

Manual for Package: mathematics

Revision 2:6M

Karl Kästner

October 29, 2019

Contents

1	mathematics	31
1.1	cast_byte_to_integer	31
2	complex-analysis	31
2.1	complex_exp_product_im_im	31
2.2	complex_exp_product_im_re	32
2.3	complex_exp_product_re_im	32
2.4	complex_exp_product_re_re	32
2.5	croots	33
2.6	root_complex	33
2.7	test_imroots	33
3	derivation	33
3.1	derive_acfar1	33
3.2	derive_ar2param	33
3.3	derive_arc_length	34
3.4	derive_fourier_power	34
3.5	derive_fourier_power_exp	34
3.6	derive_laplacian_curvilinear	34
3.7	derive_laplacian_fourier_pieewise_linear	34
3.8	derive_logtripdf	34
3.9	derive_smooth1d_parametric	34
3.10	simplify_atan	34
4	fourier/@STFT	35
4.1	STFT	35
4.2	itransform	35
4.3	stft_	35
4.4	stftmat	35

4.5	transform	35
5	fourier	35
5.1	amplitude_from_peak	35
5.2	dftmtx_man	36
5.3	example_fourier_window	36
5.4	fft_derivative	36
5.5	fft_man	36
5.6	fftsmooth	37
5.7	fix_fourier	37
5.8	fourier_axis	37
5.9	fourier_coefficient_piecewise_linear	38
5.10	fourier_coefficient_piecewise_linear_1	38
5.11	fourier_coefficient_ramp3	39
5.12	fourier_coefficient_ramp_pulse	39
5.13	fourier_coefficient_ramp_step	39
5.14	fourier_coefficient_square_pulse	39
5.15	fourier_derivative	39
5.16	fourier_expand	39
5.17	fourier_fit	39
5.18	fourier_interpolate	40
5.19	fourier_matrix	40
5.20	fourier_matrix2	40
5.21	fourier_matrix3	40
5.22	fourier_power	40
5.23	fourier_power_exp	40
5.24	fourier_predict	41
5.25	fourier_range	41
5.26	fourier_regress	41
5.27	fourier_resampled_fit	41
5.28	fourier_resampled_predict	41
5.29	fourier_signed_square	42
5.30	fourier_transform	42
5.31	hyperbolic_fourier_box	42
5.32	idftmtx_man	42
5.33	laplace_2d_pwlinear	42
5.34	nanfft	43
5.35	peaks	43
5.36	roots_fourier	43
5.37	spectral_density	44
5.38	test_complex_exp_product	44
5.39	test_idftmtx	44
6	geometry/@Geometry	44

6.1	Geometry	44
6.2	arclength	44
6.3	arclength_old	44
6.4	arclength_old2	45
6.5	base_point	45
6.6	base_point_limited	45
6.7	centroid	45
6.8	cosa_min_max	45
6.9	cross2	45
6.10	curvature	45
6.11	ddot	45
6.12	distance	46
6.13	distance2	46
6.14	dot	46
6.15	edge_length	46
6.16	enclosed_angle	46
6.17	enclosing_triangle	46
6.18	hexagon	46
6.19	inPolygon	47
6.20	inTetra	47
6.21	inTetra2	47
6.22	inTriangle	47
6.23	intersect	47
6.24	lineintersect	47
6.25	lineintersect1	47
6.26	minimum_distance_lines	47
6.27	mittenpunkt	48
6.28	nagelpoint	48
6.29	onLine	48
6.30	orthocentre	48
6.31	plumb_line	48
6.32	poly_area	48
6.33	poly_edges	48
6.34	poly_set	48
6.35	poly_width	49
6.36	polyxpoly	49
6.37	project_to_curve	49
6.38	random_disk	49
6.39	random_simplex	49
6.40	sphere_volume	49
6.41	tetra_volume	49
6.42	tobarycentric	50
6.43	tobarycentric1	50
6.44	tobarycentric2	50

6.45	tobarycentric3	50
6.46	tri_angle	50
6.47	tri_area	50
6.48	tri_centroid	50
6.49	tri_distance_opposit_midpoint	50
6.50	tri_edge_length	51
6.51	tri_edge_midpoint	51
6.52	tri_excircle	51
6.53	tri_height	51
6.54	tri_incircle	51
6.55	tri_isacute	51
6.56	tri_isobtuse	51
6.57	tri_semiperimeter	51
6.58	tri_side_length	52
7	geometry	52
7.1	Polygon	52
7.2	bounding_box	52
7.3	curvature_1d	52
7.4	cvt	52
7.5	deg_to_frac	52
7.6	ellipse	53
7.7	ellipseX	53
7.8	ellipseY	53
7.9	first_intersect	53
7.10	golden_ratio	53
7.11	hypot3	53
7.12	meanangle	53
7.13	meanangle2	53
7.14	meanangle3	54
7.15	meanangle4	54
7.16	medianangle	54
7.17	medianangle2	54
7.18	pilim	54
7.19	streamline_radius_of_curvature	54
8	linear-algebra	55
8.1	averaging_matrix_2	55
8.2	colnorm	55
8.3	condest_	55
9	linear-algebra/coordinate-transformation	55
9.1	barycentric2cartesian	55
9.2	barycentric2cartesian3	55

9.3	cartesian2barycentric	55
9.4	cartesian_to_unit_triangle_basis	55
9.5	example_approximate_utm_conversion	55
9.6	latlon2utm	56
9.7	latlon2utm_simple	56
9.8	lowrance_mercator_to_wgs84	56
9.9	nmea2utm	56
9.10	sn2xy	56
9.11	unit_triangle_to_cartesian	56
9.12	utm2latlon	56
9.13	xy2nt	57
9.14	xy2sn	57
9.15	xy2sn_java	57
9.16	xy2sn_old	57
10	linear-algebra	57
10.1	det2x2	57
10.2	det3x3	57
10.3	det4x4	58
10.4	diag2x2	58
10.5	eig2x2	58
10.6	first	58
10.7	gershgorin_circle	58
10.8	hausdorff	58
10.9	ieig2x2	58
10.10	inv2x2	59
10.11	inv3x3	59
10.12	inv4x4	59
10.13	lpmean	59
10.14	lpnorm	59
10.15	matvec3	59
10.16	max2d	59
10.17	mpoweri	59
10.18	mtimes2x2	60
10.19	mtimes3x3	60
10.20	nannorm	60
10.21	nanshift	60
10.22	nl	60
10.23	normalise	60
10.24	normalize1	60
10.25	normrows	61
10.26	orth2	61
10.27	orth_man	61
10.28	orthogonalise	61

10.29	paddext	61
10.30	paddval1	61
10.31	paddval2	61
11	linear-algebra/polynomial	62
11.1	chebychev	62
11.2	piecewise_polynomial	62
11.3	roots1	62
11.4	roots2	62
11.5	vanderi_1d	62
12	linear-algebra	62
12.1	randrot	62
12.2	right	62
12.3	rot2	63
12.4	rot2dir	63
12.5	rot3	63
12.6	rownorm	63
12.7	simmilarity_matrix	63
12.8	spnorm	63
12.9	spzeros	63
12.10	transpose3	63
12.11	transposeall	63
13	logic	64
13.1	bitor_man	64
14	master/derive	64
14.1	derive_bc_one_sided	64
14.2	derive_convergence	64
14.3	derive_error_fdm	64
14.4	derive_fdm_poly	64
14.5	derive_fdm_power	64
14.6	derive_fdm_taylor	64
14.7	derive_fdm_vargrid	64
14.8	derive_fem_2d_mass	65
14.9	derive_fem_error_2d	65
14.10	derive_fem_error_3d	65
14.11	derive_fem_sym_2d	65
14.12	derive_grid_constants	65
14.13	derive_interpolation	65
14.14	derive_laplacian	65
14.15	derive_limit	65
14.16	derive_nc_1d	65

14.17	derive_nc_1d_	65
14.18	derive_nc_2d	66
14.19	derive_nonuniform_symmetric	66
14.20	derive_richardson	66
14.21	derive_sum	66
14.22	nn	66
14.23	test_derive	66
14.24	test_derive_fdm_poly	66
14.25	test_filter	66
14.26	test_vargrid	66
15	master/eigenvalue	67
15.1	eig_bisection	67
15.2	eig_inverse	67
15.3	eig_inverse_iteration	67
15.4	eig_power_iteration	67
16	master/eigenvalue/jacobi-davidson/JDQR	67
16.1	Example1	67
16.2	Example2	67
16.3	ILU	67
16.4	jdqr	68
16.5	testA	70
16.6	testB	70
17	master/eigenvalue/jacobi-davidson	71
17.1	afun_jdm	71
17.2	davidson	71
17.3	jacobi_davidson	71
17.4	jacobi_davidson_qr	71
17.5	jacobi_davidson_qz	71
17.6	jacobi_davidson_simple	71
17.7	jdqr	72
17.8	jdqr_sleijpen	75
17.9	jdqr_vorst	78
17.10	jdqz	81
17.11	mfunc_jdm	86
17.12	mgs	86
17.13	minres_	86
17.14	mv_jacobi_davidson	86
18	master/fdm	86
18.1	fdm_adaptive_grid	86
18.2	fdm_adaptive_refinement_old	86

18.3	fdm_assemble_d1_2d	86
18.4	fdm_assemble_d2_2d	87
18.5	fdm_confinement	87
18.6	fdm_d_vargrid	87
18.7	fdm_h_unstructured	87
18.8	fdm_hydrogen_vargrid	87
18.9	fdm_mark_unstructured_2d	87
18.10	fdm_plot	87
18.11	fdm_plot_series	87
18.12	fdm_refine_2d	87
18.13	fdm_refine_3d	87
18.14	fdm_refine_unstructured_2d	88
18.15	fdm_schroedinger_2d	88
18.16	fdm_schroedinger_3d	88
18.17	relocate	88
19	master/fem	88
19.1	Mesh_2d.java	88
19.2	Tree_2d.java	88
19.3	assemble_1d_dphi_dphi	88
19.4	assemble_1d_phi_phi	88
19.5	assemble_2d_dphi_dphi.java	88
19.6	assemble_2d_phi_phi.java	89
19.7	assemble_3d_dphi_dphi.java	89
19.8	assemble_3d_phi_phi.java	89
19.9	boundary_1d	89
19.10	boundary_2d	89
19.11	boundary_3d	89
19.12	check_area_2d	89
19.13	circmesh	89
19.14	cropradius	89
19.15	display_2d	89
19.16	display_3d	90
19.17	distort	90
19.18	err_2d	90
19.19	estimate_err_2d_3	90
19.20	example_1d	90
19.21	example_2d	90
19.22	explode	90
19.23	fem_2d	90
19.24	fem_2d_heuristic_mesh	90
19.25	fem_get_2d_radial	90
19.26	fem_interpolation	91
19.27	fem_plot_1d	91

19.28	fem_plot_1d_series	91
19.29	fem_plot_2d	91
19.30	fem_plot_2d_series	91
19.31	fem_plot_3d	91
19.32	fem_plot_3d_series	91
19.33	fem_plot_confine_series	91
19.34	fem_radial	91
19.35	flip_2d	92
19.36	get_mesh_arrays	92
19.37	hashkey	92
20	master/fem/int	92
20.1	int_1d_gauss	92
20.2	int_1d_gauss_1	92
20.3	int_1d_gauss_2	92
20.4	int_1d_gauss_3	92
20.5	int_1d_gauss_4	92
20.6	int_1d_gauss_5	92
20.7	int_1d_gauss_6	93
20.8	int_1d_gauss_lobatto	93
20.9	int_1d_nc_2	93
20.10	int_1d_nc_3	93
20.11	int_1d_nc_4	93
20.12	int_1d_nc_5	93
20.13	int_1d_nc_6	93
20.14	int_1d_nc_7	93
20.15	int_1d_nc_7_hardy	93
20.16	int_2d_gauss_1	93
20.17	int_2d_gauss_12	94
20.18	int_2d_gauss_13	94
20.19	int_2d_gauss_16	94
20.20	int_2d_gauss_25	94
20.21	int_2d_gauss_3	94
20.22	int_2d_gauss_33	94
20.23	int_2d_gauss_6	94
20.24	int_2d_gauss_7	94
20.25	int_2d_gauss_9	94
20.26	int_2d_nc_10	94
20.27	int_2d_nc_15	95
20.28	int_2d_nc_21	95
20.29	int_2d_nc_3	95
20.30	int_2d_nc_6	95
20.31	int_3d_gauss_1	95
20.32	int_3d_gauss_11	95

