

Manual for Package: mathematics

Revision 20M

Karl Kästner

December 22, 2022

Contents

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

5.1	derive_bc_one_sided	5
5.2	derive_convergence	5
5.3	derive_error_fdm	5
5.4	derive_fdm_poly	5
5.5	derive_fdm_power	5
5.6	derive_fdm_taylor	5
5.7	derive_fdm_vargrid	5
5.8	derive_fem_2d_mass	5
5.9	derive_fem_error_2d	5
5.10	derive_fem_error_3d	6
5.11	derive_fem_sym_2d	6
5.12	derive_grid_constants	6
5.13	derive_interpolation	6
5.14	derive_laplacian	6
5.15	derive_limit	6
5.16	derive_nc_1d	6
5.17	derive_nc_1d_	6
5.18	derive_nc_2d	6
5.19	derive_nonuniform_symmetric	6
5.20	derive_richardson	7
5.21	derive_sum	7
5.22	nn	7
5.23	test_derive	7
5.24	test_derive_fdm_poly	7
5.25	test_filter	7
5.26	test_vargrid	7
6	derivation	7
6.1	simplify_atan	7
7	mathematics	8
7.1	entropy	8
8	finance	8
8.1	derive_skewrnd_walsh_paramter	8
8.2	gbm_cdf	8
8.3	gbm_fit	8
8.4	gbm_fit_old	8
8.5	gbm_inv	8
8.6	gbm_mean	8
8.7	gbm_median	8
8.8	gbm_pdf	9
8.9	gbm_simulate	9
8.10	gbm_skewness	9

8.11	gbm_std	9
8.12	gbm_transform_time_step	9
8.13	put_price_black_scholes	9
8.14	skewgbm_simulate	9
8.15	skewrnd_walsh	9
9	finance/test	9
9.1	test_gbm	9
9.2	test_gbm_pdf	10
9.3	test_skewrnd_walsh	10
10	fourier/@STFT	10
10.1	STFT	10
10.2	itransform	10
10.3	stft_	10
10.4	stftmat	10
10.5	transform	10
11	fourier	11
11.1	amplitude_from_peak	11
11.2	caesaro_weight	11
11.3	dftmtx_man	11
11.4	example_fourier_window	11
11.5	fft_man	11
11.6	fftsmooth	12
11.7	fix_fourier	12
11.8	fourier_axis	12
11.9	fourier_axis_2d	13
11.10	fourier_cesaro_correction	13
11.11	fourier_coefficient_piecewise_linear	13
11.12	fourier_coefficient_piecewise_linear_1	13
11.13	fourier_coefficient_ramp3	14
11.14	fourier_coefficient_ramp_pulse	14
11.15	fourier_coefficient_ramp_step	14
11.16	fourier_coefficient_square_pulse	14
11.17	fourier_complete_negative_half_plane	14
11.18	fourier_cubic_interaction_coefficients	14
11.19	fourier_derivative	15
11.20	fourier_derivative_matrix_1d	15
11.21	fourier_derivative_matrix_2d	15
11.22	fourier_expand	15
11.23	fourier_fit	15
11.24	fourier_interpolate	16
11.25	fourier_matrix	16

11.26	fourier_matrix2	16
11.27	fourier_matrix3	16
11.28	fourier_matrix_exp	16
11.29	fourier_multiplicative_interaction_coefficients	16
11.30	fourier_power	16
11.31	fourier_power_exp	17
11.32	fourier_predict	17
11.33	fourier_quadratic_interaction_coefficients	17
11.34	fourier_random_phase_walk	17
11.35	fourier_range	17
11.36	fourier_regress	17
11.37	fourier_resampled_fit	18
11.38	fourier_resampled_predict	18
11.39	fourier_signed_square	18
11.40	fourier_transform	18
11.41	fourier_transform_fractional	18
11.42	fourier_truncate_negative_half_plane	19
11.43	hyperbolic_fourier_box	19
11.44	idftmtx_man	19
11.45	laplace_2d_pwlinear	19
11.46	mean_fourier_power	19
11.47	moments_fourier_power	19
11.48	nanfft	20
11.49	peaks	20
11.50	roots_fourier	20
11.51	spectral_density	20
11.52	std_fourier_power	20
11.53	test_complex_exp_product	21
11.54	test_fourier_filter	21
11.55	test_idftmtx	21
11.56	var_fourier_power	21
12 mathematics		21
12.1	gaussfit_quantile	21
13 geometry/@Geometry		21
13.1	Geometry	21
13.2	arclength	21
13.3	arclength_old	22
13.4	arclength_old2	22
13.5	base_point	22
13.6	base_point_limited	22
13.7	centroid	22
13.8	cosa_min_max	22

13.9	cross2	22
13.10	curvature	22
13.11	ddot	23
13.12	distance	23
13.13	distance2	23
13.14	dot	23
13.15	edge_length	23
13.16	enclosed_angle	23
13.17	enclosing_triangle	23
13.18	hexagon	23
13.19	inPolygon	24
13.20	inTetra	24
13.21	inTetra2	24
13.22	inTriangle	24
13.23	intersect	24
13.24	lineintersect	24
13.25	lineintersect1	24
13.26	minimum_distance_lines	24
13.27	mittenpunkt	25
13.28	nagelpoint	25
13.29	onLine	25
13.30	orthocentre	25
13.31	plumb_line	25
13.32	poly_area	25
13.33	poly_edges	25
13.34	poly_set	25
13.35	poly_width	26
13.36	polyxpoly	26
13.37	project_to_curve	26
13.38	quad_isconvex	26
13.39	random_disk	26
13.40	random_simplex	26
13.41	sphere_volume	26
13.42	tetra_volume	27
13.43	tobarycentric	27
13.44	tobarycentric1	27
13.45	tobarycentric2	27
13.46	tobarycentric3	27
13.47	tri_angle	27
13.48	tri_area	27
13.49	tri_centroid	27
13.50	tri_distance_opposit_midpoint	28
13.51	tri_edge_length	28
13.52	tri_edge_midpoint	28

13.53	tri_excircle	28
13.54	tri_height	28
13.55	tri_incircle	28
13.56	tri_isacute	28
13.57	tri_isobtuse	28
13.58	tri_semiperimeter	29
13.59	tri_side_length	29
14	geometry	29
14.1	Polygon	29
14.2	bounding_box	29
14.3	curvature_1d	29
14.4	cvt	29
14.5	deg_to_frac	30
14.6	ellipse	30
14.7	ellipseX	30
14.8	ellipseY	30
14.9	first_intersect	30
14.10	golden_ratio	30
14.11	hypot3	30
14.12	meanangle	30
14.13	meanangle2	31
14.14	meanangle3	31
14.15	meanangle4	31
14.16	medianangle	31
14.17	medianangle2	31
14.18	pilim	31
14.19	streamline_radius_of_curvature	32
15	histogram/@Histogram	32
15.1	2x	32
15.2	Histogram	32
15.3	bimodes	32
15.4	cdf	32
15.5	cdfS	32
15.6	chi2test	32
15.7	cmoment	32
15.8	cmomentS	32
15.9	entropy	33
15.10	entropyS	33
15.11	export_csv	33
15.12	iquantile	33
15.13	kstest	33
15.14	kurtosis	33

15.15	kurtosisS	33
15.16	mean	33
15.17	meanS	33
15.18	median	33
15.19	medianS	34
15.20	mode	34
15.21	modeS	34
15.22	moment	34
15.23	momentS	34
15.24	pdf	34
15.25	quantile	34
15.26	quantileS	34
15.27	resample	34
15.28	setup	34
15.29	skewness	35
15.30	skewnessS	35
15.31	stairs	35
15.32	stairsS	35
15.33	std	35
15.34	stdS	35
15.35	var	35
15.36	varS	35
16	histogram	35
16.1	hist_man	35
16.2	histadapt	36
16.3	histconst	36
16.4	pdf_poly	36
16.5	plotcdf	36
16.6	test_histogram	36
17	mathematics	36
17.1	imrotmat	36
18	linear-algebra	36
18.1	averaging_matrix_2	36
18.2	colnorm	36
18.3	condest_	37
18.4	connectivity_matrix	37
19	linear-algebra/coordinate-transformation	37
19.1	barycentric2cartesian	37
19.2	barycentric2cartesian3	37
19.3	cartesian2barycentric	37

19.4	cartesian_to_unit_triangle_basis	37
19.5	ellipsoid2geoid	37
19.6	example_approximate_utm_conversion	37
19.7	latlon2utm	38
19.8	latlon2utm_simple	38
19.9	lowrance_mercator_to_wgs84	38
19.10	nmea2utm	38
19.11	sn2xy	38
19.12	unit_triangle_to_cartesian	38
19.13	utm2latlon	38
19.14	xy2nt	39
19.15	xy2sn	39
19.16	xy2sn_java	39
19.17	xy2sn_old	39
20	linear-algebra	39
20.1	det2x2	39
20.2	det3x3	39
20.3	det4x4	40
20.4	diag2x2	40
20.5	down	40
20.6	eig2x2	40
21	linear-algebra/eigenvalue	40
21.1	eig_bisection	40
21.2	eig_inverse	40
21.3	eig_inverse_iteration	40
21.4	eig_power_iteration	40
22	linear-algebra/eigenvalue/jacobi-davidson	41
22.1	afun_jdm	41
22.2	davidson	41
22.3	jacobi_davidson	41
22.4	jacobi_davidson_qr	41
22.5	jacobi_davidson_qz	41
22.6	jacobi_davidson_simple	41
22.7	jdqr	41
22.8	jdqr_sleijpen	45
22.9	jdqr_vorst	48
22.10	jdqz	51
22.11	mfunc_jdm	55
22.12	mgs	56
22.13	minres_	56
22.14	mv_jacobi_davidson	56

23	linear-algebra	56
23.1	first	56
23.2	gershgorin_circle	56
23.3	hausdorff	56
23.4	ieig2x2	56
23.5	inv2x2	57
23.6	inv3x3	57
23.7	inv4x4	57
23.8	kernel2matrix	57
24	linear-algebra/lanczos	57
24.1	arnoldi	57
24.2	arnoldi_new	57
24.3	eigs_lanczos_man	57
24.4	lanczos	57
24.5	lanczos_	57
24.6	lanczos_biorthogonal	58
24.7	lanczos_biorthogonal_improved	58
24.8	lanczos_ghep	58
24.9	mv_lanczos	58
24.10	reorthogonalise	58
24.11	test_lanczos	58
25	linear-algebra	58
25.1	laplacian_eigenvalue	58
25.2	laplacian_eigenvector	58
25.3	laplacian_power	58
25.4	least_squares_perpendicular_offset	59
25.5	left	59
26	linear-algebra/linear-systems	59
26.1	gmres_man	59
26.2	minres_recycle	59
27	linear-algebra	59
27.1	lpmean	59
27.2	lpnorm	59
27.3	matvec3	59
27.4	max2d	59
27.5	mid	60
27.6	mpoweri	60
27.7	mtimes2x2	60
27.8	mtimes3x3	60
27.9	nannorm	60

27.10	nanshift	60
27.11	nl	60
27.12	normalise	61
27.13	normalize1	61
27.14	normrows	61
27.15	orth2	61
27.16	orth_man	61
27.17	orthogonalise	61
27.18	padd2	61
27.19	paddext	62
27.20	paddval1	62
27.21	paddval2	62
28	linear-algebra/polynomial	62
28.1	chebychev	62
28.2	piecewise_polynomial	62
28.3	roots1	62
28.4	roots2	62
28.5	roots2poly	63
28.6	roots3	63
28.7	roots4	63
28.8	roots_piecewise_linear	63
28.9	test_roots4	63
28.10	vanderi_1d	63
29	linear-algebra	63
29.1	randrot	63
29.2	right	63
29.3	rot2	64
29.4	rot2dir	64
29.5	rot3	64
29.6	rotR	64
29.7	rownorm	64
29.8	simmilarity_matrix	64
29.9	spnorm	64
29.10	spzeros	64
29.11	test_roots3	64
29.12	transform_minmax	65
29.13	transpose3	65
29.14	transposeall	65
29.15	up	65
29.16	vander_nd	65
29.17	vanderd_2d	65

30	logic	65
30.1	bitor_man	65
31	master/plot	65
31.1	attach_boundary_value	65
31.2	cartesian_polar	66
31.3	img_vargrid	66
31.4	plot_basis_functions	66
31.5	plot_convergence	66
31.6	plot_dof	66
31.7	plot_eigenbar	66
31.8	plot_error_estimation	66
31.9	plot_error_estimation_2	66
31.10	plot_error_fem	66
31.11	plot_fdm_kernel	66
31.12	plot_fdm_vs_fem	67
31.13	plot_fem_accuracy	67
31.14	plot_function_and_grid	67
31.15	plot_hat	67
31.16	plot_hydrogen_wf	67
31.17	plot_mesh	67
31.18	plot_mesh_2	67
31.19	plot_refine	67
31.20	plot_refine_3d	67
31.21	plot_runtime	67
31.22	plot_spectrum	68
31.23	plot_wavefunction	68
32	master/ported	68
32.1	assemble_2d_phi_phi	68
32.2	assemble_3d_dphi_dphi	68
32.3	assemble_3d_phi_phi	68
32.4	dV_2d_	68
32.5	derivative_2d	68
32.6	derivative_3d	68
32.7	element_neighbour_2d	68
32.8	prefetch_2d_	69
32.9	promote_2d_3_10	69
32.10	promote_2d_3_15	69
32.11	promote_2d_3_21	69
32.12	promote_2d_3_6	69
32.13	promote_3d_4_10	69
32.14	promote_3d_4_20	69
32.15	promote_3d_4_35	69

32.16	vander_2d	69
32.17	vander_3d	69
33	mathematics	70
33.1	nearest_fractional_timestep	70
34	number-theory	70
34.1	ceiln	70
34.2	digitsb	70
34.3	floorn	70
34.4	iseven	70
34.5	multichoosek	70
34.6	nchoosek_man	71
34.7	pythagorean_triple	71
34.8	roundn	71
35	numerical-methods	71
35.1	advect_analytic	71
36	numerical-methods/differentiation	71
36.1	derivative1	71
36.2	derivative2	71
37	numerical-methods	71
37.1	diffuse_analytic	71
38	numerical-methods/finite-difference	72
38.1	cdiff	72
38.2	cdiffb	72
38.3	central_difference	72
38.4	cmean	72
38.5	cmean2	72
38.6	derivative_matrix_1_1d	72
38.7	derivative_matrix_2_1d	72
38.8	derivative_matrix_2d	73
38.9	derivative_matrix_curvilinear	73
38.10	derivative_matrix_curvilinear_2	73
38.11	difference_kernel	73
38.12	diffusion_matrix_2d_anisotropic	73
38.13	diffusion_matrix_2d_anisotropic2	73
38.14	directional_neighbour	73
38.15	distmat	73
38.16	downwind_difference	74
38.17	gradpde2d	74
38.18	laplacian	74

38.19	laplacian_fdm	74
38.20	lrmean	74
39	numerical-methods/finite-difference/master	74
39.1	fdm_adaptive_grid	74
39.2	fdm_adaptive_refinement_old	74
39.3	fdm_assemble_d1_2d	74
39.4	fdm_assemble_d2_2d	75
39.5	fdm_confinement	75
39.6	fdm_d_vargrid	75
39.7	fdm_h_unstructured	75
39.8	fdm_hydrogen_vargrid	75
39.9	fdm_mark_unstructured_2d	75
39.10	fdm_plot	75
39.11	fdm_plot_series	75
39.12	fdm_refine_2d	75
39.13	fdm_refine_3d	75
39.14	fdm_refine_unstructured_2d	76
39.15	fdm_schroedinger_2d	76
39.16	fdm_schroedinger_3d	76
39.17	relocate	76
40	numerical-methods/finite-difference	76
40.1	mid	76
40.2	pwmid	76
40.3	ratio	76
40.4	steplength	76
40.5	swapoddeven	77
40.6	test_derivative_matrix_2d	77
40.7	test_derivative_matrix_curvilinear	77
40.8	test_difference_kernel	77
40.9	upwind_difference	77
41	numerical-methods/finite-element	77
41.1	Mesh_2d.java	77
41.2	Tree_2d.java	77
41.3	assemble_1d_dphi_dphi	77
41.4	assemble_1d_phi_phi	77
41.5	assemble_2d_dphi_dphi_java	78
41.6	assemble_2d_phi_phi_java	78
41.7	assemble_3d_dphi_dphi_java	78
41.8	assemble_3d_phi_phi_java	78
41.9	boundary_1d	78
41.10	boundary_2d	78

