 2, 2021

Contents

1	mathematics	35
1.1	acf_decay_rate	35
2	calendar	35
2.1	days_per_month	35
2.2	isnight	35
3	mathematics	35
3.1	cast_byte_to_integer	35
4	complex-analysis	35
4.1	complex_exp_product_im_im	35
4.2	complex_exp_product_im_re	36
4.3	complex_exp_product_re_im	36
4.4	complex_exp_product_re_re	36
4.5	croots	37
4.6	root_complex	37
4.7	test_imroots	37
5	derivation	37
5.1	derive_acfar1	37
5.2	derive_ar2param	38
5.3	derive_arc_length	38
5.4	derive_fourier_power	38
5.5	derive_fourier_power_exp	38
5.6	derive_laplacian_curvilinear	38
5.7	derive_laplacian_fourier_pieewise_linear	38
5.8	derive_logtripdf	38
5.9	derive_smooth1d_parametric	38

6	derivation/master	38
6.1	derive_bc_one_sided	38
6.2	derive_convergence	39
6.3	derive_error_fdm	39
6.4	derive_fdm_poly	39
6.5	derive_fdm_power	39
6.6	derive_fdm_taylor	39
6.7	derive_fdm_vargrid	39
6.8	derive_fem_2d_mass	39
6.9	derive_fem_error_2d	39
6.10	derive_fem_error_3d	39
6.11	derive_fem_sym_2d	39
6.12	derive_grid_constants	40
6.13	derive_interpolation	40
6.14	derive_laplacian	40
6.15	derive_limit	40
6.16	derive_nc_1d	40
6.17	derive_nc_1d_	40
6.18	derive_nc_2d	40
6.19	derive_nonuniform_symmetric	40
6.20	derive_richardson	40
6.21	derive_sum	40
6.22	nn	41
6.23	test_derive	41
6.24	test_derive_fdm_poly	41
6.25	test_filter	41
6.26	test_vargrid	41
7	derivation	41
7.1	simplify_atan	41
8	mathematics	41
8.1	diffusion_matrix_2d_anisotropic	41
8.2	diffusion_matrix_2d_anisotropic2	41
8.3	entropy	42
8.4	euler	42
8.5	exp10	42
8.6	filter_twosided	42
9	finance	42
9.1	derive_skewrnd_walsh_paramter	42
9.2	gbm_cdf	42
9.3	gbm_fit	42
9.4	gbm_fit_old	42

9.5	gbm_inv	42
9.6	gbm_mean	43
9.7	gbm_median	43
9.8	gbm_pdf	43
9.9	gbm_simulate	43
9.10	gbm_skewness	43
9.11	gbm_std	43
9.12	gbm_transform_time_step	43
9.13	put_price_black_scholes	43
9.14	skewgbm_simulate	43
9.15	skewrnd_walsh	43
10	finance/test	44
10.1	test_gbm	44
10.2	test_gbm_pdf	44
10.3	test_skewrnd_walsh	44
11	fourier/@STFT	44
11.1	STFT	44
11.2	itransform	44
11.3	stft_	44
11.4	stftmat	44
11.5	transform	45
12	fourier	45
12.1	amplitude_from_peak	45
12.2	dftmtx_man	45
12.3	example_fourier_window	45
12.4	fft_derivative	46
12.5	fft_man	46
12.6	fftsmooth	46
12.7	fix_fourier	47
12.8	fourier_axis	47
12.9	fourier_cesaro_correction	47
12.10	fourier_coefficient_piecewise_linear	47
12.11	fourier_coefficient_piecewise_linear_1	48
12.12	fourier_coefficient_ramp3	48
12.13	fourier_coefficient_ramp_pulse	48
12.14	fourier_coefficient_ramp_step	48
12.15	fourier_coefficient_square_pulse	48
12.16	fourier_cubic_interaction_coefficients	48
12.17	fourier_derivative	49
12.18	fourier_expand	49
12.19	fourier_fit	49

12.20	fourier_interpolate	49
12.21	fourier_matrix	49
12.22	fourier_matrix2	49
12.23	fourier_matrix3	49
12.24	fourier_matrix_exp	50
12.25	fourier_multiplicative_interaction_coefficients	50
12.26	fourier_power	50
12.27	fourier_power_exp	50
12.28	fourier_predict	50
12.29	fourier_quadratic_interaction_coefficients	51
12.30	fourier_range	51
12.31	fourier_regress	51
12.32	fourier_resampled_fit	51
12.33	fourier_resampled_predict	51
12.34	fourier_signed_square	51
12.35	fourier_transform	52
12.36	hyperbolic_fourier_box	52
12.37	idftmtx_man	52
12.38	laplace_2d_pwlinear	52
12.39	nanfft	53
12.40	peaks	53
12.41	roots_fourier	53
12.42	spectral_density	53
12.43	test_complex_exp_product	53
12.44	test_fourier_filter	54
12.45	test_idftmtx	54
13 mathematics		54
13.1	fourier_derivative_matrix_1d	54
13.2	fourier_derivative_matrix_2d	54
13.3	fractional_fourier_transform	54
13.4	gaussfit_quantile	54
14 geometry/@Geometry		54
14.1	Geometry	54
14.2	arclength	55
14.3	arclength_old	55
14.4	arclength_old2	55
14.5	base_point	55
14.6	base_point_limited	55
14.7	centroid	55
14.8	cosa_min_max	55
14.9	cross2	56
14.10	curvature	56

14.11	ddot	56
14.12	distance	56
14.13	distance2	56
14.14	dot	56
14.15	edge_length	56
14.16	enclosed_angle	56
14.17	enclosing_triangle	57
14.18	hexagon	57
14.19	inPolygon	57
14.20	inTetra	57
14.21	inTetra2	57
14.22	inTriangle	57
14.23	intersect	57
14.24	lineintersect	57
14.25	lineintersect1	58
14.26	minimum_distance_lines	58
14.27	mittenpunkt	58
14.28	nagelpoint	58
14.29	onLine	58
14.30	orthocentre	58
14.31	plumb_line	58
14.32	poly_area	58
14.33	poly_edges	59
14.34	poly_set	59
14.35	poly_width	59
14.36	polyxpoly	59
14.37	project_to_curve	59
14.38	quad_isconvex	59
14.39	random_disk	59
14.40	random_simplex	60
14.41	sphere_volume	60
14.42	tetra_volume	60
14.43	tobarycentric	60
14.44	tobarycentric1	60
14.45	tobarycentric2	60
14.46	tobarycentric3	60
14.47	tri_angle	60
14.48	tri_area	61
14.49	tri_centroid	61
14.50	tri_distance_opposit_midpoint	61
14.51	tri_edge_length	61
14.52	tri_edge_midpoint	61
14.53	tri_excircle	61
14.54	tri_height	61

14.55	tri_incircle	61
14.56	tri_isacute	62
14.57	tri_isobtuse	62
14.58	tri_semiperimeter	62
14.59	tri_side_length	62
15	geometry	62
15.1	Polygon	62
15.2	bounding_box	62
15.3	curvature_1d	63
15.4	cvt	63
15.5	deg_to_frac	63
15.6	ellipse	63
15.7	ellipseX	63
15.8	ellipseY	63
15.9	first_intersect	63
15.10	golden_ratio	63
15.11	hypot3	64
15.12	meanangle	64
15.13	meanangle2	64
15.14	meanangle3	64
15.15	meanangle4	64
15.16	medianangle	64
15.17	medianangle2	64
15.18	pilim	65
15.19	streamline_radius_of_curvature	65
16	mathematics	65
16.1	hexagonal_pattern	65
17	histogram/@Histogram	65
17.1	2x	65
17.2	Histogram	65
17.3	bimodes	65
17.4	cdf	65
17.5	cdfS	65
17.6	chi2test	66
17.7	cmoment	66
17.8	cmomentS	66
17.9	entropy	66
17.10	entropyS	66
17.11	export_csv	66
17.12	iquantile	66
17.13	kstest	66

17.14	kurtosis	66
17.15	kurtosisS	66
17.16	mean	67
17.17	meanS	67
17.18	median	67
17.19	medianS	67
17.20	mode	67
17.21	modeS	67
17.22	moment	67
17.23	momentS	67
17.24	pdf	67
17.25	quantile	67
17.26	quantileS	68
17.27	resample	68
17.28	setup	68
17.29	skewness	68
17.30	skewnessS	68
17.31	stairs	68
17.32	stairsS	68
17.33	std	68
17.34	stdS	68
17.35	var	68
17.36	varS	69
18	histogram	69
18.1	hist_man	69
18.2	histadapt	69
18.3	histconst	69
18.4	pdf_poly	69
18.5	plotcdf	69
18.6	test_histogram	69
19	mathematics	69
19.1	implicit_advection	69
19.2	imrotmat	70
19.3	interp1_circular	70
19.4	kernel2matrix	70
20	linear-algebra	70
20.1	averaging_matrix_2	70
20.2	colnorm	70
20.3	condest_	70
21	linear-algebra/coordinate-transformation	70

21.1	barycentric2cartesian	70
21.2	barycentric2cartesian3	70
21.3	cartesian2barycentric	71
21.4	cartesian_to_unit_triangle_basis	71
21.5	ellipsoid2geoid	71
21.6	example_approximate_utm_conversion	71
21.7	latlon2utm	71
21.8	latlon2utm_simple	71
21.9	lowrance_mercator_to_wgs84	71
21.10	nmea2utm	71
21.11	sn2xy	72
21.12	unit_triangle_to_cartesian	72
21.13	utm2latlon	72
21.14	xy2nt	72
21.15	xy2sn	72
21.16	xy2sn_java	72
21.17	xy2sn_old	72
22	linear-algebra	73
22.1	det2x2	73
22.2	det3x3	73
22.3	det4x4	73
22.4	diag2x2	73
22.5	down	73
22.6	eig2x2	73
23	linear-algebra/eigenvalue	73
23.1	eig_bisection	73
23.2	eig_inverse	73
23.3	eig_inverse_iteration	74
23.4	eig_power_iteration	74
24	linear-algebra/eigenvalue/jacobi-davidson	74
24.1	afun_jdm	74
24.2	davidson	74
24.3	jacobi_davidson	74
24.4	jacobi_davidson_qr	74
24.5	jacobi_davidson_qz	74
24.6	jacobi_davidson_simple	74
24.7	jdqr	75
24.8	jdqr_sleijpen	78
24.9	jdqr_vorst	81
24.10	jdqz	84
24.11	mfunc_jdm	89

24.12	mgs	89
24.13	minres_	89
24.14	mv_jacobi_davidson	89
25	linear-algebra	89
25.1	first	89
25.2	gershgorin_circle	89
25.3	hausdorff	90
25.4	ieig2x2	90
25.5	inv2x2	90
25.6	inv3x3	90
25.7	inv4x4	90
26	linear-algebra/lanczos	90
26.1	arnoldi	90
26.2	arnoldi_new	90
26.3	eigs_lanczos_man	91
26.4	lanczos	91
26.5	lanczos_	91
26.6	lanczos_biorthogonal	91
26.7	lanczos_biorthogonal_improved	91
26.8	lanczos_ghep	91
26.9	mv_lanczos	91
26.10	reorthogonalise	91
26.11	test_lanczos	91
27	linear-algebra	92
27.1	left	92
28	linear-algebra/linear-systems	92
28.1	gmres_man	92
28.2	minres_recycle	92
29	linear-algebra	92
29.1	lpmean	92
29.2	lpnorm	92
29.3	matvec3	92
29.4	max2d	92
29.5	mid	93
29.6	mpoweri	93
29.7	mtimes2x2	93
29.8	mtimes3x3	93
29.9	nannorm	93
29.10	nanshift	93

29.11	nl	93
29.12	normalise	94
29.13	normalize1	94
29.14	normrows	94
29.15	orth2	94
29.16	orth_man	94
29.17	orthogonalise	94
29.18	paddext	94
29.19	paddval1	95
29.20	paddval2	95
30	linear-algebra/polynomial	95
30.1	chebychev	95
30.2	piecewise_polynomial	95
30.3	roots1	95
30.4	roots2	95
30.5	roots2poly	95
30.6	roots3	95
30.7	roots4	96
30.8	roots_piecewise_linear	96
30.9	test_roots4	96
30.10	vanderi_1d	96
31	linear-algebra	96
31.1	randrot	96
31.2	right	96
31.3	rot2	96
31.4	rot2dir	96
31.5	rot3	97
31.6	rotR	97
31.7	rownorm	97
31.8	simmilarity_matrix	97
31.9	spnorm	97
31.10	spzeros	97
31.11	test_roots3	97
31.12	transform_minmax	97
31.13	transpose3	97
31.14	transposeall	98
31.15	up	98
32	logic	98
32.1	bitor_man	98
33	mathematics	98

33.1	lowpass2d_anisotropic	98
34	master/plot	98
34.1	attach_boundary_value	98
34.2	cartesian_polar	98
34.3	img_vargrid	98
34.4	plot_basis_functions	99
34.5	plot_convergence	99
34.6	plot_dof	99
34.7	plot_eigenbar	99
34.8	plot_error_estimation	99
34.9	plot_error_estimation_2	99
34.10	plot_error_fem	99
34.11	plot_fdm_kernel	99
34.12	plot_fdm_vs_fem	99
34.13	plot_fem_accuracy	99
34.14	plot_function_and_grid	100
34.15	plot_hat	100
34.16	plot_hydrogen_wf	100
34.17	plot_mesh	100
34.18	plot_mesh_2	100
34.19	plot_refine	100
34.20	plot_refine_3d	100
34.21	plot_runtime	100
34.22	plot_spectrum	100
34.23	plot_wavefunction	100
35	master/ported	101
35.1	assemble_2d_phi_phi	101
35.2	assemble_3d_dphi_dphi	101
35.3	assemble_3d_phi_phi	101
35.4	dV_2d_	101
35.5	derivative_2d	101
35.6	derivative_3d	101
35.7	element_neighbour_2d	101
35.8	prefetch_2d_	101
35.9	promote_2d_3_10	101
35.10	promote_2d_3_15	102
35.11	promote_2d_3_21	102
35.12	promote_2d_3_6	102
35.13	promote_3d_4_10	102
35.14	promote_3d_4_20	102
35.15	promote_3d_4_35	102
35.16	vander_2d	102