20.33	int_3d_gauss_14	95
20.34	int_3d_gauss_15	95
20.35	int_3d_gauss_24	95
20.36	int_3d_gauss_4	95
20.37	int_3d_gauss_45	96
20.38	int_3d_gauss_5	96
20.39	int_3d_nc_11	96
20.40	int_3d_nc_4	96
20.41	int_3d_nc_6	96
20.42	int_3d_nc_8	96
21	master/fem	96
21.1	interpolation_matrix	96
21.2	mark	96
21.3	mark_1d	96
21.4	mesh_1d_uniform	97
21.5	mesh_3d_uniform	97
21.6	mesh_interpolate	97
21.7	neighbour_1d	97
21.8	old	97
21.9	pdeeig_1d	97
21.10	pdeeig_2d	97
21.11	pdeeig_3d	97
21.12	polynomial_derivative_1d	97
21.13	potential_const	97
21.14	potential_coulomb	98
21.15	potential_harmonic_oscillator	98
21.16	project_circle	98
21.17	project_rectangle	98
21.18	promote_1d_2.3	98
21.19	promote_1d_2.4	98
21.20	promote_1d_2.5	98
21.21	promote_1d_2.6	98
21.22	quadrilaterate	98
21.23	recalculate_regularity_2d	98
21.24	refine_1d	99
21.25	refine_2d_21	99
21.26	refine_2d_structural	99
21.27	regularity_1d	99
21.28	regularity_2d	99
21.29	regularity_3d	99
21.30	relocate_2d	99
21.31	test_circmesh	99
21.32	test_hermite	99

21.33	tri_assign_points	100
21.34	triangulation_uniform	100
21.35	vander_1d	100
21.36	vanderd_1d	100
21.37	vanderi_1d	100
22	master/hydrogen-spectrum	100
22.1	hydrogen_spectrum_1d	100
22.2	hydrogen_spectrum_2012_12_02	100
22.3	hydrogen_spectrum_2d	100
22.4	hydrogen_spectrum_3d	100
23	master/lanczos	101
23.1	arnoldi	101
23.2	arnoldi_new	101
23.3	eigs_lanczos_man	101
23.4	lanczos	101
23.5	lanczos_	101
23.6	lanczos_biorthogonal	101
23.7	lanczos_biorthogonal_improved	101
23.8	lanczos_ghep	101
23.9	mv_lanczos	101
23.10	reorthogonalise	102
23.11	test_lanczos	102
24	master/linear-systems	102
24.1	gmres_man	102
24.2	minres_recycle	102
25	master/plot	102
25.1	attach_boundary_value	102
25.2	cartesian_polar	102
25.3	img_vargrid	102
25.4	plot_basis_functions	102
25.5	plot_convergence	102
25.6	plot_dof	103
25.7	plot_eigenbar	103
25.8	plot_error_estimation	103
25.9	plot_error_estimation_2	103
25.10	plot_error_fem	103
25.11	plot_fdm_kernel	103
25.12	plot_fdm_vs_fem	103
25.13	plot_fem_accuracy	103
25.14	plot_function_and_grid	103

25.15	plot_hat	103
25.16	plot_hydrogen_wf	104
25.17	plot_mesh	104
25.18	plot_mesh_2	104
25.19	plot_refine	104
25.20	plot_refine_3d	104
25.21	plot_runtime	104
25.22	plot_spectrum	104
25.23	plot_wavefunction	104
26	master/ported	104
26.1	assemble_2d_dphi_dphi	104
26.2	assemble_2d_phi_phi	105
26.3	assemble_3d_dphi_dphi	105
26.4	assemble_3d_phi_phi	105
26.5	dV_2d_	105
26.6	derivative_2d	105
26.7	derivative_3d	105
26.8	element_neighbour_2d	105
26.9	prefetch_2d_	105
26.10	promote_2d_3_10	105
26.11	promote_2d_3_15	105
26.12	promote_2d_3_21	106
26.13	promote_2d_3_6	106
26.14	promote_3d_4_10	106
26.15	promote_3d_4_20	106
26.16	promote_3d_4_35	106
26.17	vander_2d	106
26.18	vander_3d	106
27	master/sandbox	106
27.1	adapt	106
27.2	assoc_laguerre	106
27.3	assoc_legendre	107
27.4	c23	107
28	master/sandbox/cg	107
28.1	cg	107
28.2	cg_coef_to_poly	107
28.3	errmat	107
28.4	lanczos	107
28.5	laplacian_2d	107
28.6	test_cg_eigs	107
28.7	test_lanczos	107

29	master/sandbox	108
29.1	condition_number_higher_order	108
29.2	confinement_dat	108
29.3	convergence_2d_3d	108
29.4	convergence_matrix_powers	108
29.5	cut_out	108
29.6	derivative_2d	108
29.7	derivative_3d	108
29.8	dummy	108
29.9	eig_error	108
29.10	eigs_fix	109
29.11	energy_level	109
29.12	equalise	109
29.13	example_int64	109
30	master/sandbox/fem-matlab	109
30.1	boundary_circle	109
30.2	boundary_rectangle	109
30.3	geometry_circle_with_hole	109
30.4	geometry_rectangle	110
31	master/sandbox	110
31.1	fem_2d_estimate_error	110
31.2	fem_assemble_scratch	110
31.3	fem_s	110
31.4	fourier_h	110
31.5	grad_2d	110
31.6	grad_3d	110
31.7	gradient	110
31.8	harmonic_oscillator	110
31.9	hydrogen_2d_analytic	111
31.10	hydrogen_boxed	111
31.11	hydrogen_boxed_old	111
31.12	hydrogen_wave	111
31.13	hydrogen_wf	111
31.14	ichol_man	111
31.15	known_eigenvalue	111
31.16	kron_man	111
31.17	laguerre	111
31.18	laplacian_arbitrary_order_old	111
31.19	laplacian_convergence	112
31.20	laplacian_cut_out	112
31.21	laplacian_cylindrical	112
31.22	laplacian_non_uniform_old	112

31.23	laplacian_polar	112
31.24	laplacian_simple	112
31.25	lderivative_3d	112
31.26	list_dat	112
31.27	matlab-horner	112
31.28	mesh_to_grid_2d_3	112
31.29	mg_mat	113
31.30	mv	113
31.31	orth2	113
31.32	partial_derivative_2d	113
31.33	partition_function	113
31.34	partition_function_old	113
31.35	poisson	113
31.36	poisson_fem	113
31.37	potential	113
31.38	powerc	113
31.39	quick_newihbour	114
31.40	radial	114
31.41	radial_convergence	114
31.42	radial_wafefunction	114
31.43	refine_2d	114
31.44	refine_3d	114
31.45	relerr	114
31.46	restore_cw	114
31.47	runtime_bm	114
31.48	rydberg	114
31.49	s_old	115
31.50	snorm	115
31.51	spherical_harmonic	115
31.52	split_eig	115
31.53	sum1	115
31.54	sum3	115
32	master/sandbox/summation	115
32.1	acc	115
32.2	add	115
32.3	ape	115
32.4	mmul_accurately	116
32.5	sum_kahan	116
32.6	sum_pairwise	116
32.7	test_sum	116
33	master/sandbox	116
33.1	test_convergence_ill_conditioned	116

33.2	test_fem_1d	116
33.3	test_fem_2d	116
33.4	test_fem_3d	116
33.5	test_increase	116
33.6	test_lanczos_shift	117
33.7	test_ldl	117
33.8	test_power	117
33.9	trefethen_p8_fdm	117
33.10	wavefunc	117
33.11	xgrid	117
34	master/test	117
34.1	dat_test_lanczos_3d_k_20_n_40	117
34.2	poisson2d_blk	117
34.3	qr_implicit_givens_2	117
34.4	spectral_derivative_2d	118
34.5	test_2d_eigensolver_hydrogen	118
34.6	test_2d_refine	118
34.7	test_3d_eigensolver_hydrogen	118
34.8	test_FEM	118
34.9	test_Mesh_3d	118
34.10	test_arnoldi	118
34.11	test_arpackc	118
34.12	test_assemble	118
34.13	test_assembly_performance	118
34.14	test_bc_one_sided	119
34.15	test_compare_solvers	119
34.16	test_complete	119
34.17	test_convergence	119
34.18	test_convergence_b	119
34.19	test_df_2d	119
34.20	test_eig_algs	119
34.21	test_eig_inverse	119
34.22	test_eigs_lanczos	119
34.23	test_eigs_lanczos_1	119
34.24	test_eigs_lanczos_2	120
34.25	test_eigs_lanczos_performance	120
34.26	test_fdm	120
34.27	test_fdm_d_vargrid	120
34.28	test_fdm_spectral	120
34.29	test_fem	120
34.30	test_fem_1d	120
34.31	test_fem_1d_higher_order	120
34.32	test_fem_2d_adaptive	120

34.33	test_fem_2d_higher_order	120
34.34	test_fem_3d_higher_order	121
34.35	test_fem_3d_refine	121
34.36	test_fem_b	121
34.37	test_fem_derivative	121
34.38	test_fem_quadrature	121
34.39	test_final	121
34.40	test_fix_substitution	121
34.41	test_forward	121
34.42	test_get_sparse_arrays	121
34.43	test_harmonic_oscillator	121
34.44	test_high_order_fdm_periodic_bc	122
34.45	test_hydrogen_wf	122
34.46	test_ichol	122
34.47	test_interpolation	122
34.48	test_inverse_problem	122
34.49	test_it_vs_exact	122
34.50	test_jama	122
34.51	test_jd	122
34.52	test_jdqz	122
34.53	test_lanczos_2	122
34.54	test_lanczos_biorthogonal	123
34.55	test_laplacian	123
34.56	test_laplacian_non_uniform	123
34.57	test_laplacian_simple	123
34.58	test_mesh_2d_uniform	123
34.59	test_mesh_2d_uniform_2	123
34.60	test_mesh_circle	123
34.61	test_mesh_generation	123
34.62	test_mesh_interpolate	123
34.63	test_mg	123
34.64	test_minres_recycle	124
34.65	test_multigrid	124
34.66	test_nc	124
34.67	test_nonuniform_symmetric	124
34.68	test_pde	124
34.69	test_permutation	124
34.70	test_poison_fem	124
34.71	test_polar	124
34.72	test_potential	124
34.73	test_powers	124
34.74	test_precondition	125
34.75	test_project_rectangle	125
34.76	test_qr	125

34.77	test_quantum_well	125
34.78	test_radial_adaptive	125
34.79	test_radial_confinement	125
34.80	test_radial_fixes	125
34.81	test_refine_2d	125
34.82	test_refine_2d_b	125
34.83	test_refine_3d	125
34.84	test_refine_structural	126
34.85	test_regularisation	126
34.86	test_round_off	126
34.87	test_schrödinger_potentials	126
34.88	test_uniform_mesh	126
34.89	test_vargrid	126
35	number-theory	126
35.1	ceiln	126
35.2	digitsb	126
35.3	floorn	126
35.4	iseven	127
35.5	multichoosek	127
35.6	nchoosek_man	127
35.7	pythagorean_triple	127
35.8	roundn	127
36	numerical-methods/differentiation	127
36.1	derivative1	127
36.2	derivative2	128
37	numerical-methods/finite-difference	128
37.1	cdiff	128
37.2	cdiffb	128
37.3	cmean	128
37.4	derivative_matrix_1_1d	128
37.5	derivative_matrix_2_1d	128
37.6	derivative_matrix_2d	128
37.7	derivative_matrix_curvilinear	129
37.8	derivative_matrix_curvilinear_2	129
37.9	difference_kernel	129
37.10	distmat	129
37.11	gradpde2d	129
37.12	laplacian	129
37.13	laplacian_fdm	129
37.14	left	130
37.15	lrmean	130