41.11	boundary_3d	78
41.12	check_area_2d	78
41.13	circmesh	78
41.14	cropradius	78
41.15	display_2d	79
41.16	display_3d	79
41.17	distort	79
41.18	err_2d	79
41.19	estimate_err_2d_3	79
41.20	example_1d	79
41.21	example_2d	79
41.22	explode	79
41.23	fem_2d	79
41.24	fem_2d_heuristic_mesh	79
41.25	fem_get_2d_radial	80
41.26	fem_interpolation	80
41.27	fem_plot_1d	80
41.28	fem_plot_1d_series	80
41.29	fem_plot_2d	80
41.30	fem_plot_2d_series	80
41.31	fem_plot_3d	80
41.32	fem_plot_3d_series	80
41.33	fem_plot_confine_series	80
41.34	fem_radial	81
41.35	flip_2d	81
41.36	get_mesh_arrays	81
41.37	hashkey	81
42	numerical-methods/finite-element/int	81
42.1	int_1d_gauss	81
42.2	int_1d_gauss_1	81
42.3	int_1d_gauss_2	81
42.4	int_1d_gauss_3	81
42.5	int_1d_gauss_4	81
42.6	int_1d_gauss_5	82
42.7	int_1d_gauss_6	82
42.8	int_1d_gauss_lobatto	82
42.9	int_1d_gauss_n	82
42.10	int_1d_nc_2	82
42.11	int_1d_nc_3	82
42.12	int_1d_nc_4	82
42.13	int_1d_nc_5	82
42.14	int_1d_nc_6	82
42.15	int_1d_nc_7	82

42.16	int_1d_nc_7_hardy	83
42.17	int_2d_gauss_1	83
42.18	int_2d_gauss_12	83
42.19	int_2d_gauss_13	83
42.20	int_2d_gauss_16	83
42.21	int_2d_gauss_19	83
42.22	int_2d_gauss_25	83
42.23	int_2d_gauss_3	83
42.24	int_2d_gauss_33	83
42.25	int_2d_gauss_4	83
42.26	int_2d_gauss_6	84
42.27	int_2d_gauss_7	84
42.28	int_2d_gauss_9	84
42.29	int_2d_nc_10	84
42.30	int_2d_nc_15	84
42.31	int_2d_nc_21	84
42.32	int_2d_nc_3	84
42.33	int_2d_nc_6	84
42.34	int_3d_gauss_1	84
42.35	int_3d_gauss_11	84
42.36	int_3d_gauss_14	85
42.37	int_3d_gauss_15	85
42.38	int_3d_gauss_24	85
42.39	int_3d_gauss_4	85
42.40	int_3d_gauss_45	85
42.41	int_3d_gauss_5	85
42.42	int_3d_nc_11	85
42.43	int_3d_nc_4	85
42.44	int_3d_nc_6	85
42.45	int_3d_nc_8	85
43	numerical-methods/finite-element	86
43.1	interpolation_matrix	86
43.2	mark	86
43.3	mark_1d	86
43.4	mesh_1d_uniform	86
43.5	mesh_3d_uniform	86
43.6	mesh_interpolate	86
43.7	neighbour_1d	86
43.8	old	86
43.9	pdeeig_1d	86
43.10	pdeeig_2d	87
43.11	pdeeig_3d	87
43.12	polynomial_derivative_1d	87

43.13	potential_const	87
43.14	potential_coulomb	87
43.15	potential_harmonic_oscillator	87
43.16	project_circle	87
43.17	project_rectangle	87
43.18	promote_1d_2_3	87
43.19	promote_1d_2_4	87
43.20	promote_1d_2_5	88
43.21	promote_1d_2_6	88
43.22	quadrilaterate	88
43.23	recalculate_regularity_2d	88
43.24	refine_1d	88
43.25	refine_2d_21	88
43.26	refine_2d_structural	88
43.27	regularity_1d	88
43.28	regularity_2d	88
43.29	regularity_3d	89
43.30	relocate_2d	89
43.31	test_circmesh	89
43.32	test_hermite	89
43.33	tri_assign_points	89
43.34	triangulation_uniform	89
43.35	vander_1d	89
43.36	vanderd_1d	89
43.37	vanderi_1d	89
44	numerical-methods/finite-volume/@Advection	90
44.1	Advection	90
44.2	dot_advection	90
45	numerical-methods/finite-volume/@Burgers	90
45.1	burgers_split	90
45.2	dot_burgers_fdm	90
45.3	dot_burgers_fft	90
46	numerical-methods/finite-volume/@Finite_Volume	90
46.1	Finite_Volume	90
46.2	apply_bc	91
46.3	solve	91
46.4	step_split_strang	91
46.5	step_unsplit	91
47	numerical-methods/finite-volume/@Flux_Limiter	91
47.1	Flux_Limiter	91

47.2	beam_warming	91
47.3	fromm	92
47.4	lax_wendroff	92
47.5	minmod	92
47.6	monotized_central	92
47.7	muscl	92
47.8	superbee	92
47.9	upwind	92
47.10	vanLeer	93
48	numerical-methods/finite-volume/@KDV	93
48.1	dot_kdv_fdm	93
48.2	dot_kdv_fft	93
48.3	kdv_split	93
49	numerical-methods/finite-volume/@Reconstruct_Average_Evolve	93
49.1	Reconstruct_Average_Evolve	93
49.2	advect_highres	94
49.3	advect_lowress	94
50	numerical-methods/finite-volume	94
50.1	Godunov	94
50.2	Lax_Friedrich	94
50.3	Measure	94
50.4	Roe	94
50.5	fv_swe	95
50.6	staggered_euler	95
50.7	staggered_grid	95
51	numerical-methods	95
51.1	grid2quad	95
52	numerical-methods/integration	95
52.1	cumintL	95
52.2	cumintR	95
52.3	int_trapezoidal	95
53	numerical-methods/interpolation/@Kriging	96
53.1	Kriging	96
53.2	estimate_semivariance	96
53.3	interpolate_	96
54	numerical-methods/interpolation/@RegularizedInterpolator1	96
54.1	RegularizedInterpolator1	96
54.2	init	96

55	numerical-methods/interpolation/@RegularizedInterpolator2	97
55.1	RegularizedInterpolator2	97
55.2	init	97
56	numerical-methods/interpolation/@RegularizedInterpolator3	97
56.1	RegularizedInterpolator3	97
56.2	init	97
57	numerical-methods/interpolation	97
57.1	IDW	97
57.2	IPoly	97
57.3	IRBM	98
57.4	ISparse	98
57.5	Inn	98
57.6	Interpolator	98
57.7	fixnan	98
57.8	idw1	98
57.9	idw2	98
57.10	inner2outer	99
57.11	inner2outer2	99
57.12	interp1_circular	99
57.13	interp1_limited	99
57.14	interp1_man	99
57.15	interp1_piecewise_linear	99
57.16	interp1_save	99
57.17	interp1_slope	100
57.18	interp1_smooth	100
57.19	interp1_unique	100
57.20	interp2_man	100
57.21	interp_angle	100
57.22	interp_fourier	100
57.23	interp_fourier_batch	100
57.24	interp_sn	101
57.25	interp_sn2	101
57.26	interp_sn3	101
57.27	interp_sn_	101
57.28	limit_by_distance_1d	101
57.29	resample1	101
57.30	resample_d_min	101
57.31	resample_vector	102
57.32	test_interp1_limited	102
58	numerical-methods	102
58.1	inverse_complex	102

58.2	maccormack_step	102
58.3	minmod	102
59	numerical-methods/multigrid	102
59.1	mg_interpolate	102
59.2	mg_restrict	102
60	numerical-methods/ode/@BVPS_Characteristic	102
60.1	BVPS_Characteristic	102
60.2	assemble1_A	103
60.3	assemble1_A_Q	103
60.4	assemble2_A	103
60.5	assemble_AA	103
60.6	assemble_AAA	103
60.7	assemble_Ic	103
60.8	bvp1c	103
60.9	check_arguments	103
60.10	couple_junctions	104
60.11	derivative	104
60.12	init	104
60.13	inner2outer_bvp2c	104
60.14	reconstruct	104
60.15	resample	104
60.16	solve	104
60.17	test_assemble1_A	105
60.18	test_assemble2_A	105
61	numerical-methods/ode/@Time_Stepper	105
61.1	Time_Stepper	105
61.2	solve	105
62	numerical-methods/ode	105
62.1	bvp2fdm	105
62.2	bvp2wavetrain	106
62.3	bvp2wavetwopass	106
62.4	ivp_euler_forward	106
62.5	ivp_euler_forward2	106
62.6	ivprk2	106
62.7	ode2_matrix	106
62.8	ode2characteristic	106
62.9	step_trapezoidal	107
62.10	test_bvp2	107
63	numerical-methods/optimisation	107

63.1	aitken_iteration	107
63.2	anderson_iteration	107
63.3	armijo_stopping_criterion	107
63.4	astar	107
63.5	binsearch	107
63.6	bisection	107
63.7	box1	108
63.8	box2	108
63.9	cauchy	108
63.10	cauchy2	108
63.11	directional_derivative	108
63.12	dud	108
63.13	extreme3	108
63.14	extreme_quadratic	109
63.15	ftest	109
63.16	fzero_bisect	109
63.17	fzero_newton	109
63.18	grad	109
63.19	hessian	109
63.20	hessian_from_gradient	110
63.21	hessian_projected	110
63.22	line_search	110
63.23	line_search2	110
63.24	line_search_polynomial	110
63.25	line_search_polynomial2	111
63.26	line_search_quadratic	111
63.27	line_search_quadratic2	111
63.28	line_search_wolfe	111
63.29	ls_bgfs	111
63.30	ls_broyden	112
63.31	ls_generalized_secant	112
63.32	nlgc	112
63.33	nlls	112
63.34	picard	112
63.35	poly_extrema	113
63.36	quadratic_function	113
63.37	quadratic_programming	113
63.38	quadratic_step	113
63.39	rosenbrock	113
63.40	sqrt_heron	113
63.41	test_directional_derivative	113
63.42	test_dud	113
63.43	test_fzero_newton	113
63.44	test_line_search_quadratic2	114

63.45	test_ls_generalized_secant	114
63.46	test_nlcg_6_order	114
63.47	test_nlls	114
64	numerical-methods/pde	114
64.1	laplacian2d_fundamental_solution	114
65	numerical-methods/piecewise-polynomials	114
65.1	Hermite1	114
65.2	hp2_fit	114
65.3	hp2_predict	115
65.4	hp_predict	115
65.5	hp_regress	115
65.6	lp_count	115
65.7	lp_predict	115
65.8	lp_regress	115
65.9	lp_regress_	115
66	numerical-methods	115
66.1	test_adams_bashforth	115
67	mathematics	116
67.1	oversampleNZ	116
68	regression/@PolyOLS	116
68.1	PolyOLS	116
68.2	coefftest	116
68.3	detrend	116
68.4	fit	116
68.5	fit_	116
68.6	predict	116
68.7	predict_	117
68.8	slope	117
69	regression/@PowerLS	117
69.1	PowerLS	117
69.2	fit	117
69.3	predict	117
69.4	predict_	117
70	regression/@Theil	117
70.1	Theil	117
70.2	detrend	118
70.3	fit	118
70.4	predict	118

70.5	slope	118
71	regression	118
71.1	Theil_Multivariate	118
71.2	areg	118
71.3	ginireg	119
71.4	hesssimplereg	119
71.5	l1lin	119
71.6	lsq_sparam	119
71.7	polyfitd	119
71.8	regression_method_of_moments	119
71.9	robustlinreg	120
71.10	theil2	120
71.11	theil_generalised	120
71.12	total_least_squares	120
71.13	weighted_median_regression	120
72	set-theory	120
72.1	issubset	120
73	mathematics	121
73.1	shuffle_index	121
74	signal-processing	121
74.1	asymwin	121
75	signal-processing/autocorrelation	121
75.1	acf_radial	121
75.2	acfar1	121
75.3	acfar1_2	121
75.4	acfar2	121
75.5	acfar2_2	122
75.6	ar1_cutoff_frequency	122
75.7	ar1_effective_sample_size	122
75.8	ar1_mse_mu_single_sample	122
75.9	ar1_mse_pop	122
75.10	ar1_mse_range	122
75.11	ar1_spectrum	122
75.12	ar1_to_tikhonov	122
75.13	ar1_var_factor	123
75.14	ar1_var_factor_	123
75.15	ar1_var_range2	123
75.16	ar1delay	123
75.17	ar1delay_old	123

75.18	ar2_acf2c	123
75.19	ar2conv	124
75.20	ar2dof	124
75.21	ar2param	124
75.22	autocorr2	124
75.23	autocorr_bandpass	124
75.24	autocorr_brownian_phase	124
75.25	autocorr_brownian_phase_2d	124
75.26	autocorr_brownian_phase_across	124
75.27	autocorr_decay_rate	125
75.28	autocorr_effective_sample_size	125
75.29	autocorr_fft	125
75.30	autocorr_forest	125
75.31	autocorr_genton	125
75.32	autocorr_highpass	125
75.33	autocorr_lowpass	125
75.34	autocorr_periodic_additive_noise	125
75.35	autocorr_periodic_windowed	125
75.36	autocorr_radial	126
76 signal-processing		126
76.1	average_wave_shape	126
76.2	bandpass	126
76.3	bartlett	126
76.4	bin1d	126
76.5	bin2d	126
76.6	binormrnd	127
76.7	coherence	127
76.8	conv1_man	127
76.9	conv2_man	127
76.10	conv2z	127
76.11	conv30	127
76.12	conv_	127
76.13	conv_centered	127
76.14	convz	128
76.15	cosexpdelay	128
76.16	csmooth	128
76.17	daniell_window	128
76.18	db2neper	128
76.19	db2power	128
76.20	derive_bandpass_continuous_scale	128
76.21	derive_danielle_weight	128
76.22	derive_limit_0_acfar	129
76.23	detect_peak	129

76.24	determine_phase_shift	129
76.25	determine_phase_shift1	129
76.26	doublesum_ij	129
76.27	effective_sample_size_to_ar1	129
76.28	fcut2Lw_gausswin	129
76.29	fcut_gausswin	129
76.30	filt_hodges_lehman	129
77	signal-processing/filters	130
77.1	circfilt2	130
77.2	filter1	130
77.3	filter2	130
77.4	filter_	130
77.5	filter_r_to_f0	130
77.6	filter_rho_to_f0	130
77.7	filter_twosided	130
77.8	filteriir	131
77.9	filterp	131
77.10	filterp1	131
77.11	filterstd	132
77.12	gaussfilt2	132
77.13	lowpass_discrete	132
77.14	meanfilt2	132
77.15	medfilt1_man	132
77.16	medfilt1_man2	132
77.17	medfilt1_padded	132
77.18	medfilt1_reduced	133
77.19	trifilt1	133
77.20	trifilt2	133
78	signal-processing	133
78.1	firls_man	133
78.2	fit_spectral_density	133
78.3	fit_spectral_density_2d	133
78.4	fit_spectral_density_radial	133
78.5	flattopwin	134
78.6	frequency_response_boxcar	134
78.7	freqz_boxcar	134
78.8	gaussfilt1	134
78.9	hanchangewin	134
78.10	hanchangewin2	134
78.11	hanwin	134
78.12	hanwin_	134
78.13	high_pass_1d_simple	135

78.14	kaiserwin	135
78.15	kalman	135
78.16	lanczoswin	135
78.17	last	135
78.18	maxfilt1	135
78.19	meanfilt1	135
78.20	mid_term_single_sample	135
78.21	minfilt1	135
78.22	minmax	136
78.23	mu2ar1	136
78.24	mysmooth	136
78.25	nanautocorr	136
78.26	nanmedfilt1	136
78.27	neper2db	136
78.28	oscillator_noisy	136
79 signal-processing/passes		136
79.1	bandpass1d	136
79.2	bandpass1d_fft	137
79.3	bandpass1d_implicit	137
79.4	bandpass2	137
79.5	bandpass2d	137
79.6	bandpass2d_2	137
79.7	bandpass2d_fft	137
79.8	bandpass2d_ideal	137
79.9	bandpass2d_implicit	137
79.10	bandpass2d_iso	137
79.11	bandpass_arg	138
79.12	bandpass_f0_to_rho	138
79.13	bandpass_max	138
79.14	bandpass_max2	138
79.15	highpass	138
79.16	highpass1d_fft_cos	138
79.17	highpass1d_implicit	138
79.18	highpass2d_fft	138
79.19	highpass2d_ideal	139
79.20	highpass2d_implicit	139
79.21	highpass_arg	139
79.22	highpass_fc_to_rho	139
79.23	lowpass	139
79.24	lowpass1d_fft	139
79.25	lowpass1d_implicit	139
79.26	lowpass2	139
79.27	lowpass2d_2	139

