

Manual for Package: mathematics

Karl Kastner

October 8, 2019

Contents

1	complex-analysis	1
1.1	complex_exp_product_im_im	1
1.2	complex_exp_product_im_re	1
1.3	complex_exp_product_re_im	2
1.4	complex_exp_product_re_re	2
1.5	croots	2
1.6	root_complex	3
1.7	test_imroots	3
2	derivation	3
2.1	derive_acfar1	3
2.2	derive_ar2param	3
2.3	derive_arc_length	3
2.4	derive_fourier_power	3
2.5	derive_fourier_power_exp	3
2.6	derive_laplacian_curvilinear	3
2.7	derive_laplacian_fourier_pieewise_linear	3
2.8	derive_logtripdf	4
2.9	derive_smooth1d_parametric	4
2.10	simplify_atan	4
3	fourier/@STFT	4
3.1	STFT	4
3.2	itransform	4
3.3	stft_	4
3.4	stftmat	4
3.5	transform	5
4	fourier	5
4.1	amplitude_from_peak	5
4.2	dftmtx_man	5

4.3	example_fourier_window	5
4.4	fft_derivative	6
4.5	fft_man	6
4.6	fftsmooth	6
4.7	fix_fourier	7
4.8	fourier_axis	7
4.9	fourier_coefficient_piecewise_linear	7
4.10	fourier_coefficient_piecewise_linear_1	8
4.11	fourier_coefficient_ramp3	8
4.12	fourier_coefficient_ramp_pulse	8
4.13	fourier_coefficient_ramp_step	8
4.14	fourier_coefficient_square_pulse	8
4.15	fourier_derivative	8
4.16	fourier_expand	9
4.17	fourier_fit	9
4.18	fourier_interpolate	9
4.19	fourier_matrix	9
4.20	fourier_matrix2	9
4.21	fourier_matrix3	9
4.22	fourier_power	9
4.23	fourier_power_exp	10
4.24	fourier_predict	10
4.25	fourier_range	10
4.26	fourier_regress	10
4.27	fourier_resampled_fit	11
4.28	fourier_resampled_predict	11
4.29	fourier_signed_square	11
4.30	fourier_transform	11
4.31	hyperbolic_fourier_box	11
4.32	idftmtx_man	12
4.33	laplace_2d_pwlinear	12
4.34	nanfft	12
4.35	peaks	12
4.36	roots_fourier	13
4.37	spectral_density	13
4.38	test_complex_exp_product	13
4.39	test_idftmtx	13
5	geometry/@Geometry	13
5.1	Geometry	13
5.2	arclength	13
5.3	arclength_old	14
5.4	arclength_old2	14
5.5	base_point	14

5.6	base_point_limited	14
5.7	centroid	14
5.8	cross2	14
5.9	curvature	14
5.10	ddot	15
5.11	distance	15
5.12	distance2	15
5.13	dot	15
5.14	edge_length	15
5.15	enclosed_angle	15
5.16	enclosing_triangle	15
5.17	hexagon	15
5.18	inPolygon	16
5.19	inTetra	16
5.20	inTetra2	16
5.21	inTriangle	16
5.22	intersect	16
5.23	lineintersect	16
5.24	lineintersect1	16
5.25	minimum_distance_lines	16
5.26	mittenpunkt	17
5.27	nagelpoint	17
5.28	onLine	17
5.29	orthocentre	17
5.30	plumb_line	17
5.31	poly_area	17
5.32	poly_edges	17
5.33	poly_set	17
5.34	poly_width	18
5.35	polyxpoly	18
5.36	project_to_curve	18
5.37	random_disk	18
5.38	random_simplex	18
5.39	sphere_volume	18
5.40	tetra_volume	18
5.41	tobarycentric	19
5.42	tobarycentric1	19
5.43	tobarycentric2	19
5.44	tobarycentric3	19
5.45	tri_angle	19
5.46	tri_area	19
5.47	tri_centroid	19
5.48	tri_distance_opposit_midpoint	19
5.49	tri_edge_length	20

5.50	tri_edge_midpoint	20
5.51	tri_excircle	20
5.52	tri_height	20
5.53	tri_incircle	20
5.54	tri_isacute	20
5.55	tri_isobtuse	20
5.56	tri_semiperimeter	20
5.57	tri_side_length	21
6	geometry	21
6.1	Polygon	21
6.2	bounding_box	21
6.3	curvature_1d	21
6.4	cvt	21
6.5	deg_to_frac	21
6.6	ellipse	22
6.7	ellipseX	22
6.8	ellipseY	22
6.9	first_intersect	22
6.10	golden_ratio	22
6.11	hypot3	22
6.12	meanangle	22
6.13	meanangle2	22
6.14	meanangle3	23
6.15	meanangle4	23
6.16	medianangle	23
6.17	medianangle2	23
6.18	pilim	23
6.19	streamline_radius_of_curvature	23
7	linear-algebra	24
7.1	averaging_matrix_2	24
7.2	colnorm	24
7.3	condest_	24
8	linear-algebra/coordinate-transformation	24
8.1	barycentric2cartesian	24
8.2	barycentric2cartesian3	24
8.3	cartesian2barycentric	24
8.4	cartesian_to_unit_triangle_basis	24
8.5	latlon2utm	25
8.6	lowrance_mercator_to_wgs84	25
8.7	nmea2utm	25
8.8	sn2xy	25

8.9	unit_triangle_to_cartesian	25
8.10	utm2latlon	25
8.11	xy2nt	25
8.12	xy2sn	26
8.13	xy2sn_java	26
8.14	xy2sn_old	26
9	linear-algebra	26
9.1	det2x2	26
9.2	det3x3	26
9.3	det4x4	26
9.4	diag2x2	26
9.5	eig2x2	27
9.6	first	27
9.7	gershgorin_circle	27
9.8	haussdorff	27
9.9	ieig2x2	27
9.10	inv2x2	27
9.11	inv3x3	27
9.12	inv4x4	28
9.13	lpmean	28
9.14	lpnorm	28
9.15	matvec3	28
9.16	max2d	28
9.17	mpoweri	28
9.18	mtimes2x2	28
9.19	mtimes3x3	28
9.20	nannorm	29
9.21	nanshift	29
9.22	nl	29
9.23	normalise	29
9.24	normalize1	29
9.25	normrows	29
9.26	orth2	29
9.27	orth_man	30
9.28	orthogonalise	30
9.29	paddext	30
9.30	paddval1	30
9.31	paddval2	30
10	linear-algebra/polynomial	30
10.1	chebychev	30
10.2	piecewise_polynomial	30
10.3	roots1	31

10.4	roots2	31
10.5	vanderi_1d	31
10.6	vandermonde	31
11	linear-algebra	31
11.1	randrot	31
11.2	right	31
11.3	rot2	31
11.4	rot2dir	32
11.5	rot3	32
11.6	rownorm	32
11.7	simmilarity_matrix	32
11.8	spnorm	32
11.9	spzeros	32
11.10	transpose3	32
11.11	transposeall	32
12	logic	33
12.1	bitor_man	33
13	number-theory	33
13.1	ceiln	33
13.2	digitsb	33
13.3	floorn	33
13.4	iseven	33
13.5	multichoosek	33
13.6	nchoosek_man	34
13.7	pythagorean_triple	34
13.8	roundn	34
14	numerical-methods/differentiation	34
14.1	derivative1	34
14.2	derivative2	34
15	numerical-methods/finite-difference	34
15.1	cdiff	34
15.2	cdiffb	35
15.3	cmean	35
15.4	derivative_matrix_1_1d	35
15.5	derivative_matrix_2_1d	35
15.6	derivative_matrix_2d	35
15.7	derivative_matrix_curvilinear	35
15.8	derivative_matrix_curvilinear_2	35
15.9	difference_kernel	36