37.16	mid	130
37.17	pwmid	130
37.18	ratio	130
37.19	steplength	130
37.20	swapoddeven	130
37.21	test_derivative_matrix_2d	130
37.22	test_derivative_matrix_curvilinear	131
37.23	test_difference_kernel	131
38	numerical-methods/finite-volume/@Advection	131
38.1	Advection	131
38.2	dot_advection	131
39	numerical-methods/finite-volume/@Burgers	131
39.1	burgers_split	131
39.2	dot_burgers_fdm	131
39.3	dot_burgers_fft	131
40	numerical-methods/finite-volume/@Finite_Volume	132
40.1	Finite_Volume	132
40.2	apply_bc	132
40.3	solve	132
40.4	step_split_strang	132
40.5	step_unsplit	132
41	numerical-methods/finite-volume/@Flux_Limiter	132
41.1	Flux_Limiter	132
41.2	beam_warming	133
41.3	fromm	133
41.4	lax_wendroff	133
41.5	minmod	133
41.6	monotized_central	133
41.7	muscl	133
41.8	superbee	133
41.9	upwind	134
41.10	vanLeer	134
42	numerical-methods/finite-volume/@KDV	134
42.1	dot_kdv_fdm	134
42.2	dot_kdv_fft	134
42.3	kdv_split	134
43	numerical-methods/finite-volume/@Reconstruct_Average_Evolve	134
43.1	Reconstruct_Average_Evolve	134
43.2	advect_highres	135

43.3	advect_lowress	135
44	numerical-methods/finite-volume	135
44.1	Godunov	135
44.2	Lax_Friedrich	135
44.3	Measure	135
44.4	Roe	135
44.5	fv_swe	136
44.6	staggered_euler	136
44.7	staggered_grid	136
45	numerical-methods	136
45.1	grid2quad	136
46	numerical-methods/integration	136
46.1	cumintL	136
46.2	cumintR	136
46.3	int_trapezoidal	136
47	numerical-methods/interpolation/@Kriging	137
47.1	Kriging	137
47.2	estimate_semivariance	137
47.3	interpolate_	137
48	numerical-methods/interpolation/@RegularizedInterpolator1	137
48.1	RegularizedInterpolator1	137
48.2	init	137
49	numerical-methods/interpolation/@RegularizedInterpolator2	138
49.1	RegularizedInterpolator2	138
49.2	init	138
50	numerical-methods/interpolation/@RegularizedInterpolator3	138
50.1	RegularizedInterpolator3	138
50.2	init	138
51	numerical-methods/interpolation	138
51.1	IDW	138
51.2	IPoly	138
51.3	IRBM	139
51.4	ISparse	139
51.5	Inn	139
51.6	Interpolator	139
51.7	fixnan	139
51.8	idw1	139

51.9	idw2	139
51.10	inner2outer	140
51.11	inner2outer2	140
51.12	interp1_limited	140
51.13	interp1_man	140
51.14	interp1_save	140
51.15	interp1_slope	140
51.16	interp1_smooth	141
51.17	interp1_unique	141
51.18	interp2_man	141
51.19	interp_angle	141
51.20	interp_fourier	141
51.21	interp_fourier_batch	141
51.22	interp_sn	141
51.23	interp_sn2	142
51.24	interp_sn3	142
51.25	interp_sn_	142
51.26	limit_by_distance_1d	142
51.27	resample1	142
51.28	resample_d_min	142
51.29	resample_vector	142
51.30	test_interp1_limited	142
52	numerical-methods	143
52.1	inverse_complex	143
53	numerical-methods/ode	143
53.1	bvp2_check_arguments	143
53.2	bvp2c	143
53.3	bvp2c2	143
53.4	bvp2fdm	144
53.5	bvp2wavetrain	144
53.6	bvp2wavetwopass	144
53.7	ivp_euler_forward	144
53.8	ivprk2	145
53.9	ode2_matrix	145
53.10	ode2characteristic	145
53.11	step_trapezoidal	145
53.12	test_bvp2	145
54	numerical-methods/optimisation	145
54.1	armijo_stopping_criterion	145
54.2	astar	145
54.3	binsearch	146

54.4	bisection	146
54.5	box1	146
54.6	box2	146
54.7	cauchy	146
54.8	cauchy2	146
54.9	directional_derivative	146
54.10	dud	147
54.11	extreme3	147
54.12	extreme_quadratic	147
54.13	ftest	147
54.14	grad	147
54.15	hessian	147
54.16	hessian_from_gradient	148
54.17	hessian_projected	148
54.18	line_search	148
54.19	line_search2	148
54.20	line_search_polynomial	148
54.21	line_search_polynomial2	149
54.22	line_search_quadratic	149
54.23	line_search_quadratic2	149
54.24	line_search_wolfe	149
54.25	ls_bgfs	149
54.26	ls_broyden	150
54.27	ls_generalized_secant	150
54.28	nlcg	150
54.29	nlls	150
54.30	picard	150
54.31	poly_extrema	151
54.32	quadratic_function	151
54.33	quadratic_programming	151
54.34	quadratic_step	151
54.35	rosenbrock	151
54.36	sqrt_heron	151
54.37	test_directional_derivative	151
54.38	test_dud	151
54.39	test_line_search_quadratic2	151
54.40	test_ls_generalized_secant	152
54.41	test_nlcg_6_order	152
54.42	test_nlls	152
55	numerical-methods/piecewise-polynomials	152
55.1	Hermite1	152
55.2	hp2_fit	152
55.3	hp2_predict	152

55.4	hp_predict	152
55.5	hp_regress	153
55.6	lp_count	153
55.7	lp_predict	153
55.8	lp_regress	153
55.9	lp_regress_	153
56	regression/@PolyOLS	153
56.1	PolyOLS	153
56.2	coefftest	153
56.3	detrend	153
56.4	fit	154
56.5	fit_	154
56.6	predict	154
56.7	predict_	154
56.8	slope	154
57	regression/@PowerLS	154
57.1	PowerLS	154
57.2	fit	154
57.3	predict	155
57.4	predict_	155
58	regression/@Theil	155
58.1	Theil	155
58.2	detrend	155
58.3	fit	155
58.4	predict	155
58.5	slope	156
59	regression	156
59.1	Theil_Multivariate	156
59.2	areg	156
59.3	ginireg	156
59.4	hessimplereg	156
59.5	l1lin	156
59.6	lsq_sparam	157
59.7	polyfitd	157
59.8	regression_method_of_moments	157
59.9	robustlinreg	157
59.10	theil2	157
59.11	theil_generalised	157
59.12	total_least_squares	158
59.13	weighted_median_regression	158

60 set-theory	158
60.1 issubset	158
61 signal-processing	158
61.1 acf_effective_sample_size	158
61.2 acf_genton	158
61.3 acfar1	159
61.4 acfar1_2	159
61.5 acfar2	159
61.6 acfar2_2	159
61.7 ar1_cutoff_frequency	159
61.8 ar1_effective_sample_size	159
61.9 ar1_mse_mu_single_sample	159
61.10 ar1_mse_pop	160
61.11 ar1_mse_range	160
61.12 ar1_spectrum	160
61.13 ar1_to_tikhonov	160
61.14 ar1_var_factor	160
61.15 ar1_var_factor_	160
61.16 ar1_var_range2	160
61.17 ar1delay	161
61.18 ar1delay_old	161
61.19 ar2conv	161
61.20 ar2dof	161
61.21 ar2param	161
61.22 asymwin	161
61.23 autocorr_fft	161
61.24 bandpass	162
61.25 bandpass2	162
61.26 bartlett	162
61.27 bartlett_spectrogram	162
61.28 bin1d	162
61.29 bin2d	162
61.30 binormrnd	163
61.31 conv1_man	163
61.32 conv2_man	163
61.33 conv2z	163
61.34 conv30	163
61.35 conv_	163
61.36 conv_centered	163
61.37 convz	163
61.38 cosexpdelay	164
61.39 csmooth	164
61.40 daniell_window	164

61.41	danielle_window	164
61.42	db2neper	164
61.43	db2power	164
61.44	derive_danielle_weight	164
61.45	derive_limit_0_acfar	164
61.46	detect_peak	165
61.47	digital_low_pass_filter	165
61.48	doublesum_ij	165
61.49	effective_sample_size_to_ar1	165
61.50	filt_hodges_lehman	165
61.51	filter1	165
61.52	filter2	165
61.53	filter_	166
61.54	filteriir	166
61.55	filterp	166
61.56	filterp1	167
61.57	filterstd	167
61.58	firls_man	167
61.59	flattopwin	167
61.60	frequency_response_boxcar	167
61.61	freqz_boxcar	167
61.62	gaussfilt1	167
61.63	hanchangewin	167
61.64	hanchangewin2	168
61.65	hanwin	168
61.66	hanwin_	168
61.67	highpass	168
61.68	kaiserwin	168
61.69	kalman	168
61.70	lanczoswin	168
61.71	last	168
61.72	lowpass	169
61.73	lowpass2	169
61.74	lowpass_iir	169
61.75	lowpass_iir_symmetric	169
61.76	lowpassfilter2	169
61.77	maxfilt1	169
61.78	meanfilt1	169
61.79	medfilt1_man	169
61.80	medfilt1_man2	170
61.81	medfilt1_padded	170
61.82	medfilt1_reduced	170
61.83	mid_term_single_sample	170
61.84	minfilt1	170

61.85	mu2ar1	170
61.86	nanautocorr	170
61.87	nanmedfilt1	170
61.88	neper2db	171
61.89	peaks_man	171
61.90	polyfilt1	171
61.91	qmedfilt1	171
61.92	randar1	171
61.93	randar1_dual	171
61.94	randar2	171
61.95	randarp	171
61.96	range_window	172
61.97	rectwin	172
61.98	recursive_sum	172
61.99	select_range	172
61.100	smooth1d_parametric	172
61.101	smooth2	172
61.102	smooth_man	172
61.103	smooth_parametric	172
61.104	smooth_parametric2	173
61.105	smoothfft	173
61.106	spectrogram	173
61.107	std_window	173
61.108	sum_i_lag	173
61.109	sum_ii	173
61.110	sum_ii_	173
61.111	sum_ij	173
61.112	sum_ij_	174
61.113	sum_ij_partial_	174
61.114	sum_multivar	174
61.115	test_acfar1	174
61.116	test_acfar1_2	174
61.117	test_acfar1_3	174
61.118	test_acfar1_4	174
61.119	test_acfar2	174
61.120	test_ar1_var_factor	174
61.121	test_ar1_var_factor_2	174
61.122	test_ar1_var_mu_single_sample	175
61.123	test_ar1_var_pop	175
61.124	test_ar1_var_pop_1	175
61.125	test_ar1delay	175
61.126	test_bivariate_covariance_term	175
61.127	test_convexity	175
61.128	test_lanczoswin	175

61.129	test_madcorr	175
61.130	test_randar1	175
61.131	test_randar1_multivariate	175
61.132	test_randar2	176
61.133	test_sum_ij	176
61.134	test_sum_multivar	176
61.135	test_trifilt1	176
61.136	test_wautocorr	176
61.137	test_wavelet_transform	176
61.138	test_wordfilt	176
61.139	test_xar1_mid_term	176
61.140	tikhonov_to_ar1	176
61.141	trapwin	177
61.142	trifilt1	177
61.143	triwin	177
61.144	triwin2	177
61.145	varar1	177
61.146	welch_spectrogram	177
61.147	wfilt	177
61.148	winbandpass	177
61.149	window_make_odd	178
61.150	winfilt0	178
61.151	winlength	178
61.152	wmeanfilt	178
61.153	wmedfilt	178
61.154	wordfilt	178
61.155	wordfilt_edgeworth	178
61.156	xar1	178
61.157	xcorr_man	179
62 sorting		179
62.1	sort2	179
62.2	sort2d	179
63 special-functions		179
63.1	bessel_sphere	179
63.2	hankel_sphere	179
63.3	hermite	179
63.4	legendre_man	180
63.5	neumann_sphere	180
64 statistics		180
64.1	atan_s2	180
64.2	beta_mode_to_parameter	180