35.17	vander_3d	102
36	number-theory	102
36.1	ceiln	102
36.2	digitsb	103
36.3	floorn	103
36.4	iseven	103
36.5	multichoosek	103
36.6	nchoosek_man	103
36.7	pythagorean_triple	103
36.8	roundn	103
37	numerical-methods/differentiation	104
37.1	derivative1	104
37.2	derivative2	104
38	numerical-methods/finite-difference	104
38.1	cdiff	104
38.2	cdiffb	104
38.3	central_difference	104
38.4	cmean	104
38.5	cmean2	104
38.6	derivative_matrix_1_1d	105
38.7	derivative_matrix_2_1d	105
38.8	derivative_matrix_2d	105
38.9	derivative_matrix_curvilinear	105
38.10	derivative_matrix_curvilinear_2	105
38.11	difference_kernel	105
38.12	distmat	105
38.13	downwind_difference	106
38.14	gradpde2d	106
38.15	laplacian	106
38.16	laplacian_fdm	106
38.17	lrmean	106
39	numerical-methods/finite-difference/master	106
39.1	fdm_adaptive_grid	106
39.2	fdm_adaptive_refinement_old	106
39.3	fdm_assemble_d1_2d	106
39.4	fdm_assemble_d2_2d	107
39.5	fdm_confinement	107
39.6	fdm_d_vargrid	107
39.7	fdm_h_unstructured	107
39.8	fdm_hydrogen_vargrid	107

39.9	fdm_mark_unstructured_2d	107
39.10	fdm_plot	107
39.11	fdm_plot_series	107
39.12	fdm_refine_2d	107
39.13	fdm_refine_3d	107
39.14	fdm_refine_unstructured_2d	108
39.15	fdm_schroedinger_2d	108
39.16	fdm_schroedinger_3d	108
39.17	relocate	108
40	numerical-methods/finite-difference	108
40.1	mid	108
40.2	pwmid	108
40.3	ratio	108
40.4	steplength	108
40.5	swapoddeven	109
40.6	test_derivative_matrix_2d	109
40.7	test_derivative_matrix_curvilinear	109
40.8	test_difference_kernel	109
40.9	upwind_difference	109
41	numerical-methods/finite-element	109
41.1	Mesh_2d.java	109
41.2	Tree_2d.java	109
41.3	assemble_1d_dphi_dphi	109
41.4	assemble_1d_phi_phi	109
41.5	assemble_2d_dphi_dphi_java	110
41.6	assemble_2d_phi_phi_java	110
41.7	assemble_3d_dphi_dphi_java	110
41.8	assemble_3d_phi_phi_java	110
41.9	boundary_1d	110
41.10	boundary_2d	110
41.11	boundary_3d	110
41.12	check_area_2d	110
41.13	circmesh	110
41.14	cropradius	110
41.15	display_2d	111
41.16	display_3d	111
41.17	distort	111
41.18	err_2d	111
41.19	estimate_err_2d_3	111
41.20	example_1d	111
41.21	example_2d	111
41.22	explode	111

41.23	fem_2d	111
41.24	fem_2d_heuristic_mesh	111
41.25	fem_get_2d_radial	112
41.26	fem_interpolation	112
41.27	fem_plot_1d	112
41.28	fem_plot_1d_series	112
41.29	fem_plot_2d	112
41.30	fem_plot_2d_series	112
41.31	fem_plot_3d	112
41.32	fem_plot_3d_series	112
41.33	fem_plot_confine_series	112
41.34	fem_radial	113
41.35	flip_2d	113
41.36	get_mesh_arrays	113
41.37	hashkey	113
42	numerical-methods/finite-element/int	113
42.1	int_1d_gauss	113
42.2	int_1d_gauss_1	113
42.3	int_1d_gauss_2	113
42.4	int_1d_gauss_3	113
42.5	int_1d_gauss_4	113
42.6	int_1d_gauss_5	114
42.7	int_1d_gauss_6	114
42.8	int_1d_gauss_lobatto	114
42.9	int_1d_gauss_n	114
42.10	int_1d_nc_2	114
42.11	int_1d_nc_3	114
42.12	int_1d_nc_4	114
42.13	int_1d_nc_5	114
42.14	int_1d_nc_6	114
42.15	int_1d_nc_7	114
42.16	int_1d_nc_7_hardy	115
42.17	int_2d_gauss_1	115
42.18	int_2d_gauss_12	115
42.19	int_2d_gauss_13	115
42.20	int_2d_gauss_16	115
42.21	int_2d_gauss_19	115
42.22	int_2d_gauss_25	115
42.23	int_2d_gauss_3	115
42.24	int_2d_gauss_33	115
42.25	int_2d_gauss_4	115
42.26	int_2d_gauss_6	116
42.27	int_2d_gauss_7	116

42.28	int_2d_gauss_9	116
42.29	int_2d_nc_10	116
42.30	int_2d_nc_15	116
42.31	int_2d_nc_21	116
42.32	int_2d_nc_3	116
42.33	int_2d_nc_6	116
42.34	int_3d_gauss_1	116
42.35	int_3d_gauss_11	116
42.36	int_3d_gauss_14	117
42.37	int_3d_gauss_15	117
42.38	int_3d_gauss_24	117
42.39	int_3d_gauss_4	117
42.40	int_3d_gauss_45	117
42.41	int_3d_gauss_5	117
42.42	int_3d_nc_11	117
42.43	int_3d_nc_4	117
42.44	int_3d_nc_6	117
42.45	int_3d_nc_8	117
43 numerical-methods/finite-element		118
43.1	interpolation_matrix	118
43.2	mark	118
43.3	mark_1d	118
43.4	mesh_1d_uniform	118
43.5	mesh_3d_uniform	118
43.6	mesh_interpolate	118
43.7	neighbour_1d	118
43.8	old	118
43.9	pdeeig_1d	118
43.10	pdeeig_2d	119
43.11	pdeeig_3d	119
43.12	polynomial_derivative_1d	119
43.13	potential_const	119
43.14	potential_coulomb	119
43.15	potential_harmonic_oscillator	119
43.16	project_circle	119
43.17	project_rectangle	119
43.18	promote_1d_2_3	119
43.19	promote_1d_2_4	119
43.20	promote_1d_2_5	120
43.21	promote_1d_2_6	120
43.22	quadrilaterate	120
43.23	recalculate_regularity_2d	120
43.24	refine_1d	120

43.25	refine_2d_21	120
43.26	refine_2d_structural	120
43.27	regularity_1d	120
43.28	regularity_2d	120
43.29	regularity_3d	121
43.30	relocate_2d	121
43.31	test_circmesh	121
43.32	test_hermite	121
43.33	tri_assign_points	121
43.34	triangulation_uniform	121
43.35	vander_1d	121
43.36	vanderd_1d	121
43.37	vanderi_1d	121
44	numerical-methods/finite-volume/@Advection	122
44.1	Advection	122
44.2	dot_advection	122
45	numerical-methods/finite-volume/@Burgers	122
45.1	burgers_split	122
45.2	dot_burgers_fdm	122
45.3	dot_burgers_fft	122
46	numerical-methods/finite-volume/@Finite_Volume	122
46.1	Finite_Volume	122
46.2	apply_bc	123
46.3	solve	123
46.4	step_split_strang	123
46.5	step_unsplit	123
47	numerical-methods/finite-volume/@Flux_Limiter	123
47.1	Flux_Limiter	123
47.2	beam_warming	123
47.3	fromm	124
47.4	lax_wendroff	124
47.5	minmod	124
47.6	monotized_central	124
47.7	muscl	124
47.8	superbee	124
47.9	upwind	124
47.10	vanLeer	125
48	numerical-methods/finite-volume/@KDV	125
48.1	dot_kdv_fdm	125

48.2	dot_kdv_fft	125
48.3	kdv_split	125
49	numerical-methods/finite-volume/@Reconstruct_Average_Evolve	125
49.1	Reconstruct_Average_Evolve	125
49.2	advect_highres	126
49.3	advect_lowress	126
50	numerical-methods/finite-volume	126
50.1	Godunov	126
50.2	Lax_Friedrich	126
50.3	Measure	126
50.4	Roe	126
50.5	fv_swe	127
50.6	staggered_euler	127
50.7	staggered_grid	127
51	numerical-methods	127
51.1	grid2quad	127
52	numerical-methods/integration	127
52.1	cumintL	127
52.2	cumintR	127
52.3	int_trapezoidal	127
53	numerical-methods/interpolation/@Kriging	128
53.1	Kriging	128
53.2	estimate_semivariance	128
53.3	interpolate_	128
54	numerical-methods/interpolation/@RegularizedInterpolator	128
54.1	RegularizedInterpolator1	128
54.2	init	128
55	numerical-methods/interpolation/@RegularizedInterpolator	129
55.1	RegularizedInterpolator2	129
55.2	init	129
56	numerical-methods/interpolation/@RegularizedInterpolator	129
56.1	RegularizedInterpolator3	129
56.2	init	129
57	numerical-methods/interpolation	129
57.1	IDW	129
57.2	IPoly	129

57.3	IRBM	130
57.4	ISparse	130
57.5	Inn	130
57.6	Interpolator	130
57.7	fixnan	130
57.8	idw1	130
57.9	idw2	130
57.10	inner2outer	131
57.11	inner2outer2	131
57.12	interp1_limited	131
57.13	interp1_man	131
57.14	interp1_piecewise_linear	131
57.15	interp1_save	131
57.16	interp1_slope	132
57.17	interp1_smooth	132
57.18	interp1_unique	132
57.19	interp2_man	132
57.20	interp_angle	132
57.21	interp_fourier	132
57.22	interp_fourier_batch	132
57.23	interp_sn	133
57.24	interp_sn2	133
57.25	interp_sn3	133
57.26	interp_sn_	133
57.27	limit_by_distance_1d	133
57.28	resample1	133
57.29	resample_d_min	133
57.30	resample_vector	134
57.31	test_interp1_limited	134
58	numerical-methods	134
58.1	inverse_complex	134
58.2	maccormack_step	134
58.3	minmod	134
59	numerical-methods/multigrid	134
59.1	mg_interpolate	134
59.2	mg_restrict	134
60	numerical-methods/ode/@BVPS_Characteristic	134
60.1	BVPS_Characteristic	134
60.2	assemble1_A	135
60.3	assemble1_A_Q	135
60.4	assemble2_A	135

60.5	assemble_AA	135
60.6	assemble_AAA	135
60.7	assemble_Ic	135
60.8	bvp1c	135
60.9	check_arguments	135
60.10	couple_junctions	136
60.11	derivative	136
60.12	init	136
60.13	inner2outer_bvp2c	136
60.14	reconstruct	136
60.15	resample	136
60.16	solve	136
60.17	test_assemble1_A	137
60.18	test_assemble2_A	137
61	numerical-methods/ode/@Time_Stepper	137
61.1	Time_Stepper	137
61.2	solve	137
62	numerical-methods/ode	137
62.1	bvp2fdm	137
62.2	bvp2wavetrain	138
62.3	bvp2wavetwopass	138
62.4	ivp_euler_forward	138
62.5	ivprk2	138
62.6	ode2_matrix	138
62.7	ode2characteristic	138
62.8	step_trapezoidal	138
62.9	test_bvp2	139
63	numerical-methods/optimisation	139
63.1	aitken_iteration	139
63.2	anderson_iteration	139
63.3	armijo_stopping_criterion	139
63.4	astar	139
63.5	binsearch	139
63.6	bisection	139
63.7	box1	139
63.8	box2	140
63.9	cauchy	140
63.10	cauchy2	140
63.11	directional_derivative	140
63.12	dud	140
63.13	extreme3	140

63.14	extreme-quadratic	141
63.15	ftest	141
63.16	fzero_bisect	141
63.17	fzero_newton	141
63.18	grad	141
63.19	hessian	141
63.20	hessian_from_gradient	141
63.21	hessian_projected	141
63.22	line_search	142
63.23	line_search2	142
63.24	line_search_polynomial	142
63.25	line_search_polynomial2	142
63.26	line_search_quadratic	143
63.27	line_search_quadratic2	143
63.28	line_search_wolfe	143
63.29	ls_bgfs	143
63.30	ls_broyden	143
63.31	ls_generalized_secant	144
63.32	nlcg	144
63.33	nlls	144
63.34	picard	144
63.35	poly_extrema	144
63.36	quadratic_function	144
63.37	quadratic_programming	144
63.38	quadratic_step	145
63.39	rosenbrock	145
63.40	sqrt_heron	145
63.41	test_directional_derivative	145
63.42	test_dud	145
63.43	test_fzero_newton	145
63.44	test_line_search_quadratic2	145
63.45	test_ls_generalized_secant	145
63.46	test_nlcg_6_order	145
63.47	test_nlls	146
64	numerical-methods/pde	146
64.1	laplacian2d_fundamental_solution	146
65	numerical-methods/piecewise-polynomials	146
65.1	Hermite1	146
65.2	hp2_fit	146
65.3	hp2_predict	146
65.4	hp_predict	146
65.5	hp_regress	147