15.10	distmat	36
15.11	gradpde2d	36
15.12	laplacian	36
15.13	laplacian_fdm	36
15.14	left	36
15.15	lrmean	36
15.16	mid	37
15.17	pwmid	37
15.18	ratio	37
15.19	steplength	37
15.20	swapoddeven	37
15.21	test_derivative_matrix_2d	37
15.22	test_derivative_matrix_curvilinear	37
15.23	test_difference_kernel	37
16	numerical-methods/finite-volume/@Advection	38
16.1	Advection	38
16.2	dot_advection	38
17	numerical-methods/finite-volume/@Burgers	38
17.1	burgers_split	38
17.2	dot_burgers_fdm	38
17.3	dot_burgers_fft	38
18	numerical-methods/finite-volume/@Finite_Volume	38
18.1	Finite_Volume	38
18.2	apply_bc	39
18.3	solve	39
18.4	step_split_strang	39
18.5	step_unsplit	39
19	numerical-methods/finite-volume/@Flux_Limiter	39
19.1	Flux_Limiter	39
19.2	beam_warming	39
19.3	fromm	40
19.4	lax_wendroff	40
19.5	minmod	40
19.6	monotized_central	40
19.7	muscl	40
19.8	superbee	40
19.9	upwind	40
19.10	vanLeer	41
20	numerical-methods/finite-volume/@KDV	41

20.1	dot_kdv_fdm	41
20.2	dot_kdv_fft	41
20.3	kdv_split	41
21	numerical-methods/finite-volume/@Reconstruct_Average_Evolve	41
21.1	Reconstruct_Average_Evolve	41
21.2	advect_highres	42
21.3	advect_lowress	42
22	numerical-methods/finite-volume	42
22.1	Godunov	42
22.2	Lax_Friedrich	42
22.3	Measure	42
22.4	Roe	42
22.5	fv_swe	43
22.6	staggered_euler	43
22.7	staggered_grid	43
23	numerical-methods	43
23.1	grid2quad	43
24	numerical-methods/integration	43
24.1	cumintL	43
24.2	cumintR	43
24.3	int_trapezoidal	43
25	numerical-methods/interpolation/@Kriging	44
25.1	Kriging	44
25.2	estimate_semivariance	44
25.3	interpolate_	44
26	numerical-methods/interpolation/@RegularizedInterpolator1	44
26.1	RegularizedInterpolator1	44
26.2	init	44
27	numerical-methods/interpolation/@RegularizedInterpolator2	45
27.1	RegularizedInterpolator2	45
27.2	init	45
28	numerical-methods/interpolation/@RegularizedInterpolator3	45
28.1	RegularizedInterpolator3	45
28.2	init	45
29	numerical-methods/interpolation	45
29.1	IDW	45

29.2	IPoly	45
29.3	IRBM	46
29.4	ISparse	46
29.5	Inn	46
29.6	Interpolator	46
29.7	fixnan	46
29.8	idw1	46
29.9	idw2	46
29.10	inner2outer	47
29.11	inner2outer2	47
29.12	interp1_limited	47
29.13	interp1_man	47
29.14	interp1_save	47
29.15	interp1_smooth	47
29.16	interp1_unique	48
29.17	interp2_man	48
29.18	interp_angle	48
29.19	interp_fourier	48
29.20	interp_fourier_batch	48
29.21	interp_sn	48
29.22	interp_sn2	48
29.23	interp_sn3	49
29.24	interp_sn_	49
29.25	limit_by_distance_1d	49
29.26	resample1	49
29.27	resample_d_min	49
29.28	resample_vector	49
29.29	test_interp1_limited	49
30	numerical-methods	49
30.1	inverse_complex	49
31	numerical-methods/ode	50
31.1	bvp2c	50
31.2	bvp2c2	50
31.3	bvp2fdm	51
31.4	bvp2wavetrain	51
31.5	bvp2wavetwopass	51
31.6	ivp_euler_forward	51
31.7	ivprk2	51
31.8	ode2_matrix	52
31.9	ode2characteristic	52
31.10	step_trapezoidal	52

32	numerical-methods/optimisation	52
32.1	armijo_stopping_criterion	52
32.2	astar	52
32.3	binsearch	52
32.4	bisection	52
32.5	box1	53
32.6	box2	53
32.7	cauchy	53
32.8	cauchy2	53
32.9	directional_derivative	53
32.10	dud	53
32.11	extreme3	53
32.12	extreme_quadratic	54
32.13	ftest	54
32.14	grad	54
32.15	hessian	54
32.16	hessian_from_gradient	54
32.17	hessian_projected	54
32.18	line_search	55
32.19	line_search2	55
32.20	line_search_polynomial	55
32.21	line_search_polynomial2	55
32.22	line_search_quadratic	56
32.23	line_search_quadratic2	56
32.24	line_search_wolfe	56
32.25	ls_bgfs	56
32.26	ls_broyden	56
32.27	ls_generalized_secant	57
32.28	nlcg	57
32.29	nlls	57
32.30	picard	57
32.31	poly_extrema	57
32.32	quadratic_function	57
32.33	quadratic_programming	57
32.34	quadratic_step	58
32.35	rosenbrock	58
32.36	sqrt_heron	58
32.37	test_directional_derivative	58
32.38	test_dud	58
32.39	test_line_search_quadratic2	58
32.40	test_ls_generalized_secant	58
32.41	test_nlcg_6_order	58
32.42	test_nlls	58

33	numerical-methods/piecewise-polynomials	59
33.1	Hermite1	59
33.2	hp2_fit	59
33.3	hp2_predict	59
33.4	hp_predict	59
33.5	hp_regress	59
33.6	lp_count	59
33.7	lp_predict	60
33.8	lp_regress	60
33.9	lp_regress_	60
34	regression/@PolyOLS	60
34.1	PolyOLS	60
34.2	coefftest	60
34.3	detrend	60
34.4	fit	60
34.5	fit_	60
34.6	predict	61
34.7	predict_	61
34.8	slope	61
35	regression/@PowerLS	61
35.1	PowerLS	61
35.2	fit	61
35.3	predict	61
35.4	predict_	61
36	regression/@Theil	62
36.1	Theil	62
36.2	detrend	62
36.3	fit	62
36.4	predict	62
36.5	slope	62
37	regression	62
37.1	Theil_Multivariate	62
37.2	areg	63
37.3	ginireg	63
37.4	hesssimplereg	63
37.5	l1lin	63
37.6	lsq_sparam	63
37.7	polyfitd	63
37.8	regression_method_of_moments	64
37.9	robustlinreg	64

37.10	theil2	64
37.11	theil_generalised	64
37.12	total_least_squares	64
37.13	weighted_median_regression	64
38	set-theory	65
38.1	issubset	65
39	signal-processing	65
39.1	acf_effective_sample_size	65
39.2	acf_genton	65
39.3	acfar1	65
39.4	acfar1_2	65
39.5	acfar2	65
39.6	acfar2_2	66
39.7	ar1_cutoff_frequency	66
39.8	ar1_effective_sample_size	66
39.9	ar1_mse_mu_single_sample	66
39.10	ar1_mse_pop	66
39.11	ar1_mse_range	66
39.12	ar1_spectrum	66
39.13	ar1_to_tikhonov	66
39.14	ar1_var_factor	67
39.15	ar1_var_factor_	67
39.16	ar1_var_range2	67
39.17	ar1delay	67
39.18	ar1delay_old	67
39.19	ar2conv	67
39.20	ar2dof	68
39.21	ar2param	68
39.22	asymwin	68
39.23	autocorr_fft	68
39.24	bandpass	68
39.25	bandpass2	68
39.26	bartlett	68
39.27	bartlett_spectrogram	69
39.28	bin1d	69
39.29	bin2d	69
39.30	binormrnd	69
39.31	conv1_man	69
39.32	conv2_man	69
39.33	conv2z	69
39.34	conv30	70
39.35	conv_	70

39.36	conv_centered	70
39.37	convz	70
39.38	cosexpdelay	70
39.39	csmooth	70
39.40	daniell_window	70
39.41	danielle_window	71
39.42	db2neper	71
39.43	db2power	71
39.44	derive_danielle_weight	71
39.45	derive_limit_0_acfar	71
39.46	detect_peak	71
39.47	digital_low_pass_filter	71
39.48	doublesum_ij	72
39.49	effective_sample_size_to_ar1	72
39.50	filt_hodges_lehman	72
39.51	filter1	72
39.52	filter2	72
39.53	filter_	72
39.54	filteriir	72
39.55	filterp	73
39.56	filterp1	73
39.57	filterstd	73
39.58	firls_man	73
39.59	flattopwin	74
39.60	frequency_response_boxcar	74
39.61	freqz_boxcar	74
39.62	gaussfilt1	74
39.63	hanchangewin	74
39.64	hanchangewin2	74
39.65	hanwin	74
39.66	hanwin_	74
39.67	highpass	75
39.68	kaiserwin	75
39.69	kalman	75
39.70	lanczoswin	75
39.71	last	75
39.72	lowpass	75
39.73	lowpass2	75
39.74	lowpass_iir	75
39.75	lowpass_iir_symmetric	76
39.76	lowpassfilter2	76
39.77	maxfilt1	76
39.78	meanfilt1	76
39.79	medfilt1_man	76