64.3	correlation_confidence_pearson	180
65	statistics/distributions	180
65.1	PDF	180
65.2	binorm_separation_coefficient	180
65.3	binormcdf	181
65.4	binormfit	181
65.5	binormpdf	181
65.6	edgeworth_cdf	181
65.7	edgeworth_pdf	181
65.8	logn_mode2param	181
65.9	logn_param2mode	181
65.10	lognpdf_	182
65.11	pdfsample	182
65.12	t2cdf	182
65.13	t2inv	182
66	statistics	182
66.1	example_standard_error_of_sample_quantiles	182
66.2	f_var_finite	182
66.3	gamma_mode_to_parameter	182
66.4	hodges_lehmann_correlation	183
66.5	hodges_lehmann_dispersion	183
67	statistics/information-theory	183
67.1	akaike_information_criterion	183
67.2	bayesian_information_criterion	183
68	statistics	184
68.1	kurtncdf	184
68.2	kurtnpdf	184
68.3	kurtosis_bias_corrected	184
68.4	limit	184
68.5	logfactorial	184
68.6	loglogpdf	184
68.7	logskewcdf	184
68.8	logskewpdf	184
69	statistics/logu	185
69.1	lambertw_numeric	185
69.2	logtrialtcd	185
69.3	logtrialtinv	185
69.4	logtrialtmean	185
69.5	logtrialtpdf	185

69.6	logtrialtrnd	185
69.7	logtricdf	186
69.8	logtriinv	186
69.9	logtrimean	186
69.10	logtripdf	186
69.11	logtirnd	186
69.12	logucdf	186
69.13	logucm	186
69.14	loguinv	186
69.15	logumean	187
69.16	logupdf	187
69.17	logurnd	187
69.18	loguvar	187
69.19	medlogu	187
69.20	test_logurnd	187
69.21	tricdf	187
69.22	triinv	187
69.23	trimean	188
69.24	tripdf	188
69.25	trirnd	188
70 statistics		188
70.1	maxnnormals	188
70.2	midrange	188
70.3	minavg	188
70.4	mode_man	188
71 statistics/moment-statistics		189
71.1	autocorr_man3	189
71.2	autocorr_man4	189
71.3	autocorr_man5	189
71.4	blockserr	189
71.5	comoment	189
71.6	corr_man	190
71.7	cov_man	190
71.8	dof	190
71.9	edgeworth_quantile	190
71.10	effective_sample_size	190
71.11	f_correlation	190
71.12	f_finite	191
71.13	lmean	191
71.14	lmoment	191
71.15	maskmean	191
71.16	masknanmean	191

71.17	mean1	191
71.18	mean_man	191
71.19	mse	191
71.20	nanautocorr_man1	192
71.21	nanautocorr_man2	192
71.22	nanautocorr_man4	192
71.23	nancorr	192
71.24	nancumsum	192
71.25	nanlmean	192
71.26	nanr2	192
71.27	nanrms	193
71.28	nanrmse	193
71.29	nanserr	193
71.30	nanwmean	193
71.31	nanwstd	193
71.32	nanwvar	193
71.33	nanxcorr	193
71.34	pearson	194
71.35	pearson_to_kendall	194
71.36	pool_samples	194
71.37	qmean	194
71.38	range_mean	194
71.39	rmse	194
71.40	serr	194
71.41	serr1	194
71.42	test_qskew	195
71.43	test_qstd_qskew_optimal_p	195
71.44	wautocorr	195
71.45	wcorr	195
71.46	wcov	195
71.47	wdof	195
71.48	wkurt	195
71.49	wmean	196
71.50	wrms	196
71.51	wserr	196
71.52	wskew	196
71.53	wstd	196
71.54	wvar	196
72	statistics	197
72.1	nangeomean	197
72.2	nangeostd	197
73	statistics/nonparametric-statistics	197

73.1	kernel1d	197
73.2	kernel2d	197
74	statistics	197
74.1	normmoment	197
74.2	normpdf2	197
75	statistics/order-statistics	198
75.1	hodges_lehmann_location	198
75.2	kendall	198
75.3	kendall_to_pearson	198
75.4	mad2sd	198
75.5	madcorr	198
75.6	median2_holder	198
75.7	median_ci	199
75.8	median_man	199
75.9	mediani	199
75.10	nanmadcorr	199
75.11	nanwmedian	199
75.12	nanwquantile	199
75.13	oja_median	199
75.14	qkurtosis	200
75.15	qmoments	200
75.16	qskew	200
75.17	qskewq	200
75.18	qstdq	200
75.19	quantile1_optimisation	200
75.20	quantile2_breckling	201
75.21	quantile2_chaudhuri	201
75.22	quantile2_projected	201
75.23	quantile2_projected2	201
75.24	quantile_envelope	201
75.25	quantile_regression_simple	201
75.26	ranking	201
75.27	spatial_median	201
75.28	spatial_quantile	202
75.29	spatial_quantile2	202
75.30	spatial_quantile3	202
75.31	spatial_rank	202
75.32	spatial_sign	202
75.33	spatial_signed_rank	202
75.34	spearman	202
75.35	spearman_rank	202
75.36	spearman_to_pearson	203

75.37	wmedian	203
75.38	wquantile	203
76	statistics	203
76.1	qstd	203
76.2	quantile_extrap	203
77	statistics/random-number-generation	203
77.1	laplacernd	203
77.2	randc	203
77.3	skewrnd	204
77.4	skewrnd2	204
78	statistics	204
78.1	range	204
78.2	resample_with_replacement	204
79	statistics/resampling-statistics/@Jackknife	204
79.1	Jackknife	204
79.2	estimated_STATIC	205
79.3	matrix1_STATIC	205
79.4	matrix2	205
80	statistics/resampling-statistics	205
80.1	block_jackknife	205
80.2	jackknife_moments	205
80.3	moving_block_jackknife	206
80.4	randblockserr	206
80.5	resample	206
81	statistics	206
81.1	scale_quantile_sd	206
81.2	skewpdf	207
81.3	trimmed_mean	207
81.4	ttest2_man	207
81.5	ttest_man	207
81.6	ttest_paired	207
81.7	wharmean	207
82	wavelet	208
82.1	contiuous_wavelet_transform	208
82.2	cwt_man	208
82.3	example_wavelets	208
82.4	phasewrap	208
82.5	test_cwt_man	208

82.6	test_phasewrap	208
82.7	test_wavelet	208
82.8	test_wavelet2	208
82.9	test_wavelet_analysis	209
82.10	test_wavelet_reconstruct	209
82.11	test_wtc	209
82.12	wavelet	209
82.13	wavelet_reconstruct	209
82.14	wavelet_transform	209
83	mathematics	209
83.1	wrapphase	209

1 mathematics

mathematical functions of various kind

1.1 cast_byte_to_integer

cast byte to integer

2 complex-analysis

operations on complex numbers

2.1 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

2.2 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

2.3 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

2.4 complex_exp_product_re_re

product of the real part of two complex exponentials

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

the product has two frequency components

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

2.5 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

2.6 root_complex

root of a complex number

2.7 test_imroots

3 derivation

derivation of several functions by means of symbolic computation

3.1 derive_acfar1

3.2 derive_ar2param

3.3 derive_arc_length

3.4 derive_fourier_power

3.5 `derive_fourier_power_exp`

3.6 `derive_laplacian_curvilinear`

3.7 `derive_laplacian_fourier_piecewise_linear`

3.8 `derive_logtripdf`

3.9 `derive_smooth1d_parametric`

3.10 `simplify_atan`

symbolic simplification of the arcus tangent

4 `fourier/@STFT`

4.1 `STFT`

class for short time fourier transform

Note: the interval `Ti` 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

4.2 itransform

inverse of the short time fourier transform

4.3 stft_

static wrapper for STFT

4.4 stftmat

transformation matrix for the short time fourier transform

4.5 transform

short time fourier transform

5 fourier

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

5.1 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 (!)

5.2 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

5.3 example_fourier_window

5.4 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

5.5 fft_man

fast fourier transform for complex input data

input:
F : data in real space

output :

F : fourier transformation of F

5.6 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

5.7 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

5.8 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

5.9 `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

5.10 `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

5.11 `fourier_coefficient_ramp3`

fourier series coefficient of a ramp

5.12 `fourier_coefficient_ramp_pulse`

fourier series coefficient of a ramp pulses

5.13 `fourier_coefficient_ramp_step`

fourier coefficient of a ramp-step

5.14 `fourier_coefficient_square_pulse`

fourier series coefficients of a square pulse

5.15 `fourier_derivative`

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

5.16 `fourier_expand`

expand values of fourier series

5.17 `fourier_fit`

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

5.18 `fourier_interpolate`

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

5.19 `fourier_matrix`

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

5.20 fourier_matrix2

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

5.21 fourier_matrix3

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

5.22 fourier_power

powers of a continuous fourier series in sin/cos form

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

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

5.23 fourier_power_exp

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

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

$y = a_0 + \sum (a_j \sin(j\omega t) + b_j \cos(j\omega t))$
 $= \text{Real}(\sum_{i=0}^{\infty} c_i \exp(1i\omega t)), c_i = a_i + b_i$