65.6	lp_count	147
65.7	lp_predict	147
65.8	lp_regress	147
65.9	lp_regress_	147
66	numerical-methods	147
66.1	test_adams_bashforth	147
67	mathematics	147
67.1	quantile_sin	147
68	regression/@PolyOLS	148
68.1	PolyOLS	148
68.2	coefftest	148
68.3	detrend	148
68.4	fit	148
68.5	fit_	148
68.6	predict	148
68.7	predict_	148
68.8	slope	148
69	regression/@PowerLS	149
69.1	PowerLS	149
69.2	fit	149
69.3	predict	149
69.4	predict_	149
70	regression/@Theil	149
70.1	Theil	149
70.2	detrend	149
70.3	fit	150
70.4	predict	150
70.5	slope	150
71	regression	150
71.1	Theil_Multivariate	150
71.2	areg	150
71.3	ginireg	150
71.4	hessimplereg	151
71.5	l1lin	151
71.6	lsq_sparam	151
71.7	polyfitd	151
71.8	regression_method_of_moments	151
71.9	robustlinreg	151
71.10	theil2	152

71.11	theil_generalised	152
71.12	total_least_squares	152
71.13	weighted_median_regression	152
72	set-theory	152
72.1	issubset	152
73	mathematics	152
73.1	shuffle_index	152
74	signal-processing	153
74.1	acf_bp	153
74.2	acf_effective_sample_size	153
74.3	acf_genton	153
74.4	acf_hp	153
74.5	acf_lp	153
74.6	acfar1	153
74.7	acfar1_2	153
74.8	acfar2	153
74.9	acfar2_2	154
74.10	ar1_cutoff_frequency	154
74.11	ar1_effective_sample_size	154
74.12	ar1_mse_mu_single_sample	154
74.13	ar1_mse_pop	154
74.14	ar1_mse_range	154
74.15	ar1_spectrum	154
74.16	ar1_to_tikhonov	154
74.17	ar1_var_factor	155
74.18	ar1_var_factor_	155
74.19	ar1_var_range2	155
74.20	ar1delay	155
74.21	ar1delay_old	155
74.22	ar2conv	155
74.23	ar2dof	156
74.24	ar2param	156
74.25	asymwin	156
74.26	autocorr2	156
74.27	autocorr_fft	156
74.28	bandpass	156
74.29	bandpass1d	156
74.30	bandpass1d_implicit	156
74.31	bandpass2	157
74.32	bandpass2d	157
74.33	bandpass2d_2	157

74.34	bandpass2d_fft	157
74.35	bandpass2d_implicit	157
74.36	bandpass2d_iso	157
74.37	bartlett	157
74.38	bartlett_periodogram	157
74.39	bartlett_spectrogram	158
74.40	bin1d	158
74.41	bin2d	158
74.42	binormrnd	158
74.43	coherence	158
74.44	conv1_man	158
74.45	conv2_man	158
74.46	conv2z	159
74.47	conv30	159
74.48	conv_	159
74.49	conv_centered	159
74.50	convz	159
74.51	cosexpdelay	159
74.52	csmooth	159
74.53	daniell_window	160
74.54	danielle_window	160
74.55	db2neper	160
74.56	db2power	160
74.57	derive_bandpass_normalization_and_zeros	160
74.58	derive_danielle_weight	160
74.59	derive_limit_0_acfar	160
74.60	detect_peak	160
74.61	digital_low_pass_filter	161
74.62	doublesum_ij	161
74.63	effective_sample_size_to_ar1	161
74.64	filt_hodges_lehman	161
74.65	filter1	161
74.66	filter2	161
74.67	filter_	161
74.68	filter_arg	161
74.69	filter_f0_to_rho	162
74.70	filter_rho_to_f0	162
74.71	filteriir	162
74.72	filterp	163
74.73	filterp1	163
74.74	filterstd	163
74.75	firls_man	163
74.76	flattopwin	163
74.77	frequency_response_boxcar	163

74.78	freqz_boxcar	163
74.79	gaussfilt1	163
74.80	hanchangewin	164
74.81	hanchangewin2	164
74.82	hanwin	164
74.83	hanwin_	164
74.84	highpass	164
74.85	highpass1d_implicit	164
74.86	highpass2d_fft	164
74.87	highpass2d_implicit	164
74.88	kaiserwin	164
74.89	kalman	165
74.90	lanczoswin	165
74.91	last	165
74.92	lowpass	165
74.93	lowpass1d_fft	165
74.94	lowpass1d_implicit	165
74.95	lowpass2	165
74.96	lowpass2d_2	165
74.97	lowpass2d_fft	165
74.98	lowpass2d_implicit	166
74.99	lowpass_iir	166
74.100	lowpass_iir_symmetric	166
74.101	lowpassfilter2	166
74.102	maxfilt1	166
74.103	meanfilt1	166
74.104	medfilt1_man	166
74.105	medfilt1_man2	166
74.106	medfilt1_padded	167
74.107	medfilt1_reduced	167
74.108	mid_term_single_sample	167
74.109	minfilt1	167
74.110	mu2ar1	167
74.111	mysmooth	167
74.112	nanautocorr	167
74.113	nanmedfilt1	167
74.114	neper2db	168
74.115	peaks_man	168
74.116	polyfilt1	168
74.117	qmedfilt1	168
74.118	randar1	168
74.119	randar1_dual	168
74.120	randar2	168
74.121	randarp	168

74.122	range_window	169
74.123	rectwin	169
74.124	recursive_sum	169
74.125	select_range	169
74.126	smooth1d_parametric	169
74.127	smooth2	169
74.128	smooth_man	169
74.129	smooth_parametric	169
74.130	smooth_parametric2	170
74.131	smooth_with_splines	170
74.132	smoothfft	170
74.133	spectral_density_bp	170
74.134	spectral_density_hp	170
74.135	spectral_density_lp	170
74.136	spectrogram	170
74.137	std_window	170
74.138	sum_i_lag	170
74.139	sum_ii	171
74.140	sum_ii_	171
74.141	sum_ij	171
74.142	sum_ij_	171
74.143	sum_ij_partial_	171
74.144	sum_multivar	171
74.145	test_acfar1	171
74.146	test_acfar1_2	171
74.147	test_acfar1_3	172
74.148	test_acfar1_4	172
74.149	test_acfar2	172
74.150	test_ar1_var_factor	172
74.151	test_ar1_var_factor_2	172
74.152	test_ar1_var_mu_single_sample	172
74.153	test_ar1_var_pop	172
74.154	test_ar1_var_pop_1	172
74.155	test_ar1delay	172
74.156	test_bivariate_covariance_term	172
74.157	test_convexity	173
74.158	test_lanczoswin	173
74.159	test_madcorr	173
74.160	test_randar1	173
74.161	test_randar1_multivariate	173
74.162	test_randar2	173
74.163	test_sum_ij	173
74.164	test_sum_multivar	173
74.165	test_trifilt1	173

74.166	test_wautocorr	173
74.167	test_wavelet_transform	174
74.168	test_wordfilt	174
74.169	test_xar1_mid_term	174
74.170	tikhonov_to_ar1	174
74.171	trapwin	174
74.172	trifilt1	174
74.173	triwin	174
74.174	triwin2	174
74.175	varar1	175
74.176	welch_spectrogram	175
74.177	wfilt	175
74.178	winbandpass	175
74.179	window_make_odd	175
74.180	winfilt0	175
74.181	winlength	175
74.182	wmeanfilt	176
74.183	wmedfilt	176
74.184	wordfilt	176
74.185	wordfilt_edgeworth	176
74.186	xar1	176
74.187	xcorr_man	176
75	mathematics	176
75.1	skew_generalized_normal_fit	176
76	sorting	177
76.1	sort2	177
76.2	sort2d	177
77	special-functions	177
77.1	bessel_sphere	177
77.2	digamma_man	177
77.3	hankel_sphere	177
77.4	hermite	177
77.5	legendre_man	178
77.6	neumann_sphere	178
78	statistics	178
78.1	atan_s2	178
78.2	beta_mode_to_parameter	178
78.3	coefficient_of_determination	178
78.4	conditional_expectation_normal	178
78.5	correlation_confidence_pearson	178

79	statistics/distributions	179
79.1	PDF	179
79.2	binorm_separation_coefficient	179
79.3	binormcdf	179
79.4	binormfit	179
79.5	binormpdf	179
79.6	edgeworth_cdf	179
79.7	edgeworth_pdf	179
79.8	logn_mode2param	180
79.9	logn_param2mode	180
79.10	lognpdf_	180
79.11	pdfsample	180
79.12	t2cdf	180
79.13	t2inv	180
80	statistics	180
80.1	example_standard_error_of_sample_quantiles	180
80.2	f_var_finite	181
80.3	gamma_mode_to_parameter	181
80.4	gaussfit3	181
80.5	gaussfit_quantile	181
80.6	geoserr	181
80.7	geostd	181
80.8	hodges_lehmann_correlation	181
80.9	hodges_lehmann_dispersion	182
81	statistics/information-theory	182
81.1	akaike_information_criterion	182
81.2	bayesian_information_criterion	182
82	statistics	182
82.1	kurtncdf	182
82.2	kurtnpdf	182
82.3	kurtosis_bias_corrected	183
82.4	limit	183
82.5	logfactorial	183
82.6	loglogpdf	183
82.7	lognfit_quantile	183
82.8	logskewcdf	183
82.9	logskewpdf	183
83	statistics/logu	183
83.1	lambertw_numeric	183
83.2	logtrialtcd	184

83.3	logtrialtinv	184
83.4	logtrialtmean	184
83.5	logtrialtpdf	184
83.6	logtrialtrnd	184
83.7	logtricdf	184
83.8	logtriinv	185
83.9	logtrimean	185
83.10	logtripdf	185
83.11	logtrirnd	185
83.12	logucdf	185
83.13	logucm	185
83.14	loguinv	185
83.15	logumean	185
83.16	logupdf	186
83.17	logurnd	186
83.18	loguvar	186
83.19	medlogu	186
83.20	test_logurnd	186
83.21	tricdf	186
83.22	triinv	186
83.23	trimediam	186
83.24	tripdf	187
83.25	trirnd	187
84 statistics		187
84.1	max_exprnd	187
84.2	maxnormals	187
84.3	mean_generalized_gampdf	187
84.4	midrange	187
84.5	minavg	187
84.6	mode_man	188
85 statistics/moment-statistics		188
85.1	autocorr_man3	188
85.2	autocorr_man4	188
85.3	autocorr_man5	188
85.4	blockserr	188
85.5	comoment	188
85.6	corr_man	189
85.7	cov_man	189
85.8	dof	189
85.9	edgeworth_quantile	189
85.10	effective_sample_size	189
85.11	f_correlation	190

85.12	f_finite	190
85.13	lmean	190
85.14	lmoment	190
85.15	maskmean	190
85.16	masknanmean	190
85.17	mean1	190
85.18	mean_man	190
85.19	mse	191
85.20	nanautocorr_man1	191
85.21	nanautocorr_man2	191
85.22	nanautocorr_man4	191
85.23	nancorr	191
85.24	nancumsum	191
85.25	nanlmean	191
85.26	nanr2	192
85.27	nanrms	192
85.28	nanrmse	192
85.29	nanserr	192
85.30	nanwmean	192
85.31	nanwstd	192
85.32	nanwvar	192
85.33	nanxcorr	193
85.34	pearson	193
85.35	pearson_to_kendall	193
85.36	pool_samples	193
85.37	qmean	193
85.38	range_mean	193
85.39	rmse_	193
85.40	serr	193
85.41	serr1	194
85.42	test_kskew	194
85.43	test_qstd_kskew_optimal_p	194
85.44	wautocorr	194
85.45	wcorr	194
85.46	wcov	194
85.47	wdof	194
85.48	wkurt	195
85.49	wmean	195
85.50	wrms	195
85.51	wserr	195
85.52	wskew	195
85.53	wstd	195
85.54	wvar	195

86 statistics	196
86.1 nangeomean	196
86.2 nangeostd	196
87 statistics/nonparametric-statistics	196
87.1 kernel1d	196
87.2 kernel2d	196
88 statistics	196
88.1 normmoment	196
88.2 normpdf2	197
89 statistics/order-statistics	197
89.1 hodges_lehmann_location	197
89.2 kendall	197
89.3 kendall_to_pearson	197
89.4 mad2sd	197
89.5 madcorr	197
89.6 median2_holder	198
89.7 median_ci	198
89.8 median_man	198
89.9 mediani	198
89.10 nanmadcorr	198
89.11 nanwmedian	198
89.12 nanwquantile	198
89.13 oja_median	199
89.14 qkurtosis	199
89.15 qmoments	199
89.16 qskew	199
89.17 qskewq	199
89.18 qstdq	200
89.19 quantile1_optimisation	200
89.20 quantile2_breckling	200
89.21 quantile2_chaudhuri	200
89.22 quantile2_projected	200
89.23 quantile2_projected2	200
89.24 quantile_envelope	200
89.25 quantile_regression_simple	200
89.26 ranking	201
89.27 spatial_median	201
89.28 spatial_quantile	201
89.29 spatial_quantile2	201
89.30 spatial_quantile3	201
89.31 spatial_rank	201