39.80	medfilt1_man2	76
39.81	medfilt1_padded	76
39.82	medfilt1_reduced	76
39.83	mid_term_single_sample	77
39.84	minfilt1	77
39.85	mu2ar1	77
39.86	nanautocorr	77
39.87	nanmedfilt1	77
39.88	neper2db	77
39.89	peaks_man	77
39.90	polyfilt1	78
39.91	qmedfilt1	78
39.92	randar1	78
39.93	randar1_dual	78
39.94	randar2	78
39.95	randarp	78
39.96	range_window	78
39.97	rectwin	78
39.98	recursive_sum	79
39.99	select_range	79
39.100	smooth1d_parametric	79
39.101	smooth2	79
39.102	smooth_man	79
39.103	smooth_parametric	79
39.104	smooth_parametric2	79
39.105	smoothfft	79
39.106	spectrogram	80
39.107	std_window	80
39.108	sum_i_lag	80
39.109	sum_ii	80
39.110	sum_ii_	80
39.111	sum_ij	80
39.112	sum_ij_	80
39.113	sum_ij_partial_	80
39.114	sum_multivar	81
39.115	test_acfar1	81
39.116	test_acfar1_2	81
39.117	test_acfar1_3	81
39.118	test_acfar1_4	81
39.119	test_acfar2	81
39.120	test_ar1_var_factor	81
39.121	test_ar1_var_factor_2	81
39.122	test_ar1_var_mu_single_sample	81
39.123	test_ar1_var_pop	81

39.124	test_ar1_var_pop_1	82
39.125	test_ar1delay	82
39.126	test_bivariate_covariance_term	82
39.127	test_convexity	82
39.128	test_lanczoswin	82
39.129	test_madcorr	82
39.130	test_randar1	82
39.131	test_randar1_multivariate	82
39.132	test_randar2	82
39.133	test_sum_ij	82
39.134	test_sum_multivar	83
39.135	test_trifilt1	83
39.136	test_wautocorr	83
39.137	test_wavelet_transform	83
39.138	test_wordfilt	83
39.139	test_xar1_mid_term	83
39.140	tikhonov_to_ar1	83
39.141	trapwin	83
39.142	trifilt1	83
39.143	triwin	84
39.144	triwin2	84
39.145	varar1	84
39.146	welch_spectrogram	84
39.147	wfilt	84
39.148	winbandpass	84
39.149	window_make_odd	84
39.150	winfilt0	84
39.151	winlength	85
39.152	wmeanfilt	85
39.153	wmedfilt	85
39.154	wordfilt	85
39.155	wordfilt_edgeworth	85
39.156	xcorr_man	85
40	sorting	85
40.1	sort2	85
40.2	sort2d	86
41	special-functions	86
41.1	bessel_sphere	86
41.2	hankel_sphere	86
41.3	hermite	86
41.4	legendre_man	86
41.5	neumann_sphere	86

42	statistics	87
42.1	atan_s2	87
42.2	beta_mode_to_parameter	87
42.3	correlation_confidence_pearson	87
43	statistics/distributions	87
43.1	PDF	87
43.2	binorm_separation_coefficient	87
43.3	binormcdf	87
43.4	binormfit	87
43.5	binormpdf	88
43.6	edgeworth_cdf	88
43.7	edgeworth_pdf	88
43.8	logn_mode2param	88
43.9	logn_param2mode	88
43.10	lognpdf_	88
43.11	pdfsample	88
43.12	t2cdf	89
43.13	t2inv	89
44	statistics	89
44.1	example_standard_error_of_sample_quantiles	89
44.2	f_var_finite	89
44.3	gamma_mode_to_parameter	89
44.4	hodges_lehmann_correlation	89
44.5	hodges_lehmann_dispersion	90
45	statistics/information-theory	90
45.1	akaike_information_criterion	90
45.2	bayesian_information_criterion	90
46	statistics	90
46.1	kurtncdf	90
46.2	kurtnpdf	90
46.3	kurtosis_bias_corrected	91
46.4	limit	91
46.5	logfactorial	91
46.6	loglogpdf	91
46.7	logskewcdf	91
46.8	logskewpdf	91
47	statistics/logu	91
47.1	lambertw_numeric	91
47.2	logtrialtcd	91

47.3	logtrialtinv	92
47.4	logtrialtmean	92
47.5	logtrialtpdf	92
47.6	logtrialtrnd	92
47.7	logtricdf	92
47.8	logtriinv	92
47.9	logtrimean	93
47.10	logtripdf	93
47.11	logtrirnd	93
47.12	logucdf	93
47.13	logucm	93
47.14	loguinv	93
47.15	logumean	93
47.16	logupdf	93
47.17	logurnd	94
47.18	loguvar	94
47.19	medlogu	94
47.20	test_logurnd	94
47.21	tricdf	94
47.22	triinv	94
47.23	trimedian	94
47.24	tripdf	94
47.25	trirnd	95
48 statistics		95
48.1	maxnnormals	95
48.2	midrange	95
48.3	minavg	95
48.4	mode_man	95
49 statistics/moment-statistics		95
49.1	autocorr_man3	95
49.2	autocorr_man4	96
49.3	autocorr_man5	96
49.4	blockserr	96
49.5	comoment	96
49.6	corr_man	96
49.7	cov_man	97
49.8	dof	97
49.9	edgeworth_quantile	97
49.10	effective_sample_size	97
49.11	f_correlation	97
49.12	f_finite	97
49.13	lmean	97

49.14	lmoment	98
49.15	maskmean	98
49.16	masknanmean	98
49.17	mean1	98
49.18	mean_man	98
49.19	mse	98
49.20	nanautocorr_man1	98
49.21	nanautocorr_man2	98
49.22	nanautocorr_man4	99
49.23	nancorr	99
49.24	nancumsum	99
49.25	nanlmean	99
49.26	nanr2	99
49.27	nanrms	99
49.28	nanrmse	99
49.29	nanserr	100
49.30	nanwmean	100
49.31	nanwstd	100
49.32	nanwvar	100
49.33	nanxcorr	100
49.34	pearson	100
49.35	pearson_to_kendall	100
49.36	pool_samples	101
49.37	qmean	101
49.38	range_mean	101
49.39	rmse	101
49.40	serr	101
49.41	serr1	101
49.42	test_qskew	101
49.43	test_qstd_qskew_optimal_p	101
49.44	wautocorr	102
49.45	wcorr	102
49.46	wcov	102
49.47	wdof	102
49.48	wkurt	102
49.49	wmean	102
49.50	wrms	103
49.51	wserr	103
49.52	wskew	103
49.53	wstd	103
49.54	wvar	103

50 statistics 103

50.1	nangeomean	103
------	----------------------	-----

50.2	nangeostd	103
51	statistics/nonparametric-statistics	104
51.1	kernel1d	104
51.2	kernel2d	104
52	statistics	104
52.1	normmoment	104
52.2	normpdf2	104
53	statistics/order-statistics	104
53.1	hodges_lehmann_location	104
53.2	kendall	105
53.3	kendall_to_pearson	105
53.4	mad2sd	105
53.5	madcorr	105
53.6	median2_holder	105
53.7	median_ci	105
53.8	median_man	105
53.9	mediani	106
53.10	nanmadcorr	106
53.11	nanwmedian	106
53.12	nanwquantile	106
53.13	oja_median	106
53.14	qkurtosis	106
53.15	qmoments	107
53.16	qskew	107
53.17	qskewq	107
53.18	qstdq	107
53.19	quantile1_optimisation	107
53.20	quantile2_breckling	107
53.21	quantile2_chaudhuri	107
53.22	quantile2_projected	108
53.23	quantile2_projected2	108
53.24	quantile_envelope	108
53.25	quantile_regression_simple	108
53.26	ranking	108
53.27	spatial_median	108
53.28	spatial_quantile	108
53.29	spatial_quantile2	108
53.30	spatial_quantile3	109
53.31	spatial_rank	109
53.32	spatial_sign	109
53.33	spatial_signed_rank	109