79.28	lowpass2d_anisotropic	140
79.29	lowpass2d_fft	140
79.30	lowpass2d_ideal	140
79.31	lowpass2d_implicit	140
79.32	lowpass_arg	140
79.33	lowpass_fc_to_rho	140
79.34	lowpass_iir	140
79.35	lowpass_iir_symmetric	140
79.36	lowpassfilter2	140
80	signal-processing	141
80.1	peaks_man	141
81	signal-processing/periodogram	141
81.1	periodogram	141
81.2	periodogram_2d	141
81.3	periodogram_annular	141
81.4	periodogram_bartlett	141
81.5	periodogram_bootstrap	141
81.6	periodogram_confidence_interval	141
81.7	periodogram_filter	141
81.8	periodogram_median	142
81.9	periodogram_p_value	142
81.10	periodogram_qq	142
81.11	periodogram_quantiles	142
81.12	periodogram_radial	142
81.13	periodogram_std	142
81.14	periodogram_test_periodicity	142
81.15	periodogram_test_periodicity_2d	143
81.16	periodogram_test_stationarity	144
81.17	periodogram_welsh	144
82	signal-processing	144
82.1	polyfilt1	144
82.2	qmedfilt1	144
82.3	quadratfilt1	144
82.4	quadratwin	144
82.5	randar1	144
82.6	randar1_dual	145
82.7	randar2	145
82.8	randarp	145
82.9	rectwin	145
82.10	recursive_sum	145
82.11	select_range	145

82.12	smooth1d_parametric	145
82.13	smooth2	145
82.14	smooth_man	146
82.15	smooth_parametric	146
82.16	smooth_parametric2	146
82.17	smooth_with_splines	146
82.18	smoothfft	146
83	signal-processing/spectral-density	146
83.1	spectral_density_ar2	146
83.2	spectral_density_area	146
83.3	spectral_density_bandpass2d_ideal	146
83.4	spectral_density_bandpass_2d	147
83.5	spectral_density_bandpass_2d_max2par	147
83.6	spectral_density_bandpass_2d_scale	147
83.7	spectral_density_bandpass_2d_scale_old	147
83.8	spectral_density_bandpass_continuous	147
83.9	spectral_density_bandpass_continuous_max	147
83.10	spectral_density_bandpass_continuous_max2par	148
83.11	spectral_density_bandpass_continuous_scale	148
83.12	spectral_density_bandpass_discrete	148
83.13	spectral_density_brownian_phase	148
83.14	spectral_density_brownian_phase_2d	148
83.15	spectral_density_brownian_phase_across	148
83.16	spectral_density_brownian_phase_across_max	148
83.17	spectral_density_brownian_phase_across_max2par	148
83.18	spectral_density_brownian_phase_across_mode2par	149
83.19	spectral_density_brownian_phase_mode	149
83.20	spectral_density_brownian_phase_mode2par	149
83.21	spectral_density_brownian_phase_scale	149
83.22	spectral_density_estimate_2d	149
83.23	spectral_density_flat	149
83.24	spectral_density_forest	149
83.25	spectral_density_gausswin	149
83.26	spectral_density_highpass	149
83.27	spectral_density_highpass2d_ideal	150
83.28	spectral_density_highpass_2d	150
83.29	spectral_density_highpass_cos	150
83.30	spectral_density_lorentzian	150
83.31	spectral_density_lorentzian_max	150
83.32	spectral_density_lorentzian_max2par	150
83.33	spectral_density_lorentzian_scale	150
83.34	spectral_density_lowpass2d_ideal	150
83.35	spectral_density_lowpass_2d	150

83.36	spectral_density_lowpass_continuous	151
83.37	spectral_density_lowpass_continuous_scale	151
83.38	spectral_density_lowpass_discrete	151
83.39	spectral_density_lowpass_one_sided	151
83.40	spectral_density_maximum_bias_corrected	151
83.41	spectral_density_periodic_additive_noise	151
83.42	spectral_density_rectwin	151
83.43	spectral_density_wperiodic	151
84	signal-processing	151
84.1	spectral_density_brownian_phase_reg2par	151
84.2	spectrogram	152
84.3	sum_i_lag	152
84.4	sum_ii	152
84.5	sum_ii_	152
84.6	sum_ij	152
84.7	sum_ij_	152
84.8	sum_ij_partial_	152
84.9	sum_multivar	152
84.10	test_acfar1	153
84.11	tikhonov_to_ar1	153
84.12	trapwin	153
84.13	triwin	153
84.14	triwin2	153
84.15	tukeywin_man	153
84.16	varar1	153
84.17	welch_spectrogram	153
84.18	wfilt	154
84.19	winbandpass	154
85	signal-processing/windows	154
85.1	circwin	154
85.2	danielle_window	154
85.3	gausswin	154
85.4	gausswin2	154
85.5	radial_window	154
85.6	range_window	154
85.7	rectwin_cutoff_frequency	155
85.8	std_window	155
85.9	window2d	155
85.10	window_make_odd	155
86	signal-processing	155
86.1	winfilt0	155

86.2	winlength	155
86.3	wmeanfilt	155
86.4	wmedfilt	156
86.5	wordfilt	156
86.6	wordfilt_edgeworth	156
86.7	wrapphase	156
86.8	xar1	156
86.9	xcorr_man	156
87	sorting	156
87.1	sort2	156
87.2	sort2d	156
88	spatial-pattern-analysis/@Spatial_Pattern	157
88.1	Spatial_Pattern	157
88.2	analyze_grid	157
88.3	analyze_transect	157
88.4	extract_improfile	157
88.5	imread	157
88.6	plot	157
88.7	tabulate	157
89	spatial-pattern-analysis	157
89.1	analyze_grid2	157
89.2	approximate_ratio_distribution	157
89.3	banded_pattern	158
89.4	brownian_phase_patch_size_distribution	158
89.5	hexagonal_pattern	158
89.6	isisotropic	158
89.7	patch_size_1d	158
89.8	patch_size_2d	158
89.9	separate_isotropic_from_anisotropic_density	158
89.10	suppress_low_frequency_lobe	158
90	special-functions	158
90.1	bessel_sphere	158
90.2	beta_man	159
90.3	betainc_man	159
90.4	digamma_man	159
90.5	exp10	159
90.6	hankel_sphere	159
90.7	hermite	159
90.8	legendre_man	159
90.9	neumann_sphere	160

91	statistics	160
91.1	atan_s2	160
91.2	beta_kurt	160
91.3	beta_mean	160
91.4	beta_moment2param	160
91.5	beta_skew	160
91.6	beta_std	160
91.7	chi2_kurt	160
91.8	chi2_mean	161
91.9	chi2_skew	161
91.10	chi2_std	161
91.11	coefficient_of_determination	161
91.12	conditional_expectation_normal	161
91.13	correlation_confidence_pearson	161
92	statistics/distributions	161
92.1	PDF	161
92.2	binorm_separation_coefficient	161
92.3	binormcdf	162
92.4	binormfit	162
92.5	binormpdf	162
92.6	edgeworth_cdf	162
92.7	edgeworth_pdf	162
92.8	gam_moment2param	162
92.9	logn_mean	162
92.10	logn_mode	163
92.11	logn_mode2param	163
92.12	logn_moment2param	163
92.13	logn_param2moment	163
92.14	logn_std	163
92.15	lognpdf_	163
92.16	pdfsample	163
92.17	t2cdf	163
92.18	t2inv	164
93	statistics	164
93.1	error_propagation_fraction	164
93.2	error_propagation_product	164
93.3	example_standard_error_of_sample_quantiles	164
93.4	f_var_finite	164
93.5	fisher_mean	164
93.6	fisher_moment2param	164
93.7	fisher_std	164
93.8	gam_mean	164

93.9	gam_std	165
93.10	gamma_mode_to_parameter	165
93.11	gamma_stirling	165
93.12	gaussfit3	165
93.13	gaussfit_quantile	165
93.14	geoserr	165
93.15	geostd	165
93.16	hodges_lehmann_correlation	165
93.17	hodges_lehmann_dispersion	166
94	statistics/information-theory	166
94.1	akaike_information_criterion	166
94.2	bayesian_information_criterion	166
95	statistics	166
95.1	jackknife_block	166
95.2	kurtncdf	166
95.3	kurtnpdf	167
95.4	kurtosis_bias_corrected	167
95.5	limit	167
95.6	logfactorial	167
95.7	loglogpdf	167
95.8	lognfit_quantile	167
95.9	logskewcdf	167
95.10	logskewpdf	167
96	statistics/logu	167
96.1	lambertw_numeric	167
96.2	logtrialtcdf	168
96.3	logtrialtinv	168
96.4	logtrialtmean	168
96.5	logtrialtpdf	168
96.6	logtrialtrnd	168
96.7	logtricdf	168
96.8	logtriinv	169
96.9	logtrimean	169
96.10	logtripdf	169
96.11	logtrirnd	169
96.12	logucdf	169
96.13	logucm	169
96.14	loguinv	169
96.15	logumean	169
96.16	logupdf	170
96.17	logurnd	170

96.18	loguvar	170
96.19	medlogu	170
96.20	test_logurnd	170
96.21	tricdf	170
96.22	triinv	170
96.23	trimedian	170
96.24	tripdf	171
96.25	trirnd	171
97	statistics	171
97.1	max_exprnd	171
97.2	maxnnormals	171
97.3	mean_angle	171
97.4	mean_generalized_gampdf	171
97.5	midrange	171
97.6	minavg	172
97.7	mode_man	172
98	statistics/moment-statistics	172
98.1	autocorr_man3	172
98.2	autocorr_man4	172
98.3	autocorr_man5	172
98.4	blockserr	172
98.5	comoment	173
98.6	corr_man	173
98.7	cov_man	173
98.8	dof	173
98.9	edgeworth_quantile	173
98.10	effective_sample_size	173
98.11	f_correlation	174
98.12	f_finite	174
98.13	lmean	174
98.14	lmoment	174
98.15	maskmean	174
98.16	masknanmean	174
98.17	mean1	174
98.18	mean_man	174
98.19	mse	175
98.20	nanautocorr_man1	175
98.21	nanautocorr_man2	175
98.22	nanautocorr_man4	175
98.23	nancorr	175
98.24	nancumsum	175
98.25	nanlmean	175

98.26	nanr2	176
98.27	nanrms	176
98.28	nanrmse	176
98.29	nanserr	176
98.30	nanwmean	176
98.31	nanwstd	176
98.32	nanwvar	176
98.33	nanxcorr	177
98.34	pearson	177
98.35	pearson_to_kendall	177
98.36	pool_samples	177
98.37	qmean	177
98.38	range_mean	177
98.39	rmse_	177
98.40	serr	177
98.41	serr1	178
98.42	test_kskew	178
98.43	test_qstd_kskew_optimal_p	178
98.44	wautocorr	178
98.45	wcorr	178
98.46	wcov	178
98.47	wdof	178
98.48	wkurt	179
98.49	wmean	179
98.50	wrms	179
98.51	wserr	179
98.52	wskew	179
98.53	wstd	179
98.54	wvar	179
99	statistics	180
99.1	nangeomean	180
99.2	nangeostd	180
100	statistics/nonparametric-statistics	180
100.1	kernel1d	180
100.2	kernel2d	180
101	statistics	180
101.1	normmoment	180
101.2	normpdf2	181
102	statistics/order-statistics	181
102.1	hodges_lehmann_location	181

102.2	kendall	181
102.3	kendall_to_pearson	181
102.4	mad2sd	181
102.5	madcorr	181
102.6	median2_holder	182
102.7	median_ci	182
102.8	median_man	182
102.9	mediani	182
102.10	nanmadcorr	182
102.11	nanwmedian	182
102.12	nanwquantile	182
102.13	oja_median	183
102.14	qkurtosis	183
102.15	qmoments	183
102.16	qskew	183
102.17	qskewq	183
102.18	qstdq	184
102.19	quantile1_optimisation	184
102.20	quantile2_breckling	184
102.21	quantile2_chaudhuri	184
102.22	quantile2_projected	184
102.23	quantile2_projected2	184
102.24	quantile_envelope	184
102.25	quantile_regression_simple	184
102.26	ranking	185
102.27	spatial_median	185
102.28	spatial_quantile	185
102.29	spatial_quantile2	185
102.30	spatial_quantile3	185
102.31	spatial_rank	185
102.32	spatial_sign	185
102.33	spatial_signed_rank	185
102.34	spearman	186
102.35	spearman_rank	186
102.36	spearman_to_pearson	186
102.37	wmedian	186
102.38	wquantile	186
103	statistics	186
103.1	qstd	186
103.2	quantile_extrap	186
103.3	quantile_sin	186
104	statistics/random-number-generation	187

104.1	laplacernd	187
104.2	randc	187
104.3	skewness2param	187
104.4	skewpdf_central_moments	187
104.5	skewrnd	187
105	statistics	187
105.1	range	187
105.2	resample_with_replacement	187
106	statistics/resampling-statistics/@Jackknife	188
106.1	Jackknife	188
106.2	estimated_STATIC	188
106.3	matrix1_STATIC	188
106.4	matrix2	188
107	statistics/resampling-statistics	189
107.1	block_jackknife	189
107.2	jackknife_moments	189
107.3	moving_block_jackknife	189
107.4	randblockserr	189
107.5	resample	189
108	statistics	190
108.1	scale_quantile_sd	190
108.2	sd_sample_quantiles	190
108.3	skew_generalized_normal_fit	190
108.4	skew_generalized_normpdf	190
108.5	skewcdf	190
108.6	skewparam_to_central_moments	190
108.7	skewpdf	190
108.8	spatialrnd	191
108.9	test_mean_generalized_gampdf	191
108.10	test_skew_generalized_normpdf	191
108.11	trimmed_mean	191
108.12	ttest2_man	191
108.13	ttest_man	191
108.14	ttest_paired	191
108.15	uniformnpdf	192
108.16	wgeomean	192
108.17	wgeovar	192
108.18	wharmean	192
108.19	wharstd	192
108.20	wharvar	192

108.21	wnormpdf	192
109	stochastic	192
109.1	brownian_drift_hitting_probability	192
109.2	brownian_drift_hitting_probability2	193
109.3	brownian_field	193
109.4	brownian_motion_1d_acf	193
109.5	brownian_motion_1d_cov	193
109.6	brownian_motion_1d_fft	193
109.7	brownian_motion_1d_fourier	193
109.8	brownian_motion_1d_interleave	193
109.9	brownian_motion_1d_laplacian	193
109.10	brownian_motion_2d_cov	193
109.11	brownian_motion_2d_fft	194
109.12	brownian_motion_2d_fft_old	194
109.13	brownian_motion_2d_fourier	194
109.14	brownian_motion_2d_interleave	194
109.15	brownian_motion_2d_interleaving	194
109.16	brownian_motion_2d_kahunen	194
109.17	brownian_motion_2d_laplacian	194
109.18	brownian_motion_with_drift_hitting_probability	194
110	mathematics	194
110.1	ternary_diagram	194
111	test/master	195
111.1	dat_test_lanczos_3d_k_20_n_40	195
111.2	poisson2d_blk	195
111.3	qr_implicit_givens_2	195
111.4	spectral_derivative_2d	195
111.5	test_2d_eigensolver_hydrogen	195
111.6	test_2d_refine	195
111.7	test_3d_eigensolver_hydrogen	195
111.8	test_FEM	195
111.9	test_Mesh_3d	195
111.10	test_arnoldi	196
111.11	test_arpackc	196
111.12	test_assemble	196
111.13	test_assembly_performance	196
111.14	test_bc_one_sided	196
111.15	test_compare_solvers	196
111.16	test_complete	196
111.17	test_convergence	196
111.18	test_convergence_b	196

111.19	test_df_2d	196
111.20	test_eig_algs	197
111.21	test_eig_inverse	197
111.22	test_eigs_lanczos	197
111.23	test_eigs_lanczos_1	197
111.24	test_eigs_lanczos_2	197
111.25	test_eigs_lanczos_performance	197
111.26	test_fdm	197
111.27	test_fdm_d_vargrid	197
111.28	test_fdm_spectral	197
111.29	test_fem	197
111.30	test_fem_1d	198
111.31	test_fem_1d_higher_order	198
111.32	test_fem_2d_adaptive	198
111.33	test_fem_2d_higher_order	198
111.34	test_fem_3d_higher_order	198
111.35	test_fem_3d_refine	198
111.36	test_fem_b	198
111.37	test_fem_derivative	198
111.38	test_fem_quadrature	198
111.39	test_final	198
111.40	test_fix_substitution	199
111.41	test_forward	199
111.42	test_get_sparse_arrays	199
111.43	test_harmonic_oscillator	199
111.44	test_high_order_fdm_periodic_bc	199
111.45	test_hydrogen_wf	199
111.46	test_ichol	199
111.47	test_interpolation	199
111.48	test_inverse_problem	199
111.49	test_it_vs_exact	199
111.50	test_jama	200
111.51	test_jd	200
111.52	test_jdqz	200
111.53	test_lanczos_2	200
111.54	test_lanczos_biorthogonal	200
111.55	test_laplacian	200
111.56	test_laplacian_non_uniform	200
111.57	test_laplacian_simple	200
111.58	test_mesh_2d_uniform	200
111.59	test_mesh_2d_uniform_2	200
111.60	test_mesh_circle	201
111.61	test_mesh_generation	201
111.62	test_mesh_interpolate	201