89.32	spatial_sign	201
89.33	spatial_signed_rank	201
89.34	spearman	202
89.35	spearman_rank	202
89.36	spearman_to_pearson	202
89.37	wmedian	202
89.38	wquantile	202
90	statistics	202
90.1	qstd	202
90.2	quantile_extrap	202
91	statistics/random-number-generation	202
91.1	laplacernd	202
91.2	randc	203
91.3	skewness2param	203
91.4	skewpdf_central_moments	203
91.5	skewrnd	203
91.6	skewrnd2	203
92	statistics	203
92.1	range	203
92.2	resample_with_replacement	203
93	statistics/resampling-statistics/@Jackknife	204
93.1	Jackknife	204
93.2	estimated_STATIC	204
93.3	matrix1_STATIC	204
93.4	matrix2	204
94	statistics/resampling-statistics	205
94.1	block_jackknife	205
94.2	jackknife_moments	205
94.3	moving_block_jackknife	205
94.4	randblockserr	205
94.5	resample	205
95	statistics	206
95.1	scale_quantile_sd	206
95.2	sd_sample_quantiles	206
95.3	skew_generalized_normpdf	206
95.4	skewcdf	206
95.5	skewparam_to_central_moments	206
95.6	skewpdf	206
95.7	test_mean_generalized_gampdf	206

95.8	test_skew_generalized_normpdf	207
95.9	trimmed_mean	207
95.10	ttest2_man	207
95.11	ttest_man	207
95.12	ttest_paired	207
95.13	wgeomean	207
95.14	wgeovar	207
95.15	wharmean	208
95.16	wharstd	208
95.17	wharvar	208
96	mathematics	208
96.1	ternary_diagram	208
97	test/master	208
97.1	dat_test_lanczos_3d_k_20_n_40	208
97.2	poisson2d_blk	208
97.3	qr_implicit_givens_2	208
97.4	spectral_derivative_2d	208
97.5	test_2d_eigensolver_hydrogen	209
97.6	test_2d_refine	209
97.7	test_3d_eigensolver_hydrogen	209
97.8	test_FEM	209
97.9	test_Mesh_3d	209
97.10	test_arnoldi	209
97.11	test_arpackc	209
97.12	test_assemble	209
97.13	test_assembly_performance	209
97.14	test_bc_one_sided	209
97.15	test_compare_solvers	210
97.16	test_complete	210
97.17	test_convergence	210
97.18	test_convergence_b	210
97.19	test_df_2d	210
97.20	test_eig_algs	210
97.21	test_eig_inverse	210
97.22	test_eigs_lanczos	210
97.23	test_eigs_lanczos_1	210
97.24	test_eigs_lanczos_2	210
97.25	test_eigs_lanczos_performance	211
97.26	test_fdm	211
97.27	test_fdm_d_vargrid	211
97.28	test_fdm_spectral	211
97.29	test_fem	211

97.30	test_fem_1d	211
97.31	test_fem_1d_higher_order	211
97.32	test_fem_2d_adaptive	211
97.33	test_fem_2d_higher_order	211
97.34	test_fem_3d_higher_order	211
97.35	test_fem_3d_refine	212
97.36	test_fem_b	212
97.37	test_fem_derivative	212
97.38	test_fem_quadrature	212
97.39	test_final	212
97.40	test_fix_substitution	212
97.41	test_forward	212
97.42	test_get_sparse_arrays	212
97.43	test_harmonic_oscillator	212
97.44	test_high_order_fdm_periodic_bc	212
97.45	test_hydrogen_wf	213
97.46	test_ichol	213
97.47	test_interpolation	213
97.48	test_inverse_problem	213
97.49	test_it_vs_exact	213
97.50	test_jama	213
97.51	test_jd	213
97.52	test_jdqz	213
97.53	test_lanczos_2	213
97.54	test_lanczos_biorthogonal	213
97.55	test_laplacian	214
97.56	test_laplacian_non_uniform	214
97.57	test_laplacian_simple	214
97.58	test_mesh_2d_uniform	214
97.59	test_mesh_2d_uniform_2	214
97.60	test_mesh_circle	214
97.61	test_mesh_generation	214
97.62	test_mesh_interpolate	214
97.63	test_mg	214
97.64	test_minres_recycle	214
97.65	test_multigrid	215
97.66	test_nc	215
97.67	test_nonuniform_symmetric	215
97.68	test_pde	215
97.69	test_permutation	215
97.70	test_poison_fem	215
97.71	test_polar	215
97.72	test_potential	215
97.73	test_powers	215

97.74	test_precondition	215
97.75	test_project_rectangle	216
97.76	test_qr	216
97.77	test_quantum_well	216
97.78	test_radial_adaptive	216
97.79	test_radial_confinement	216
97.80	test_radial_fixes	216
97.81	test_refine_2d	216
97.82	test_refine_2d_b	216
97.83	test_refine_3d	216
97.84	test_refine_structural	216
97.85	test_regularisation	217
97.86	test_round_off	217
97.87	test_schrödinger_potentials	217
97.88	test_uniform_mesh	217
97.89	test_vargrid	217
98	test	217
98.1	test_bandpass2d	217
98.2	test_gaussfit3	217
98.3	test_geoserr	217
98.4	test_lognfit_quantile	217
98.5	test_lowpass2d_fft	218
98.6	test_max_normal	218
98.7	test_mtimes3x3	218
99	mathematics	218
99.1	test_hexagonal_pattern	218
99.2	test_lowpass1d_implicit	218
99.3	test_lowpass2d_anisotropic	218
99.4	vanderd_2d	218
100	wavelet	218
100.1	contiuous_wavelet_transform	218
100.2	cwt_man	219
100.3	example_wavelets	219
100.4	phasewrap	219
100.5	test_cwt_man	219
100.6	test_phasewrap	219
100.7	test_wavelet	219
100.8	test_wavelet2	219
100.9	test_wavelet_analysis	219
100.10	test_wavelet_reconstruct	219
100.11	test_wtc	220

100.12	wavelet	220
100.13	wavelet_reconstruct	220
100.14	wavelet_transform	220
101	mathematics	220
101.1	wnormpdf	220
101.2	wrapphase	220

1 mathematics

mathematical functions of various kind

1.1 acf_decay_rate

2 calendar

2.1 days_per_month

2.2 isnight

3 mathematics

mathematical functions of various kind

3.1 cast_byte_to_integer

cast byte to integer

4 complex-analysis

operations on complex numbers

4.1 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
```

4.2 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
```

4.3 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
```

4.4 complex_exp_product_re_re

product of the real part of two complex exponentials

```
re(c1 exp(io1x))*re(c2 exp(io2x)) =  
    1/2*(    real(c1*c2*exp(i*(n1+n2)*o*x)) ...  
          + real(conj(c1)*c2*exp(i*(n2-n1)*o*x)) )
```

the product has two frequency components

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

4.5 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
```

4.6 root_complex

root of a complex number

4.7 test_imroots

5 derivation

derivation of several functions by means of symbolic computation

5.1 derive_acfar1

5.2 `derive_ar2param`

5.3 `derive_arc_length`

5.4 `derive_fourier_power`

5.5 `derive_fourier_power_exp`

5.6 `derive_laplacian_curvilinear`

5.7 `derive_laplacian_fourier_piecewise_linear`

5.8 `derive_logtripdf`

5.9 `derive_smooth1d_parametric`

6 `derivation/master`

6.1 `derive_bc_one_sided`

- 6.2 `derive_convergence`
- 6.3 `derive_error_fdm`
- 6.4 `derive_fdm_poly`
- 6.5 `derive_fdm_power`
- 6.6 `derive_fdm_taylor`
- 6.7 `derive_fdm_vargrid`
- 6.8 `derive_fem_2d_mass`
- 6.9 `derive_fem_error_2d`
- 6.10 `derive_fem_error_3d`
- 6.11 `derive_fem_sym_2d`

6.12 `derive_grid_constants`

6.13 `derive_interpolation`

6.14 `derive_laplacian`

6.15 `derive_limit`

6.16 `derive_nc_1d`

6.17 `derive_nc_1d_`

6.18 `derive_nc_2d`

6.19 `derive_nonuniform_symmetric`

%

6.20 `derive_richardson`

6.21 `derive_sum`

6.22 `nn`

6.23 `test_derive`

6.24 `test_derive_fdm_poly`

6.25 `test_filter`

6.26 `test_vargrid`

7 `derivation`

derivation of several functions by means of symbolic computation

7.1 `simplify_atan`

symbolic simplification of the arcus tangent

8 `mathematics`

mathematical functions of various kind

8.1 `diffusion_matrix_2d_anisotropic`

8.2 `diffusion_matrix_2d_anisotropic2`

8.3 `entropy`

8.4 `euler`

8.5 `exp10`

8.6 `filter_twosided`

9 `finance`

9.1 `derive_skewrnd_walsh_paramter`

9.2 `gbm_cdf`

9.3 `gbm_fit`

9.4 `gbm_fit_old`

9.5 `gbm_inv`

9.6 `gbm_mean`

9.7 `gbm_median`

9.8 `gbm_pdf`

9.9 `gbm_simulate`

9.10 `gbm_skewness`

9.11 `gbm_std`

9.12 `gbm_transform_time_step`

9.13 `put_price_black_scholes`

9.14 `skewgbm_simulate`

9.15 `skewrnd_walsh`

10 finance/test

10.1 test_gbm

10.2 test_gbm_pdf

10.3 test_skewrnd_walsh

11 fourier/@STFT

11.1 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

11.2 itransform

inverse of the short time fourier transform

11.3 stft_

static wrapper for STFT

11.4 stftmat

transformation matrix for the short time fourier transform

11.5 transform

short time fourier transform

12 fourier

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

12.1 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 (!)

12.2 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

12.3 example_fourier_window

12.4 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

12.5 fft_man

fast fourier transform for complex input data

input:
F : data in real space

output :

F : fourier transformation of F

12.6 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

12.7 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

12.8 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

12.9 fourier_cesaro_correction

12.10 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

12.11 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

12.12 fourier_coefficient_ramp3

fourier series coefficient of a ramp

12.13 fourier_coefficient_ramp_pulse

fourier series coefficient of a ramp pules

12.14 fourier_coefficient_ramp_step

fourier coefficient of a ramp-step

12.15 fourier_coefficient_square_pulse

fourier series coefficients of a square pulse

12.16 fourier_cubic_interaction_coefficients

12.17 `fourier_derivative`

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

12.18 `fourier_expand`

expand values of fourier series

12.19 `fourier_fit`

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

12.20 `fourier_interpolate`

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

12.21 `fourier_matrix`

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

12.22 `fourier_matrix2`

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

12.23 `fourier_matrix3`

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

12.24 `fourier_matrix_exp`

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

12.25 `fourier_multiplicative_interaction_coefficients`

12.26 `fourier_power`

powers of a continuous fourier series in sin/cos form

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

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

12.27 `fourier_power_exp`

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

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

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

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

12.28 `fourier_predict`

expand a continuous fourier series at times t

12.29 `fourier_quadratic_interaction_coefficients`

12.30 `fourier_range`

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

12.31 `fourier_regress`

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

12.32 `fourier_resampled_fit`

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

12.33 `fourier_resampled_predict`

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

12.34 `fourier_signed_square`

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

12.35 fourier_transform

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

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

output :
 A : fourier matrix
 p : fourier transformation of b
 tt : TODO

12.36 hyperbolic_fourier_box

12.37 idftmtx_man

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

12.38 laplace_2d_pwlinear

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

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

12.39 nanfft

discrete fourier transform of a data series with gaps

12.40 peaks

peaks of the power spectrum of a discrete fourier transform

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

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

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)

12.41 roots_fourier

zeros of continuous fourier series series

$$f = a_0 + \sum_j^n a_i \cos(j x) + b_i \sin(j x)$$

12.42 spectral_density

spectral density

12.43 test_complex_exp_product

12.44 `test_fourier_filter`

12.45 `test_idftmtx`

13 `mathematics`

mathematical functions of various kind

13.1 `fourier_derivative_matrix_1d`

13.2 `fourier_derivative_matrix_2d`

13.3 `fractional_fourier_transform`

13.4 `gaussfit_quantile`

14 `geometry/@Geometry`

14.1 `Geometry`

14.2 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

14.3 arclength_old

arc length of a two dimensional function

14.4 arclength_old2

arc length of a two dimensional function

14.5 base_point

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

14.6 base_point_limited

base point (Fusspunkt) of a point on a line

14.7 centroid

centroid of a polygone

14.8 cosa_min_max

14.9 cross2

cross product in two dimensions

14.10 curvature

curvature of a function in two dimensions

14.11 ddot

sum of squares of cos of inner angles of triangle

14.12 distance

euclidan distance between two points

14.13 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

14.14 dot

dot product

14.15 edge_length

edge length

14.16 enclosed_angle

angle enclosed between two lines

14.17 enclosing_triangle

smallest enclosing equilateral triangle with bottom side parallel to
X axis

14.18 hexagon

coordinates of a hexagon, scaled and rotated

14.19 inPolygon

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

14.20 inTetra

flag points contained in tetrahedron

14.21 inTetra2

flag points contained in tetrahedron

14.22 inTriangle

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

14.23 intersect

intersect between two lines

14.24 lineintersect

intersect of two lines

14.25 lineintersect1

intersect of two lines

14.26 minimum_distance_lines

minimum distance of two lines in three dimensions

14.27 mittenpunkt

mittenpunkt of a triangle

14.28 nagelpoint

nagelpoint of a triangle

14.29 onLine

14.30 orthocentre

orthocentre of triangle

14.31 plumb_line

14.32 poly_area

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

14.33 poly_edges

edges of a polygon

14.34 poly_set

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

14.35 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

14.36 polyxpoly

intersections of two polygons

14.37 project_to_curve

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

14.38 quad_isconvex

14.39 random_disk

draw random points on the unit disk

14.40 random_simplex

random point inside of a triangle

14.41 sphere_volume

volume of a sphere
function v = sphere_volume(r)