53.34	spearman	109
53.35	spearman_rank	109
53.36	spearman_to_pearson	109
53.37	wmedian	109
53.38	wquantile	110
54	statistics/random-number-generation	110
54.1	laplacernd	110
54.2	randc	110
54.3	skewrnd	110
54.4	skewrnd2	110
55	statistics	110
55.1	range	110
56	statistics/resampling-statistics/@Jackknife	110
56.1	Jackknife	110
56.2	estimated_STATIC	111
56.3	matrix1_STATIC	111
56.4	matrix2	111
57	statistics/resampling-statistics	111
57.1	block_jackknife	111
57.2	jackknife_moments	112
57.3	moving_block_jackknife	112
57.4	randblockserr	112
57.5	resample	112
58	statistics	113
58.1	scale_quantile_sd	113
58.2	skewpdf	113
58.3	trimmed_mean	113
58.4	ttest2_man	113
58.5	ttest_man	113
58.6	ttest_paired	113
58.7	wharmean	114
59	wavelet	114
59.1	contiuous_wavelet_transform	114
59.2	cwt_man	114
59.3	example_wavelets	114
59.4	phasewrap	114
59.5	test_cwt_man	114
59.6	test_phasewrap	114
59.7	test_wavelet	114

59.8	test_wavelet2	115
59.9	test_wavelet_analysis	115
59.10	test_wavelet_reconstruct	115
59.11	test_wtc	115
59.12	wavelet	115
59.13	wavelet_reconstruct	115
59.14	wavelet_transform	115
60		115
60.1	wrapphase	115

1 complex-analysis

operations on complex numbers

1.1 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
```

1.2 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
```

1.3 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
```

1.4 complex_exp_product_re_re

product of the real part of two complex exponentials

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

the product has two frequency components

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

1.5 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
```

1.6 root_complex

root of a complex number

1.7 test_imroots

2 derivation

derivation of several functions by means of symbolic computation

2.1 derive_acfar1

2.2 derive_ar2param

2.3 derive_arc_length

2.4 derive_fourier_power

2.5 derive_fourier_power_exp

2.6 derive_laplacian_curvilinear

2.7 derive_laplacian_fourier_piecewise_linear

2.8 `derive_logtripdf`

2.9 `derive_smooth1d_parametric`

2.10 `simplify_atan`

symbolic simplification of the arcus tangent

3 `fourier/@STFT`

3.1 `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

3.2 `itransform`

inverse of the short time fourier transform

3.3 `stft_`

static wrapper for STFT

3.4 `stftmat`

transformation matrix for the short time fourier transform

3.5 transform

short time fourier transform

4 fourier

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

4.1 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 (!)

4.2 dftmtx_man

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

input :

n : number of samples

nr : number of columns

output :

F : fourier matrix

4.3 example_fourier_window

4.4 `fft_derivative`

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

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

dx : kth-derivative of x

4.5 `fft_man`

fast fourier transform for complex input data

input:
F : data in real space

output :

F : fourier transformation of F

4.6 `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

4.7 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

4.8 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

4.9 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

4.10 `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

4.11 `fourier_coefficient_ramp3`

fourier series coefficient of a ramp

4.12 `fourier_coefficient_ramp_pulse`

fourier series coefficient of a ramp pulses

4.13 `fourier_coefficient_ramp_step`

fourier coefficient of a ramp-step

4.14 `fourier_coefficient_square_pulse`

fourier series coefficients of a square pulse

4.15 `fourier_derivative`

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

4.16 `fourier_expand`

expand values of fourier series

4.17 `fourier_fit`

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

4.18 `fourier_interpolate`

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

4.19 `fourier_matrix`

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

4.20 `fourier_matrix2`

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

4.21 `fourier_matrix3`

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

4.22 `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

4.23 fourier_power_exp

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

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

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

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

4.24 fourier_predict

expand a continuous fourier series at times t

4.25 fourier_range

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

4.26 fourier_regress

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

4.27 `fourier_resampled_fit`

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

4.28 `fourier_resampled_predict`

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

4.29 `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

4.30 `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

4.31 `hyperbolic_fourier_box`

4.32 idftmtx_man

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

4.33 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

```

4.34 nanfft

discrete fourier transform of a data series with gaps

4.35 peaks

```

peaks of the power spectrum of a discrete fourier transform

rule for peaks: there is no higher value left or right of the "peak
"
                until the signal drops to p*y_peak, p = 0.5