111.63	test_mg	201
111.64	test_minres_recycle	201
111.65	test_multigrid	201
111.66	test_nc	201
111.67	test_nonuniform_symmetric	201
111.68	test_pde	201
111.69	test_permutation	201
111.70	test_poison_fem	202
111.71	test_polar	202
111.72	test_potential	202
111.73	test_powers	202
111.74	test_precondition	202
111.75	test_project_rectangle	202
111.76	test_qr	202
111.77	test_quantum_well	202
111.78	test_radial_adaptive	202
111.79	test_radial_confinement	202
111.80	test_radial_fixes	203
111.81	test_refine_2d	203
111.82	test_refine_2d.b	203
111.83	test_refine_3d	203
111.84	test_refine_structural	203
111.85	test_regularisation	203
111.86	test_round_off	203
111.87	test_schrödinger_potentials	203
111.88	test_uniform_mesh	203
111.89	test_vargrid	203
112test/signal-processing/autocorrelation		204
112.1	test_acf	204
112.2	test_acf_bias	204
112.3	test_acf_brownian_phase	204
112.4	test_acfar1_2	204
112.5	test_acfar1_3	204
112.6	test_acfar1_4	204
112.7	test_acfar2	204
112.8	test_ar1_var_factor	204
112.9	test_ar1_var_factor_2	204
112.10	test_ar1_var_mu_single_sample	205
112.11	test_ar1_var_pop	205
112.12	test_ar1_var_pop_1	205
112.13	test_ar1delay	205
112.14	test_ar2	205

113	test/signal-processing/passes	205
113.1	test_bandpass2d_ideal	205
113.2	test_lowpass1d_fft	205
113.3	test_lowpass1d_implicit	205
113.4	test_lowpass2d_anisotropic	205
113.5	test_lowpass2d_fft	206
113.6	test_lowpass2d_rho	206
114	test/signal-processing/periodogram	206
114.1	test_periodicity_test_2d	206
114.2	test_periodogram_bartlett_se	206
114.3	test_periodogram_gauss	206
114.4	test_periodogram_radial	206
114.5	test_periodogram_test	206
114.6	test_periodogram_test_periodicity_2d	206
114.7	test_periogogram_significance	206
115	test/signal-processing/spectral-density	207
115.1	test_spectral_density_2	207
115.2	test_spectral_density_bandpass_2d	207
115.3	test_spectral_density_bandpass_2d_max2par	207
115.4	test_spectral_density_bandpass_continuous	207
115.5	test_spectral_density_bandpass_continuous_1	207
115.6	test_spectral_density_bandpass_maximum	207
115.7	test_spectral_density_bandpass_scale	207
115.8	test_spectral_density_bp	207
115.9	test_spectral_density_bp_2d	207
115.10	test_spectral_density_bp_approx	208
115.11	test_spectral_density_brownian_phase	208
115.12	test_spectral_density_brownian_phase_2d	208
115.13	test_spectral_density_brownian_phase_across	208
115.14	test_spectral_density_brownian_phase_across_mode2par	208
115.15	test_spectral_density_brownian_phase_mode	208
115.16	test_spectral_density_brownian_phase_mode2par	208
115.17	test_spectral_density_brownian_phase_scale	208
115.18	test_spectral_density_flat	208
115.19	test_spectral_density_hp_cos	208
115.20	test_spectral_density_lorentzian_max	209
115.21	test_spectral_density_lorentzian_scale	209
115.22	test_spectral_density_lowpass	209
115.23	test_spectral_density_lowpass_continuous	209
115.24	test_spectral_density_lowpass_continuous_1	209
115.25	test_spectral_density_maxiumum_bias_corrected	209

116	test/spatial-pattern-analysis	209
116.1	test_approximate_ratio_distribution	209
116.2	test_approximate_ratio_quantile	209
116.3	test_spearate_isotropic_density	209
117	test/statistics/moment-statistics	210
117.1	test_wmean	210
118	test/statistics	210
118.1	test_fisher_moment2param	210
119	test/stochastics	210
119.1	test_brownian_surface	210
120	test	210
120.1	test_S	210
120.2	test_advect_analytic	210
120.3	test_asymbp	210
120.4	test_bandpass2d	210
120.5	test_bandwidth	210
120.6	test_bartlett_angle	211
120.7	test_bartlett_distribution	211
120.8	test_bartlett_expansion	211
120.9	test_beta	211
120.10	test_betainc	211
120.11	test_bivariate_covariance_term	211
120.12	test_brownian_drift_hitting_probability	211
120.13	test_brownian_drift_hitting_probability2	211
120.14	test_brownian_motion_1d	211
120.15	test_brownian_motion_2d_cov	211
120.16	test_brownian_motion_2d_fft	212
120.17	test_brownian_noise_1d	212
120.18	test_brownian_noise_2d	212
120.19	test_brownian_noise_interleave	212
120.20	test_coherence	212
120.21	test_combined_spectral_density	212
120.22	test_continuous_fourier_transform	212
120.23	test_convexity	212
120.24	test_d2	212
120.25	test_determine_phase_shift	212
120.26	test_diffuse_analytic	213
120.27	test_diffusion_matrix	213
120.28	test_ellipse	213
120.29	test_error_propagation_fraction	213

120.30	test_f	213
120.31	test_f2	213
120.32	test_fit_2d_spectral_density	213
120.33	test_fourier	213
120.34	test_fourier_derivative	213
120.35	test_fourier_derivative_1	213
120.36	test_fourier_integral	214
120.37	test_fourier_mask_covariance_matrix	214
120.38	test_ft_bp	214
120.39	test_gam	214
120.40	test_gamma_distribution	214
120.41	test_gaussfit3	214
120.42	test_gaussian_flat	214
120.43	test_geoserr	214
120.44	test_hexagonal_pattern	214
120.45	test_iafrate	214
120.46	test_implicit_ode	215
120.47	test_imrotmat	215
120.48	test_integration	215
120.49	test_ivp	215
120.50	test_jacobian	215
120.51	test_lanczoswin	215
120.52	test_laplacian_power	215
120.53	test_lognfit_quantile	215
120.54	test_ls_perpendicular_offset	215
120.55	test_madcorr	215
120.56	test_mask	216
120.57	test_max_normal	216
120.58	test_moments	216
120.59	test_moments_fourier_power	216
120.60	test_mtimes3x3	216
120.61	test_noisy_oscillator	216
120.62	test_nonperiodic_pattern	216
120.63	test_normalization	216
120.64	test_ols	216
120.65	test_parcorr	216
120.66	test_positivity_preserving	217
120.67	test_randar1	217
120.68	test_randar1_multivariate	217
120.69	test_randar2	217
120.70	test_ratio_distributions	217
120.71	test_sd_rectwin	217
120.72	test_spatialrnd	217
120.73	test_spectrum_additivity	217

120.74	test_stationarity	217
120.75	test_stationarity2	217
120.76	test_sum_ij	218
120.77	test_sum_multivar	218
120.78	test_trifilt1	218
120.79	test_wautocorr	218
120.80	test_wavelet_transform	218
120.81	test_whittle	218
120.82	test_window	218
120.83	test_wordfilt	218
120.84	test_xar1_mid_term	218
121	mathematics	219
121.1	trapezoidal_fixed	219
122	wavelet	219
122.1	contiuous_wavelet_transform	219
122.2	cwt_man	219
122.3	cwt_man2	219
122.4	example_wavelets	219
122.5	phasewrap	219
122.6	test_cwt_man	219
122.7	test_phasewrap	220
122.8	test_wavelet	220
122.9	test_wavelet2	220
122.10	test_wavelet_analysis	220
122.11	test_wavelet_reconstruct	220
122.12	test_wtc	220
122.13	wavelet	220
122.14	wavelet_reconstruct	220
122.15	wavelet_transform	220

1 calendar

1.1 days_per_month

1.2 isnight

2 mathematics

mathematical functions of various kind

2.1 cast_byte_to_integer

cast byte to integer

3 complex-analysis

operations on complex numbers

3.1 complex_exp_product_im_im

product of the imaginary part of two complex exponentials

the product has two frequency components

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

3.2 complex_exp_product_im_re

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

the product has two frequency components

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

3.3 complex_exp_product_re_im

the product has two frequency components

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

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

3.4 complex_exp_product_re_re

product of the real part of two complex exponentials

$$\begin{aligned} \text{re}(c1 \exp(io1x)) * \text{re}(c2 \exp(io2x)) = \\ 1/2 * (\text{real}(c1 * c2 * \exp(i * (n1 + n2) * o * x)) \dots \\ + \text{real}(\text{conj}(c1) * c2 * \exp(i * (n2 - n1) * o * x))) \end{aligned}$$

the product has two frequency components

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

3.5 croots

nth-roots of a complex number

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

r : roots of the complex number

3.6 root_complex

root of a complex number

3.7 test_imroots

4 derivation

derivation of several functions by means of symbolic computation

4.1 derive_acfar1

4.2 derive_ar1_spectral_density

4.3 derive_ar2param

4.4 derive_arc_length

4.5 derive_fourier_power

4.6 derive_fourier_power_exp

4.7 derive_laplacian_curvilinear

4.8 `derive_laplacian_fourier_piecewise_linear`

4.9 `derive_logtripdf`

4.10 `derive_smooth1d_parametric`

4.11 `derive_spectral_density_bandpass_initial_condition`

5 `derivation/master`

5.1 `derive_bc_one_sided`

5.2 `derive_convergence`

5.3 `derive_error_fdm`

5.4 `derive_fdm_poly`

5.5 `derive_fdm_power`

5.6 `derive_fdm_taylor`

5.7 `derive_fdm_vargrid`

5.8 `derive_fem_2d_mass`

5.9 `derive_fem_error_2d`

5.10 `derive_fem_error_3d`

5.11 `derive_fem_sym_2d`

5.12 `derive_grid_constants`

5.13 `derive_interpolation`

5.14 `derive_laplacian`

5.15 `derive_limit`

5.16 `derive_nc_1d`

5.17 `derive_nc_1d_`

5.18 `derive_nc_2d`

5.19 `derive_nonuniform_symmetric`

%

5.20 `derive_richardson`

5.21 `derive_sum`

5.22 `nn`

5.23 `test_derive`

5.24 `test_derive_fdm_poly`

5.25 `test_filter`

5.26 test_vargrid

6 derivation

derivation of several functions by means of symbolic computation

6.1 simplify_atan

symbolic simplification of the arcus tangent

7 mathematics

mathematical functions of various kind

7.1 entropy

8 finance

8.1 derive_skewrnd_walsh_paramter

8.2 gbm_cdf

8.3 gbm_fit

8.4 gbm_fit_old

8.5 `gbm_inv`

8.6 `gbm_mean`

8.7 `gbm_median`

8.8 `gbm_pdf`

8.9 `gbm_simulate`

8.10 `gbm_skewness`

8.11 `gbm_std`

8.12 `gbm_transform_time_step`

8.13 `put_price_black_scholes`

8.14 `skewgbm_simulate`

8.15 skewrnd_walsh

9 finance/test

9.1 test_gbm

9.2 test_gbm_pdf

9.3 test_skewrnd_walsh

10 fourier/@STFT

10.1 STFT

class for short time fourier transform

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

tend to oscillate in the presence of noise

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

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

10.2 itransform

inverse of the short time fourier transform

10.3 stft_

static wrapper for STFT

10.4 stftmat

transformation matrix for the short time fourier transform

10.5 transform

short time fourier transform

11 fourier

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

11.1 amplitude_from_peak

amplitude and standard deviation of the amplitude of a frequency component

represented by a peak in the fourier domain

input :

h : peak height

w : peak width at half height

output:

a : amplitude in real space

s : standard deviation of the frequency (!)

11.2 caesaro_weight

11.3 dftmtx_man

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

input :

n : number of samples

nr : number of columns

output :

F : fourier matrix

11.4 example_fourier_window

11.5 fft_man

fast fourier transform for complex input data

input:

F : data in real space

output :

F : fourier transformation of F

11.6 fftsmooth

smooth the fourier transform and determine upper and lower bound confidence intervals

input :

f :

sfunc : a smoothing function (for example fir convolution with rectangular window)

returns filtered (mean) value and normalized fir window

nf : window length

nsigma : number of standard deviations for confidence intervals

output :

ff : filtered fourier transform

l : lower bound

u : upper bound

11.7 fix_fourier

fill gaps (missing data) by means of fourier extrapolation

fix periodic data series with fourier interpolation

longest gap should not exceed 1/2 of the shortest time span of interest (1/cutoff frequency)

note: this limit equals the position of first side lobe of the ft of a rectangular window with gap length

11.8 `fourier_axis`

return axis of frequencies and periods for the discrete fourier transform
as computed by fft (matlab-style)

input:
X : sample locations (equal interval)
L : length of samples
n : number of samples

output :
f : frequencies
T : periods
mask : mask for 1/2 of the fourier transform
(as both halves are complex conjugates)
N : frequency id

11.9 `fourier_axis_2d`

11.10 `fourier_cesaro_correction`

11.11 `fourier_coefficient_piecewise_linear`

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

input :
l,r : end points of piecewise linear function
lval, rval : values at end points
L : length of domain
n : number of samples/highest frequency

output :
a, b : coefficients for frequency components

11.12 `fourier_coefficient_piecewise_linear_1`

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

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

output :
ab : coefficients for frequency components

11.13 `fourier_coefficient_ramp3`

fourier series coefficient of a ramp

11.14 `fourier_coefficient_ramp_pulse`

fourier series coefficient of a ramp pules

11.15 `fourier_coefficient_ramp_step`

fourier coefficient of a ramp-step

11.16 `fourier_coefficient_square_pulse`

fourier series coefficients of a square pulse

11.17 `fourier_complete_negative_half_plane`

11.18 `fourier_cubic_interaction_coefficients`

11.19 `fourier_derivative`

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

input:
x : data, sampled in equal intervals
k : order of the derivative

dx : kth-derivative of x

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

11.20 `fourier_derivative_matrix_1d`

11.21 `fourier_derivative_matrix_2d`

11.22 `fourier_expand`

expand values of fourier series

11.23 `fourier_fit`

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

11.24 `fourier_interpolate`

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

11.25 `fourier_matrix`

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

11.26 `fourier_matrix2`

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

11.27 `fourier_matrix3`

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

11.28 `fourier_matrix_exp`

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

11.29 `fourier_multiplicative_interaction_coefficients`

11.30 `fourier_power`

powers of a continuous fourier series in sin/cos form

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

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

11.31 `fourier_power_exp`

powers of the continuous fourier series

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

phase of first component assumed 0

higher orders than 2 ignored input

higher order than 3 not computed in output

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

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

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

11.32 `fourier_predict`

expand a continuous fourier series at times t

11.33 `fourier_quadratic_interaction_coefficients`

11.34 `fourier_random_phase_walk`

evaluate fourier series where the phase undergoes a brownian motion

11.35 `fourier_range`

approximate range of a continuous Fourier series with 2 components

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

11.36 `fourier_regress`

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

11.37 `fourier_resampled_fit`

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

11.38 `fourier_resampled_predict`

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

11.39 `fourier_signed_square`

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

11.40 `fourier_transform`

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

input:

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

output :

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

11.41 `fourier_transform_fractional`

11.42 `fourier_truncate_negative_half_plane`

11.43 `hyperbolic_fourier_box`

11.44 `idftmtx_man`

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

11.45 `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

11.46 `mean_fourier_power`

11.47 `moments_fourier_power`

11.48 nanfft

discrete fourier transform of a data series with gaps

11.49 peaks

peaks of the power spectrum of a discrete fourier transform

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

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

works best, when spectrum has been smoothened

input :

f : frequency

y : absolute value of fourier transform (power spectrum)

L : length in space or time of series

output :

a0 : amplitude

s0 : standard deviation (error?) of amplitude

w0 : width of peak

lambda = wave length (period?)