14.42 tetra_volume

volume of a tetrahedron

14.43 tobarycentric

cartesian to barycentric coordinates

14.44 tobarycentric1

cartesian to barycentric coordinates

14.45 tobarycentric2

cartesian to barycentric coordinates

14.46 tobarycentric3

cartesian to barycentric coordinates

14.47 tri_angle

cos of angles of a triangle

14.48 tri_area

angle of a triangle

14.49 tri_centroid

centroid of a triangle

14.50 tri_distance_opposit_midpoint

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

14.51 tri_edge_length

edge length of a triangle

14.52 tri_edge_midpoint

mid point of a triangle

14.53 tri_excircle

excircle of a triangle

14.54 tri_height

height of a triangle

14.55 tri_incircle

incircle of a triangle

14.56 tri_isacute

flag acute triangles

14.57 tri_isobtuse

flag obtuse triangles

14.58 tri_semiperimeter

semiperimeter of a triangle

14.59 tri_side_length

edge length of triangle

15 geometry

15.1 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

15.2 bounding_box

bounding box of X

15.3 `curvature_1d`

curvature of a sampled parametric curve in two dimensions

15.4 `cvt`

centroidal voronoi tessellation

15.5 `deg_to_frac`

degree, minutes and seconds to fractions

15.6 `ellipse`

n-points on an ellipse

15.7 `ellipseX`

x-coordinates of y-coordinates of an ellipse

15.8 `ellipseY`

15.9 `first_intersect`

get first intersection between lines in A and B

15.10 `golden_ratio`

golden ratio

15.11 hypot3

hypothenuse in 3D

15.12 meanangle

weighted mean of angles

15.13 meanangle2

mean angle

15.14 meanangle3

mean angle

15.15 meanangle4

mean angle

15.16 medianangle

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

15.17 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

15.18 pilim

limit to $\pm \pi$

15.19 streamline_radius_of_curvature

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

16 mathematics

mathematical functions of various kind

16.1 hexagonal_pattern

17 histogram/@Histogram

17.1 2x

17.2 Histogram

17.3 bimodes

17.4 cdf

17.5 cdfS

17.6 `chi2test`

17.7 `cmoment`

17.8 `cmomentS`

17.9 `entropy`

17.10 `entropyS`

17.11 `export_csv`

17.12 `iquantile`

17.13 `kstest`

17.14 `kurtosis`

17.15 `kurtosisS`

17.16 mean

17.17 meanS

17.18 median

17.19 medianS

17.20 mode

17.21 modeS

17.22 moment

17.23 momentS

17.24 pdf

17.25 quantile

17.26 quantileS

17.27 resample

17.28 setup

17.29 skewness

17.30 skewnessS

17.31 stairs

17.32 stairsS

17.33 std

17.34 stdS

17.35 var

17.36 varS

18 histogram

18.1 hist_man

18.2 histadapt

18.3 histconst

18.4 pdf_poly

18.5 plotcdf

18.6 test_histogram

19 mathematics

mathematical functions of various kind

19.1 implicit_advection

19.2 imrotmat

19.3 interp1_circular

19.4 kernel2matrix

20 linear-algebra

20.1 averaging_matrix_2

20.2 colnorm

norms of columns

20.3 condest_

estimation of the condition number

21 linear-algebra/coordinate-transformation

21.1 barycentric2cartesian

barycentric to cartesian coordinates

21.2 barycentric2cartesian3

convert barycentric to cartesian coordinates

21.3 cartesian2barycentric

cartesian to barycentric coordinates

21.4 cartesian_to_unit_triangle_basis

transform coordinates into unit triangle

21.5 ellipsoid2geoid

21.6 example_approximate_utm_conversion

21.7 latlon2utm

transform latitude and longitude to WGS84 UTM

21.8 latlon2utm_simple

21.9 lowrance_mercator_to_wgs84

convert lowrance coordinates to wgs84

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

21.10 nmea2utm

convert nmea messages to utm coordinates

21.11 sn2xy

convert sn to xy coordinates

21.12 unit_triangle_to_cartesian

transform coordinates in unit triangle to cartesian coordinates

21.13 utm2latlon

convert wgs84 utm to latitude and longitude

21.14 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

21.15 xy2sn

convert cartesian to streamwise coordiantes

21.16 xy2sn_java

use java port for speed up

21.17 xy2sn_old

transform points from cartesian into streamwise coordinates

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

22 linear-algebra

22.1 det2x2

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

22.2 det3x3

determinant of stacked 3x3 matrices

22.3 det4x4

determinant of stacked 4x4 matrices

22.4 diag2x2

diagonal of stacked 2x2 matrices

22.5 down

22.6 eig2x2

eigenvalues of stacked 2x2 matrices

23 linear-algebra/eigenvalue

23.1 eig_bisection

23.2 eig_inverse

23.3 eig_inverse_iteration

23.4 eig_power_iteration

24 linear-algebra/eigenvalue/jacobi-davidson

24.1 afun_jdm

24.2 davidson

24.3 jacobi_davidson

24.4 jacobi_davidson_qr

24.5 jacobi_davidson_qz

24.6 jacobi_davidson_simple

24.7 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
%    $V*V=eye(j)$ ,  $Qschur'*V=0$ ,
%    $W=A*V$ ,  $M=W'*W$ 
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
  Rschur=[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

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

```

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

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

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

```

```

% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
W=V*Q; V=V(:,1:j)/R; E=E/R; R=eye(j); M=Q(1:j,:)' /R;
W=V*H; V(:,j+1)=[];R=R'*R; M=H(1:j,:)' ;
%===== ARNOLDI (for initializing spaces)
=====
%===== END ARNOLDI
=====
% not accurate enough M=Rw'\(M/Rv);
%===== COMPUTE SORTED JORDAN FORM
=====
% compute vectors and matrices for skew projection
% solve preconditioned system
% 0 step of bicgstab eq. 1 step of bicgstab
% Then x is a multiple of b
% HIST=[0,1];
% explicit preconditioning
% compute norm in l-space
% HIST=[HIST;[nmv,rnorm/snorm]];
% sufficient accuracy. No need to update r,u
% implicit preconditioning
% collect the updates for x in l-space
% but, do the orth to Q implicitly
% compute norm in l-space
% HIST=[HIST;[nmv,rnorm/snorm]];
% sufficient accuracy. No need to update r,u
% Do the orth to Q explicitly
% In exact arithmetic not needed, but
% appears to be more stable.
% plot(HIST(:,1),log10(HIST(:,2)+eps),'*'), drawnow, pause
% 0 step of gmres eq. 1 step of gmres
% Then x is a multiple of b
%=====

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

```

```

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

```

24.8 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
%    $V^*V = \text{eye}(j)$ ,  $Qschur' * V = 0$ ,  $W' * W = \text{eye}(j)$ ,  $Qschur' * W = 0$ 
%    $(A * V - \tau * V) = W * R + Qschur * E$ ,  $E = Qschur' * (A * V - \tau * V)$ ,  $M = W' * V$ 
%
% 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,  $AV = A * V - \tau * V$ 
%    $V^*V = \text{eye}(j)$ ,  $W' * W = \text{eye}(j)$ ,  $Qschur' * V = 0$ ,  $Qschur' * W = 0$ ,
%    $(I - Qschur * Qschur') * AV = W * R$ ,  $M = W' * V$ ;  $R = W' * AV$ ;
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
Rschur=[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

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

```

```

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

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

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

% compute vectors and matrices for skew projection
% solve preconditioned system
% 0 step of bicgstab eq. 1 step of bicgstab
% Then x is a multiple of b
% HIST=[0,1];
% explicit preconditioning
% compute norm in l-space
% HIST=[HIST;[nmv,rnorm/snorm]];
% sufficient accuracy. No need to update r,u
% implicit preconditioning
% collect the updates for x in l-space
% but, do the orth to Q implicitly
% compute norm in l-space

```



```

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

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

```

24.9 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. precondition. 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

```

24.10 jdqz

```

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

```

```

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

```

```

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

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

%=====
%=====

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

% solve preconditioned system
%=====

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

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

```

```

% Then x is a multiple of b
%=====

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

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

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

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

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

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

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

```

```

%=====

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

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

%=====

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

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

```



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

24.11 mfunc_jdm

24.12 mgs

24.13 minres_

24.14 mv_jacobi_davidson

25 linear-algebra

25.1 first

25.2 gershgorin_circle

range of eigenvalues determined by the gershgorin circle theorem

25.3 haussdorff

haussdorf dimension

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

Koch snow flake 3:4 -> 1.2619

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

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

25.4 eig2x2

reconstruct matrix from eigenvalue decomposition

25.5 inv2x2

2x2 inverse of stacked matrices

25.6 inv3x3

25.7 inv4x4

inverse of stacked 4x4 matrices

26 linear-algebra/lanczos

26.1 arnoldi

26.2 arnoldi_new

26.3 eigs_lanczos_man

26.4 lanczos

26.5 lanczos_

26.6 lanczos_biorthogonal

26.7 lanczos_biorthogonal_improved

26.8 lanczos_ghep

26.9 mv_lanczos

26.10 reorthogonalise

26.11 test_lanczos

27 linear-algebra

27.1 left

left element of vector, leftmost column is extrapolated

28 linear-algebra/linear-systems

28.1 gmres_man

break on convergence

28.2 minres_recycle

29 linear-algebra

29.1 lpmean

mean of pth-power of a

29.2 lpnorm

norm of lth-power of a

29.3 matvec3

matrix-vector product of stacked matrices and vectors

29.4 max2d

maximum value and i-j index for matrix

29.5 mid

mid point between neighbouring vector elements

29.6 mpoweri

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

29.7 mtimes2x2

29.8 mtimes3x3

product of stacked 3x3 matrices

29.9 nannorm

norm of a vector, skips nan-values

29.10 nanshift

shift vector, but set out of range values to NaN

29.11 nl

number rows (lines) of a matrix

analogue to unix nl command

29.12 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)

29.13 normalize1

normalize columns in x to [-1,1]

29.14 normrows

29.15 orth2

make matrix A orhogonal to B

29.16 orth_man

orthogonalize the columns of A

29.17 orthogonalise

make x orthogonal to Y

29.18 paddext

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

29.19 paddval1

padd values at end of x

29.20 paddval2

padd values to x

30 linear-algebra/polynomial

30.1 chebychev

chebycheff polynomials

30.2 piecewise_polynomial

evaluate piecewise polynomial

30.3 roots1

roots of linear functions

30.4 roots2

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

30.5 roots2poly

30.6 roots3

30.7 roots4

30.8 roots_piecewise_linear

30.9 test_roots4

30.10 vanderi_1d

vandermonde matrix of an integral

31 linear-algebra

31.1 randrot

random rotation matrix

31.2 right

get right column by shifting columns to left
extrapolate rightmost column

31.3 rot2

rotation matrix from angle

31.4 rot2dir

rotation matrix from direction vector

31.5 rot3

31.6 rotR

31.7 rownorm

31.8 simmilarity_matrix

31.9 spnorm

frobenius norm

31.10 spzeros

allocate a sparze matrix of zeros

31.11 test_roots3

31.12 transform_minmax

31.13 transpose3

transpose stacked 3x3 matrices

31.14 transposeall

31.15 up

32 logic

bitwise operations on integers

32.1 bitor_man

bitwise OR of the numbers of the columns of A

input:
A (positive integer)

33 mathematics

mathematical functions of various kind

33.1 lowpass2d_anisotropic

34 master/plot

34.1 attach_boundary_value

34.2 cartesian_polar

34.3 img_vargrid

34.4 `plot_basis_functions`

34.5 `plot_convergence`

34.6 `plot_dof`

34.7 `plot_eigenbar`

34.8 `plot_error_estimation`

34.9 `plot_error_estimation_2`

34.10 `plot_error_fem`

34.11 `plot_fdm_kernel`

34.12 `plot_fdm_vs_fem`

34.13 `plot_fem_accuracy`

34.14 `plot_function_and_grid`

34.15 `plot_hat`

34.16 `plot_hydrogen_wf`

34.17 `plot_mesh`

34.18 `plot_mesh_2`

34.19 `plot_refine`

34.20 `plot_refine_3d`

34.21 `plot_runtime`

34.22 `plot_spectrum`

34.23 `plot_wavefunction`

35 master/portcd

35.1 assemble_2d_phi_phi

35.2 assemble_3d_dphi_dphi

35.3 assemble_3d_phi_phi

35.4 dV_2d_

35.5 derivative_2d

35.6 derivative_3d

35.7 element_neighbour_2d

35.8 prefetch_2d_

35.9 promote_2d_3_10

35.10 promote_2d_3_15

35.11 promote_2d_3_21

35.12 promote_2d_3_6

35.13 promote_3d_4_10

35.14 promote_3d_4_20

35.15 promote_3d_4_35

35.16 vander_2d

35.17 vander_3d

36 number-theory

36.1 ceiln

floor to leading n-digits

36.2 digitsb

number of digits with respect to specified base

36.3 floorn

floor to n-digits

36.4 iseven

true for even numbers in X

36.5 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

36.6 nchoosek_man

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

36.7 pythagorean_triple

pythagorean triple

36.8 roundn

round to n digits

37 numerical-methods/differentiation

37.1 derivative1

first derivative on variable mesh
second order accurate

37.2 derivative2

second derivative on a variable mesh

38 numerical-methods/finite-difference

38.1 cdiff

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

38.2 cdiffb

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

38.3 central_difference

38.4 cmean

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

38.5 cmean2

38.6 derivative_matrix_1_1d

finite difference matrix of first derivative in one dimensions
n : number of grid points
h = L/(n+1) constant step with
function [D1, d1] = derivative_matrix_1d(n,L,order)