works best, when spectrum has been smoothened

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

```


output :

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

4.36 roots_fourier

zeros of continuous fourier series series

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

4.37 spectral_density

spectral density

4.38 test_complex_exp_product

4.39 test_idftmtx

5 geometry/@Geometry

5.1 Geometry

5.2 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

5.3 arclength_old

arc length of a two dimensional function

5.4 arclength_old2

arc length of a two dimensional function

5.5 base_point

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

5.6 base_point_limited

base point (Fusspunkt) of a point on a line

5.7 centroid

centroid of a polygone

5.8 cross2

cross product in two dimensions

5.9 curvature

curvature of a function in two dimensions

5.10 ddot

sum of squares of cos of inner angles of triangle

5.11 distance

euclidan distance between two points

5.12 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

5.13 dot

dot product

5.14 edge_length

edge length

5.15 enclosed_angle

angle enclosed between two lines

5.16 enclosing_triangle

smallest enclosing equilateral triangle with bottom side paralle to
X axis

5.17 hexagon

coordinates of a hexagon, scaled and rotated

5.18 inPolygon

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

5.19 inTetra

flag points contained in tetrahedron

5.20 inTetra2

flag points contained in tetrahedron

5.21 inTriangle

flag points contained in triangle

5.22 intersect

intersect between two lines

5.23 lineintersect

intersect of two lines

5.24 lineintersect1

intersect of two lines

5.25 minimum_distance_lines

minimum distance of two lines in three dimensions

5.26 mittenpunkt

mittenpunkt of a triangle

5.27 nagelpoint

nagelpoint of a triangle

5.28 onLine

5.29 orthocentre

orthocentre of triangle

5.30 plumb_line

5.31 poly_area

area of a polygon

5.32 poly_edges

edges of a polygon

5.33 poly_set

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

5.34 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

5.35 polyxpoly

intersections of two polygons

5.36 project_to_curve

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

5.37 random_disk

draw random points on the unit disk

5.38 random_simplex

random point inside of a triangle

5.39 sphere_volume

volume of a sphere

5.40 tetra_volume

volume of a tetrahedron

5.41 tobarycentric

cartesian to barycentric coordinates

5.42 tobarycentric1

cartesian to barycentric coordinates

5.43 tobarycentric2

cartesian to barycentric coordinates

5.44 tobarycentric3

cartesian to barycentric coordinates

5.45 tri_angle

cos of angles of a triangle

5.46 tri_area

angle of a triangle

5.47 tri_centroid

centroid of a triangle

5.48 tri_distance_opposit_midpoint

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

5.49 `tri_edge_length`

edge length of a triangle

5.50 `tri_edge_midpoint`

mid point of a triangle

5.51 `tri_excircle`

excircle of a triangle

5.52 `tri_height`

height of a triangle

5.53 `tri_incircle`

incircle of a triangle

5.54 `tri_isacute`

flag acute triangles

5.55 `tri_isobtuse`

flag obtuse triangles

5.56 `tri_semiperimeter`

semiperimeter of a triangle

5.57 tri_side_length

edge length of triangle

6 geometry

6.1 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

6.2 bounding_box

bounding box of X

6.3 curvature_1d

curvature of a sampled parametric curve in two dimensions

6.4 cvt

centroidal voronoi tessellation

6.5 deg_to_frac

degree, minutes and seconds to fractions

6.6 ellipse

n-points on an ellipse

6.7 ellipseX

x-coordinates of y-coordinates of an ellipse

6.8 ellipseY

6.9 first_intersect

get first intersection between lines in A and B

6.10 golden_ratio

golden ratio

6.11 hypot3

hypothenuse in 3D

6.12 meanangle

weighted mean of angles

6.13 meanangle2

mean angle

6.14 meanangle3

mean angle

6.15 meanangle4

mean angle

6.16 medianangle

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

6.17 medianangle2

median angle

input
alpha : x*m, [rad] angle

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

6.18 pilim

limit to +- pi

6.19 streamline_radius_of_curvature

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

7 linear-algebra

7.1 averaging_matrix_2

7.2 colnorm

norms of columns

7.3 condest_

estimation of the condition number

8 linear-algebra/coordinate-transformation

8.1 barycentric2cartesian

barycentric to cartesian coordinates

8.2 barycentric2cartesian3

convert barycentric to cartesian coordinates

8.3 cartesian2barycentric

cartesian to barycentric coordinates

8.4 cartesian_to_unit_triangle_basis

transform coordinates into unit triangle

8.5 latlon2utm

transform latitude and longitude to WGS84 UTM

8.6 lowrance_mercator_to_wgs84

convert lowrance coordinates to wgs84

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

8.7 nmea2utm

convert nmea messages to utm coordinates

8.8 sn2xy

convert sn to xy coordinates

8.9 unit_triangle_to_cartesian

transform coordinates in unit triangle to cartesian coordinates

8.10 utm2latlon

convert wgs84 utm to latitude and longitude

8.11 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

8.12 xy2sn

convert cartesian to streamwise coordiantes

8.13 xy2sn.java

use java port for speed up

8.14 xy2sn_old

transform points from cartesian into streamwise coordinates

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

9 linear-algebra

9.1 det2x2

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

9.2 det3x3

determinant of stacked 3x3 matrices

9.3 det4x4

determinant of stacked 4x4 matrices

9.4 diag2x2

diagonal of stacked 2x2 matrices

9.5 eig2x2

eigenvalues of stacked 2x2 matrices

9.6 first

9.7 gershgorin_circle

range of eigenvalues determined by the gershgorin circle theorem

9.8 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

9.9 ieig2x2

reconstruct matrix from eigenvalue decomposition

9.10 inv2x2

2x2 inverse of stacked matrices

9.11 inv3x3

9.12 inv4x4

inverse of stacked 4x4 matrices

9.13 lpmean

mean of pth-power of a

9.14 lpnorm

norm of lth-power of a

9.15 matvec3

matrix-vector product of stacked matrices and vectors

9.16 max2d

maximum value and i-j index for matrix

9.17 mpoweri

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

9.18 mtimes2x2

9.19 mtimes3x3

product of stacked 3x3 matrices

9.20 nannorm

norm of a vector, skips nan-values

9.21 nanshift

shift vector, but set out of range values to NaN

9.22 nl

number rows (lines) of a matrix

analogue to unix nl command

9.23 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)

9.24 normalize1

normalize columns in x to [-1,1]

9.25 normrows

9.26 orth2

make matrix A orhogonal to B

9.27 `orth_man`

orthogonalize the columns of A

9.28 `orthogonalise`

make x orthogonal to Y

9.29 `paddext`

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

9.30 `paddval1`

padd values at end of x

9.31 `paddval2`

padd values to x

10 linear-algebra/polynomial

10.1 `chebychev`

chebycheff polynomials

10.2 `piecewise_polynomial`

evaluate piecewise polynomial

10.3 roots1

roots of linear functions

10.4 roots2

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

10.5 vanderi_1d

vandermonde matrix of an integral

10.6 vandermonde

van der monde matrix

11 linear-algebra

11.1 randrot

random rotation matrix

11.2 right

get right column by shifting columns to left
extrapolate rightmost column

11.3 rot2

rotation matrix from angle

11.4 rot2dir

rotation matrix from direction vector

11.5 rot3

11.6 rownorm

11.7 simmilarity_matrix

11.8 spnorm

frobenius norm

11.9 spzeros

allocate a sparze matrix of zeros

11.10 transpose3

transpose stacked 3x3 matrices

11.11 transposeall

12 logic

bitwise operations on integers

12.1 bitor_man

bitwise OR of the numbers of the columns of A

input:

A (positive integer)

13 number-theory

13.1 ceiln

floor to leading n-digits

13.2 digitsb

number of digits with respect to specified base

13.3 floorn

floor to n-digits

13.4 iseven

true for even numbers in X

13.5 multichoosek

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

input :

x : scalar integer or vector of arbitrary numbers

k : length of subsets

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

13.6 nchoosek_man

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

13.7 pythagorean_triple

```
pythagorean triple
```

13.8 roundn

```
round to n digits
```

14 numerical-methods/differentiation

14.1 derivative1

```
first derivative on variable mesh  
second order accurate
```

14.2 derivative2

```
second derivative on a variable mesh
```

15 numerical-methods/finite-difference

15.1 cdiff

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

15.2 cdiffb

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

15.3 cmean

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

15.4 derivative_matrix_1_1d

finite difference matrix of first derivative in one dimensions

15.5 derivative_matrix_2_1d

finite derivative matrix of second derivative in one dimension

15.6 derivative_matrix_2d

finite difference derivative matrix in two dimensions

15.7 derivative_matrix_curvilinear

derivative matrix on a curvilinear grid

15.8 derivative_matrix_curvilinear_2

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

15.9 difference_kernel

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

15.10 distmat

distance matrix for a 2 dimensional rectangular matrix

15.11 gradpde2d

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

15.12 laplacian

15.13 laplacian_fdm

finite difference matrix of the laplacian
BC

15.14 left

left element of vector, leftmost column is extrapolated

15.15 lrmean

mean of the left and right element

15.16 mid

mid point between neighbouring vector elements

15.17 pwmid

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

15.18 ratio

ratio of two subsequent values

15.19 steplength

step length of a vector if it were equispaced

15.20 swapoddeven

swap odd and even elements in a vector

15.21 test_derivative_matrix_2d

15.22 test_derivative_matrix_curvilinear

15.23 test_difference_kernel

16 numerical-methods/finite-volume/@Advection

16.1 Advection

FVM treatment of the Advection equation

16.2 dot_advection

advection equation

17 numerical-methods/finite-volume/@Burgers

17.1 burgers_split

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

17.2 dot_burgers_fdm

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

17.3 dot_burgers_fft

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

18 numerical-methods/finite-volume/@Finite_Volume

18.1 Finite_Volume

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

18.2 apply_bc

apply boundary conditions

18.3 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);
```

18.4 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

18.5 step_unsplit

step in time, without splitting the inhomogeneous term

19 numerical-methods/finite-volume/@Flux_Limiter

19.1 Flux_Limiter

class of flux limiters

19.2 beam_warming

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

19.3 fromm

fromme limiter
low res

19.4 lax_wendroff

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

19.5 minmod

min-mod schock limiter

19.6 monotized_central

monotonized central flux limiter

19.7 muscl

muscl flux limiter

19.8 superbee

superbee limiter

19.9 upwind

godunov scheme
godunov, first order accurate

19.10 vanLeer

van Leer limiter

20 numerical-methods/finite-volume/@KDV

20.1 dot_kdv_fdm

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

20.2 dot_kdv_fft

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

20.3 kdv_split

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

21 numerical-methods/finite-volume/@Reconstruct_Average_Evolve

21.1 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;
```

21.2 advect_highres

single time step for the reconstruct evolve algorithm

21.3 advect_lowres

single time step
low resolution

22 numerical-methods/finite-volume

22.1 Godunov

Godunov, upwind method for systems of pdes

22.2 Lax_Friedrich

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

22.3 Measure

22.4 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

22.5 fv_swe

wrapper for solving SWE

22.6 staggered_euler

forward euler method with staggered grid

22.7 staggered_grid

staggered grid approximation to the SWE

23 numerical-methods

23.1 grid2quad

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

24 numerical-methods/integration

24.1 cumintL

cumulative integral from left to right

24.2 cumintR

cumulative integral from right to left

24.3 int_trapezoidal

integrate y along x with the trapezoidal rule

25 numerical-methods/interpolation/@Kriging

25.1 Kriging

```
class for Kriging interpolation
```

25.2 estimate_semivariance

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

25.3 interpolate_

```
interpolate with Krieking method

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

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

26 numerical-methods/interpolation/@RegularizedInterpolator

26.1 RegularizedInterpolator1

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

26.2 init

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


27 numerical-methods/interpolation/@RegularizedInterpolator2

27.1 RegularizedInterpolator2

class for regularized interpolation on an unstructures mesh (
interpolation)