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

5.24 fourier_predict

expand a continuous fourier series at times t

5.25 `fourier_range`

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

5.26 `fourier_regress`

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

5.27 `fourier_resampled_fit`

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

5.28 `fourier_resampled_predict`

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

5.29 `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

5.30 `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

```

5.31 hyperbolic_fourier_box

5.32 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)

5.33 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

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

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

5.34 nanfft

discrete fourier transform of a data series with gaps

5.35 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 $p \cdot y_{\text{peak}}$, $p = 0.5$

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)

5.36 roots_fourier

zeros of continuous fourier series series

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

5.37 spectral_density

spectral density

5.38 test_complex_exp_product

5.39 test_idftmtx

6 geometry/@Geometry

6.1 Geometry

6.2 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

6.3 arclength_old

arc length of a two dimensional function

6.4 arclength_old2

arc length of a two dimensional function

6.5 base_point

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

6.6 base_point_limited

base point (Fusspunkt) of a point on a line

6.7 centroid

centroid pf a polygone

6.8 cosa_min_max

6.9 cross2

cross product in two dimensions

6.10 curvature

curvature of a function in two dimensions

6.11 ddot

sum of squares of cos of inner angles of triangle

6.12 distance

equclidan distance between two points

6.13 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

6.14 dot

dot product

6.15 `edge_length`

edge length

6.16 `enclosed_angle`

angle enclosed between two lines

6.17 `enclosing_triangle`

smallest enclosing equilateral triangle with bottom side parallel to
X axis

6.18 `hexagon`

coordinates of a hexagon, scaled and rotated

6.19 `inPolygon`

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

6.20 `inTetra`

flag points contained in tetrahedron

6.21 `inTetra2`

flag points contained in tetrahedron

6.22 `inTriangle`

flag points contained in triangle

6.23 intersect

intersect between two lines

6.24 lineintersect

intersect of two lines

6.25 lineintersect1

intersect of two lines

6.26 minimum_distance_lines

minimum distance of two lines in three dimensions

6.27 mittenpunkt

mittenpunkt of a triangle

6.28 nagelpoint

nagelpoint of a triangle

6.29 onLine

6.30 orthocentre

orthocentre of triangle

6.31 `plumb_line`

6.32 `poly_area`

area of a polygon

6.33 `poly_edges`

edges of a polygon

6.34 `poly_set`

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

6.35 `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

6.36 `polyxpoly`

intersections of two polygons

6.37 `project_to_curve`

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

6.38 random_disk

draw random points on the unit disk

6.39 random_simplex

random point inside of a triangle

6.40 sphere_volume

volume of a sphere

6.41 tetra_volume

volume of a tetrahedron

6.42 tobarycentric

cartesian to barycentric coordinates

6.43 tobarycentric1

cartesian to barycentric coordinates

6.44 tobarycentric2

cartesian to barycentric coordinates

6.45 tobarycentric3

cartesian to barycentric coordinates

6.46 tri_angle

cos of angles of a triangle

6.47 tri_area

angle of a triangle

6.48 tri_centroid

centroid of a triangle

6.49 tri_distance_opposit_midpoint

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

6.50 tri_edge_length

edge length of a triangle

6.51 tri_edge_midpoint

mid point of a triangle

6.52 tri_excircle

excircle of a triangle

6.53 tri_height

height of a triangle

6.54 tri_incircle

incircle of a triangle

6.55 tri_isacute

flag acute triangles

6.56 tri_isobtuse

flag obtuse triangles

6.57 tri_semiperimeter

semiperimeter of a triangle

6.58 tri_side_length

edge length of triangle

7 geometry

7.1 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

7.2 bounding_box

bounding box of X

7.3 curvature_1d

curvature of a sampled parametric curve in two dimensions

7.4 cvt

centroidal voronoi tessellation

7.5 deg_to_frac

degree, minutes and seconds to fractions

7.6 ellipse

n-points on an ellipse

7.7 ellipseX

x-coordinates of y-coordinates of an ellipse

7.8 ellipseY

7.9 first_intersect

get first intersection between lines in A and B

7.10 golden_ratio

golden ratio

7.11 hypot3

hypotenuse in 3D

7.12 meanangle

weighted mean of angles

7.13 meanangle2

mean angle

7.14 meanangle3

mean angle

7.15 meanangle4

mean angle

7.16 medianangle

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

7.17 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

7.18 pilim

limit to +- pi

7.19 streamline_radius_of_curvature

streamline radius of curvature

simplifies when rotatate to streamwise coordinates to $R = 1/dv/ds * u$

8 linear-algebra

8.1 averaging_matrix_2

8.2 colnorm

norms of columns

8.3 condest_

estimation of the condition number

9 linear-algebra/coordinate-transformation

9.1 barycentric2cartesian

barycentric to cartesian coordinates

9.2 barycentric2cartesian3

convert barycentric to cartesian coordinates

9.3 cartesian2barycentric

cartesian to barycentric coordinates

9.4 cartesian_to_unit_triangle_basis

transform coodinates into unit triangle

9.5 example_approximate_utm_conversion

9.6 latlon2utm

transform latitude and longitude to WGS84 UTM

9.7 latlon2utm_simple

9.8 lowrance_mercator_to_wgs84

convert lowrance coordinates to wgs84

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

9.9 nmea2utm

convert nmea messages to utm coordinates

9.10 sn2xy

convert sn to xy coordinates

9.11 unit_triangle_to_cartesian

transform coordinates in unit triangle to cartesian coordinates

9.12 utm2latlon

convert wgs84 utm to latitude and longitude

9.13 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

9.14 xy2sn

convert cartesian to streamwise coordiantes

9.15 xy2sn.java

use java port for speed up

9.16 xy2sn_old

transform points from cartesian into streamwise coordinates

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

10 linear-algebra

10.1 det2x2

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

10.2 det3x3

determinant of stacked 3x3 matrices

10.3 det4x4

determinant of stacked 4x4 matrices

10.4 diag2x2

diagonal of stacked 2x2 matrices

10.5 eig2x2

eigenvalues of stacked 2x2 matrices

10.6 first

10.7 gershgorin_circle

range of eigenvalues determined by the gershgorin circle theorem

10.8 haussdorff

hausdorff 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

10.9 ieig2x2

reconstruct matrix from eigenvalue decomposition

10.10 inv2x2

2x2 inverse of stacked matrices

10.11 inv3x3

10.12 inv4x4

inverse of stacked 4x4 matrices

10.13 lpmean

mean of pth-power of a

10.14 `lpnorm`

norm of lth-power of a

10.15 `matvec3`

matrix-vector product of stacked matrices and vectors

10.16 `max2d`

maximum value and i-j index for matrix

10.17 `mpoweri`

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

10.18 `mtimes2x2`

10.19 `mtimes3x3`

product of stacked 3x3 matrices

10.20 `nannorm`

norm of a vector, skips nan-values

10.21 `nanshift`

shift vector, but set out of range values to NaN

10.22 nl

number rows (lines) of a matrix

analogue to unix nl command

10.23 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)

10.24 normalize1

normalize columns in x to [-1,1]

10.25 normrows

10.26 orth2

make matrix A orthogonal to B

10.27 orth_man

orthogonalize the columns of A

10.28 orthogonalise

make x orthogonal to Y

10.29 paddext

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

10.30 paddval1

padd values at end of x

10.31 paddval2

padd values to x

11 linear-algebra/polynomial

11.1 chebychev

chebycheff polynomials

11.2 piecewise_polynomial

evaluate piecewise polynomial

11.3 roots1

roots of linear functions

11.4 roots2

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

11.5 vanderi_1d

vandermonde matrix of an integral

12 linear-algebra

12.1 randrot

random rotation matrix

12.2 right

get right column by shifting columns to left
extrapolate rightmost column

12.3 rot2

rotation matrix from angle

12.4 rot2dir

rotation matrix from direction vector

12.5 rot3

12.6 rownorm

12.7 simmilarity_matrix

12.8 spnorm

frobenius norm

12.9 spzeros

allocate a sparze matrix of zeros

12.10 transpose3

transpose stacked 3x3 matrices

12.11 transposeall

13 logic

bitwise operations on integers

13.1 bitor_man

bitwise OR of the numbers of the columns of A

input:
A (positive integer)

14 master/derive

14.1 derive_bc_one_sided

14.2 derive_convergence

14.3 `derive_error_fdm`

14.4 `derive_fdm_poly`

14.5 `derive_fdm_power`

14.6 `derive_fdm_taylor`

14.7 `derive_fdm_vargrid`

14.8 `derive_fem_2d_mass`

14.9 `derive_fem_error_2d`

14.10 `derive_fem_error_3d`

14.11 `derive_fem_sym_2d`

14.12 `derive_grid_constants`

14.13 `derive_interpolation`

14.14 `derive_laplacian`

14.15 `derive_limit`

14.16 `derive_nc_1d`

14.17 `derive_nc_1d_`

14.18 `derive_nc_2d`

14.19 `derive_nonuniform_symmetric`

%

14.20 `derive_richardson`

14.21 `derive_sum`

14.22 `nn`

14.23 test_derive

14.24 test_derive_fdm_poly

14.25 test_filter

14.26 test_vargrid

15 master/eigenvalue

15.1 eig_bisection

15.2 eig_inverse

15.3 eig_inverse_iteration

15.4 eig_power_iteration

16 master/eigenvalue/jacobi-davidson/JDQR

16.1 Example1

16.2 Example2

% dimension of the matrix operation

16.3 ILU

16.4 jdqr

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

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

```

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

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

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

```

```

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

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

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

% 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

```

16.5 testA

16.6 testB

1
2
3
4
5
6
7
8
9
10

17 master/eigenvalue/jacobi-davidson

17.1 afun_jdm

17.2 davidson

17.3 jacobi_davidson

17.4 jacobi_davidson_qr

17.5 jacobi_davidson_qz

17.6 jacobi_davidson_simple

17.7 jdqr

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

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



```

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

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

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

```

```

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

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

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

% compute vectors and matrices for skew projection
% solve preconditioned system
% 0 step of bicgstab eq. 1 step of bicgstab
% Then x is a multiple of b
% HIST=[0,1];
% explicit preconditioning
% compute norm in l-space
% HIST=[HIST;[nmv,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'

```

17.8 jdqr sleipen

```

% 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'

```

17.9 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
%    $V*V=eye(j)$ ,  $Qschur'*V=0$ ,
%    $W=A*V$ ,  $M=W'*W$ 
%
% Compute approximate eigenpair and residual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check for convergence
% Expand the partial Schur form
  Rschur=[Rschur;zeros(1,k)],Qschur'*MV(u)]; k=k+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

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

```

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

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

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

```



```

% Expand preconditioned Schur matrix PinvQ
% Check for shrinking the search subspace
% Solve correction equation
% Expand the subspaces of the interaction matrix
W=V*Q; V=V(:,1:j)/R; E=E/R; R=eye(j); M=Q(1:j,:)' /R;
W=V*H; V(:,j+1)=[];R=R'*R; M=H(1:j,:)' ;
%===== ARNOLDI (for initializing spaces)
=====
%===== END ARNOLDI
=====
% not accurate enough M=Rw'\(M/Rv);
%===== COMPUTE SORTED JORDAN FORM
=====
% 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

```

17.10 jdqz

```

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

```

```

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

```

```

%=====

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

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

%=====

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

% solve preconditioned system
%=====

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

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

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

```

```

% plot(HIST(:,1),log10(HIST(:,2)+eps),'*'), drawnow
% 0 step of gmres eq. 1 step of gmres
% Then x is a multiple of b
%=====