38.7 derivative_matrix_2_1d

finite derivative matrix of second derivative in one dimension

38.8 derivative_matrix_2d

finite difference derivative matrix in two dimensions

38.9 derivative_matrix_curvilinear

derivative matrix on a curvilinear grid

38.10 derivative_matrix_curvilinear_2

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

38.11 difference_kernel

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

38.12 distmat

distance matrix for a 2 dimensional rectangular matrix

38.13 downwind_difference

38.14 gradpde2d

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

38.15 laplacian

38.16 laplacian_fdm

finite difference matrix of the laplacian
BC

38.17 lrmean

mean of the left and right element

39 numerical-methods/finite-difference/master

39.1 fdm_adaptive_grid

39.2 fdm_adaptive_refinement_old

39.3 fdm_assemble_d1_2d

39.4 `fdm_assemble_d2_2d`

39.5 `fdm_confinement`

39.6 `fdm_d_vargrid`

39.7 `fdm_h_unstructured`

39.8 `fdm_hydrogen_vargrid`

39.9 `fdm_mark_unstructured_2d`

39.10 `fdm_plot`

39.11 `fdm_plot_series`

39.12 `fdm_refine_2d`

39.13 `fdm_refine_3d`

39.14 `fdm_refine_unstructured_2d`

39.15 `fdm_schroedinger_2d`

39.16 `fdm_schroedinger_3d`

39.17 `relocate`

40 `numerical-methods/finite-difference`

40.1 `mid`

`mid point between neighbouring vector elements`

40.2 `pwmid`

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

40.3 `ratio`

`ratio of two subsequent values`

40.4 `steplength`

`step length of a vector if it were equispaced`

40.5 swapoddeven

swap odd and even elements in a vector

40.6 test_derivative_matrix_2d

40.7 test_derivative_matrix_curvilinear

40.8 test_difference_kernel

40.9 upwind_difference

41 numerical-methods/finite-element

41.1 Mesh_2d.java

41.2 Tree_2d.java

41.3 assemble_1d_dphi_dphi

41.4 assemble_1d_phi_phi

41.5 `assemble_2d_dphi_dphi_java`

41.6 `assemble_2d_phi_phi_java`

41.7 `assemble_3d_dphi_dphi_java`

41.8 `assemble_3d_phi_phi_java`

41.9 `boundary_1d`

41.10 `boundary_2d`

41.11 `boundary_3d`

41.12 `check_area_2d`

41.13 `circmesh`

41.14 `cropradius`

41.15 `display_2d`

41.16 `display_3d`

41.17 `distort`

41.18 `err_2d`

41.19 `estimate_err_2d_3`

41.20 `example_1d`

41.21 `example_2d`

41.22 `explode`

41.23 `fem_2d`

41.24 `fem_2d_heuristic_mesh`

41.25 fem_get_2d_radial

41.26 fem_interpolation

41.27 fem_plot_1d

41.28 fem_plot_1d_series

41.29 fem_plot_2d

41.30 fem_plot_2d_series

41.31 fem_plot_3d

41.32 fem_plot_3d_series

41.33 fem_plot_confine_series

41.34 fem_radial

adaptive grid
constant grid

41.35 flip_2d

41.36 get_mesh_arrays

41.37 hashkey

42 numerical-methods/finite-element/int

42.1 int_1d_gauss

42.2 int_1d_gauss_1

42.3 int_1d_gauss_2

42.4 int_1d_gauss_3

42.5 int_1d_gauss_4

42.6 int_1d_gauss_5

42.7 int_1d_gauss_6

42.8 int_1d_gauss_lobatto

42.9 int_1d_gauss_n

42.10 int_1d_nc_2

42.11 int_1d_nc_3

42.12 int_1d_nc_4

42.13 int_1d_nc_5

42.14 int_1d_nc_6

42.15 int_1d_nc_7

42.16 int_1d_nc_7_hardy

42.17 int_2d_gauss_1

42.18 int_2d_gauss_12

42.19 int_2d_gauss_13

42.20 int_2d_gauss_16

42.21 int_2d_gauss_19

42.22 int_2d_gauss_25

42.23 int_2d_gauss_3

42.24 int_2d_gauss_33

42.25 int_2d_gauss_4

42.26 int_2d_gauss_6

42.27 int_2d_gauss_7

42.28 int_2d_gauss_9

42.29 int_2d_nc_10

42.30 int_2d_nc_15

42.31 int_2d_nc_21

42.32 int_2d_nc_3

42.33 int_2d_nc_6

42.34 int_3d_gauss_1

42.35 int_3d_gauss_11

42.36 int_3d_gauss_14

42.37 int_3d_gauss_15

42.38 int_3d_gauss_24

42.39 int_3d_gauss_4

42.40 int_3d_gauss_45

42.41 int_3d_gauss_5

42.42 int_3d_nc_11

42.43 int_3d_nc_4

42.44 int_3d_nc_6

42.45 int_3d_nc_8

43 numerical-methods/finite-element

43.1 interpolation_matrix

43.2 mark

43.3 mark_1d

43.4 mesh_1d_uniform

43.5 mesh_3d_uniform

43.6 mesh_interpolate

43.7 neighbour_1d

43.8 old

43.9 pdeeig_1d

43.10 pde eig_2d

43.11 pde eig_3d

43.12 polynomial_derivative_1d

43.13 potential_const

43.14 potential_coulomb

43.15 potential_harmonic_oscillator

43.16 project_circle

43.17 project_rectangle

43.18 promote_1d_2_3

43.19 promote_1d_2_4

43.20 promote_1d_2_5

43.21 promote_1d_2_6

43.22 quadrilaterate

43.23 recalculate_regularity_2d

43.24 refine_1d

43.25 refine_2d_21

43.26 refine_2d_structural

43.27 regularity_1d

43.28 regularity_2d

43.29 regularity_3d

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

43.30 relocate_2d

43.31 test_circmesh

43.32 test_hermite

43.33 tri_assign_points

43.34 triangulation_uniform

43.35 vander_1d

van der Monde matrix

43.36 vanderd_1d

43.37 vanderi_1d

44 numerical-methods/finite-volume/@Advection

44.1 Advection

FVM treatment of the Advection equation

44.2 dot_advection

advection equation

45 numerical-methods/finite-volume/@Burgers

45.1 burgers_split

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

45.2 dot_burgers_fdm

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

45.3 dot_burgers_fft

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

46 numerical-methods/finite-volume/@Finite_Volume

46.1 Finite_Volume

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

46.2 apply_bc

apply boundary conditions

46.3 solve

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

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

46.4 step_split_strang

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

46.5 step_unsplit

step in time, without splitting the inhomogeneous term

47 numerical-methods/finite-volume/@Flux_Limiter

47.1 Flux_Limiter

class of flux limiters

47.2 beam_warming

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

47.3 fromm

fromme limiter
low res

47.4 lax_wendroff

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

47.5 minmod

min-mod schock limiter

47.6 monotized_central

monotonized central flux limiter

47.7 muscl

muscl flux limiter

47.8 superbee

superbee limiter

47.9 upwind

godunov scheme
godunov, first order accurate

47.10 vanLeer

van Leer limiter

48 numerical-methods/finite-volume/@KDV

48.1 dot_kdv_fdm

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

48.2 dot_kdv_fft

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

48.3 kdv_split

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

49 numerical-methods/finite-volume/@Reconstruct_Average_Evolve

49.1 Reconstruct_Average_Evolve

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

McCronack Scheme

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

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

49.2 advect_highres

single time step for the reconstruct evolve algorithm

49.3 advect_lowres

single time step
low resolution

50 numerical-methods/finite-volume

50.1 Godunov

Godunov, upwind method for systems of pdes

50.2 Lax_Friedrich

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

50.3 Measure

50.4 Roe

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

The roe solver guarantess:

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

50.5 fv_swe

wrapper for solving SWE

50.6 staggered_euler

forward euler method with staggered grid

50.7 staggered_grid

staggered grid approximation to the SWE

51 numerical-methods

51.1 grid2quad

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

52 numerical-methods/integration

52.1 cumintL

cumulative integral from left to right

52.2 cumintR

cumulative integral from right to left

52.3 int_trapezoidal

integrate y along x with the trapezoidal rule

53 numerical-methods/interpolation/@Kriging

53.1 Kriging

```
class for Kriging interpolation
```

53.2 estimate_semivariance

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

53.3 interpolate_

```
interpolate with Krieking method

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

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

54 numerical-methods/interpolation/@RegularizedInterpolator

54.1 RegularizedInterpolator1

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

54.2 init

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


55 numerical-methods/interpolation/@RegularizedInterpolator2

55.1 RegularizedInterpolator2

class for regularized interpolation on an unstructures mesh (
interpolation)

55.2 init

initialize the interpolator with a set of point samples

56 numerical-methods/interpolation/@RegularizedInterpolator3

56.1 RegularizedInterpolator3

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

56.2 init

initialize the interpolator with a set of sampling points

57 numerical-methods/interpolation

57.1 IDW

spatial averaging by inverse distance weighting

57.2 IPoly

polynomial interpolation class

57.3 IRBM

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

57.4 ISparse

```
sparse interpolation class
```

57.5 Inn

```
nearest neighbour interpolation
```

57.6 Interpolator

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

57.7 fixnan

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

57.8 idw1

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

57.9 idw2

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

57.10 inner2outer

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

57.11 inner2outer2

interpolate from element (segment) centres to edge points

57.12 interp1_limited

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

57.13 interp1_man

interpolate

57.14 interp1_piecewise_linear

57.15 interp1_save

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

57.16 `interp1_slope`

quadratic interpolation returning value and derivative(s)

57.17 `interp1_smooth`

57.18 `interp1_unique`

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

57.19 `interp2_man`

nearest neighbour interpolation in two dimensions

57.20 `interp_angle`

interpolate an angle

57.21 `interp_fourier`

interpolation by the fourier method

57.22 `interp_fourier_batch`

batch interpolation by the fourier interpolation

57.23 interp_sn

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

57.24 interp_sn2

interpolation in streamwise coordinates

57.25 interp_sn3

57.26 interp_sn_

57.27 limit_by_distance_1d

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

57.28 resample1

interpolation along a parametric curve with variable step width

57.29 resample_d_min

resample a function

57.30 `resample_vector`

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

57.31 `test_interp1_limited`

58 `numerical-methods`

58.1 `inverse_complex`

58.2 `maccormack_step`

58.3 `minmod`

59 `numerical-methods/multigrid`

59.1 `mg_interpolate`

59.2 `mg_restrict`

60 `numerical-methods/ode/@BVPS_Characteristic`

60.1 `BVPS_Characteristic`

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

60.2 assemble1_A

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

60.3 assemble1_A_Q

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

60.4 assemble2_A

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

60.5 assemble_AA

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

60.6 assemble_AAA

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

60.7 assemble_Ic

60.8 bvp1c

60.9 check_arguments

60.10 couple_junctions

60.11 derivative

60.12 init

60.13 inner2outer_bvp2c

60.14 reconstruct

60.15 resample

60.16 solve

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

odefun provides ode coefficients c:

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

subject to the boundary conditions

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

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

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

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

60.17 test_assemble1_A

60.18 test_assemble2_A

61 numerical-methods/ode/@Time_Stepper

61.1 Time_Stepper

61.2 solve

62 numerical-methods/ode

62.1 bvp2fdm

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

as boundary value problems by the finite difference method

odefun provides ode coefficients c:

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

subject to the boundary conditions

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

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

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

62.2 bvp2wavetrain

solve second order boundary value problem by repeated integration

62.3 bvp2wavetwopass

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

62.4 ivp_euler_forward

solve initial value problem by the euler forward method

62.5 ivprk2

solve initial value problem by the two step runge kutta method

62.6 ode2_matrix

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

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

62.7 ode2characteristic

second order odes
transmitted and reflected wave

62.8 step_trapezoidal

single trapezoidal step

62.9 test_bvp2

63 numerical-methods/optimisation

63.1 aitken_iteration

63.2 anderson_iteration

63.3 armijo_stopping_criterion

armijo stopping criterion for optimizations

63.4 astar

astar path finding algorithm

63.5 binsearch

binary search on a line

63.6 bisection

bisection

63.7 box1

test objective function for optimisation routines

63.8 box2

63.9 cauchy

63.10 cauchy2

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

63.11 directional_derivative

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

63.12 dud

optimization by the dud algorithm

63.13 extreme3

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

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

tdx : index where extremum should be computed
t0 : location of the extremum
val0 : value of extremum

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

63.14 extreme_quadratic

63.15 ftest

63.16 fzero_bisect

63.17 fzero_newton

63.18 grad

numerical gradient

63.19 hessian

numerical hessian

63.20 hessian_from_gradient

numerical hessian from gradient

63.21 hessian_projected

numerical hessian projected to one dimension

63.22 line_search

bisection routine

63.23 line_search2

bisection method

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

63.24 line_search_polynomial

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

63.25 line_search_polynomial2

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

63.26 line_search_quadratic

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

63.27 line_search_quadratic2

```
quadratic line search
```

63.28 line_search_wolfe

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

63.29 ls_bgfs

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

63.30 ls_broyden

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