27.2 init

initialize the interpolator with a set of point samples

28 numerical-methods/interpolation/@RegularizedInterpolator3

28.1 RegularizedInterpolator3

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

28.2 init

initialize the interpolator with a set of sampling points

29 numerical-methods/interpolation

29.1 IDW

spatial averaging by inverse distance weighting

29.2 IPoly

polynomial interpolation class

29.3 IRBM

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

29.4 ISparse

```
sparse interpolation class
```

29.5 Inn

```
nearest neighbour interpolation
```

29.6 Interpolator

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

29.7 fixnan

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

29.8 idw1

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

29.9 idw2

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

29.10 inner2outer

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

29.11 inner2outer2

interpolate from element (segment) centres to edge points

29.12 interp1_limited

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

29.13 interp1_man

interpolate

29.14 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

29.15 interp1_smooth

29.16 interp1_unique

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

29.17 interp2_man

nearest neighbour interpolation in two dimensions

29.18 interp_angle

interpolate an angle

29.19 interp_fourier

interpolation by the fourier method

29.20 interp_fourier_batch

batch interpolation by the fourier interpolation

29.21 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_);

29.22 interp_sn2

interpolation in streamwise coordinates

29.23 interp_sn3

29.24 interp_sn_

29.25 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
 $ratio^1 = \exp(a*1)$

29.26 resample1

interpolation along a parametric curve with variable step width

29.27 resample_d_min

resample a function

29.28 resample_vector

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

29.29 test_interp1_limited

30 numerical-methods

30.1 inverse_complex

31 numerical-methods/ode

31.1 bvp2c

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

odefun provides ode coefficients c:
$$c(x,1) y''(x) + c(x,2) y'(x) + c(x,3) y = c(x,4)$$
$$c_1 y'' + c_2 y' + c_3 y + c_4 = c_4$$

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

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

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

31.2 bvp2c2

solve second order boundary value problem via roots of the characteristic polynomial

input:

x : [nx1] discretized domain
n : number of vertices
ns = n-1 : number of segments

bc : struct : boundary condition
$$bc.p(1)*y(0) + bc.pd(2)*y'(0) = bc.val(1)$$
$$bc.p(2)*y(L) + bc.pd(2)*y'(L) = bc.val(2)$$

output:

A : [2*ns x 2*ns] discretisation matrix
rhs : [2*ns x 1] right hand side

$y = A^{-1} rhs$

31.3 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])

31.4 bvp2wavetrain

solve second order boundary value problem by repeated integration

31.5 bvp2wavetwopass

two pass solution for the linearised wave equation

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

31.6 ivp_euler_forward

solve initial value problem by the euler forward method

31.7 ivprk2

solve initial value problem by the two step runge kutta method

31.8 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]
```

31.9 ode2characteristic

second order odes
transmitted and reflected wave

31.10 step_trapezoidal

single trapezoidal step

32 numerical-methods/optimisation

32.1 armijo_stopping_criterion

armijo stopping criterion for optimizations

32.2 astar

astar path finding algorithm

32.3 binsearch

binary search on a line

32.4 bisection

bisection

32.5 box1

test objective function for optimisation routines

32.6 box2

32.7 cauchy

32.8 cauchy2

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

32.9 directional_derivative

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

32.10 dud

optimization by the dud algorithm

32.11 extreme3

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

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

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

32.12 extreme_quadratic

32.13 ftest

32.14 grad

numerical gradient

32.15 hessian

numerical hessian

32.16 hessian_from_gradient

numerical hessian from gradient

32.17 hessian_projected

numerical hessian projected to one dimension

32.18 line_search

bisection routine

32.19 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

32.20 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

32.21 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

32.22 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
```