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

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

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

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

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

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

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

```

```

%=====

%=====

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

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

%=====

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

```

```

%=====

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

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

%=====

%=====

%=====

```

17.11 mfunc_jdm

17.12 mgs

17.13 minres_

17.14 mv_jacobi_davidson

18 master/fdm

18.1 fdm_adaptive_grid

18.2 fdm_adaptive_refinement_old

18.3 `fdm_assemble_d1_2d`

18.4 `fdm_assemble_d2_2d`

18.5 `fdm_confinement`

18.6 `fdm_d_vargrid`

18.7 `fdm_h_unstructured`

18.8 `fdm_hydrogen_vargrid`

18.9 `fdm_mark_unstructured_2d`

18.10 `fdm_plot`

18.11 `fdm_plot_series`

18.12 `fdm_refine_2d`

18.13 `fdm_refine_3d`

18.14 `fdm_refine_unstructured_2d`

18.15 `fdm_schroedinger_2d`

18.16 `fdm_schroedinger_3d`

18.17 `relocate`

19 `master/fem`

19.1 `Mesh_2d.java`

19.2 `Tree_2d.java`

19.3 `assemble_1d_dphi_dphi`

19.4 `assemble_1d_phi_phi`

19.5 assemble_2d_dphi_dphi_java

19.6 assemble_2d_phi_phi_java

19.7 assemble_3d_dphi_dphi_java

19.8 assemble_3d_phi_phi_java

19.9 boundary_1d

19.10 boundary_2d

19.11 boundary_3d

19.12 check_area_2d

19.13 circmesh

19.14 cropradius

19.15 `display_2d`

19.16 `display_3d`

19.17 `distort`

19.18 `err_2d`

19.19 `estimate_err_2d_3`

19.20 `example_1d`

19.21 `example_2d`

19.22 `explode`

19.23 `fem_2d`

19.24 `fem_2d_heuristic_mesh`

19.25 fem_get_2d_radial

19.26 fem_interpolation

19.27 fem_plot_1d

19.28 fem_plot_1d_series

19.29 fem_plot_2d

19.30 fem_plot_2d_series

19.31 fem_plot_3d

19.32 fem_plot_3d_series

19.33 fem_plot_confine_series

19.34 fem_radial

adaptive grid
constant grid

19.35 flip_2d

19.36 get_mesh_arrays

19.37 hashkey

20 master/fem/int

20.1 int_1d_gauss

20.2 int_1d_gauss_1

20.3 int_1d_gauss_2

20.4 int_1d_gauss_3

20.5 int_1d_gauss_4

20.6 int_1d_gauss_5

20.7 int_1d_gauss_6

20.8 int_1d_gauss_lobatto

20.9 int_1d_nc_2

20.10 int_1d_nc_3

20.11 int_1d_nc_4

20.12 int_1d_nc_5

20.13 int_1d_nc_6

20.14 int_1d_nc_7

20.15 int_1d_nc_7_hardy

20.16 int_2d_gauss_1

20.17 int_2d_gauss_12

20.18 int_2d_gauss_13

20.19 int_2d_gauss_16

20.20 int_2d_gauss_25

20.21 int_2d_gauss_3

20.22 int_2d_gauss_33

20.23 int_2d_gauss_6

20.24 int_2d_gauss_7

20.25 int_2d_gauss_9

20.26 int_2d_nc_10

20.27 int_2d_nc_15

20.28 int_2d_nc_21

20.29 int_2d_nc_3

20.30 int_2d_nc_6

20.31 int_3d_gauss_1

20.32 int_3d_gauss_11

20.33 int_3d_gauss_14

20.34 int_3d_gauss_15

20.35 int_3d_gauss_24

20.36 int_3d_gauss_4

20.37 int_3d_gauss_45

20.38 int_3d_gauss_5

20.39 int_3d_nc_11

20.40 int_3d_nc_4

20.41 int_3d_nc_6

20.42 int_3d_nc_8

21 master/fem

21.1 interpolation_matrix

21.2 mark

21.3 `mark_1d`

21.4 `mesh_1d_uniform`

21.5 `mesh_3d_uniform`

21.6 `mesh_interpolate`

21.7 `neighbour_1d`

21.8 `old`

21.9 `pdeeig_1d`

21.10 `pdeeig_2d`

21.11 `pdeeig_3d`

21.12 `polynomial_derivative_1d`

21.13 potential_const

21.14 potential_coulomb

21.15 potential_harmonic_oscillator

21.16 project_circle

21.17 project_rectangle

21.18 promote_1d_2_3

21.19 promote_1d_2_4

21.20 promote_1d_2_5

21.21 promote_1d_2_6

21.22 quadrilaterate

21.23 `recalculate_regularity_2d`

21.24 `refine_1d`

21.25 `refine_2d_21`

21.26 `refine_2d_structural`

21.27 `regularity_1d`

21.28 `regularity_2d`

21.29 `regularity_3d`

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

21.30 `relocate_2d`

21.31 `test_circmesh`

21.32 `test_hermit`

21.33 `tri_assign_points`

21.34 `triangulation_uniform`

21.35 `vander_1d`

`van der Monde matrix`

21.36 `vanderd_1d`

21.37 `vanderi_1d`

22 `master/hydrogen-spectrum`

22.1 `hydrogen_spectrum_1d`

22.2 `hydrogen_spectrum_2012_12_02`

22.3 `hydrogen_spectrum_2d`

22.4 hydrogen_spectrum_3d

23 master/lanczos

23.1 arnoldi

23.2 arnoldi_new

23.3 eigs_lanczos_man

23.4 lanczos

23.5 lanczos_

23.6 lanczos_biorthogonal

23.7 lanczos_biorthogonal_improved

23.8 lanczos_ghep

23.9 mv_lanczos

23.10 reorthogonalise

23.11 test_lanczos

24 master/linear-systems

24.1 gmres_man

break on convergence

24.2 minres_recycle

25 master/plot

25.1 attach_boundary_value

25.2 cartesian_polar

25.3 img_vargrid

25.4 plot_basis_functions

25.5 `plot_convergence`

25.6 `plot_dof`

25.7 `plot_eigenbar`

25.8 `plot_error_estimation`

25.9 `plot_error_estimation_2`

25.10 `plot_error_fem`

25.11 `plot_fdm_kernel`

25.12 `plot_fdm_vs_fem`

25.13 `plot_fem_accuracy`

25.14 `plot_function_and_grid`

25.15 `plot_hat`

25.16 `plot_hydrogen_wf`

25.17 `plot_mesh`

25.18 `plot_mesh_2`

25.19 `plot_refine`

25.20 `plot_refine_3d`

25.21 `plot_runtime`

25.22 `plot_spectrum`

25.23 `plot_wavefunction`

26 master/ported

26.1 assemble_2d_dphi_dphi

26.2 assemble_2d_phi_phi

26.3 assemble_3d_dphi_dphi

26.4 assemble_3d_phi_phi

26.5 dV_2d_

26.6 derivative_2d

26.7 derivative_3d

26.8 element_neighbour_2d

26.9 prefetch_2d_

26.10 promote_2d_3_10

26.11 promote_2d_3_15

26.12 promote_2d_3_21

26.13 promote_2d_3_6

26.14 promote_3d_4_10

26.15 promote_3d_4_20

26.16 promote_3d_4_35

26.17 vander_2d

26.18 vander_3d

27 master/sandbox

27.1 adapt

27.2 assoc_laguerre

27.3 assoc_legendre

27.4 c23

28 master/sandbox/cg

28.1 cg

28.2 cg_coef_to_poly

28.3 errmat

28.4 lanczos

28.5 laplacian_2d

28.6 test_cg_eigs

28.7 test_lanczos

29 master/sandbox

29.1 condition_number_higher_order

29.2 confinement_dat

29.3 convergence_2d_3d

29.4 convergence_matrix_powers

29.5 cut_out

29.6 derivative_2d

29.7 derivative_3d

29.8 `dummy`

29.9 `eig_error`

29.10 `eigs_fix`

29.11 `energy_level`

29.12 `equalise`

29.13 `example_int64`

Basic operations

Matrix multiplication

Timing

30 `master/sandbox/fem-matlab`

30.1 `boundary_circle`

30.2 boundary_rectangle

30.3 geometry_circle_with_hole

30.4 geometry_rectangle

31 master/sandbox

31.1 fem_2d_estimate_error

31.2 fem_assemble_scratch

31.3 fem_s

31.4 fourier_h

31.5 grad_2d

31.6 grad_3d

31.7 gradient

31.8 harmonic_oscillator

31.9 hydrogen_2d_analytic

31.10 hydrogen_boxed

31.11 hydrogen_boxed_old

31.12 hydrogen_wave

% Hydrogen atom

31.13 hydrogen_wf

31.14 ichol_man

31.15 known_eigenvalue

31.16 kron_man

31.17 laguerre

31.18 laplacian_arbitrary_order_old

31.19 laplacian_convergence

31.20 laplacian_cut_out

31.21 laplacian_cylindrical

31.22 laplacian_non_uniform_old

31.23 laplacian_polar

31.24 laplacian_simple

31.25 lderivative_3d

31.26 list_dat

31.27 matlab-horner

31.28 mesh_to_grid_2d_3

31.29 mg_mat

31.30 mv

31.31 orth2

31.32 partial_derivative_2d

31.33 partition_function

31.34 partition_function_old

31.35 poisson

31.36 poisson_fem

31.37 `potential`

31.38 `powerc`

31.39 `quick_newihsbour`

31.40 `radial`

31.41 `radial_convergence`

31.42 `radial_wafefunction`

31.43 `refine_2d`

31.44 `refine_3d`

31.45 `relerr`

31.46 `restore_cw`

31.47 runtime_bm

31.48 rydberg

31.49 s_old

31.50 snorm

31.51 spherical_harmonic

31.52 split_eig

31.53 sum1

31.54 sum3

32 master/sandbox/summation

32.1 acc

32.2 `add`

32.3 `ape`

32.4 `mmul_accurately`

32.5 `sum_kahan`

32.6 `sum_pairwise`

32.7 `test_sum`

33 `master/sandbox`

33.1 `test_convergence_ill_conditioned`

33.2 `test_fem_1d`

33.3 `test_fem_2d`

33.4 test_fem_3d

33.5 test_increase

33.6 test_lanczos_shift

33.7 test_ldl

33.8 test_power

33.9 trefethen_p8_fdm

33.10 wavefunc

33.11 xgrid

34 master/test

34.1 dat_test_lanczos_3d_k_20_n_40

34.2 poisson2d_blk

34.3 qr_implicit_givens_2

34.4 spectral_derivative_2d

34.5 test_2d_eigensolver_hydrogen

34.6 test_2d_refine

34.7 test_3d_eigensolver_hydrogen

34.8 test_FEM

34.9 test_Mesh_3d

34.10 test_arnoldi

34.11 test_arpackc

34.12 test_assemble

34.13 test_assembly_performance

34.14 test_bc_one_sided

34.15 test_compare_solvers

34.16 test_complete

34.17 test_convergence

34.18 test_convergence_b

34.19 test_df_2d

34.20 test_eig_algs

34.21 test_eig_inverse

34.22 test_eigs_lanczos

34.23 test_eigs_lanczos_1

34.24 test_eigs_lanczos_2

34.25 test_eigs_lanczos_performance

34.26 test_fdm

34.27 test_fdm_d_vargrid

34.28 test_fdm_spectral

34.29 test_fem

34.30 test_fem_1d

34.31 test_fem_1d_higher_order

34.32 test_fem_2d_adaptive

34.33 test_fem_2d_higher_order

34.34 test_fem_3d_higher_order

34.35 test_fem_3d_refine

34.36 test_fem_b

34.37 test_fem_derivative

34.38 test_fem_quadrature

34.39 test_final

34.40 test_fix_substitution

34.41 test_forward

34.42 test_get_sparse_arrays

34.43 test_harmonic_oscillator

34.44 test_high_order_fdm_periodic_bc

34.45 test_hydrogen_wf

34.46 test_ichol

34.47 test_interpolation

34.48 test_inverse_problem

34.49 test_it_vs_exact

34.50 test_jama

34.51 test_jd

34.52 test_jdqz

34.53 test_lanczos_2

34.54 test_lanczos_biorthogonal

34.55 test_laplacian

34.56 test_laplacian_non_uniform

34.57 test_laplacian_simple

34.58 test_mesh_2d_uniform

34.59 test_mesh_2d_uniform_2

34.60 test_mesh_circle

34.61 test_mesh_generation

34.62 test_mesh_interpolate

34.63 test_mg

34.64 test_minres_recycle

34.65 test_multigrid

34.66 test_nc

34.67 test_nonuniform_symmetric

34.68 test_pde

34.69 test_permutation

34.70 test_poison_fem

34.71 test_polar

34.72 test_potential

34.73 test_powers

34.74 test_precondition

34.75 test_project_rectangle

34.76 test_qr

34.77 test_quantum_well

34.78 test_radial_adaptive

34.79 test_radial_confinement

34.80 test_radial_fixes

34.81 test_refine_2d

34.82 test_refine_2d_b

34.83 test_refine_3d

34.84 test_refine_structural

34.85 test_regularisation

34.86 test_round_off

34.87 test_schrödinger_potentials

34.88 test_uniform_mesh

34.89 test_vargrid

35 number-theory

35.1 ceiln

floor to leading n-digits

35.2 digitsb

number of digits with respect to specified base

35.3 floorn

floor to n-digits

35.4 iseven

true for even numbers in X

35.5 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

35.6 nchoosek_man

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

35.7 pythagorean_triple

pythagorean triple

35.8 roundn

round to n digits

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/finite-difference

37.1 cdiff

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

37.2 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

37.3 cmean

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

37.4 derivative_matrix_1_1d

finite difference matrix of first derivative in one dimensions

37.5 derivative_matrix_2_1d

finite derivative matrix of second derivative in one dimension

37.6 derivative_matrix_2d

finite difference derivative matrix in two dimensions

37.7 derivative_matrix_curvilinear

derivative matrix on a curvilinear grid

37.8 derivative_matrix_curvilinear_2

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

37.9 difference_kernel

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

37.10 distmat

distance matrix for a 2 dimensional rectangular matrix

37.11 gradpde2d

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

37.12 laplacian

37.13 laplacian_fdm

finite difference matrix of the laplacian
BC

37.14 left

left element of vector, leftmost column is extrapolated

37.15 lrmean

mean of the left and right element

37.16 mid

mid point between neighbouring vector elements

37.17 pwmid

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

37.18 ratio

ratio of two subsequent values

37.19 steplength

step length of a vector if it were equispaced

37.20 swapoddeven

swap odd and even elements in a vector

37.21 test_derivative_matrix_2d

37.22 test_derivative_matrix_curvilinear

37.23 test_difference_kernel

38 numerical-methods/finite-volume/@Advection

38.1 Advection

FVM treatment of the Advection equation

38.2 dot_advection

advection equation

39 numerical-methods/finite-volume/@Burgers

39.1 burgers_split

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

39.2 dot_burgers_fdm

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

39.3 dot_burgers_fft

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

40 numerical-methods/finite-volume/@Finite_Volume

40.1 Finite_Volume

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

40.2 apply_bc

apply boundary conditions

40.3 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);
```

40.4 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

40.5 step_unsplit

step in time, without splitting the inhomogeneous term

41 numerical-methods/finite-volume/@Flux_Limiter

41.1 Flux_Limiter

class of flux limiters

41.2 beam_warming

beam warming scheme

low resolution

note: works only if sign of eigenvalues point into the same
direction according to RL