```
BGFS:
Broyden 1965
Fletcher 1970
Goldfarb 1970
Shanno 1970
```

63.31 ls_generalized_secant

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

63.32 nlcg

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

63.33 nlls

non-linear least squares

63.34 picard

picard iteration

63.35 poly_extrema

extrema of a polynomial

63.36 quadratic_function

evaluate quadratic function in higher dimensions

63.37 quadratic_programming

optimize by quadratic programming

63.38 quadratic_step

single step of the quadratic programming

63.39 rosenbrock

rosenbrock test function

63.40 sqrt_heron

Heron's method for the square root

63.41 test_directional_derivative

63.42 test_dud

63.43 test_fzero_newton

63.44 test_line_search_quadratic2

63.45 test_ls_generalized_secant

63.46 test_nlcg_6_order

63.47 test_nlls

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

64 numerical-methods/pde

64.1 laplacian2d_fundamental_solution

65 numerical-methods/piecewise-polynomials

65.1 Hermite1

hermite polynomial interpolation in 1d

65.2 hp2_fit

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

65.3 hp2_predict

prediction with pw hermite polynomial
c are values at support points

65.4 hp_predict

predict with piecewise hermite polynomial

65.5 hp_regress

fit piecewise hermite polynomial
coefficients are values and derivatives

65.6 lp_count

lagrangian basis for interpolation
count number of valid samples

65.7 lp_predict

lagrangian basis piecwie interpolation, predicor

65.8 lp_regress

65.9 lp_regress_

66 numerical-methods

66.1 test_adams_bashforth

67 mathematics

mathematical functions of various kind

67.1 quantile_sin

68 regression/@PolyOLS

68.1 PolyOLS

class for polynomial least squares

68.2 coefftest

68.3 detrend

detrending by polynomial regression

68.4 fit

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

68.5 fit_

fit a polynomial function

68.6 predict

predict polynomial function values

68.7 predict_

68.8 slope

slope by linear regression

69 regression/@PowerLS

69.1 PowerLS

class for power law regression

69.2 fit

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

69.3 predict

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

69.4 predict_

70 regression/@Theil

70.1 Theil

Kendal-Theil-Sen robust regression

70.2 detrend

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

70.3 fit

fit slope and intercept to a set of sample with the Theil-Sen method

c : confidence interval $c = 2*ns*normcdf(1)$ for ns-sigma intervals
param : itercept and slope
P : confidence interval

70.4 predict

predict values and confidence intervals with the Theil-Sen method

70.5 slope

fit the slope with the Theil-Sen method

71 regression

linear and non-linear regression

71.1 Theil_Multivariate

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

71.2 areg

regression using the pth-fraction of samples with smallest residual

71.3 ginireg

gini regression

71.4 hesssimplereg

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

71.5 llin

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

71.6 lsq_sparam

parameter covariance of the least squares regression

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

71.7 polyfitd

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

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

71.8 regression_method_of_moments

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

71.9 robustlinreg

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

71.10 theil2

Theil senn-estimator for two dimensions (glm)

71.11 theil_generalised

generalization of the Theil-Senn operator to higher dimensions,
for arbitrary functions such as polynomials and multivariate
regression
either higher order polynomials or glm
c.f. "On theil's fitting method", Pegoraro, 1991

71.12 total_least_squares

total least squares

71.13 weighted_median_regression

weighted median regression
c.f. Scholz, 1978

72 set-theory

72.1 issubset

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

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

73 mathematics

mathematical functions of various kind

73.1 shuffle_index

74 signal-processing

74.1 acf_bp

74.2 acf_effective_sample_size

effective sample size from acf

74.3 acf_genton

autocorrelation function

74.4 acf_hp

74.5 acf_lp

74.6 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
```

74.7 acfar1_2

autocorrelation of the ar1 process

74.8 acfar2

impulse response of the ar2 process

74.9 acfar2_2

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

74.10 ar1_cutoff_frequency

74.11 ar1_effective_sample_size

effective sample size correction for autocorrelated series

74.12 ar1_mse_mu_single_sample

standard error of a single sample of an ar1 correlated process

74.13 ar1_mse_pop

variance of the population mean of a single realisation around zero

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

74.14 ar1_mse_range

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

74.15 ar1_spectrum

spectrum of the ar1 process

74.16 ar1_to_tikhonov

convert ar1 correlation to tikhonovs lambda

74.17 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

74.18 ar1_var_factor_

variance of an autocorrelated finite process

74.19 ar1_var_range2

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

74.20 ar1delay

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

74.21 ar1delay_old

autocorrelation of the residual

74.22 ar2conv

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

74.23 ar2dof

effective samples size for the ar2 process

74.24 ar2param

ar2 parameter estimation from first two terms of acf

acf = [1 a1 a2 ...]

74.25 asymwin

creates asymmetrical filter windows

filter will always have negative weights

74.26 autocorr2

74.27 autocorr_fft

autocorrelation function

74.28 bandpass

bandpass filter

74.29 bandpass1d

74.30 bandpass1d_implicit

74.31 bandpass2

bandpass filter

74.32 bandpass2d

74.33 bandpass2d_2

74.34 bandpass2d_fft

74.35 bandpass2d_implicit

74.36 bandpass2d_iso

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

74.38 bartlett_periodogram

74.39 bartlett_spectrogram

bartlett spectrogramm
TODO sliding window

74.40 bin1d

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

74.41 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

74.42 binormrnd

generate two correlated normally distributed vectors

74.43 coherence

74.44 conv1_man

convolutions with padding

74.45 conv2_man

convolution in 2d

74.46 `conv2z`

74.47 `conv30`

convolve with rectangular window of length `n`
circular boundaries

74.48 `conv_`

convolution of `a` with `b`

74.49 `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

74.50 `convz`

74.51 `cosexpdelay`

74.52 `csmooth`

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

74.53 daniell_window

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

74.54 danielle_window

danielle fourier window

74.55 db2neper

convert decibel to neper

74.56 db2power

power ratio from db

74.57 derive_bandpass_normalization_and_zeros

74.58 derive_danielle_weight

74.59 derive_limit_0_acfar

74.60 detect_peak

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

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

74.62 doublesum_ij

double sum of r^i

74.63 effective_sample_size_to_ar1

convert effective sample size to ar1 correlation

74.64 filt_hodges_lehman

74.65 filter1

filter along one dimension

74.66 filter2

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

74.67 filter_

invalidate values that exceed n-times the robust standard deviation

74.68 filter_arg

74.69 filter_f0_to_rho

74.70 filter_rho_to_f0

74.71 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

74.72 filterp

74.73 filterp1

`fir filter with some fancy extras`

74.74 filterstd

74.75 firls_man

`design finite impulse response filter by the least squares method`

74.76 flattopwin

`the flat top window`

74.77 frequency_response_boxcar

`frequency response of a boxcar filter`

74.78 freqz_boxcar

`frequency response of a boxcar filter`

74.79 gaussfilt1

`filter data series with a gaussian window`

74.80 hanchangewin

hanning window for change point detection

74.81 hanchangewin2

nanning window for chage point detection

74.82 hanwin

hanning filter window

74.83 hanwin_

hanning filter window

74.84 highpass

high pass filter

74.85 highpass1d_implicit

74.86 highpass2d_fft

74.87 highpass2d_implicit

74.88 kaiserwin

kaiser filter window

74.89 kalman

Kalman filter

74.90 lanczoswin

Lanczos window

74.91 last

lake tail, but for matrices

74.92 lowpass

low pass filter

74.93 lowpass1d_fft

74.94 lowpass1d_implicit

74.95 lowpass2

design low pass filter with cutoff-frequency f1

74.96 lowpass2d_2

74.97 lowpass2d_fft

74.98 `lowpass2d_implicit`

74.99 `lowpass_iir`

`iir-low pass`

74.100 `lowpass_iir_symmetric`

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

74.101 `lowpassfilter2`

`low-pass filter of data`

74.102 `maxfilt1`

74.103 `meanfilt1`

`moving average filter with special treatment of the boundaries`

74.104 `medfilt1_man`

`moving median filter, supports columnwise operation`

74.105 `medfilt1_man2`

`moving median filter with special treatment of boundaries`

74.106 medfilt1_padded

median filter with padding

74.107 medfilt1_reduced

median filter with padding

74.108 mid_term_single_sample

variance of single sample, mid term

74.109 minfilt1

74.110 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 $\rho \rightarrow 1$

74.111 mysmooth

74.112 nanautocorr

autocorrelation with nan-values

74.113 nanmedfilt1

medfilt1, skipping nans

74.114 neper2db

convert neper to db

74.115 peaks_man

peaks of a periodogram

74.116 polyfilt1

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

74.117 qmedfilt1

medfilt1, after fitting a quadratic polynomial

74.118 randar1

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

74.119 randar1_dual

draw random variables of two correlated ar1 processes

74.120 randar2

generate ar2 process

74.121 randarp

randomly generate the instance of an ar-p process

74.122 range_window

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

74.123 rectwin

rectangular window

74.124 recursive_sum

74.125 select_range

74.126 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

74.127 smooth2

smooth vectos of X

74.128 smooth_man

74.129 smooth_parametric

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

74.130 smooth_parametric2

parametrically smooth the curve

74.131 smooth_with_splines

74.132 smoothfft

filter with fast fourier transform

74.133 spectral_density_bp

74.134 spectral_density_hp

74.135 spectral_density_lp

74.136 spectrogram

spectrogram

74.137 std_window

moving block standard deviation

74.138 sum_i_lag

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

74.139 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

74.140 sum_ii_

74.141 sum_ij

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

74.142 sum_ij_

74.143 sum_ij_partial_

74.144 sum_multivar

sum of matrix entries of bivariate ar1 process

74.145 test_acfar1

74.146 test_acfar1.2

74.147 test_acfar1_3

74.148 test_acfar1_4

74.149 test_acfar2

74.150 test_ar1_var_factor

74.151 test_ar1_var_factor_2

74.152 test_ar1_var_mu_single_sample

74.153 test_ar1_var_pop

74.154 test_ar1_var_pop_1

74.155 test_ar1delay

74.156 test_bivariate_covariance_term

74.157 test_convexity

74.158 test_lanczoswin

74.159 test_madcorr

74.160 test_randar1

74.161 test_randar1_multivariate

74.162 test_randar2

74.163 test_sum_ij

74.164 test_sum_multivar

74.165 test_trifilt1

74.166 test_wautocorr

74.167 `test_wavelet_transform`

74.168 `test_wordfilt`

74.169 `test_xar1_mid_term`

74.170 `tikhonov_to_ar1`

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

74.171 `trapwin`

trapezoidal filter window

74.172 `trifilt1`

filter with triangular window

74.173 `triwin`

triangular filter window

74.174 `triwin2`

triangular filter window

74.175 varar1

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

74.176 welch_spectrogram

welch spectrogram

74.177 wfilt

filter with window

74.178 winbandpass

filter with bandpass

74.179 window_make_odd

74.180 winfilt0

filter with window

74.181 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

74.182 wmeanfilt

mean filter with window

74.183 wmedfilt

median filter with window

74.184 wordfilt

weighted order filter

74.185 wordfilt_edgeworth

weighed order filter

74.186 xar1

74.187 xcorr_man

cross correlation of two sampled ar1 processes

75 mathematics

mathematical functions of various kind

75.1 skew_generalized_normal_fit

76 sorting

76.1 sort2

sort two numbers

76.2 sort2d

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

77 special-functions

77.1 `bessel_sphere`

spherical Bessel function of the first kind

77.2 `digamma_man`

77.3 `hankel_sphere`

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

77.4 `hermite`

probabilistic's hermite polynomial by recurrence relation

input :
n : order
x : value

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

77.5 legendre_man

legendre polynomials

77.6 neumann_sphere

spherical Neumann function
Bessel function of the second kind

78 statistics

78.1 atan_s2

stadard deviation of the arcus tangens by means of taylor expansion

78.2 beta_mode_to_parameter

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

78.3 coefficient_of_determination

78.4 conditional_expectation_normal

78.5 correlation_confidence_pearson

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

79 statistics/distributions

79.1 PDF

class for quasi-distributions from a set of sampling points

79.2 binorm_separation_coefficient

separation coefficient of a bimodal normal distribution

79.3 binormcdf

bio-modal gaussian distribution

79.4 binormfit

fit sum of to normal distribution to a histogram

79.5 binormpdf

79.6 edgeworth_cdf

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

79.7 edgeworth_pdf

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

79.8 logn_mode2param

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

79.9 logn_param2mode

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

79.10 lognpdf_

log normal distribution called by modes rather than parameters

79.11 pdfsample

pdf from sample distribution
Note: better use kernel density estimates

79.12 t2cdf

Hotelling's T-squared cumulative distribution

79.13 t2inv

inverse of Hotelling's T-squared cumulative distribution

80 statistics

80.1 example_standard_error_of_sample_quantiles

80.2 f_var_finite

reduction of variance when sampling from a finite population
without replacement

80.3 gamma_mode_to_parameter

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

80.4 gaussfit3

80.5 gaussfit_quantile

80.6 geoserr

80.7 geostd

80.8 hodges_lehmann_correlation

hodges_lehmann correlatoon coefficient
c.f. Shamos 1976
c.f. Bickel and Lehmann 1976
c.f. rousseeuw 1993
c.f. Shevlyakov 2011

80.9 hodges_lehmann_dispersion

dispersion determined by the hodges lehman method
asymptotic efficiency of dispersion estimates:
standard deviation: $E(s - \hat{s})/s = 2/\sqrt{2n} \sim 0.707/\sqrt{n}$
(100%)
hodges lehmann dispersion $E(s - \hat{s})/s = (\pi/3)^2 / (\sqrt{2n}) \sim$
 $0.775/\sqrt{n}$ (91%)
mad $E(s - \hat{s})/s \sim 1.17 s/\sqrt{n}$
(60%)
c.f. Shamos 1976
c.f. Bickel and Lehmann 1976
c.f. rousseeuw 1993
nb: rousseeuw uses the 25th percentile, which is more efficient for
small sample sizes