pdx : index of peak

f : frequency (if not given as input)

11.50 roots_fourier

zeros of continuous fourier series series

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

11.51 spectral_density

spectral density

11.52 std_fourier_power

11.53 test_complex_exp_product

11.54 test_fourier_filter

11.55 test_idftmtx

11.56 var_fourier_power

12 mathematics

mathematical functions of various kind

12.1 gaussfit_quantile

13 geometry/@Geometry

13.1 Geometry

13.2 arclength

arc length of a two dimensional curve

8th order accurate

does not require the segments length to vary smoothly

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

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

13.3 arclength_old

arc length of a two dimensional function

13.4 arclength_old2

arc length of a two dimensional function

13.5 base_point

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

13.6 base_point_limited

base point (Fusspunkt) of a point on a line

13.7 centroid

centroid of a polygone

13.8 cosa_min_max

13.9 cross2

cross product in two dimensions

13.10 curvature

curvature of a function in two dimensions

13.11 ddot

sum of squares of cos of inner angles of triangle

13.12 distance

euclidan distance between two points

13.13 distance2

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

13.14 dot

dot product

13.15 edge_length

edge length

13.16 enclosed_angle

angle enclosed between two lines

13.17 enclosing_triangle

smallest enclosing equilateral triangle with bottom side paralle to
X axis

13.18 hexagon

coordinates of a hexagon, scaled and rotated

13.19 inPolygon

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

13.20 inTetra

flag points contained in tetrahedron

13.21 inTetra2

flag points contained in tetrahedron

13.22 inTriangle

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

13.23 intersect

intersect between two lines

13.24 lineintersect

intersect of two lines

13.25 lineintersect1

intersect of two lines

13.26 minimum_distance_lines

minimum distance of two lines in three dimensions

13.27 mittenpunkt

mittenpunkt of a triangle

13.28 nagelpoint

nagelpoint of a triangle

13.29 onLine

13.30 orthocentre

orthocentre of triangle

13.31 plumb_line

13.32 poly_area

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

13.33 poly_edges

edges of a polygon

13.34 poly_set

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

13.35 poly_width

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

13.36 polyxpoly

intersections of two polygons

13.37 project_to_curve

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

13.38 quad_isconvex

13.39 random_disk

draw random points on the unit disk

13.40 random_simplex

random point inside of a triangle

13.41 sphere_volume

volume of a sphere
function v = sphere_volume(r)

13.42 tetra_volume

volume of a tetrahedron

13.43 tobarycentric

cartesian to barycentric coordinates

13.44 tobarycentric1

cartesian to barycentric coordinates

13.45 tobarycentric2

cartesian to barycentric coordinates

13.46 tobarycentric3

cartesian to barycentric coordinates

13.47 tri_angle

cos of angles of a triangle

13.48 tri_area

angle of a triangle

13.49 tri_centroid

centroid of a triangle

13.50 tri_distance_opposit_midpoint

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

13.51 tri_edge_length

edge length of a triangle

13.52 tri_edge_midpoint

mid point of a triangle

13.53 tri_excircle

excircle of a triangle

13.54 tri_height

height of a triangle

13.55 tri_incircle

incircle of a triangle

13.56 tri_isacute

flag acute triangles

13.57 tri_isobtuse

flag obtuse triangles

13.58 tri_semiperimeter

semiperimeter of a triangle

13.59 tri_side_length

edge length of triangle

14 geometry

14.1 Polygon

Simple 2D polygon class

Polygon properties:

x - x coordinates of polygon

y - y coordinates of polygon

nnodes - number of nodes in the polygon

Polygon methods:

in - checks whether given points lie inside, on the edge, or
outside of the polygon

area - returns the area of the polygon

centerline - computes the centerline of the river

iscw - check whether polygon is clockwise

reverse - reverse the order of the polygon

14.2 bounding_box

bounding box of X

14.3 curvature_1d

curvature of a sampled parametric curve in two dimensions

14.4 cvt

centroidal voronoi tessellation

14.5 deg_to_frac

degree, minutes and seconds to fractions

14.6 ellipse

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

14.7 ellipseX

x-coordinates of y-coordinates of an ellipse

14.8 ellipseY

14.9 first_intersect

get first intersection between lines in A and B

14.10 golden_ratio

golden ratio

14.11 hypot3

hypothenuse in 3D

14.12 meanangle

weighted mean of angles

14.13 meanangle2

mean angle

14.14 meanangle3

mean angle

14.15 meanangle4

mean angle

14.16 medianangle

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

14.17 medianangle2

median angle

input
alpha : x*m, [rad] angle

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

14.18 pilim

limit to +- pi

14.19 streamline_radius_of_curvature

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

15 histogram/@Histogram

15.1 2x

15.2 Histogram

15.3 bimodes

15.4 cdf

15.5 cdfS

15.6 chi2test

15.7 cmoment

15.8 cmomentS

15.9 entropy

15.10 entropyS

15.11 export_csv

15.12 iquantile

15.13 kstest

15.14 kurtosis

15.15 kurtosisS

15.16 mean

15.17 meanS

15.18 median

15.19 medianS

15.20 mode

15.21 modeS

15.22 moment

15.23 momentS

15.24 pdf

15.25 quantile

15.26 quantileS

15.27 resample

15.28 setup

15.29 skewness

15.30 skewnessS

15.31 stairs

15.32 stairsS

15.33 std

15.34 stdS

15.35 var

15.36 varS

16 histogram

16.1 hist_man

16.2 **histadapt**

16.3 **histconst**

16.4 **pdf_poly**

16.5 **plotcdf**

16.6 **test_histogram**

17 **mathematics**

mathematical functions of various kind

17.1 **inrotmat**

18 **linear-algebra**

18.1 **averaging_matrix_2**

18.2 **colnorm**

norms of columns

18.3 condest_

estimation of the condition number

18.4 connectivity_matrix

19 linear-algebra/coordinate-transformation

19.1 barycentric2cartesian

barycentric to cartesian coordinates

19.2 barycentric2cartesian3

convert barycentric to cartesian coordinates

19.3 cartesian2barycentric

cartesian to barycentric coordinates

19.4 cartesian_to_unit_triangle_basis

transform coordinates into unit triangle

19.5 ellipsoid2geoid

19.6 example_approximate_utm_conversion

19.7 latlon2utm

transform latitude and longitude to WGS84 UTM

19.8 latlon2utm_simple

19.9 lowrance_mercator_to_wgs84

convert lowrance coordinates to wgs84

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

19.10 nmea2utm

convert nmea messages to utm coordinates

19.11 sn2xy

convert sn to xy coordinates

19.12 unit_triangle_to_cartesian

transform coordinates in unit triangle to cartesian coordinates

19.13 utm2latlon

convert wgs84 utm to latitude and longitude

19.14 xy2nt

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

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

19.15 xy2sn

convert cartesian to streamwise coordiantes

19.16 xy2sn_java

use java port for speed up

19.17 xy2sn_old

transform points from cartesian into streamwise coordinates

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

20 linear-algebra

20.1 det2x2

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

20.2 det3x3

determinant of stacked 3x3 matrices

20.3 det4x4

determinant of stacked 4x4 matrices

20.4 diag2x2

diagonal of stacked 2x2 matrices

20.5 down

20.6 eig2x2

eigenvalues of stacked 2x2 matrices

21 linear-algebra/eigenvalue

21.1 eig_bisection

21.2 eig_inverse

21.3 eig_inverse_iteration

21.4 eig_power_iteration

22 linear-algebra/eigenvalue/jacobi-davidson

22.1 afun_jdm

22.2 davidson

22.3 jacobi_davidson

22.4 jacobi_davidson_qr

22.5 jacobi_davidson_qz

22.6 jacobi_davidson_simple

22.7 jdqr

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

```

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

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

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

```

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

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

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

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

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

```

```

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

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

```

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

22.8 jdqr_sleijpen

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

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

```

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

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

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

```

```

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

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

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

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

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

```



```

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

```

22.9 jdqr_vorst

```

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

```

```

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

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

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

```

```

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

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

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

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

% accepted separation between eigenvalues:
% no preconditioning
% solve left preconditioned system
% compute vectors and matrices for skew projection
% precondition and project r
% solve preconditioned system

```

```

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

```

22.10 jdqz

```

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

```

```

% Check for stagnation
% compute approximate eigenpair
% Compute approximate eigenpair and residual
% display history
% save history
% check convergence
% expand Schur form
% ZastQ=Z'*Q0
% the final Qschur
% check for conjugate pair
% t perp Zschur, t in span(Q0,imag(q))
% To detect whether another eigenpair is accurate enough
% restart if dim(V)> jmax
%===== END JDQZ
=====
%=====

%===== PREPROCESSING
=====
%=====

%===== ARNOLDI (for initial spaces)
=====
%% then precondition=I and target = 0: apply Arnoldi with A
%===== END ARNOLDI
=====
%=====

%===== POSTPROCESSING
=====
%=====

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

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

%=====

```

```

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

% solve preconditioned system
%=====

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

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

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

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

```

```

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

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

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

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

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

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

%=====

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

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

```

```

%=====

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

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

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

%=====

%=====

%=====

```

22.11 mfunc_jdm

22.12 mgs

22.13 minres_

22.14 mv_jacobi_davidson

23 linear-algebra

23.1 first

23.2 gershgorin_circle

range of eigenvalues determined by the gershgorin circle theorem

23.3 haussdorff

haussdorf dimension

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

Koch snow flake 3:4 -> 1.2619

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

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

23.4 eig2x2

reconstruct matrix from eigenvalue decomposition

23.5 inv2x2

2x2 inverse of stacked matrices

23.6 inv3x3

23.7 inv4x4

inverse of stacked 4x4 matrices

23.8 kernel2matrix

24 linear-algebra/lanczos

24.1 arnoldi

24.2 arnoldi_new

24.3 eigs_lanczos_man

24.4 lanczos

24.5 lanczos_

24.6 `lanczos_biorthogonal`

24.7 `lanczos_biorthogonal_improved`

24.8 `lanczos_ghep`

24.9 `mv_lanczos`

24.10 `reorthogonalise`

24.11 `test_lanczos`

25 `linear-algebra`

25.1 `laplacian_eigenvalue`

25.2 `laplacian_eigenvector`

25.3 `laplacian_power`

25.4 `least_squares_perpendicular_offset`

25.5 `left`

left element of vector, leftmost column is extrapolated

26 `linear-algebra/linear-systems`

26.1 `gmres_man`

break on convergence

26.2 `minres_recycle`

27 `linear-algebra`

27.1 `lpmean`

mean of pth-power of a

27.2 `lpnorm`

norm of lth-power of a

27.3 `matvec3`

matrix-vector product of stacked matrices and vectors

27.4 `max2d`

maximum value and i-j index for matrix

27.5 mid

mid point between neighbouring vector elements

27.6 mpoweri

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

27.7 mtimes2x2

27.8 mtimes3x3

product of stacked 3x3 matrices

27.9 nannorm

norm of a vector, skips nan-values

27.10 nanshift

shift vector, but set out of range values to NaN

27.11 nl

number rows (lines) of a matrix

analogue to unix nl command

27.12 normalise

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

27.13 normalize1

normalize columns in x to [-1,1]

27.14 normrows

27.15 orth2

make matrix A orthogonal to B

27.16 orth_man

orthogonalize the columns of A

27.17 orthogonalise

make x orthogonal to Y

27.18 padd2

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

27.19 paddext

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

27.20 paddval1

padd values at end of x

27.21 paddval2

padd values to x

28 linear-algebra/polynomial

28.1 chebychev

chebycheff polynomials

28.2 piecewise_polynomial

evaluate piecewise polynomial

28.3 roots1

roots of linear functions

28.4 roots2

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

28.5 roots2poly

28.6 roots3

28.7 roots4

28.8 roots_piecewise_linear

28.9 test_roots4

28.10 vanderi_1d

vandermonde matrix of an integral

29 linear-algebra

29.1 randrot

random rotation matrix

29.2 right

get right column by shifting columns to left
extrapolate rightmost column

29.3 rot2

rotation matrix from angle

29.4 rot2dir

rotation matrix from direction vector

29.5 rot3

29.6 rotR

29.7 rownorm

29.8 simmilarity_matrix

29.9 spnorm

frobenius norm

29.10 spzeros

allocate a sparze matrix of zeros

29.11 test_roots3

29.12 `transform_minmax`

29.13 `transpose3`

`transpose` stacked 3x3 matrices

29.14 `transposeall`

29.15 `up`

29.16 `vander_nd`

29.17 `vanderd_2d`

30 `logic`

bitwise operations on integers

30.1 `bitor_man`

bitwise OR of the numbers of the columns of A

input:
 A (positive integer)

31 `master/plot`

31.1 `attach_boundary_value`

31.2 `cartesian_polar`

31.3 `img_vargrid`

31.4 `plot_basis_functions`

31.5 `plot_convergence`

31.6 `plot_dof`

31.7 `plot_eigenbar`

31.8 `plot_error_estimation`

31.9 `plot_error_estimation_2`

31.10 `plot_error_fem`

31.11 `plot_fdm_kernel`

31.12 `plot_fdm_vs_fem`

31.13 `plot_fem_accuracy`

31.14 `plot_function_and_grid`

31.15 `plot_hat`

31.16 `plot_hydrogen_wf`

31.17 `plot_mesh`

31.18 `plot_mesh_2`

31.19 `plot_refine`

31.20 `plot_refine_3d`

31.21 `plot_runtime`

31.22 `plot_spectrum`

31.23 `plot_wavefunction`

32 `master/ported`

32.1 `assemble_2d_phi_phi`

32.2 `assemble_3d_dphi_dphi`

32.3 `assemble_3d_phi_phi`

32.4 `dV_2d_`

32.5 `derivative_2d`

32.6 `derivative_3d`

32.7 `element_neighbour_2d`

32.8 `prefetch_2d_`

32.9 `promote_2d_3_10`

32.10 `promote_2d_3_15`

32.11 `promote_2d_3_21`

32.12 `promote_2d_3_6`

32.13 `promote_3d_4_10`

32.14 `promote_3d_4_20`

32.15 `promote_3d_4_35`

32.16 `vander_2d`

32.17 `vander_3d`

33 mathematics

mathematical functions of various kind

33.1 nearest_fractional_timestep

34 number-theory

34.1 ceiln

floor to leading n-digits

34.2 digitsb

number of digits with respect to specified base

34.3 floorn

floor to n-digits

34.4 iseven

true for even numbers in X

34.5 multichoosek

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

input :

x : scalar integer or vector of arbitrary numbers
k : length of subsets

output :

if x scalar : number of combinations
if x vector : the exact combinations

34.6 nchoosek_man

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

34.7 pythagorean_triple

pythagorean triple

34.8 roundn

round to n digits

35 numerical-methods

35.1 advect_analytic

36 numerical-methods/differentiation

36.1 derivative1

first derivative on variable mesh
second order accurate

36.2 derivative2

second derivative on a variable mesh

37 numerical-methods

37.1 diffuse_analytic

38 numerical-methods/finite-difference

38.1 cdiff

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

38.2 cdifb

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

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_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_2d

finite derivative matrix of second derivative in one dimension

38.8 `derivative_matrix_2d`

finite difference derivative matrix in two dimensions

38.9 `derivative_matrix_curvilinear`

derivative matrix on a curvilinear grid

38.10 `derivative_matrix_curvilinear_2`

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

38.11 `difference_kernel`

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

38.12 `diffusion_matrix_2d_anisotropic`

38.13 `diffusion_matrix_2d_anisotropic2`

38.14 `directional_neighbour`

38.15 `distmat`

distance matrix for a 2 dimensional rectangular matrix

38.16 downwind_difference

38.17 gradpde2d

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

38.18 laplacian

38.19 laplacian_fdm

finite difference matrix of the laplacian
BC

38.20 lrmean

mean of the left and right element

39 numerical-methods/finite-difference/master

39.1 fdm_adaptive_grid

39.2 fdm_adaptive_refinement_old

39.3 fdm_assemble_d1_2d

39.4 `fdm_assemble_d2_2d`

39.5 `fdm_confinement`

39.6 `fdm_d_vargrid`

39.7 `fdm_h_unstructured`

39.8 `fdm_hydrogen_vargrid`

39.9 `fdm_mark_unstructured_2d`

39.10 `fdm_plot`

39.11 `fdm_plot_series`

39.12 `fdm_refine_2d`

39.13 `fdm_refine_3d`

39.14 `fdm_refine_unstructured_2d`

39.15 `fdm_schroedinger_2d`

39.16 `fdm_schroedinger_3d`

39.17 `relocate`

40 numerical-methods/finite-difference

40.1 `mid`

mid point between neighbouring vector elements

40.2 `pwmid`

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

40.3 `ratio`

ratio of two subsequent values

40.4 `steplength`

step length of a vector if it were equispaced

40.5 swapoddeven

swap odd and even elements in a vector

40.6 test_derivative_matrix_2d

40.7 test_derivative_matrix_curvilinear

40.8 test_difference_kernel

40.9 upwind_difference

41 numerical-methods/finite-element

41.1 Mesh_2d.java

41.2 Tree_2d.java

41.3 assemble_1d_dphi_dphi

41.4 assemble_1d_phi_phi

41.5 `assemble_2d_dphi_dphi_java`

41.6 `assemble_2d_phi_phi_java`

41.7 `assemble_3d_dphi_dphi_java`

41.8 `assemble_3d_phi_phi_java`

41.9 `boundary_1d`

41.10 `boundary_2d`

41.11 `boundary_3d`

41.12 `check_area_2d`

41.13 `circmesh`

41.14 `cropradius`

41.15 `display_2d`

41.16 `display_3d`

41.17 `distort`

41.18 `err_2d`

41.19 `estimate_err_2d_3`

41.20 `example_1d`

41.21 `example_2d`

41.22 `explode`

41.23 `fem_2d`

41.24 `fem_2d_heuristic_mesh`

41.25 fem_get_2d_radial

41.26 fem_interpolation

41.27 fem_plot_1d

41.28 fem_plot_1d_series

41.29 fem_plot_2d

41.30 fem_plot_2d_series

41.31 fem_plot_3d

41.32 fem_plot_3d_series

41.33 fem_plot_confine_series

41.34 fem_radial

adaptive grid
constant grid

41.35 flip_2d

41.36 get_mesh_arrays

41.37 hashkey

42 numerical-methods/finite-element/int

42.1 int_1d_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 pdeeig_2d

43.11 pdeeig_3d

43.12 polynomial_derivative_1d

43.13 potential_const

43.14 potential_coulomb

43.15 potential_harmonic_oscillator

43.16 project_circle

43.17 project_rectangle

43.18 promote_1d_2_3

43.19 promote_1d_2_4

43.20 promote_1d_2_5

43.21 promote_1d_2_6

43.22 quadrilaterate

43.23 recalculate_regularity_2d

43.24 refine_1d

43.25 refine_2d_21

43.26 refine_2d_structural

43.27 regularity_1d

43.28 regularity_2d

43.29 regularity_3d

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

43.30 relocate_2d

43.31 test_circmesh

43.32 test_hermite

43.33 tri_assign_points

43.34 triangulation_uniform

43.35 vander_1d

van der Monde matrix

43.36 vanderd_1d

43.37 vanderi_1d

44 numerical-methods/finite-volume/@Advection

44.1 Advection

FVM treatment of the Advection equation

44.2 dot_advection

advection equation

45 numerical-methods/finite-volume/@Burgers

45.1 burgers_split

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

45.2 dot_burgers_fdm

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

45.3 dot_burgers_fft

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

46 numerical-methods/finite-volume/@Finite_Volume

46.1 Finite_Volume

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

46.2 apply_bc

apply boundary conditions

46.3 solve

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

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

46.4 step_split_strang

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

46.5 step_unsplit

step in time, without splitting the inhomogeneous term

47 numerical-methods/finite-volume/@Flux_Limiter

47.1 Flux_Limiter

class of flux limiters

47.2 beam_warming

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

47.3 fromm

fromme limiter
low res

47.4 lax_wendroff

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

47.5 minmod

min-mod schock limiter

47.6 monotized_central

monotonized central flux limiter

47.7 muscl

muscl flux limiter

47.8 superbee

superbee limiter

47.9 upwind

godunov scheme
godunov, first order accurate

47.10 vanLeer

van Leer limiter

48 numerical-methods/finite-volume/@KDV

48.1 dot_kdv_fdm

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

48.2 dot_kdv_fft

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

48.3 kdv_split

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

49 numerical-methods/finite-volume/@Reconstruct_Average_E

49.1 Reconstruct_Average_Evolve

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

McCronack Scheme

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

error:

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

49.2 advect_highres

single time step for the reconstruct evolve algorithm

49.3 advect_lowres

single time step
low resolution

50 numerical-methods/finite-volume

50.1 Godunov

Godunov, upwind method for systems of pdes

50.2 Lax_Friedrich

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

50.3 Measure

50.4 Roe

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

The roe solver guarantess:

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

50.5 fv_swe

wrapper for solving SWE

50.6 staggered_euler

forward euler method with staggered grid

50.7 staggered_grid

staggered grid approximation to the SWE

51 numerical-methods

51.1 grid2quad

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

52 numerical-methods/integration

52.1 cumintL

cumulative integral from left to right

52.2 cumintR

cumulative integral from right to left

52.3 int_trapezoidal

integrate y along x with the trapezoidal rule

53 numerical-methods/interpolation/@Kriging

53.1 Kriging