32.23 line_search_quadratic2

```
quadratic line search
```

32.24 line_search_wolfe

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

32.25 ls_bgfs

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

32.26 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
```

32.27 ls_generalized_secant

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

32.28 nlcg

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

32.29 nlls

non-linear least squares

32.30 picard

picard iteration

32.31 poly_extrema

extrema of a polynomial

32.32 quadratic_function

evaluate quadratic function in higher dimensions

32.33 quadratic_programming

optimize by quadratic programming

32.34 quadratic_step

single step of the quadratic programming

32.35 rosenbrock

rosenbrock test function

32.36 sqrt_heron

Heron's method for the square root

32.37 test_directional_derivative

32.38 test_dud

32.39 test_line_search_quadratic2

32.40 test_ls_generalized_secant

32.41 test_nlcg_6_order

32.42 test_nlls

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

33 numerical-methods/piecewise-polynomials

33.1 Hermite1

hermite polynomial interpolation in 1d

33.2 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)

33.3 hp2_predict

prediction with pw hermite polynomial
c are values at support points

33.4 hp_predict

predict with piecewise hermite polynomial

33.5 hp_regress

fit piecewise hermite polynomial
coefficients are values and derivatives

33.6 lp_count

lagrangian basis for interpolation
count number of valid samples

33.7 `lp_predict`

lagrangian basis piecewise interpolation, predictor

33.8 `lp_regress`

33.9 `lp_regress_`

34 `regression/@PolyOLS`

34.1 `PolyOLS`

class for polynomial least squares

34.2 `coefftest`

34.3 `detrend`

detrending by polynomial regression

34.4 `fit`

fit a polynomial function
like `polyfit`, but returns parameter error estimates

34.5 `fit_`

fit a polynomial function

34.6 predict

predict polynomial function values

34.7 predict_

34.8 slope

slope by linear regression

35 regression/@PowerLS

35.1 PowerLS

class for power law regression

35.2 fit

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

35.3 predict

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

35.4 predict_

36 regression/@Theil

36.1 Theil

Kendal-Theil-Sen robust regression

36.2 detrend

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

36.3 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

36.4 predict

predict values and confidence intervals with the Theil-Sen method

36.5 slope

fit the slope with the Theil-Sen method

37 regression

linear and non-linear regression

37.1 Theil_Multivariate

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

37.2 areg

regression using the pth-fraction of samples with smallest residual

37.3 ginireg

gini regression

37.4 hesssimplereg

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

37.5 l1lin

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

37.6 lsq_sparam

parameter covariance of the least squares regression

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

37.7 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

37.8 regression_method_of_moments

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

37.9 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

37.10 theil2

Theil senn-estimator for two dimensions (glm)

37.11 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

37.12 total_least_squares

total least squares

37.13 weighted_median_regression

weighted median regression
c.f. Scholz, 1978

38 set-theory

38.1 issubset

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

A : first set

B : second set

P : set of primes (auxiliary)

39 signal-processing

39.1 acf_effective_sample_size

effective sample size from acf

39.2 acf_genton

autocorrelation function

39.3 acfar1

Autocorrelation function of the finite AR1 process

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

pause

39.4 acfar1_2

autocorrelation of the ar1 process

39.5 acfar2

impulse response of the ar2 process

39.6 acfar2_2

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

39.7 ar1_cutoff_frequency

39.8 ar1_effective_sample_size

effective sample size correction for autocorrelated series

39.9 ar1_mse_mu_single_sample

standard error of a single sample of an ar1 correlated process

39.10 ar1_mse_pop

variance of the population mean of a single realisation around zero

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

39.11 ar1_mse_range

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

39.12 ar1_spectrum

spectrum of the ar1 process

39.13 ar1_to_tikhonov

convert ar1 correlation to tikhonovs lambda

39.14 ar1_var_factor

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

39.15 ar1_var_factor_

variance of an autocorrelated finite process

39.16 ar1_var_range2

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

39.17 ar1delay

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

39.18 ar1delay_old

autocorrelation of the residual

39.19 ar2conv

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

39.20 ar2dof

effective samples size for the ar2 process

39.21 ar2param

ar2 parameter estimation from first two terms of acf

acf = [1 a1 a2 ...]

39.22 asymwin

creates asymmetrical filter windows

filter will always have negative weights

39.23 autocorr_fft

autocorrelation function

39.24 bandpass

bandpass filter

39.25 bandpass2

bandpass filter

39.26 bartlett

Effective sample size factor for bartlett window

c.f. thiebaux

c.f spectral analysis-jenkins, eq. (6.3.27)

c = acf

note: results seems always to be 1 too low

T : reduction factor for dof

for ar1 with $a = \rho^k = \exp(-k/L)$, $T = 2L$

39.27 bartlett_spectrogram

bartlett spectrogramm
TODO sliding window

39.28 bin1d

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

39.29 bin2d

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

39.30 binormrnd

generate two correlated normally distributed vectors

39.31 conv1_man

convolutions with padding

39.32 conv2_man

convolution in 2d

39.33 conv2z

39.34 conv30

convolve with rectangular window of length n
circular boundaries

39.35 conv_

convolution of a with b

39.36 conv_centered

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

39.37 convz

39.38 cosexpdelay

39.39 csmooth

smooth recursively with [1,2,1]/4 kernel

39.40 daniell_window

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

39.41 danielle_window

danielle fourier window

39.42 db2neper

convert decibel to neper

39.43 db2power

power ratio from db

39.44 derive_danielle_weight

39.45 derive_limit_0_acfar

39.46 detect_peak

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

39.47 digital_low_pass_filter

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

39.48 doublesum_ij

double sum of r^i

39.49 effective_sample_size_to_ar1

convert effective sample size to ar1 correlation

39.50 filt_hodges_lehman

39.51 filter1

filter along one dimension

39.52 filter2

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

39.53 filter_

invalidate values that exceed n-times the robust standard deviation

39.54 filteriir

filter adcp t-n data over time

v : nz,nt : values to be filtered

H : nt,1 : depth of ensemble

last : nt,1 : last bin above bottom that can be sampled without
side lobe interference

nf : scalar : number of reweighted iterations

when samples

- distance to bed is reference (advantageous for near-bed suspended
transport)

TODO for wash load: distance to surface is more relevant
interpolate depending on z

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

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

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

39.55 filterp

39.56 filterp1

fir filter with some fancy extras

39.57 filterstd

39.58 firls_man

design finite impulse response filter by the least squares method

39.59 flattopwin

the flat top window

39.60 frequency_response_boxcar

frequency response of a boxcar filter

39.61 freqz_boxcar

frequency response of a boxcar filter

39.62 gaussfilt1

filter data series with a gaussian window

39.63 hanchangewin

hanning window for change point detection

39.64 hanchangewin2

hanning window for change point detection

39.65 hanwin

hanning filter window

39.66 hanwin_

hanning filter window

39.67 highpass

high pass filter

39.68 kaiserwin

kaiser filter window

39.69 kalman

Kalman filter

39.70 lanczoswin

Lanczos window

39.71 last

lake tail, but for matrices

39.72 lowpass

low pass filter

39.73 lowpass2

design low pass filter with cutoff-frequency f1

39.74 lowpass_iir

iir-low pass

39.75 lowpass_iir_symmetric

two-sided iir low pass filter (for symmetry)

39.76 lowpassfilter2

low-pass filter of data

39.77 maxfilt1

39.78 meanfilt1

moving average filter with special treatment of the boundaries

39.79 medfilt1_man

moving median filter, supports columnwise operation

39.80 medfilt1_man2

moving median filter with special treatment of boundaries

39.81 medfilt1_padded

median filter with padding

39.82 medfilt1_reduced

median filter with padding

39.83 mid_term_single_sample

variance of single sample, mid term

39.84 minfilt1

39.85 mu2ar1

error variance of the mean of the finite length ar1 process

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

39.86 nanautocorr

autocorrelation with nan-values

39.87 nanmedfilt1

medfilt1, skipping nans

39.88 neper2db

convert neper to db

39.89 peaks_man

peaks of a periodogram

39.90 polyfilt1

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

39.91 qmedfilt1

medfilt1, after fitting a quadratic polynomial

39.92 randar1

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

39.93 randar1_dual

draw random variables of two correlated ar1 processes

39.94 randar2

generate ar2 process

39.95 randarp

randomly generate the instance of an ar-p process

39.96 range_window

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

39.97 rectwin

rectangular window

39.98 `recursive_sum`

39.99 `select_range`

39.100 `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

39.101 `smooth2`

smooth vectos of X

39.102 `smooth_man`

39.103 `smooth_parametric`

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

39.104 `smooth_parametric2`

parametrically smooth the curve

39.105 `smoothfft`

filter with fast fourier transform

39.106 spectrogram

spectrogram

39.107 std_window

moving block standard deviation

39.108 sum_i_lag

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

39.109 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

39.110 sum_ii_

39.111 sum_ij

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

39.112 sum_ij_

39.113 sum_ij_partial_

39.114 **sum_multivar**

sum of matrix entries of bivariate ar1 process

39.115 **test_acfar1**

39.116 **test_acfar1_2**

39.117 **test_acfar1_3**

39.118 **test_acfar1_4**

39.119 **test_acfar2**

39.120 **test_ar1_var_factor**

39.121 **test_ar1_var_factor_2**

39.122 **test_ar1_var_mu_single_sample**

39.123 **test_ar1_var_pop**

39.124 test_ar1_var_pop_1

39.125 test_ar1delay

39.126 test_bivariate_covariance_term

39.127 test_convexity

39.128 test_lanczoswin

39.129 test_madcorr

39.130 test_randar1

39.131 test_randar1_multivariate

39.132 test_randar2

39.133 test_sum_ij

39.134 `test_sum_multivar`

39.135 `test_trifilt1`

39.136 `test_wautocorr`

39.137 `test_wavelet_transform`

39.138 `test_wordfilt`

39.139 `test_xar1_mid_term`

39.140 `tikhonov_to_ar1`

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

39.141 `trapwin`

trapezoidal filter window

39.142 `trifilt1`

filter with triangular window

39.143 triwin

triangular filter window

39.144 triwin2

triangular filter window

39.145 varar1

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

39.146 welch_spectrogram

welch spectrogram

39.147 wfilt

filter with window

39.148 winbandpass

filter with bandpass

39.149 window_make_odd

39.150 winfilt0

filter with window

39.151 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

39.152 wmeanfilt

mean filter with window

39.153 wmedfilt

median filter with window

39.154 wordfilt

weighted order filter

39.155 wordfilt_edgeworth

weighed order filter

39.156 xcorr_man

cross correlation of two sampled ar1 processes

40 sorting

40.1 sort2

sort two numbers

40.2 sort2d

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

41 special-functions

41.1 `bessel_sphere`