81 statistics/information-theory

81.1 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

81.2 bayesian_information_criterion

bayesian information criterion

82 statistics

82.1 kurtncdf

82.2 kurtnpdf

82.3 kurtosis_bias_corrected

bias corrected kurtosis

82.4 limit

limit a by lower and upper bound

82.5 logfactorial

approximate log of the factorial

82.6 loglogpdf

82.7 lognfit_quantile

82.8 logskewcdf

82.9 logskewpdf

83 statistics/logu

83.1 lambertw_numeric

lambert-w function

83.2 logtrialtcdf

pdf of a logarithmic triangular distribution

83.3 logtrialtinv

inverse of the logarithmic triangular distribution

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

83.4 logtrialtmean

mean of the logarithmic triangular distribution

83.5 logtrialtpdf

density of the logarithmic triangular distribution

83.6 logtrialtrnd

83.7 logtricdf

cumulative distribution of the logarithmic triangular distribution

83.8 logtriinv

inverse of the logarithmic triangular distribution

83.9 logtrimean

mean of the logarithmic triangular distribution

83.10 logtripdf

probability density of the logarithmic triangular distribution

83.11 logtrirnd

83.12 logucdf

probability density of the logarithmic uniform distribution

83.13 logucm

central moments of the log-uniform distribution

83.14 loguinv

inverse of the log-uniform distribution

83.15 logumean

mean of the log-uniform distribution

83.16 logupdf

pdf of the log uniform distribution

83.17 logurnd

random numbers following a log-uniform distribution

83.18 loguvar

variance of the log-uniform distribution

83.19 medlogu

median of the log-uniform distribution

83.20 test_logurnd

83.21 tricdf

cumulative distribution of the log-triangular distribution

83.22 triinv

inverse of the triangular distribution

83.23 trimedian

median of the triangular distribution

83.24 tripdf

probability density of the triangular distribution

83.25 trirnd

random numbers of the triangular distribution

84 statistics

84.1 max_exprnd

84.2 maxnnormals

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

84.3 mean_generalized_gampdf

84.4 midrange

mid range of columns of X

84.5 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

84.6 mode_man

85 statistics/moment-statistics

85.1 autocorr_man3

autocorrelation of the columns of X

85.2 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

85.3 autocorr_man5

autocorrellation of the columns of X

85.4 blockserr

estimate the standard error of potetially sequentilly 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

85.5 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

85.6 corr_man

correlation of two vectors

85.7 cov_man

covariance matrix of two vectors

85.8 dof

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

85.9 edgeworth_quantile

inverse edgeworth expansion

c.f. cornis fisher 1937

c.f. Rao 2010

c.f. 2.50 in hall

CHERNOZHUKOV 3.3

85.10 effective_sample_size

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

85.11 f_correlation

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

85.12 f_finite

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

85.13 lmean

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

85.14 lmoment

l-moment of vector x

85.15 maskmean

mean of the masked values of X

85.16 masknanmean

85.17 mean1

mean of x

85.18 mean_man

mean and standard error of X

85.19 mse

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

85.20 nanautocorr_man1

autocorrelation of a vector with nan-values

85.21 nanautocorr_man2

autocorrelation of a vector with nan-values

85.22 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

85.23 nancorr

(co)-correlation matrix when samples a NaN

85.24 nancumsum

cumulative sum, setting nan values to zero

85.25 nanlmean

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

85.26 nanr2

coefficient of determination when samples are invalid

85.27 nanrms

root mean square value when sample contains nan-values

85.28 nanrmse

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

85.29 nanserr

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

85.30 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)

85.31 nanwstd

weighed standard deviation

85.32 nanwvar

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

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

85.33 nanxcorr

85.34 pearson

pearson correlation coefficient

85.35 pearson_to_kendall

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

85.36 pool_samples

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

85.37 qmean

trimmed mean

85.38 range_mean

85.39 rmse_

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

85.40 serr

standard error of the mean of a set of uncorrelated samples

85.41 `serr1`

85.42 `test_qskew`

85.43 `test_qstd_qskew_optimal_p`

85.44 `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

85.45 `wcorr`

correlation of two vectors when samples are weighted

85.46 `wcov`

covariance of two vectors when samples are weighted

85.47 `wdof`

effective degrees of freedom for weighted samples

85.48 wkurt

kurtosis with weighted samples

85.49 wmean

weighted mean

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

varargin can be dim

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

85.50 wrms

weighted root mean square

85.51 wserr

weighted root mean square error

85.52 wskew

skewness of a weighted set of samples

85.53 wstd

weighed standard deviation

85.54 wvar

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

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

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

s2_mu : error of mean, s2_mu : sd of prediction

86 statistics

86.1 nangeomean

86.2 nangeostd

geometric standard deviation ignoring nan-values

87 statistics/nonparametric-statistics

87.1 kernel1d

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

87.2 kernel2d

kernel density estimate in two dimensions

88 statistics

88.1 normmoment

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

88.2 normpdf2

pdf of the bivariate normal distribution

89 statistics/order-statistics

89.1 hodges_lehmann_location

hodges lehman location estimator

Asymptotic rms efficiency of location estimate:

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

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

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

89.2 kendall

kendall correlation coefficient

89.3 kendall_to_pearson

convert kendall rank correlation coefficient to the person product
moment

correlation coefficient

c.f. Kruska, 1985

89.4 mad2sd

transform median absolute deviation to standard deviation
for normal distributed values

89.5 madcorr

proxy correlation by median absolute deviation

89.6 median2_holder

89.7 median_ci

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

89.8 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`

89.9 mediani

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

89.10 nanmadcorr

proxy correlation by median absolute deviation

89.11 nanwmedian

weighted median, skips nan-values

89.12 nanwquantile

weighted quantile, skips nan values

89.13 oja_median

two dimensional oja median

note: the multivariate median is not unique

oja 1983, for extension to multivariate function, see chaudhri

89.14 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

89.15 qmoments

moments estimated from quantiles

89.16 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

89.17 qskewq

skewness estimated by quantiles

89.18 qstdq

proxy standard deviation determined by quantiles

89.19 quantile1_optimisation

89.20 quantile2_breckling

qunatile regression

89.21 quantile2_chaudhuri

quantile regression

89.22 quantile2_projected

quantile in two dimensions

89.23 quantile2_projected2

spatial qunatile for chosen direction

89.24 quantile_envelope

89.25 quantile_regression_simple

simple quantile regression

89.26 ranking

ranking for spearman statistics

89.27 spatial_median

c.f. Oja 2008

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

89.28 spatial_quantile

spatial quantile

89.29 spatial_quantile2

spatial quantile

89.30 spatial_quantile3

spatial quantile

89.31 spatial_rank

unsigned rank

89.32 spatial_sign

spatial sign

89.33 spatial_signed_rank

signed rank

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

89.34 spearman

spearman's product moment coefficient

89.35 spearman_rank

89.36 spearman_to_pearson

conversion of spearman rank to person product moment correlation coefficient

89.37 wmedian

weighted median

89.38 wquantile

weighted quantile

90 statistics

90.1 qstd

90.2 quantile_extrap

91 statistics/random-number-generation

91.1 laplacernd

random number of laplace distribution

91.2 randc

correlate to correlated standard normally distributed vectors

91.3 skewness2param

91.4 skewpdf_central_moments

91.5 skewrnd

random numbers of the skew normal distribution

91.6 skewrnd2

random numbers of the skew normal distribution

92 statistics

92.1 range

mid range

92.2 resample_with_replacement

93 statistics/resampling-statistics/@Jackknife

93.1 Jackknife

class for leave out 1 (delete 1) Jackknife estimates

note 1 : the 1-delete jackknife does not yield consistend estimates
for all functions,
in particular it will perform poorly on robust estimation
functions
this is overcome by the d-delete jackknife, where d has to
exceed the breakdown point
of the estimating function, for example \sqrt{n} for the
median
as this leads to unreasonably large number of repetitions,
bootstrap
is recommended for large sample cases (or blocking for
sequential data)
note 2 : as a linearisation, jackknife underestimates the error
variance in case of
dependence in the data
note 3 : studentisation and the leave out 1 jackknife are related
note 4 : the double 1 sample jackknife performs iferior to the d1
jackknife

93.2 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

93.3 matrix1_STATIC

matrix of estimation for leaving out two samples at a time

93.4 matrix2

matrix of estimations for jacknive with two samples left out

94 statistics/resampling-statistics

94.1 block_jackknife

94.2 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

94.3 moving_block_jackknife

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

94.4 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

94.5 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

95 statistics

95.1 scale_quantile_sd

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

95.2 sd_sample_quantiles

95.3 skew_generalized_normpdf

95.4 skewcdf

95.5 skewparam_to_central_moments

95.6 skewpdf

skew-normal distribution
c.f. Azzalini 1985

95.7 test_mean_generalized_gampdf

95.8 test_skew_generalized_normpdf

95.9 trimmed_mean

trimmed mean

95.10 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

95.11 ttest_man

two-sample t-test

unequal sample size

equal variance

95.12 ttest_paired

paired t-test

unequal sample size

equal variance

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

95.13 wgeomean

weighted geometric mean

function mu = wgeomean(w,x)

95.14 wgeovar

variance of the weighted geometric mean

95.15 wharmean

weighted harmonic mean

95.16 wharstd

95.17 wharvar

96 mathematics

mathematical functions of various kind

96.1 ternary_diagram

97 test/master

97.1 dat_test_lanczos_3d_k_20_n_40

97.2 poisson2d_blk

97.3 qr_implicit_givens_2

97.4 spectral_derivative_2d

97.5 test_2d_eigensolver_hydrogen

97.6 test_2d_refine

97.7 test_3d_eigensolver_hydrogen

97.8 test_FEM

97.9 test_Mesh_3d

97.10 test_arnoldi

97.11 test_arpackc

97.12 test_assemble

97.13 test_assembly_performance

97.14 test_bc_one_sided

97.15 test_compare_solvers

97.16 test_complete

97.17 test_convergence

97.18 test_convergence_b

97.19 test_df_2d

97.20 test_eig_algs

97.21 test_eig_inverse

97.22 test_eigs_lanczos

97.23 test_eigs_lanczos_1

97.24 test_eigs_lanczos_2

97.25 test_eigs_lanczos_performance

97.26 test_fdm

97.27 test_fdm_d_vargrid

97.28 test_fdm_spectral

97.29 test_fem

97.30 test_fem_1d

97.31 test_fem_1d_higher_order

97.32 test_fem_2d_adaptive

97.33 test_fem_2d_higher_order

97.34 test_fem_3d_higher_order

97.35 test_fem_3d_refine

97.36 test_fem_b

97.37 test_fem_derivative

97.38 test_fem_quadrature

97.39 test_final

97.40 test_fix_substitution

97.41 test_forward

97.42 test_get_sparse_arrays

97.43 test_harmonic_oscillator

97.44 test_high_order_fdm_periodic_bc

97.45 test_hydrogen_wf

97.46 test_ichol

97.47 test_interpolation

97.48 test_inverse_problem

97.49 test_it_vs_exact

97.50 test_jama

97.51 test_jd

97.52 test_jdqz

97.53 test_lanczos_2

97.54 test_lanczos_biorthogonal

97.55 test_laplacian

97.56 test_laplacian_non_uniform

97.57 test_laplacian_simple

97.58 test_mesh_2d_uniform

97.59 test_mesh_2d_uniform_2

97.60 test_mesh_circle

97.61 test_mesh_generation

97.62 test_mesh_interpolate

97.63 test_mg

97.64 test_minres_recycle

97.65 test_multigrid

97.66 test_nc

97.67 test_nonuniform_symmetric

97.68 test_pde

97.69 test_permutation

97.70 test_poison_fem

97.71 test_polar

97.72 test_potential

97.73 test_powers

97.74 test_precondition

97.75 test_project_rectangle

97.76 test_qr

97.77 test_quantum_well

97.78 test_radial_adaptive

97.79 test_radial_confinement

97.80 test_radial_fixes

97.81 test_refine_2d

97.82 test_refine_2d_b

97.83 test_refine_3d

97.84 test_refine_structural

97.85 test_regularisation

97.86 test_round_off

97.87 test_schrödinger_potentials

97.88 test_uniform_mesh

97.89 test_vargrid

98 test

98.1 test_bandpass2d

98.2 test_gaussfit3

98.3 test_geoserr

98.4 test_lognfit_quantile

98.5 test_lowpass2d_fft

98.6 test_max_normal

98.7 test_mtimes3x3

99 mathematics

mathematical functions of various kind

99.1 test_hexagonal_pattern

99.2 test_lowpass1d_implicit

99.3 test_lowpass2d_anisotropic

99.4 vanderd_2d

100 wavelet

100.1 continuous_wavelet_transform

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

100.2 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

100.3 example_wavelets

100.4 phasewrap

wrap the phase to $\pm \pi$

100.5 test_cwt_man

100.6 test_phasewrap

100.7 test_wavelet

100.8 test_wavelet2

100.9 test_wavelet_analysis

100.10 test_wavelet_reconstruct

100.11 test_wtc

100.12 wavelet

wavelet windows

100.13 wavelet_reconstruct

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

100.14 wavelet_transform

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

101 mathematics

mathematical functions of various kind

101.1 wnormpdf

101.2 wrapphase