```
class for Kriging interpolation
```

53.2 estimate_semivariance

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

53.3 interpolate_

```
interpolate with Kriging method

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

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

54 numerical-methods/interpolation/@RegularizedInterpolator1

54.1 RegularizedInterpolator1

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

54.2 init

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

55 numerical-methods/interpolation/@RegularizedInterpolator

55.1 RegularizedInterpolator2

class for regularized interpolation on an unstructures mesh (
interpolation)

55.2 init

initialize the interpolator with a set of point samples

56 numerical-methods/interpolation/@RegularizedInterpolator

56.1 RegularizedInterpolator3

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

56.2 init

initialize the interpolator with a set of sampling points

57 numerical-methods/interpolation

57.1 IDW

spatial averaging by inverse distance weighting

57.2 IPoly

polynomial interpolation class

57.3 IRBM

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

57.4 ISparse

```
sparse interpolation class
```

57.5 Inn

```
nearest neighbour interpolation
```

57.6 Interpolator

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

57.7 fixnan

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

57.8 idw1

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

57.9 idw2

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

57.10 inner2outer

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

57.11 inner2outer2

interpolate from element (segment) centres to edge points

57.12 interp1_circular

57.13 interp1_limited

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

57.14 interp1_man

interpolate

57.15 interp1_piecewise_linear

57.16 interp1_save

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

57.17 `interp1_slope`

quadratic interpolation returning value and derivative(s)

57.18 `interp1_smooth`

57.19 `interp1_unique`

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

57.20 `interp2_man`

nearest neighbour interpolation in two dimensions

57.21 `interp_angle`

interpolate an angle

57.22 `interp_fourier`

interpolation by the fourier method

57.23 `interp_fourier_batch`

batch interpolation by the fourier interpolation

57.24 interp_sn

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

57.25 interp_sn2

interpolation in streamwise coordinates

57.26 interp_sn3

57.27 interp_sn_

57.28 limit_by_distance_1d

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

57.29 resample1

interpolation along a parametric curve with variable step width

57.30 resample_d_min

resample a function

57.31 resample_vector

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

57.32 test_interp1_limited

58 numerical-methods

58.1 inverse_complex

58.2 maccormack_step

58.3 minmod

59 numerical-methods/multigrid

59.1 mg_interpolate

59.2 mg_restrict

60 numerical-methods/ode/@BVPS_Characteristic

60.1 BVPS_Characteristic

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

60.2 assemble1_A

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

60.3 assemble1_A_Q

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

60.4 assemble2_A

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

60.5 assemble_AA

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

60.6 assemble_AAA

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

60.7 assemble_Ic

60.8 bvp1c

60.9 check_arguments

60.10 couple_junctions

60.11 derivative

60.12 init

60.13 inner2outer_bvp2c

60.14 reconstruct

60.15 resample

60.16 solve

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

odefun provides ode coefficients c:

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

subject to the boundary conditions

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

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

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

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

60.17 test_assemble1_A

60.18 test_assemble2_A

61 numerical-methods/ode/@Time_Stepper

61.1 Time_Stepper

61.2 solve

62 numerical-methods/ode

62.1 bvp2fdm

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

as boundary value problems by the finite difference method

odefun provides ode coefficients c:

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

subject to the boundary conditions

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

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

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

62.2 bvp2wavetrain

solve second order boundary value problem by repeated integration

62.3 bvp2wavetwopass

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

62.4 ivp_euler_forward

solve initial value problem by the euler forward method

62.5 ivp_euler_forward2

62.6 ivprk2

solve initial value problem by the two step runge kutta method

62.7 ode2_matrix

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

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

62.8 ode2characteristic

second order odes
transmitted and reflected wave

62.9 step_trapezoidal

single trapezoidal step

62.10 test_bvp2

63 numerical-methods/optimisation

63.1 aitken_iteration

63.2 anderson_iteration

63.3 armijo_stopping_criterion

armijo stopping criterion for optimizations

63.4 astar

astar path finding algorithm

63.5 binsearch

binary search on a line

63.6 bisection

bisection

63.7 box1

test objective function for optimisation routines

63.8 box2

63.9 cauchy

63.10 cauchy2

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

63.11 directional_derivative

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

63.12 dud

optimization by the dud algorithm

63.13 extreme3

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

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

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

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

63.14 extreme_quadratic

63.15 ftest

63.16 fzero_bisect

63.17 fzero_newton

63.18 grad

numerical gradient

63.19 hessian

numerical hessian

63.20 hessian_from_gradient

numerical hessian from gradient

63.21 hessian_projected

numerical hessian projected to one dimension

63.22 line_search

bisection routine

63.23 line_search2

bisection method

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

63.24 line_search_polynomial

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

63.25 line_search_polynomial2

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

63.26 line_search_quadratic

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

63.27 line_search_quadratic2

```
quadratic line search
```

63.28 line_search_wolfe

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

63.29 ls_bgfs

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


63.30 ls_broyden

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

BGFS:
Broyden 1965
Fletcher 1970
Goldfarb 1970
Shanno 1970

63.31 ls_generalized_secant

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

63.32 nlcg

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

63.33 nlls

non-linear least squares

63.34 picard

picard iteration

63.35 poly_extrema

extrema of a polynomial

63.36 quadratic_function

evaluate quadratic function in higher dimensions

63.37 quadratic_programming

optimize by quadratic programming

63.38 quadratic_step

single step of the quadratic programming

63.39 rosenbrock

rosenbrock test function

63.40 sqrt_heron

Heron's method for the square root

63.41 test_directional_derivative

63.42 test_dud

63.43 test_fzero_newton

63.44 test_line_search_quadratic2

63.45 test_ls_generalized_secant

63.46 test_nlcg_6_order

63.47 test_nlls

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

64 numerical-methods/pde

64.1 laplacian2d_fundamental_solution

65 numerical-methods/piecewise-polynomials

65.1 Hermite1

hermite polynomial interpolation in 1d

65.2 hp2_fit

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

65.3 hp2_predict

prediction with pw hermite polynomial
c are values at support points

65.4 hp_predict

predict with piecewise hermite polynomial

65.5 hp_regress

fit piecewise hermite polynomial
coefficients are values and derivatives

65.6 lp_count

lagrangian basis for interpolation
count number of valid samples

65.7 lp_predict

lagrangian basis piecwie interpolation, predictor

65.8 lp_regress

65.9 lp_regress_

66 numerical-methods

66.1 test_adams_bashforth

67 mathematics

mathematical functions of various kind

67.1 oversampleNZ

68 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 giniрег

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

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

71.8 regression_method_of_moments

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

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 asymwin

creates asymmetrical filter windows
filter will always have negative weights

75 signal-processing/autocorrelation

75.1 acf_radial

75.2 acfar1

Autocorrelation function of the finite AR1 process

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

75.3 acfar1_2

autocorrelation of the ar1 process

75.4 acfar2

impulse response of the ar2 process

75.5 acfar2_2

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

75.6 ar1_cutoff_frequency

75.7 ar1_effective_sample_size

effective sample size correction for autocorrelated series

75.8 ar1_mse_mu_single_sample

standard error of a single sample of an ar1 correlated process

75.9 ar1_mse_pop

variance of the population mean of a single realisation around zero

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

75.10 ar1_mse_range

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

75.11 ar1_spectrum

spectrum of the ar1 process

75.12 ar1_to_tikhonov

convert ar1 correlation to tikhonovs lambda

75.13 ar1_var_factor

variance correction factor for an autocorrelated finite process
n : [1 .. inf] population size
m : [1 .. n] samples size
rho : [-1 < rho < 1 (for convergence)] correlation of samples

75.14 ar1_var_factor_

variance of an autocorrelated finite process

75.15 ar1_var_range2

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

75.16 ar1delay

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

75.17 ar1delay_old

autocorrelation of the residual

75.18 ar2_acf2c

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

75.19 ar2conv

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

75.20 ar2dof

effective samples size for the ar2 process

75.21 ar2param

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

75.22 autocorr2

75.23 autocorr_bandpass

75.24 autocorr_brownian_phase

75.25 autocorr_brownian_phase_2d

75.26 autocorr_brownian_phase_across

75.27 autocorr_decay_rate

estimate exponential decay of the autocorrelation

75.28 autocorr_effective_sample_size

effective sample size from acf

75.29 autocorr_fft

estimate sample autocorrelation function

75.30 autocorr_forest

75.31 autocorr_genton

autocorrelation function

75.32 autocorr_highpass

75.33 autocorr_lowpass

75.34 autocorr_periodic_additive_noise

75.35 autocorr_periodic_windowed

75.36 autocorr_radial

76 signal-processing

76.1 average_wave_shape

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

76.2 bandpass

bandpass filter

76.3 bartlett

Effective sample size factor for bartlett window
c.f. thiebaux
c.f spectral analysis-jenkins, eq. (6.3.27)
 $c = acf$
note: results seams always to be 1 tac too low
T : reduction factor for dof
for ar1 with $a = \rho^k = \exp(-k/L)$, $T = 2L$

76.4 bin1d

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

76.5 bin2d

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

76.6 `binormrnd`

generate two correlated normally distributed vectors

76.7 `coherence`

76.8 `conv1_man`

convolutions with padding

76.9 `conv2_man`

convolution in 2d

76.10 `conv2z`

76.11 `conv30`

convolve with rectangular window of length n
circular boundaries

76.12 `conv_`

convolution of a with b

76.13 `conv_centered`

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

76.14 convz

76.15 cosexpdelay

76.16 csmooth

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

76.17 daniell_window

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

76.18 db2neper

convert decibel to neper

76.19 db2power

power ratio from db

76.20 derive_bandpass_continuous_scale

76.21 derive_danielle_weight

76.22 `derive_limit_0_acfar`

76.23 `detect_peak`

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

76.24 `determine_phase_shift`

76.25 `determine_phase_shift1`

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

76.26 `doublesum_ij`

double sum of r^i

76.27 `effective_sample_size_to_ar1`

convert effective sample size to ar1 correlation

76.28 `fcut2Lw_gausswin`

76.29 `fcut_gausswin`

76.30 `flt_hodges_lehman`

77 signal-processing/filters

77.1 circfilt2

Mon 19 Dec 17:03:02 CET 2022

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

77.2 filter1

filter along one dimension

77.3 filter2

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

77.4 filter_

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

77.5 filter_r_to_f0

77.6 filter_rho_to_f0

77.7 filter_twosided

77.8 filteriir

filter adcp t-n data over time

v : nz,nt : values to be filtered
H : nt,1 : depth of ensemble
last : nt,1 : last bin above bottom that can be sampled without
side lobe interference
nf : scalar : number of reweighted iterations

when samples
- distance to bed is reference (advantageous for near-bed suspended
transport)
TODO for wash load: distance to surface is more relevant
interpolate depending on z

when depth changes, neighbouring indices do not correspond to same
relative position in the water column
relative position in the column (s-coordinate) smoothes values
near the bed: absolute distance to bed is chosen
near surface: absolute distance to surface is chosen
-> cubic transformation of index

faster and avoid aliasing (smoothing along z)
resample ensemble to same number of bins in S -> filter ->
resample back
use nonlinear transform z-s coordinates
-> resampling has to be local (Hi -> H-filtered)

filtered profile coordinates to sample coordinates
zf -> zi (special transform)
corresponding indices and fractions
filtration step (update of hf and vf)
sample coordinates to updated profile coordinates
(the inverse step is actually not necessary)
write filtered value