41.3 fromm

fromme limiter

low res

41.4 lax_wendroff

lax wendroff scheme

second order accurate, but no tvd

this is effectively not a limiter

eq. 6.39 in randall, leveque

41.5 minmod

min-mod schock limiter

41.6 monotized_central

monotonized central flux limiter

41.7 muscl

muscl flux limiter

41.8 superbee

superbee limiter

41.9 upwind

godunov scheme
godunov, first order accurate

41.10 vanLeer

van Leer limiter

42 numerical-methods/finite-volume/@KDV

42.1 dot_kdv_fdm

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

42.2 dot_kdv_fft

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

42.3 kdv_split

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

43 numerical-methods/finite-volume/@Reconstruct_Average_Evolve

43.1 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;
```

43.2 advect_highres

single time step for the reconstruct evolve algorithm

43.3 advect_lowres

single time step

low resolution

44 numerical-methods/finite-volume

44.1 Godunov

Godunov, upwind method for systems of pdes

44.2 Lax-Friedrich

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

44.3 Measure

44.4 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

44.5 fv_swe

wrapper for solving SWE

44.6 staggered_euler

forward euler method with staggered grid

44.7 staggered_grid

staggered grid approximation to the SWE

45 numerical-methods

45.1 grid2quad

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

46 numerical-methods/integration

46.1 cumintL

cumulative integral from left to right

46.2 cumintR

cumulative integral from right to left

46.3 int_trapezoidal

integrate y along x with the trapezoidal rule

47 numerical-methods/interpolation/@Kriging

47.1 Kriging

class for Kriging interpolation

47.2 estimate_semivariance

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

47.3 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

48 numerical-methods/interpolation/@RegularizedInterpolator

48.1 RegularizedInterpolator1

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

48.2 init

initialize the interpolator with a set of sampling points

49 numerical-methods/interpolation/@RegularizedInterpolator

49.1 RegularizedInterpolator2

class for regularized interpolation on an unstructures mesh (interpolation)

49.2 init

initialize the interpolator with a set of point samples

50 numerical-methods/interpolation/@RegularizedInterpolator

50.1 RegularizedInterpolator3

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

50.2 init

initialize the interpolator with a set of sampling points

51 numerical-methods/interpolation

51.1 IDW

spatial averaging by inverse distance weighting

51.2 IPoly

polynomial interpolation class

51.3 IRBM

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

51.4 ISparse

sparse interpolation class

51.5 Inn

nearest neighbour interpolation

51.6 Interpolator

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

51.7 fixnan

fill nan-values in vector with gaps

51.8 idw1

spatial average by inverse distance weighting

51.9 idw2

spatial average by inverse distance weighting

51.10 inner2outer

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

51.11 inner2outer2

interpolate from element (segment) centres to edge points

51.12 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

51.13 interp1_man

interpolate

51.14 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

51.15 interp1_slope

quadratic interpolation returning value and derivative(s)

51.16 interp1_smooth

51.17 interp1_unique

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

51.18 interp2_man

nearest neighbour interpolation in two dimensions

51.19 interp_angle

interpolate an angle

51.20 interp_fourier

interpolation by the fourier method

51.21 interp_fourier_batch

batch interpolation by the fourier interpolation

51.22 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_);

51.23 interp_sn2

interpolation in streamwise coordinates

51.24 interp_sn3

51.25 interp_sn_

51.26 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
 $ratio^1 = \exp(a*1)$

51.27 resample1

interpolation along a parametric curve with variable step width

51.28 resample_d_min

resample a function

51.29 resample_vector

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

51.30 test_interp1_limited

52 numerical-methods

52.1 inverse_complex

53 numerical-methods/ode

53.1 bvp2_check_arguments

53.2 bvp2c

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

odefun provides ode coefficients c:

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

subject to the boundary conditions

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

$$\begin{aligned} b_1 y + b_2 y' &= f \\ q(x,1) * (p(x,1) y_1(x) + p(x,2) y_1'(x) \end{aligned}$$

```

+ 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])

```

53.3 bvp2c2

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

input:

```

x : [nxc] discretized domain
n : number of vertices
nxc = 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*nxc x 2*ns] discretisation matrix
rhs : [2*nxc x 1] right hand side

```

```

y = A^-1 rhs

```

53.4 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])

```


53.5 bvp2wavetrain

solve second order boundary value problem by repeated integration

53.6 bvp2wavetwopass

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

53.7 ivp_euler_forward

solve initial value problem by the euler forward method

53.8 ivprk2

solve initial value problem by the two step runge kutta method

53.9 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]
```

53.10 ode2characteristic

second order odes
transmitted and reflected wave

53.11 step_trapezoidal

single trapezoidal step

53.12 test_bvp2

54 numerical-methods/optimisation

54.1 armijo_stopping_criterion

armijo stopping criterion for optimizations

54.2 astar

astar path finding algorithm

54.3 binsearch

binary search on a line

54.4 bisection

bisection

54.5 box1

test objective function for optimisation routines

54.6 box2

54.7 cauchy

54.8 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

54.9 directional_derivative

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

54.10 dud

optimization by the dud algorithm

54.11 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
ouput:
tdx : index where extremum should be computed
t0 : location of the extremum
val0 : value of extremum

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

54.12 extreme_quadratic

54.13 ftest

54.14 grad

numerical gradient

54.15 hessian

numerical hessian

54.16 hessian_from_gradient

numerical hessian from gradient

54.17 hessian_projected

numerical hessian projected to one dimension

54.18 line_search

bisection routine

54.19 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

54.20 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
```

54.21 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
```

54.22 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
```

54.23 line_search_quadratic2

```
quadratic line search
```

54.24 line_search_wolfe

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

54.25 ls_bgfs

least squares by the bgfs method

54.26 ls_broyden

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

BGFS:
Broyden 1965
Fletcher 1970
Goldfarb 1970
Shanno 1970

54.27 ls_generalized_secant

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

54.28 nlcg

non-linear conjugate gradient
input:
x : nx1 start vectort

opt : struct options
fdx : gradient constraint

54.29 nlls

non-linear least squares

54.30 picard

picard iteration

54.31 poly_extrema

extrema of a polynomial

54.32 quadratic_function

evaluate quadratic function in higher dimensions

54.33 quadratic_programming

optimize by quadratic programming

54.34 quadratic_step

single step of the quadratic programming

54.35 rosenbrock

rosenbrock test function

54.36 `sqrt_heron`

Heron's method for the square root

54.37 `test_directional_derivative`

54.38 `test_dud`

54.39 `test_line_search_quadratic2`

54.40 `test_ls_generalized_secant`

54.41 `test_nlcg_6_order`

54.42 `test_nlls`

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

55 numerical-methods/piecewise-polynomials

55.1 `Hermite1`

hermite polynomial interpolation in 1d

55.2 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)

55.3 hp2_predict

prediction with pw hermite polynomial
c are values at support points

55.4 hp_predict

predict with piecewise hermite polynomial

55.5 hp_regress

fit piecewise hermite polynomial
coefficients are values and derivatives

55.6 lp_count

lagrangian basis for interpolation
count number of valid samples

55.7 lp_predict

lagrangian basis piecwie interpolation, predicor

55.8 `lp_regress`

55.9 `lp_regress_`

56 `regression/@PolyOLS`

56.1 `PolyOLS`

class for polynomial least squares

56.2 `coefftest`

56.3 `detrend`

detrending by polynomial regression

56.4 `fit`

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

56.5 `fit_`

fit a polynomial function

56.6 `predict`

predict polynomial function values

56.7 predict_

56.8 slope

slope by linear regression

57 regression/@PowerLS

57.1 PowerLS

class for power law regression

57.2 fit

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

57.3 predict

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

57.4 predict_

58 regression/@Theil

58.1 Theil

Kendal-Theil-Sen robust regression

58.2 detrend

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

58.3 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

58.4 predict

predict values and confidence intervals with the Theil-Sen method

58.5 slope

fit the slope with the Theil-Sen method

59 regression

linear and non-linear regression

59.1 Theil_Multivariate

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

59.2 areg

regression using the pth-fraction of samples with smallest residual

59.3 giniрег

gini regression

59.4 hesssimplereg

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

59.5 l1lin

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

59.6 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)

59.7 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

59.8 regression_method_of_moments

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

59.9 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

59.10 theil2

Theil senn-estimator for two dimensions (glm)

59.11 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

59.12 total_least_squares

total least squares

59.13 weighted_median_regression

weighted median regression
c.f. Scholz, 1978

60 set-theory

60.1 issubset

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

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

61 signal-processing

61.1 acf_effective_sample_size

effective sample size from acf

61.2 acf_genton

autocorrelation function

61.3 acfar1

Autocorrelation function of the finite AR1 process

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

61.4 acfar1_2

autocorrelation of the ar1 process

61.5 acfar2

impulse response of the ar2 process

61.6 acfar2_2

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

61.7 ar1_cutoff_frequency

61.8 `ar1_effective_sample_size`

effective sample size correction for autocorrelated series

61.9 `ar1_mse_mu_single_sample`

standard error of a single sample of an ar1 correlated process

61.10 `ar1_mse_pop`

variance of the population mean of a single realisation around zero

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

61.11 `ar1_mse_range`

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

61.12 `ar1_spectrum`

spectrum of the ar1 process

61.13 `ar1_to_tikhonov`

convert ar1 correlation to tikhonovs lambda

61.14 `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

61.15 ar1_var_factor_

variance of an autocorrelated finite process

61.16 ar1_var_range2

variance of sub sample starting at the end of the series

from the finite length first order autocorrelated process

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

61.17 ar1delay

approximate acf by the ar1 process

acf: autocovariance or autocorrelation function

nf : skip first samples (for mixed geometric-arithmetic series (ARMA))

61.18 ar1delay_old

autocorrelation of the residual

61.19 ar2conv

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

61.20 ar2dof

effective samples size for the ar2 process

61.21 ar2param

ar2 parameter estimation from first two terms of acf

```
acf = [1 a1 a2 ...]
```

61.22 asymwin

creates asymmetrical filter windows

filter will always have negative weights

61.23 autocorr_fft

autocorrelation function

61.24 bandpass

bandpass filter

61.25 bandpass2

bandpass filter

61.26 bartlett

Effective sample size factor for bartlett window

c.f. thiebaux

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

```
c = acf
```

note: results seams always to be 1 tac too low

T : reduction factor for dof

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

61.27 bartlett_spectrogram

bartlet spectrogramm

TODO sliding window

61.28 bin1d

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

61.29 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

61.30 binormrnd

generate two correlated normally distributed vectors

61.31 conv1_man

convolutions with padding

61.32 conv2_man

convolution in 2d

61.33 conv2z

61.34 conv30

convolve with rectangular window of lenght *n*
circular boundaries

61.35 conv_

convolution of a with b

61.36 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

61.37 convz

61.38 cosexpdelay

61.39 csmooth

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

61.40 daniell_window

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

61.41 danielle_window

danielle fourier window

61.42 db2neper

convert decibel to neper

61.43 db2power

power ratio from db

61.44 derive_danielle_weight

61.45 derive_limit_0_acfar

61.46 detect_peak

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

61.47 digital_low_pass_filter

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

61.48 doublesum_ij

double sum of r^i

61.49 `effective_sample_size_to_ar1`

convert effective sample size to ar1 correlation

61.50 `flt_hodges_lehman`

61.51 `filter1`

filter along one dimension

61.52 `filter2`

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

61.53 `filter_`

invalidate values that exceed n-times the robust standard deviation

61.54 `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 alising (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

61.55 filterp

61.56 filterp1

fir filter with some fancy extras

61.57 filterstd

61.58 firls_man

design finite impulse response filter by the least squares method

61.59 flattopwin

the flat top window

61.60 frequency_response_boxcar

frequency response of a boxcar filter

61.61 freqz_boxcar

frequency response of a boxcar filter

61.62 gaussfilt1

filter data series with a gaussian window

61.63 hanchangewin

hanning window for change point detection

61.64 hanchangewin2

hanning window for change point detection

61.65 hanwin

hanning filter window

61.66 hanwin_

hanning filter window

61.67 highpass

high pass filter

61.68 kaiserwin

kaiser filter window

61.69 kalman

Kalman filter

61.70 lanczoswin

Lanczos window

61.71 last

lake tail, but for matrices

61.72 lowpass

low pass filter

61.73 lowpass2

design low pass filter with cutoff-frequency f1

61.74 lowpass_iir

iir-low pass

61.75 lowpass_iir_symmetric

two-sided iir low pass filter (for symmetry)