spherical Bessel function of the first kind

41.2 `hankel_sphere`

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

41.3 `hermite`

probabilistic's hermite polynomial by recurrence relation

input :
n : order
x : value

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

41.4 `legendre_man`

legendre polynomials

41.5 `neumann_sphere`

spherical Neumann function
Bessel function of the second kind

42 statistics

42.1 atan_s2

stadard deviation of the arcus tangens by means of taylor expansion

42.2 beta_mode_to_parameter

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

42.3 correlation_confidence_pearson

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

43 statistics/distributions

43.1 PDF

class for quasi-distributions from a set of sampling points

43.2 binorm_separation_coefficient

separation coefficient of a bimodal normal distribution

43.3 binormcdf

bio-modal gaussian distribution

43.4 binormfit

fit sum of to normal distribution to a histogram

43.5 binormpdf

43.6 edgeworth_cdf

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

43.7 edgeworth_pdf

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

43.8 logn_mode2param

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

43.9 logn_param2mode

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

43.10 lognpdf_

log normal distribution called by modes rather than parameters

43.11 pdfsample

pdf from sample distribution
Note: better use kernel density estimates

43.12 t2cdf

Hotelling's T-squared cumulative distribution

43.13 t2inv

inverse of Hotelling's T-squared cumulative distribution

44 statistics

44.1 example_standard_error_of_sample_quantiles

44.2 f_var_finite

reduction of variance when sampling from a finite population
without replacement

44.3 gamma_mode_to_parameter

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

44.4 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

44.5 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

45 statistics/information-theory

45.1 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

45.2 bayesian_information_criterion

bayesian information criterion

46 statistics

46.1 kurtncdf

46.2 kurtnpdf

46.3 kurtosis_bias_corrected

bias corrected kurtosis

46.4 limit

limit a by lower and upper bound

46.5 logfactorial

approximate log of the factorial

46.6 loglogpdf

46.7 logskewcdf

46.8 logskewpdf

47 statistics/logu

47.1 lambertw_numeric

lambert-w function

47.2 logtrialtcdf

pdf of a logarithmic triangular distribution

47.3 logtrialtinv

inverse of the logarithmic triangular distribution

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

47.4 logtrialtmean

mean of the logarithmic triangular distribution

47.5 logtrialtpdf

density of the logarithmic triangular distribution

47.6 logtrialtrnd

47.7 logtricdf

cumulative distribution of the logarithmic triangular distribution

47.8 logtriinv

invere of the logarithmic triangular distribution

47.9 logtrimean

mean of the logarithmic triangular distribution

47.10 logtripdf

probability density of the logarithmic triangular distribution

47.11 logtirnd

47.12 logucdf

probability density of the logarithmic uniform distribution

47.13 logucm

central moments of the log-uniform distribution

47.14 loguinv

inverse of the log-uniform distribution

47.15 logumean

mean of the log-uniform distribution

47.16 logupdf

pdf of the log uniform distribution

47.17 logurnd

random numbers following a log-uniform distribution

47.18 loguvar

variance of the log-uniform distribution

47.19 medlogu

median of the log-uniform distribution

47.20 test_logurnd

47.21 tricdf

cumulative distribution of the log-triangular distribution

47.22 triinv

inverse of the triangular distribution

47.23 trimedian

median of the triangular distribution

47.24 tripdf

probability density of the triangular distribution

47.25 trirnd

random numbers of the triangular distribution

48 statistics

48.1 maxnnormals

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

48.2 midrange

mid range of columns of X

48.3 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

48.4 mode_man

49 statistics/moment-statistics

49.1 autocorr_man3

autocorrelation of the columns of X

49.2 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

49.3 autocorr_man5

autocorrelation of the columns of X

49.4 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

49.5 comoment

non-central higher order moments of the multivariate normal distribution

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

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

μ : $n \times 1$ mean vector

C : $n \times n$ covariance matrix

k : $n \times 1$ powers of variables in moments

49.6 corr_man

correlation of two vectors

49.7 cov_man

covariance matrix of two vectors

49.8 dof

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

49.9 edgeworth_quantile

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

49.10 effective_sample_size

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

49.11 f_correlation

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

49.12 f_finite

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

49.13 lmean

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

49.14 `lmoment`

l-moment of vector `x`

49.15 `maskmean`

mean of the masked values of `X`

49.16 `masknanmean`

49.17 `mean1`

mean of `x`

49.18 `mean_man`

mean and standard error of `X`

49.19 `mse`

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

49.20 `nanautocorr_man1`

autocorrelation of a vector with nan-values

49.21 `nanautocorr_man2`

autocorrelation of a vector with nan-values

49.22 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

49.23 nancorr

(co)-correlation matrix when samples a NaN

49.24 nancumsum

cumulative sum, setting nan values to zero

49.25 nanlmean

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

49.26 nanr2

coefficient of determination when samples are invalid

49.27 nanrms

root mean square value when sample contains nan-values

49.28 nanrmse

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

49.29 nanserr

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

49.30 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)

49.31 nanwstd

weighed standard deviation

49.32 nanwvar

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

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

49.33 nanxcorr

49.34 pearson

pearson correlation coefficient

49.35 pearson_to_kendall

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

49.36 pool_samples

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

49.37 qmean

trimmed mean

49.38 range_mean

49.39 rmse

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

49.40 serr

standard error of the mean of a set of uncorrelated samples

49.41 serr1

49.42 test_qskew

49.43 test_qstd_qskew_optimal_p

49.44 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

49.45 wcorr

correlation of two vectors when samples are weighted

49.46 wcov

covariance of two vectors when samples are weighted

49.47 wdof

effective degrees of freedom for weighted samples

49.48 wkurt

kurtosis with weighted samples

49.49 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)

49.50 wrms

weighted root mean square error

49.51 wserr

weighted root mean square error

49.52 wskew

skewness of a weighted set of samples

49.53 wstd

weighed standard deviation

49.54 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

50 statistics

50.1 nangeomean

50.2 nangeostd

geometric standard deviation ignoring nan-values

51 statistics/nonparametric-statistics

51.1 kernel1d

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

51.2 kernel2d

kernel density estimate in two dimensions

52 statistics

52.1 normmoment

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

52.2 normpdf2

pdf of the bivariate normal distribution

53 statistics/order-statistics

53.1 hodges_lehmann_location

hodges lehman location estimator

Asymptotic rms efficiency of location estimte:

mean: $1 \text{ s}/\sqrt{n}$
hodges lehman: $\sqrt{\pi/3} * s \sim 1.0233 \text{ s}/\sqrt{n}$
median: $\pi/2 \text{ s}/\sqrt{n} \sim 1.25 \text{ s} / \sqrt{n}$

53.2 kendall

kendall correlation coefficient

53.3 kendall_to_pearson

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

c.f. Kruska, 1985

53.4 mad2sd

transform median absolute deviation to standard deviation
for normal distributed values

53.5 madcorr

proxy correlation by median absolute deviation

53.6 median2_holder

53.7 median_ci

median and its confidence intervals under assumption of normality
 $se_me = \sqrt{1/2 \pi} \cdot 1.25331 \cdot sd/\sqrt{n}$

53.8 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

53.9 mediani

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

53.10 nanmadcorr

proxy correlation by median absolute deviation

53.11 nanwmedian

weighted median, skips nan-values

53.12 nanwquantile

weighted quantile, skips nan values

53.13 oja_median

two dimensional oja median

note: the multivariate median is not unique

oja 1983, for extension to multivariate function, see chaudhri

53.14 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

53.15 qmoments

moments estimated from quantiles

53.16 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

53.17 qskewq

skewness estimated by quantiles

53.18 qstdq

proxy standard deviation determined by quantiles

53.19 quantile1_optimisation

53.20 quantile2_breckling

quantile regression

53.21 quantile2_chaudhuri

quantile regression

53.22 quantile2_projected

quantile in two dimensions

53.23 quantile2_projected2

spatial quantile for chosen direction

53.24 quantile_envelope

53.25 quantile_regression_simple

simple quantile regression

53.26 ranking

ranking for spearman statistics

53.27 spatial_median

c.f. Oja 2008

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

53.28 spatial_quantile

spatial quantile

53.29 spatial_quantile2

spatial quantile

53.30 spatial_quantile3

spatial quantile

53.31 spatial_rank

unsigned rank

53.32 spatial_sign

spatial sign

53.33 spatial_signed_rank

signed rank

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

53.34 spearman

spearman's product moment coefficient

53.35 spearman_rank

53.36 spearman_to_pearson

conversion of spearman rank to person product moment correlation
coefficient

53.37 wmedian

weighted median

53.38 wquantile

weighted quantile

54 statistics/random-number-generation

54.1 laplacernd

random number of laplace distribution

54.2 randc

correlate to correlated standard normally distributed vectors

54.3 skewrnd

random numbers of the skew normal distribution

54.4 skewrnd2

random numbers of the skew normal distribution

55 statistics

55.1 range

mid range

56 statistics/resampling-statistics/@Jackknife

56.1 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

56.2 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

56.3 matrix1_STATIC

matrix of estimation for leaving out two samples at a time

56.4 matrix2

matrix of estimations for jacknive with two samples left out

57 statistics/resampling-statistics

57.1 block_jackknife

57.2 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

57.3 moving_block_jackknife

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

57.4 randblockserr

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

57.5 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

58 statistics

58.1 scale_quantile_sd

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

58.2 skewpdf

skew-normal distribution
c.f. Azzalini 1985

58.3 trimmed_mean

trimmed mean

58.4 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

58.5 ttest_man

two-sample t-test
unequal sample size
equal variance

58.6 ttest_paired

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

58.7 wharmean

weighted harmonic mean

59 wavelet

59.1 continuous_wavelet_transform

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

59.2 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

59.3 example_wavelets

59.4 phasewrap

wrap the phase to +/- pi

59.5 test_cwt_man

59.6 test_phasewrap

59.7 test_wavelet

59.8 test_wavelet2

59.9 test_wavelet_analysis

59.10 test_wavelet_reconstruct

59.11 test_wtc

59.12 wavelet

wavelet windows

59.13 wavelet_reconstruct

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

59.14 wavelet_transform

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

60

60.1 wrapphase