77.9 filterp

77.10 filterp1

fir filter with some fancy extras

77.11 filterstd

77.12 gaussfilt2

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

77.13 lowpass_discrete

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

77.14 meanfilt2

filter with a rectangular window along both dimensions

77.15 medfilt1_man

moving median filter, supports columnwise operation

77.16 medfilt1_man2

moving median filter with special treatment of boundaries

77.17 medfilt1_padded

median filter with padding

77.18 medfilt1_reduced

median filter with padding

77.19 trifilt1

filter with triangular window

trifilt1 is ident to twice applying rectfilt1 (meanfilt1) with half
the domain size

note : infinitely many convolution yield a gaussian

77.20 trifilt2

filter with a triangular window along both dimensions

78 signal-processing

78.1 firls_man

design finite impulse response filter by the least squares method

78.2 fit_spectral_density

fit spectral densities

78.3 fit_spectral_density_2d

fit spectral densities

78.4 fit_spectral_density_radial

fit spectral densities

78.5 flattopwin

the flat top window

78.6 frequency_response_boxcar

frequency response of a boxcar filter

78.7 freqz_boxcar

frequency response of a boxcar filter

78.8 gaussfilt1

filter data series with a gaussian window

78.9 hanchangewin

hanning window for change point detection

78.10 hanchangewin2

hanning window for change point detection

78.11 hanwin

hanning filter window

78.12 hanwin_

hanning filter window

78.13 `high_pass_1d_simple`

78.14 `kaiserwin`

kaiser filter window

78.15 `kalman`

Kalman filter

78.16 `lanczoswin`

Lanczos window

78.17 `last`

lake tail, but for matrices

78.18 `maxfilt1`

78.19 `meanfilt1`

moving average filter with special treatment of the boundaries

78.20 `mid_term_single_sample`

variance of single sample, mid term

78.21 `minfilt1`

78.22 minmax

78.23 mu2ar1

error variance of the mean of the finite length ar1 process

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

78.24 mysmooth

78.25 nanautocorr

autocorrelation with nan-values

78.26 nanmedfilt1

medfilt1, skipping nans

78.27 neper2db

convert neper to db

78.28 oscillator_noisy

79 signal-processing/passes

79.1 bandpass1d

79.2 bandpass1d_fft

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

79.3 bandpass1d_implicit

79.4 bandpass2

bandpass filter

79.5 bandpass2d

79.6 bandpass2d_2

79.7 bandpass2d_fft

79.8 bandpass2d_ideal

79.9 bandpass2d_implicit

79.10 bandpass2d_iso

79.11 `bandpass_arg`

determine correlation coefficient from frequency of mode for the symmetric

79.12 `bandpass_f0_to_rho`

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

79.13 `bandpass_max`

79.14 `bandpass_max2`

79.15 `highpass`

high pass filter

79.16 `highpass1d_fft_cos`

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

79.17 `highpass1d_implicit`

79.18 `highpass2d_fft`

79.19 `highpass2d_ideal`

79.20 `highpass2d_implicit`

79.21 `highpass_arg`

79.22 `highpass_fc_to_rho`

79.23 `lowpass`

`low pass filter`

79.24 `lowpass1d_fft`

79.25 `lowpass1d_implicit`

79.26 `lowpass2`

`design low pass filter with cutoff-frequency f1`

79.27 `lowpass2d_2`

79.28 `lowpass2d_anisotropic`

79.29 `lowpass2d_fft`

79.30 `lowpass2d_ideal`

79.31 `lowpass2d_implicit`

79.32 `lowpass_arg`

79.33 `lowpass_fc_to_rho`

79.34 `lowpass_iir`

`iir-low pass`

79.35 `lowpass_iir_symmetric`

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

79.36 `lowpassfilter2`

`low-pass filter of data`

80 signal-processing

80.1 peaks_man

peaks of a periodogram

81 signal-processing/periodogram

81.1 periodogram

compute the normalized periodogram

81.2 periodogram_2d

compute the normalized periodogram in two dimensions

81.3 periodogram_annular

81.4 periodogram_bartlett

estimate the spectral density nonparametrically with Bartlett's
method

81.5 periodogram_bootstrap

81.6 periodogram_confidence_interval

confidence interval for periodogram values

81.7 periodogram_filter

81.8 periodogram_median

81.9 periodogram_p_value

81.10 periodogram_qq

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

81.11 periodogram_quantiles

quantiles of a periodogram

81.12 periodogram_radial

81.13 periodogram_std

standard deviation of a periodogram

81.14 periodogram_test_periodicity

test a periodogram for hidden periodic frequency components

```
function [p, ratio, maxShat, mdx, fdx, S] = periodogram_test_periodicity  
    (fx, Shat, nf, fmin, fmax, S, mode)
```

input:

```
    fx : frequengcies  
    Shat : corresponding periodogram values  
    nf : number of bins to test for periodicity, ignored when S  
        is given
```

```

fmin, fmax : frequency range limits to test
S      : exact (a priori known theoretical spectral density,
           must not be estimated from the periodogram)
mode : automatically set to "exact", when S given
       inclusive : estimate density by smoothing including the
                   central bin
       exclusive  : estimate density by smoothing excluding the
                   central bin
note: inclusive and exclusive lead to different distribution
      but identical p-values

```

TODO pass L and not fx

81.15 periodogram_test_periodicity_2d

test a periodogram for hidden periodic frequency components

```

[p,stat,ratio] = periodogram_test_periodicity_2d(b, L, nf, frmin,
          frmax)
input:
    fx : frequengcies
    b   : image to test for presence of hidden periodicities,
          i.e. periodicities where the frequency is not known a
          priori
    nf  : radius of circular disk (in number of bins) used for
          smoothing
          the periodogram to estimate the spectral density
    bmsk : mask determining parts of the image to include in the
          analysis
          default is entire image
    fmin, fmax : (radial) frequency range limits to test (fmask)

mode : automatically set to "exact", when S given
       inclusive : estimate density by smoothing including the
                   central bin
       exclusive  : estimate density by smoothing excluding the
                   central bin
note: inclusive and exclusive lead to different distribution
      but identical p-values

influence of masking the input file:
- the root-mean-square energy of the ordinates is
  proportional
  to the number of unmasked points
- values in the periodogram are not any more linearly
  independent
  so that the dof of the filter window is not nf^2

```


TODO make frmin, frmax an fmask

81.16 periodogram_test_stationarity

test a periodogram for stationarity
note : the method works, but is of little practical use,
as it requires about 50 periods and a small dx to detect a
frequency change by a factor of 2

81.17 periodogram_welsh

82 signal-processing

82.1 polyfilt1

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

82.2 qmedfilt1

medfilt1, after fitting a quadratic polynomial

82.3 quadratfilt1

82.4 quadratwin

82.5 randar1

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

82.6 randar1_dual

draw random variables of two correlated ar1 processes

82.7 randar2

generate ar2 process

82.8 randarp

randomly generate the instance of an ar-p process

82.9 rectwin

rectangular window

82.10 recursive_sum

82.11 select_range

82.12 smooth1d_parametric

smooth position of $p_0=x_0, y_0$ between $p_1=x_1, y_1$ and $p_2=x_2, y_2$,
so that distance to p_1 and p_2 becomes equal
and the chord length remains the same

82.13 smooth2

smooth vectos of X

82.14 `smooth_man`

82.15 `smooth_parametric`

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

82.16 `smooth_parametric2`

parametrically smooth the curve

82.17 `smooth_with_splines`

82.18 `smoothfft`

filter with fast fourier transform

83 signal-processing/spectral-density

83.1 `spectral_density_ar2`

83.2 `spectral_density_area`

integrate the spectral density

83.3 `spectral_density_bandpass2d_ideal`

83.4 spectral_density_bandpass_2d

83.5 spectral_density_bandpass_2d_max2par

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

83.6 spectral_density_bandpass_2d_scale

83.7 spectral_density_bandpass_2d_scale_old

83.8 spectral_density_bandpass_continuous

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

output :

S_bp : spectral density of the bandpass filter in continuos space
limit case of the discrete bandpass for $dx \rightarrow 0$

Sc : scale factor to normalize area to 1, if noramlize = true

input :

f : frequency (abszissa)

fc : central frequeuncy, location of maximum on abszissa

order : number of times filter is applied iteratively, not
necessarily integer

normalize : normalize area under curve $\int_0^\infty S(f) df = 1$, if
not maximum $S(fc) = 1$

pp : powers for recombination of the lowpass filter

83.9 spectral_density_bandpass_continuous_max

maximum of the bandpass spectral density

83.10 spectral_density_bandpass_continuous_max2par

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

83.11 spectral_density_bandpass_continuous_scale

normaliztation scale of the spatial bandpass density

83.12 spectral_density_bandpass_discrete

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

83.13 spectral_density_brownian_phase

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

83.14 spectral_density_brownian_phase_2d

83.15 spectral_density_brownian_phase_across

83.16 spectral_density_brownian_phase_across_max

83.17 spectral_density_brownian_phase_across_max2par

83.18 `spectral_density_brownian_phase_across_mode2par`

83.19 `spectral_density_brownian_phase_mode`

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

83.20 `spectral_density_brownian_phase_mode2par`

transform mode to parameters of the brownian phase spectral density

83.21 `spectral_density_brownian_phase_scale`

normalization scale of the brownian phase spectral density

83.22 `spectral_density_estimate_2d`

83.23 `spectral_density_flat`

flat spectral density of a random vector with iid elements

83.24 `spectral_density_forest`

83.25 `spectral_density_gausswin`

83.26 `spectral_density_highpass`

83.27 `spectral_density_highpass2d_ideal`

83.28 `spectral_density_highpass_2d`

83.29 `spectral_density_highpass_cos`

consine shaped spectral density of a highpass filter

83.30 `spectral_density_lorentzian`

lorentzian spectral density

83.31 `spectral_density_lorentzian_max`

mode (maximum) of the lorentzian spectral density

83.32 `spectral_density_lorentzian_max2par`

transform maximum of the lorentzian spectral density to its
distribution parameters

83.33 `spectral_density_lorentzian_scale`

normalization scale of the lorentzian spectral density

83.34 `spectral_density_lowpass2d_ideal`

83.35 `spectral_density_lowpass_2d`

83.36 spectral_density_lowpass_continuous

83.37 spectral_density_lowpass_continuous_scale

83.38 spectral_density_lowpass_discrete

83.39 spectral_density_lowpass_one_sided

83.40 spectral_density_maximum_bias_corrected

83.41 spectral_density_periodic_additive_noise

83.42 spectral_density_rectwin

83.43 spectral_density_wperiodic

84 signal-processing

84.1 spectral_density_brownian_phase_reg2par

84.2 spectrogram

spectrogram

84.3 sum_i_lag

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

84.4 sum_ii

sum of ar1 matrix
 $\sum_{i=1}^n \sum_{j=1}^n \rho^{|i-j|}$
this is for the variance, take square root for the standard
deviation factor

84.5 sum_ii_

84.6 sum_ij

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

84.7 sum_ij_

84.8 sum_ij_partial_

84.9 sum_multivar

sum of matrix entries of bivariate ar1 process

84.10 test_acfar1

84.11 tikhonov_to_ar1

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

84.12 trapwin

trapezoidal filter window

84.13 triwin

triangular filter window

84.14 triwin2

triangular filter window

84.15 tukeywin_man

84.16 varar1

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

84.17 welch_spectrogram

welch spectrogram

84.18 wfilt

filter with window

84.19 winbandpass

filter with bandpass

85 signal-processing/windows

85.1 circwin

85.2 danielle_window

danielle fourier window

85.3 gausswin

85.4 gausswin2

85.5 radial_window

radial filter window in the 2d-frequency domain

85.6 range_window

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

85.7 rectwin_cutoff_frequency

85.8 std_window

moving block standard deviation

85.9 window2d

85.10 window_make_odd

86 signal-processing

86.1 winfilt0

filter with window

86.2 winlength

window length for desired cutoff frequency
power at f_c is halved
 $H(wf) = 1/\sqrt{2} H(f)$
if the filter window were used as a low pass filter
note: the user should prefer a windowed ideal low pass filter
TODO, relate this to DOF

86.3 wmeanfilt

mean filter with window

86.4 wmedfilt

median filter with window

86.5 wordfilt

weighted order filter

86.6 wordfilt_edgeworth

weighed order filter

86.7 wrapphase

86.8 xar1

86.9 xcorr_man

cross correlation of two sampled ar1 processes

87 sorting

87.1 sort2

sort two numbers

87.2 sort2d

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

88 spatial-pattern-analysis/@Spatial_Pattern

88.1 Spatial_Pattern

88.2 analyze_grid

88.3 analyze_transect

88.4 extract_improfile

88.5 imread

88.6 plot

88.7 tabulate

89 spatial-pattern-analysis

89.1 analyze_grid2

89.2 approximate_ratio_distribution

89.3 `banded_pattern`

89.4 `brownian_phase_patch_size_distribution`

89.5 `hexagonal_pattern`

89.6 `isisotropic`

89.7 `patch_size_1d`

89.8 `patch_size_2d`

89.9 `separate_isotropic_from_anisotropic_density`

89.10 `suppress_low_frequency_lobe`

90 `special-functions`

90.1 `bessel_sphere`

spherical Bessel function of the first kind

90.2 `beta_man`

90.3 `betainc_man`

90.4 `digamma_man`

90.5 `exp10`

90.6 `hankel_sphere`

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

90.7 `hermite`

probabilistic's hermite polynomial by recurrence relation

input :
n : order
x : value

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

90.8 `legendre_man`

legendre polynomials

90.9 neumann_sphere

spherical Neumann function
Bessel function of the second kind

91 statistics

91.1 atan_s2

stadard deviation of the arcus tangens by means of taylor expansion

91.2 beta_kurt

91.3 beta_mean

91.4 beta_moment2param

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

91.5 beta_skew

91.6 beta_std

91.7 chi2_kurt

91.8 `chi2_mean`

91.9 `chi2_skew`

91.10 `chi2_std`

91.11 `coefficient_of_determination`

91.12 `conditional_expectation_normal`

91.13 `correlation_confidence_pearson`

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

92 `statistics/distributions`

92.1 `PDF`

class for quasi-distributions from a set of sampling points

92.2 `binorm_separation_coefficient`

separation coefficient of a bimodal normal distribution

92.3 binormcdf

bio-modal gaussian distribution

92.4 binormfit

fit sum of to normal distribution to a histogram

92.5 binormpdf

92.6 edgeworth_cdf

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

92.7 edgeworth_pdf

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

92.8 gam_moment2param

92.9 logn_mean

92.10 logn_mode

mode (maximum) of the log-normal density

92.11 logn_mode2param

92.12 logn_moment2param

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

92.13 logn_param2moment

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

92.14 logn_std

92.15 lognpdf_

log normal distribution called by modes rather than parameters

92.16 pdfsample

pdf from sample distribution
Note: better use kernel density estimates

92.17 t2cdf

Hotelling's T-squared cumulative distribution

92.18 t2inv

inverse of Hotelling's T-squared cumulative distribution

93 statistics

93.1 error_propagation_fraction

93.2 error_propagation_product

93.3 example_standard_error_of_sample_quantiles

93.4 f_var_finite

reduction of variance when sampling from a finite population
without replacement

93.5 fisher_mean

93.6 fisher_moment2param

93.7 fisher_std

93.8 gam_mean

93.9 `gam_std`

93.10 `gamma_mode_to_parameter`

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

93.11 `gamma_stirling`

93.12 `gaussfit3`

93.13 `gaussfit_quantile`

93.14 `geoserr`

93.15 `geostd`

93.16 `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

93.17 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

94 statistics/information-theory

94.1 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

94.2 bayesian_information_criterion

bayesian information criterion

95 statistics

95.1 jackknife_block

95.2 kurtncdf

95.3 kurtnpdf

95.4 kurtosis_bias_corrected

bias corrected kurtosis

95.5 limit

limit a by lower and upper bound

95.6 logfactorial

approximate log of the factorial

95.7 loglogpdf

95.8 lognfit_quantile

95.9 logskewcdf

95.10 logskewpdf

96 statistics/logu

96.1 lambertw_numeric

lambert-w function

96.2 logtrialtcdf

pdf of a logarithmic triangular distribution

96.3 logtrialtinv

inverse of the logarithmic triangular distribution

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

96.4 logtrialtmean

mean of the logarithmic triangular distribution

96.5 logtrialtpdf

density of the logarithmic triangular distribution

96.6 logtrialtrnd

96.7 logtricdf

cumulative distribution of the logarithmic triangular distribution

96.8 logtriinv

inverse of the logarithmic triangular distribution

96.9 logtrimean

mean of the logarithmic triangular distribution

96.10 logtripdf

probability density of the logarithmic triangular distribution

96.11 logtirnd

96.12 logucdf

probability density of the logarithmic uniform distribution

96.13 logucm

central moments of the log-uniform distribution

96.14 loguinv

inverse of the log-uniform distribution

96.15 logumean

mean of the log-uniform distribution

96.16 logupdf

pdf of the log uniform distribution

96.17 logurnd

random numbers following a log-uniform distribution

96.18 loguvar

variance of the log-uniform distribution

96.19 medlogu

median of the log-uniform distribution

96.20 test_logurnd

96.21 tricdf

cumulative distribution of the log-triangular distribution

96.22 triinv

inverse of the triangular distribution

96.23 trimedian

median of the triangular distribution

96.24 tripdf

probability density of the triangular distribution

96.25 trirnd

random numbers of the triangular distribution

97 statistics

97.1 max_exprnd

97.2 maxnnormals

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

97.3 mean_angle

97.4 mean_generalized_gampdf

97.5 midrange

mid range of columns of X

97.6 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

97.7 mode_man

98 statistics/moment-statistics

98.1 autocorr_man3

autocorrelation of the columns of X

98.2 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

98.3 autocorr_man5

autocorrelation of the columns of X

98.4 blockserr

estimate the standard error of potentially sequentially correlated
data
by blocking
block length should be sufficiently larger than correlation length
and sufficiently smaller than data length
this uses a sliding block approach, which reduces the variation of
the error estimate