61.76 lowpassfilter2

low-pass filter of data

61.77 maxfilt1

61.78 meanfilt1

moving average filter with special treatment of the boundaries

61.79 medfilt1_man

moving median filter, supports columnwise operation

61.80 medfilt1_man2

moving median filter with special treatment of boundaries

61.81 medfilt1_padded

median filter with padding

61.82 medfilt1_reduced

median filter with padding

61.83 mid_term_single_sample

variance of single sample, mid term

61.84 minfilt1

61.85 mu2ar1

error variance of the mean of the finite length ar1 process

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

61.86 nanautocorr

autocorrelation with nan-values

61.87 nanmedfilt1

medfilt1, skipping nans

61.88 neper2db

convert neper to db

61.89 peaks_man

peaks of a periodogram

61.90 polyfilt1

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

61.91 qmedfilt1

medfilt1, after fitting a quadratic polynomial

61.92 randar1

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

61.93 randar1_dual

draw random variables of two correlated ar1 processes

61.94 randar2

generate ar2 process

61.95 randarp

randomly generate the instance of an ar-p process

61.96 range_window

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

61.97 rectwin

rectangular window

61.98 recursive_sum

61.99 `select_range`

61.100 `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

61.101 `smooth2`

smooth vectos of X

61.102 `smooth_man`

61.103 `smooth_parametric`

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

61.104 `smooth_parametric2`

parametrically smooth the curve

61.105 `smoothfft`

filter with fast fourier transform

61.106 `spectrogram`

spectrogram

61.107 std_window

moving block standard deviation

61.108 sum_i_lag

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

61.109 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

61.110 sum_ii_

61.111 sum_ij

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

61.112 sum_ij_

61.113 sum_ij_partial_

61.114 sum_multivar

sum of matrix entries of bivariate ar1 process

61.115 test_acfar1

61.116 test_acfar1_2

61.117 test_acfar1_3

61.118 test_acfar1_4

61.119 test_acfar2

61.120 test_ar1_var_factor

61.121 test_ar1_var_factor_2

61.122 test_ar1_var_mu_single_sample

61.123 test_ar1_var_pop

61.124 test_ar1_var_pop_1

61.125 test_ar1delay

61.126 test_bivariate_covariance_term

61.127 test_convexity

61.128 test_lanczoswin

61.129 test_madcorr

61.130 test_randar1

61.131 test_randar1_multivariate

61.132 test_randar2

61.133 test_sum_ij

61.134 test_sum_multivar

61.135 test_trifilt1

61.136 test_wautocorr

61.137 test_wavelet_transform

61.138 test_wordfilt

61.139 test_xar1_mid_term

61.140 tikhonov_to_ar1

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

61.141 trapwin

trapezoidal filter window

61.142 trifilt1

filter with triangular window

61.143 triwin

triangular filter window

61.144 triwin2

triangular filter window

61.145 varar1

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

61.146 welch_spectrogram

welch spectrogram

61.147 wfilt

filter with window

61.148 winbandpass

filter with bandpass

61.149 window_make_odd

61.150 winfilt0

filter with window

61.151 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

61.152 wmeanfilt

mean filter with window

61.153 wmedfilt

median filter with window

61.154 wordfilt

weighted order filter

61.155 wordfilt_edgeworth

weighed order filter

61.156 xar1

61.157 xcorr_man

cross correlation of two sampled ar1 processes

62 sorting

62.1 sort2

sort two numbers

62.2 sort2d

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

63 special-functions

63.1 bessell_sphere

spherical Bessel function of the first kind

63.2 hankel_sphere

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

63.3 hermite

probabilistic's hermite polynomial by recurrence relation

input :
n : order
x : value

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

63.4 legendre_man

legendre polynomials

63.5 neumann_sphere

spherical Neumann function
Bessel function of the second kind

64 statistics

64.1 atan_s2

stadard deviation of the arcus tangens by means of taylor expansion

64.2 beta_mode_to_parameter

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

64.3 correlation_confidence_pearson

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

65 statistics/distributions

65.1 PDF

class for quasi-distributions from a set of sampling points

65.2 binorm_separation_coefficient

separation coefficient of a bimodal normal distribution

65.3 binormcdf

bio-modal gaussian distribution

65.4 binormfit

fit sum of to normal distribution to a histogram

65.5 binormpdf

65.6 edgeworth_cdf

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

65.7 edgeworth_pdf

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

65.8 logn_mode2param

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

65.9 logn_param2mode

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

65.10 lognpdf_

log normal distribution called by modes rather than parameters

65.11 pdfsample

pdf from sample distribution
Note: better use kernel density estimates

65.12 t2cdf

Hotelling's T-squared cumulative distribution

65.13 t2inv

inverse of Hotelling's T-squared cumulative distribution

66 statistics

66.1 example_standard_error_of_sample_quantiles

66.2 f_var_finite

reduction of variance when sampling from a finite population
without replacement

66.3 gamma_mode_to_parameter

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

66.4 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

66.5 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

67 statistics/information-theory

67.1 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

67.2 bayesian_information_criterion

bayesian information criterion

68 statistics

68.1 kurtncdf

68.2 kurtnpdf

68.3 kurtosis_bias_corrected

bias corrected kurtosis

68.4 limit

limit a by lower and upper bound

68.5 logfactorial

approximate log of the factorial

68.6 loglogpdf

68.7 logskewcdf

68.8 logskewpdf

69 statistics/logu

69.1 lambertw_numeric

lambert-w function

69.2 logtrialtcdf

pdf of a logarithmic triangular distribution

69.3 logtrialtinv

inverse of the logarithmic triangular distribution

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

69.4 logtrialtmean

mean of the logarithmic triangular distribution

69.5 logtrialtpdf

density of the logarithmic triangular distribution

69.6 logtrialtrnd

69.7 logtricdf

cumulative distribution of the logarithmic triangular distribution

69.8 logtriinv

invere of the logarithmic triangular distribution

69.9 logtrimean

mean of the logarithmic triangular distribution

69.10 logtripdf

probability density of the logarithmic triangular distribution

69.11 logtirnd

69.12 logucdf

probability density of the logarithmic uniform distribution

69.13 logucm

central moments of the log-uniform distribution

69.14 loguinv

inverse of the log-uniform distribution

69.15 logumean

mean of the log-uniform distribution

69.16 logupdf

pdf of the log uniform distribution

69.17 logurnd

random numbers following a log-uniform distribution

69.18 loguvar

variance of the log-uniform distribution

69.19 medlogu

median of the log-uniform distribution

69.20 test_logurnd

69.21 tricdf

cumulative distribution of the log-triangular distribution

69.22 triinv

inverse of the triangular distribution

69.23 trimedian

median of the triangular distribution

69.24 tripdf

probability density of the triangular distribution

69.25 trirnd

random numbers of the triangular distribution

70 statistics

70.1 maxnnormals

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

70.2 midrange

mid range of columns of X

70.3 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

70.4 mode_man

71 statistics/moment-statistics

71.1 autocorr_man3

autocorrelation of the columns of X

71.2 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

71.3 autocorr_man5

autocorrelation of the columns of X

71.4 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

71.5 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

71.6 corr_man

correlation of two vectors

71.7 cov_man

covariance matrix of two vectors

71.8 dof

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

71.9 edgeworth_quantile

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

71.10 effective_sample_size

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

71.11 f_correlation

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

71.12 f_finite

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

71.13 lmean

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

71.14 lmoment

l-moment of vector x

71.15 maskmean

mean of the masked values of X

71.16 masknanmean

71.17 mean1

mean of x

71.18 mean_man

mean and standard error of X

71.19 mse

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

71.20 nanautocorr_man1

autocorrelation of a vector with nan-values

71.21 nanautocorr_man2

autocorrelation of a vector with nan-values

71.22 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

71.23 nancorr

(co)-correlation matrix when samples a NaN

71.24 nancumsum

cumulative sum, setting nan values to zero

71.25 nanlmean

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

71.26 nanr2

coefficient of determination when samples are invalid

71.27 nanrms

root mean square value when sample contains nan-values

71.28 nanrmse

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

71.29 nanserr

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

71.30 nanwmean

weighted mean
min_x sum w (x-mu)^2 => mu = sum(wx)/sum(w)
varargin can be dim
function [mu serr] = nanwmean(w,x)

71.31 nanwstd

weighed standard deviation

71.32 nanwvar

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

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

71.33 nanxcorr

71.34 pearson

pearson correlation coefficient

71.35 pearson_to_kendall

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

71.36 pool_samples

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

71.37 qmean

trimmed mean

71.38 range_mean

71.39 rmse

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

71.40 serr

standard error of the mean of a set of uncorrelated samples

71.41 serr1

71.42 test_qskew

71.43 test_qstd_qskew_optimal_p

71.44 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

71.45 wcorr

correlation of two vectors when samples are weighted

71.46 wcov

covariance of two vectors when samples are weighted

71.47 wdof

effective degrees of freedom for weighted samples

71.48 wkurt

kurtosis with weighted samples

71.49 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)

71.50 wrms

weighted root mean square error

71.51 wserr

weighted root mean square error

71.52 wskew

skewness of a weighted set of samples

71.53 wstd

weighed standard deviation

71.54 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

72 statistics

72.1 nangeomean

72.2 nangeostd

geometric standard deviation ignoring nan-values

73 statistics/nonparametric-statistics

73.1 kernel1d

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

73.2 kernel2d

kernel density estimate in two dimensions

74 statistics

74.1 normmoment

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

74.2 normpdf2

pdf of the bivariate normal distribution

75 statistics/order-statistics

75.1 hodges_lehmann_location

hodges lehman location estimator

Asymptotic rms efficiency of location estimte:

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

75.2 kendall

kendall correlation coefficient

75.3 kendall_to_pearson

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

c.f. Kruska, 1985

75.4 mad2sd

transform median absolute deviation to standard deviation
for normal distributed values

75.5 madcorr

proxy correlation by median absolute deviation

75.6 median2_holder

75.7 median_ci

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

75.8 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

75.9 mediani

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

75.10 nanmadcorr

proxy correlation by median absolute deviation

75.11 nanwmedian

weighted median, skips nan-values

75.12 nanwquantile

weighted quantile, skips nan values

75.13 oja_median

two dimensional oja median

note: the multivariate median is not unique

oja 1983, for extension to multivariate function, see chaudhri

75.14 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

75.15 qmoments

moments estimated from quantiles

75.16 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

75.17 qskewq

skewness estimated by quantiles

75.18 qstdq

proxy standard deviation determined by quantiles

75.19 quantile1_optimisation

75.20 quantile2_breckling

quantile regression

75.21 quantile2_chaudhuri

quantile regression

75.22 quantile2_projected

quantile in two dimensions

75.23 quantile2_projected2

spatial quantile for chosen direction

75.24 quantile_envelope

75.25 quantile_regression_simple

simple quantile regression

75.26 ranking

ranking for spearman statistics

75.27 spatial_median

c.f. Oja 2008

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

75.28 spatial_quantile

spatial quantile

75.29 spatial_quantile2

spatial quantile

75.30 spatial_quantile3

spatial quantile

75.31 spatial_rank

unsigned rank

75.32 spatial_sign

spatial sign

75.33 spatial_signed_rank

signed rank

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

75.34 spearman

spearman's product moment coefficient

75.35 spearman_rank

75.36 spearman_to_pearson

conversion of spearman rank to person product moment correlation
coefficient

75.37 wmedian

weighted median

75.38 wquantile

weighted quantile

76 statistics

76.1 qstd

76.2 quantile_extrap

77 statistics/random-number-generation

77.1 laplacernd

random number of laplace distribution

77.2 randc

correlate to correlated standard normally distributed vectors

77.3 skewrnd

random numbers of the skew normal distribution

77.4 skewrnd2

random numbers of the skew normal distribution

78 statistics

78.1 range

mid range

78.2 resample_with_replacement

79 statistics/resampling-statistics/@Jackknife

79.1 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

79.2 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

79.3 matrix1_STATIC

matrix of estimation for leaving out two samples at a time

79.4 matrix2

matrix of estimations for jackknife with two samples left out

80 statistics/resampling-statistics

80.1 block_jackknife

80.2 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

80.3 moving_block_jackknife

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

80.4 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

80.5 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

81 statistics

81.1 scale_quantile_sd

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

81.2 skewpdf

skew-normal distribution
c.f. Azzalini 1985

81.3 trimmed_mean

trimmed mean

81.4 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

81.5 ttest_man

two-sample t-test
unequal sample size
equal variance

81.6 ttest_paired

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

81.7 wharmean

weighted harmonic mean

82 wavelet

82.1 contiuous_wavelet_transform

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

82.2 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

82.3 example_wavelets

82.4 phasewrap

wrap the phase to $\pm \pi$

82.5 test_cwt_man

82.6 test_phasewrap

82.7 test_wavelet

82.8 test_wavelet2

82.9 test_wavelet_analysis

82.10 test_wavelet_reconstruct

82.11 test_wtc

82.12 wavelet

wavelet windows

82.13 wavelet_reconstruct

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

82.14 wavelet_transform

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

83 mathematics

mathematical functions of various kind

83.1 wrapphase