98.5 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

98.6 corr_man

correlation of two vectors

98.7 cov_man

covariance matrix of two vectors

98.8 dof

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

98.9 edgeworth_quantile

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

98.10 effective_sample_size

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

98.11 `f_correlation`

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

98.12 `f_finite`

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

98.13 `lmean`

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

98.14 `lmoment`

l -moment of vector x

98.15 `maskmean`

mean of the masked values of X

98.16 `masknanmean`

98.17 `mean1`

mean of x

98.18 `mean_man`

mean and standard error of X

98.19 mse

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

98.20 nanautocorr_man1

autocorrelation of a vector with nan-values

98.21 nanautocorr_man2

autocorrelation of a vector with nan-values

98.22 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

98.23 nancorr

(co)-correlation matrix when samples a NaN

98.24 nancumsum

cumulative sum, setting nan values to zero

98.25 nanlmean

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

98.26 nanr2

coefficient of determination when samples are invalid

98.27 nanrms

root mean square value when sample contains nan-values

98.28 nanrmse

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

98.29 nanserr

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

98.30 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)

98.31 nanwstd

weighed standard deviation

98.32 nanwvar

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

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

98.33 nanxcorr

98.34 pearson

pearson correlation coefficient

98.35 pearson_to_kendall

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

98.36 pool_samples

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

98.37 qmean

trimmed mean

98.38 range_mean

98.39 rmse_

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

98.40 serr

standard error of the mean of a set of uncorrelated samples

98.41 `serr1`

98.42 `test_qskew`

98.43 `test_qstd_qskew_optimal_p`

98.44 `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

98.45 `wcorr`

correlation of two vectors when samples are weighted

98.46 `wcov`

covariance of two vectors when samples are weighted

98.47 `wdof`

effective degrees of freedom for weighted samples

98.48 wkurt

kurtosis with weighted samples

98.49 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)

98.50 wrms

weighted root mean square

98.51 wserr

weighted root mean square error

98.52 wskew

skewness of a weighted set of samples

function sk = wskew(w,x)

98.53 wstd

weighed standard deviation

98.54 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

99 statistics

99.1 nangeomean

99.2 nangeostd

geometric standard deviation ignoring nan-values

100 statistics/nonparametric-statistics

100.1 kernel1d

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

100.2 kernel2d

kernel density estimate in two dimensions

101 statistics

101.1 normmmoment

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

101.2 normpdf2

pdf of the bivariate normal distribution

102 statistics/order-statistics

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

102.2 kendall

kendall correlation coefficient

102.3 kendall_to_pearson

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

c.f. Kruska, 1985

102.4 mad2sd

transform median absolute deviation to standard deviation
for normal distributed values

102.5 madcorr

proxy correlation by median absolute deviation

102.6 median2_holder

102.7 median_ci

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

102.8 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`

102.9 mediani

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

102.10 nanmadcorr

proxy correlation by median absolute deviation

102.11 nanwmedian

weighted median, skips nan-values

102.12 nanwquantile

weighted quantile, skips nan values

102.13 oja_median

two dimensional oja median

note: the multivariate median is not unique

oja 1983, for extension to multivariate function, see chaudhri

102.14 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

102.15 qmoments

moments estimated from quantiles

102.16 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

102.17 qskewq

skewness estimated by quantiles

102.18 qstdq

proxy standard deviation determined by quantiles

102.19 quantile1_optimisation

102.20 quantile2_breckling

qunatile regression

102.21 quantile2_chaudhuri

quantile regression

102.22 quantile2_projected

quantile in two dimensions

102.23 quantile2_projected2

spatial qunatile for chosen direction

102.24 quantile_envelope

102.25 quantile_regression_simple

simple quantile regression

102.26 ranking

ranking for spearman statistics

102.27 spatial_median

c.f. Oja 2008

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

102.28 spatial_quantile

spatial quantile

102.29 spatial_quantile2

spatial quantile

102.30 spatial_quantile3

spatial quantile

102.31 spatial_rank

unsigned rank

102.32 spatial_sign

spatial sign

102.33 spatial_signed_rank

signed rank

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

102.34 spearman

spearman's product moment coefficient

102.35 spearman_rank

102.36 spearman_to_pearson

conversion of spearman rank to person product moment correlation coefficient

102.37 wmedian

weighted median

102.38 wquantile

weighted quantile

103 statistics

103.1 qstd

103.2 quantile_extrap

103.3 quantile_sin

104 statistics/random-number-generation

104.1 laplacernd

random number of laplace distribution

104.2 randc

correlate to correlated standard normally distributed vectors

104.3 skewness2param

104.4 skewpdf_central_moments

104.5 skewrnd

random numbers of the skew normal distribution

105 statistics

105.1 range

range and mid range of input

105.2 resample_with_replacement

106 statistics/resampling-statistics/@Jackknife

106.1 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

106.2 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

106.3 matrix1_STATIC

matrix of estimation for leaving out two samples at a time

106.4 matrix2

matrix of estimations for jacknive with two samples left out

107 statistics/resampling-statistics

107.1 block_jackknife

107.2 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

107.3 moving_block_jackknife

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

107.4 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

107.5 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

108 statistics

108.1 scale_quantile_sd

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

108.2 sd_sample_quantiles

108.3 skew_generalized_normal_fit

108.4 skew_generalized_normpdf

108.5 skewcdf

108.6 skewparam_to_central_moments

108.7 skewpdf

skew-normal distribution
c.f. Azzalini 1985

108.8 spatialrnd

108.9 test_mean_generalized_gampdf

108.10 test_skew_generalized_normpdf

108.11 trimmed_mean

trimmed mean

108.12 ttest2_man

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

108.13 ttest_man

two-sample t-test
unequal sample size
equal variance

108.14 ttest_paired

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

108.15 uniformnpdf

108.16 wgeomean

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

108.17 wgeovar

variance of the weighted geometric mean

108.18 wharmean

weighted harmonic mean

108.19 wharstd

108.20 wharvar

108.21 wnormpdf

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

109 stochastic

109.1 brownian_drift_hitting_probability

109.2 brownian_drift_hitting_probability2

109.3 brownian_field

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

109.4 brownian_motion_1d_acf

109.5 brownian_motion_1d_cov

109.6 brownian_motion_1d_fft

109.7 brownian_motion_1d_fourier

109.8 brownian_motion_1d_interleave

109.9 brownian_motion_1d_laplacian

109.10 brownian_motion_2d_cov

109.11 brownian_motion_2d_fft

109.12 brownian_motion_2d_fft_old

109.13 brownian_motion_2d_fourier

109.14 brownian_motion_2d_interleave

109.15 brownian_motion_2d_interleaving

109.16 brownian_motion_2d_kahunen

109.17 brownian_motion_2d_laplacian

109.18 brownian_motion_with_drift_hitting_probability

110 mathematics

mathematical functions of various kind

110.1 ternary_diagram

111 test/master

111.1 dat_test_lanczos_3d_k_20_n_40

111.2 poisson2d_blk

111.3 qr_implicit_givens_2

111.4 spectral_derivative_2d

111.5 test_2d_eigensolver_hydrogen

111.6 test_2d_refine

111.7 test_3d_eigensolver_hydrogen

111.8 test_FEM

111.9 test_Mesh_3d

111.10 test_arnoldi

111.11 test_arpackc

111.12 test_assemble

111.13 test_assembly_performance

111.14 test_bc_one_sided

111.15 test_compare_solvers

111.16 test_complete

111.17 test_convergence

111.18 test_convergence_b

111.19 test_df_2d

111.20 test_eig_algs

111.21 test_eig_inverse

111.22 test_eigs_lanczos

111.23 test_eigs_lanczos_1

111.24 test_eigs_lanczos_2

111.25 test_eigs_lanczos_performance

111.26 test_fdm

111.27 test_fdm_d_vargrid

111.28 test_fdm_spectral

111.29 test_fem

111.30 test_fem_1d

111.31 test_fem_1d_higher_order

111.32 test_fem_2d_adaptive

111.33 test_fem_2d_higher_order

111.34 test_fem_3d_higher_order

111.35 test_fem_3d_refine

111.36 test_fem_b

111.37 test_fem_derivative

111.38 test_fem_quadrature

111.39 test_final

111.40 test_fix_substitution

111.41 test_forward

111.42 test_get_sparse_arrays

111.43 test_harmonic_oscillator

111.44 test_high_order_fdm_periodic_bc

111.45 test_hydrogen_wf

111.46 test_ichol

111.47 test_interpolation

111.48 test_inverse_problem

111.49 test_it_vs_exact

111.50 test_jama

111.51 test_jd

111.52 test_jdqz

111.53 test_lanczos_2

111.54 test_lanczos_biorthogonal

111.55 test_laplacian

111.56 test_laplacian_non_uniform

111.57 test_laplacian_simple

111.58 test_mesh_2d_uniform

111.59 test_mesh_2d_uniform_2

111.60 test_mesh_circle

111.61 test_mesh_generation

111.62 test_mesh_interpolate

111.63 test_mg

111.64 test_minres_recycle

111.65 test_multigrid

111.66 test_nc

111.67 test_nonuniform_symmetric

111.68 test_pde

111.69 test_permutation

111.70 test_poisson_fem

111.71 test_polar

111.72 test_potential

111.73 test_powers

111.74 test_precondition

111.75 test_project_rectangle

111.76 test_qr

111.77 test_quantum_well

111.78 test_radial_adaptive

111.79 test_radial_confinement

111.80 test_radial_fixes

111.81 test_refine_2d

111.82 test_refine_2d_b

111.83 test_refine_3d

111.84 test_refine_structural

111.85 test_regularisation

111.86 test_round_off

111.87 test_schrödinger_potentials

111.88 test_uniform_mesh

111.89 test_vargrid

112 test/signal-processing/autocorrelation

112.1 test_acf

112.2 test_acf_bias

112.3 test_acf_brownian_phase

112.4 test_acfar1_2

112.5 test_acfar1_3

112.6 test_acfar1_4

112.7 test_acfar2

112.8 test_ar1_var_factor

112.9 test_ar1_var_factor_2

112.10 test_ar1_var_mu_single_sample

112.11 test_ar1_var_pop

112.12 test_ar1_var_pop_1

112.13 test_ar1delay

112.14 test_ar2

113 test/signal-processing/passes

113.1 test_bandpass2d_ideal

113.2 test_lowpass1d_fft

113.3 test_lowpass1d_implicit

113.4 test_lowpass2d_anisotropic

113.5 test_lowpass2d_fft

113.6 test_lowpass2d_rho

114 test/signal-processing/periodogram

114.1 test_periodicity_test_2d

114.2 test_periodogram_bartlett_se

114.3 test_periodogram_gauss

114.4 test_periodogram_radial

114.5 test_periodogram_test

114.6 test_periodogram_test_periodicity_2d

114.7 test_periogogram_significance

115 test/signal-processing/spectral-density

115.1 test_spectral_density_2

115.2 test_spectral_density_bandpass_2d

115.3 test_spectral_density_bandpass_2d_max2par

115.4 test_spectral_density_bandpass_continuous

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

115.5 test_spectral_density_bandpass_continuous_1

115.6 test_spectral_density_bandpass_maximum

115.7 test_spectral_density_bandpass_scale

115.8 test_spectral_density_bp

115.9 test_spectral_density_bp_2d

115.10 test_spectral_density_bp_approx

115.11 test_spectral_density_brownian_phase

115.12 test_spectral_density_brownian_phase_2d

115.13 test_spectral_density_brownian_phase_across

115.14 test_spectral_density_brownian_phase_across_mode2par

115.15 test_spectral_density_brownian_phase_mode

115.16 test_spectral_density_brownian_phase_mode2par

115.17 test_spectral_density_brownian_phase_scale

115.18 test_spectral_density_flat

115.19 test_spectral_density_hp_cos

115.20 test_spectral_density_lorentzian_max

115.21 test_spectral_density_lorentzian_scale

115.22 test_spectral_density_lowpass

115.23 test_spectral_density_lowpass_continuous

115.24 test_spectral_density_lowpass_continuous_1

115.25 test_spectral_density_maxiumum_bias_corrected

116 test/spatial-pattern-analysis

116.1 test_approximate_ratio_distribution

116.2 test_approximate_ratio_quantile

116.3 test_spearate_isotropic_density

117 test/statistics/moment-statistics

117.1 test_wmean

118 test/statistics

118.1 test_fisher_moment2param

119 test/stochastics

119.1 test_brownian_surface

120 test

120.1 test_S

120.2 test_advect_analytic

120.3 test_asymbp

120.4 test_bandpass2d

120.5 test_bandwidth

120.6 test_bartlett_angle

120.7 test_bartlett_distribution

120.8 test_bartlett_expansion

120.9 test_beta

120.10 test_betainc

120.11 test_bivariate_covariance_term

120.12 test_brownian_drift_hitting_probability

120.13 test_brownian_drift_hitting_probability2

120.14 test_brownian_motion_1d

120.15 test_brownian_motion_2d_cov

120.16 test_brownian_motion_2d_fft

120.17 test_brownian_noise_1d

120.18 test_brownian_noise_2d

120.19 test_brownian_noise_interleave

120.20 test_coherence

120.21 test_combined_spectral_density

120.22 test_continuous_fourier_transform

120.23 test_convexity

120.24 test_d2

120.25 test_determine_phase_shift

120.26 test_diffuse_analytic

120.27 test_diffusion_matrix

120.28 test_ellipse

120.29 test_error_propagation_fraction

120.30 test_f

120.31 test_f2

120.32 test_fit_2d_spectral_density

120.33 test_fourier

120.34 test_fourier_derivative

120.35 test_fourier_derivative_1

120.36 test_fourier_integral

120.37 test_fourier_mask_covariance_matrix

120.38 test_ft_bp

120.39 test_gam

120.40 test_gamma_distribution

120.41 test_gaussfit3

120.42 test_gaussian_flat

120.43 test_geoserr

120.44 test_hexagonal_pattern

120.45 test_iafrate

120.46 test_implicit_ode

120.47 test_imrotmat

120.48 test_integration

120.49 test_ivp

120.50 test_jacobian

120.51 test_lanczoswin

120.52 test_laplacian_power

120.53 test_lognfit_quantile

120.54 test_ls_perpendicular_offset

120.55 test_madcorr

120.56 test_mask

120.57 test_max_normal

120.58 test_moments

120.59 test_moments_fourier_power

120.60 test_mtimes3x3

120.61 test_noisy_oscillator

120.62 test_nonperiodic_pattern

120.63 test_normaliztation

120.64 test_ols

120.65 test_parcorr

120.66 test_positivity_preserving

120.67 test_randar1

120.68 test_randar1_multivariate

120.69 test_randar2

120.70 test_ratio_distributions

120.71 test_sd_rectwin

120.72 test_spatialrnd

120.73 test_spectrum_additivity

120.74 test_stationarity

120.75 test_stationarity2

120.76 test_sum_ij

120.77 test_sum_multivar

120.78 test_trifilt1

120.79 test_wautocorr

120.80 test_wavelet_transform

120.81 test_whittle

120.82 test_window

120.83 test_wordfilt

120.84 test_xar1_mid_term

121 mathematics

mathematical functions of various kind

121.1 trapezoidal_fixed

122 wavelet

122.1 contiuous_wavelet_transform

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

122.2 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

122.3 cwt_man2

122.4 example_wavelets

122.5 phasewrap

wrap the phase to +/- pi

122.6 test_cwt_man

122.7 test_phasewrap

122.8 test_wavelet

122.9 test_wavelet2

122.10 test_wavelet_analysis

122.11 test_wavelet_reconstruct

122.12 test_wtc

122.13 wavelet

wavelet windows

122.14 wavelet_reconstruct

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

122.15 wavelet_transform

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