

IntegrationRoutines

5.3

Generated by Doxygen 1.8.14

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Data Structure Index	7
4.1	Data Structures	7
5	File Index	11
5.1	File List	11
6	Module Documentation	15
6.1	Models	15
6.1.1	Detailed Description	15
6.2	Utils	16
6.2.1	Detailed Description	16
6.3	Integration	17
6.3.1	Detailed Description	17
6.4	GaussJackson	18
6.4.1	Detailed Description	19
6.5	Lsode	20
6.5.1	Detailed Description	24

6.5.2	Function Documentation	24
6.5.2.1	allocate_arrays() [1/2]	24
6.5.2.2	allocate_arrays() [2/2]	25
6.5.2.3	calculate_epsilon()	25
6.5.2.4	calculate_integration_coefficients()	25
6.5.2.5	check_interface_data()	26
6.5.2.6	create_constructor()	26
6.5.2.7	create_copy() [1/5]	27
6.5.2.8	create_copy() [2/5]	27
6.5.2.9	create_copy() [3/5]	27
6.5.2.10	create_copy() [4/5]	27
6.5.2.11	create_copy() [5/5]	28
6.5.2.12	create_first_order_ode_integrator()	28
6.5.2.13	create_generalized_deriv_second_order_ode_integrator()	28
6.5.2.14	create_integration_controls()	29
6.5.2.15	create_second_order_ode_integrator()	29
6.5.2.16	destroy_allocated_arrays() [1/2]	30
6.5.2.17	destroy_allocated_arrays() [2/2]	30
6.5.2.18	gauss_elim_factor()	30
6.5.2.19	index_of_max_magnitude()	31
6.5.2.20	integrate() [1/4]	31
6.5.2.21	integrate() [2/4]	31
6.5.2.22	integrate() [3/4]	32
6.5.2.23	integrate() [4/4]	33
6.5.2.24	integrator_compute_new_order()	33
6.5.2.25	integrator_compute_new_order_check_step_error()	34
6.5.2.26	integrator_compute_new_order_prep()	34
6.5.2.27	integrator_core()	35
6.5.2.28	integrator_corrector_converged()	36
6.5.2.29	integrator_corrector_failed_part1()	36

6.5.2.30	integrator_corrector_failed_part2()	37
6.5.2.31	integrator_corrector_iteration()	37
6.5.2.32	integrator_error_test_failed()	38
6.5.2.33	integrator_fail_reset_order_1_part1()	38
6.5.2.34	integrator_fail_reset_order_1_part2()	39
6.5.2.35	integrator_predict()	39
6.5.2.36	integrator_reset_iteration_loop_part1()	40
6.5.2.37	integrator_reset_iteration_loop_part2()	40
6.5.2.38	integrator_reset_method_coeffs()	40
6.5.2.39	integrator_reset_yh()	41
6.5.2.40	integrator_set_new_order()	41
6.5.2.41	integrator_terminate()	41
6.5.2.42	integrator_test_stepsize_change()	42
6.5.2.43	integrator_wrapup()	42
6.5.2.44	interpolate_y()	42
6.5.2.45	jacobian_prep_init()	43
6.5.2.46	jacobian_prep_loop()	44
6.5.2.47	jacobian_prep_wrap_up()	45
6.5.2.48	linear_chord_iteration()	45
6.5.2.49	linear_solver()	46
6.5.2.50	load_derivatives()	46
6.5.2.51	load_ew_values()	46
6.5.2.52	LsodeControlDataInterface()	47
6.5.2.53	LsodeFirstOrderODEIntegrator() ^[1/2]	47
6.5.2.54	LsodeFirstOrderODEIntegrator() ^[2/2]	47
6.5.2.55	LsodeGeneralizedDerivSecondOrderODEIntegrator() ^[1/2]	47
6.5.2.56	LsodeGeneralizedDerivSecondOrderODEIntegrator() ^[2/2]	48
6.5.2.57	LsodeIntegrationControls()	48
6.5.2.58	LsodeIntegratorConstructor()	48
6.5.2.59	LsodeSecondOrderODEIntegrator() ^[1/2]	48

6.5.2.60	LsodeSecondOrderODEIntegrator() [2/2]	49
6.5.2.61	LsodeSimpleSecondOrderODEIntegrator()	49
6.5.2.62	magnitude_of_weighted_array() [1/2]	50
6.5.2.63	magnitude_of_weighted_array() [2/2]	50
6.5.2.64	manager_check_stop_conditions()	51
6.5.2.65	manager_initialize_calculation_part1()	51
6.5.2.66	manager_initialize_calculation_part2()	51
6.5.2.67	manager_integration_loop_part1()	52
6.5.2.68	manager_integration_loop_part2()	52
6.5.2.69	manager_integration_loop_part3()	52
6.5.2.70	manager_set_calculation_phase_eq_2_reload()	53
6.5.2.71	process_entry_point_cycle_start()	53
6.5.2.72	reset_integrator()	53
6.5.2.73	set_abs_tol()	53
6.5.2.74	set_rel_tol()	54
6.5.2.75	update_control_data()	54
6.5.2.76	~LsodeFirstOrderODEIntegrator()	54
6.5.2.77	~LsodeGeneralizedDerivSecondOrderODEIntegrator()	55
6.5.2.78	~LsodeSecondOrderODEIntegrator()	55
7	Namespace Documentation	57
7.1	er7_utils Namespace Reference	57
7.2	jeod Namespace Reference	57
7.2.1	Detailed Description	59
7.2.2	Typedef Documentation	59
7.2.2.1	GaussJacksonIntegratorBaseFirst	60
7.2.2.2	GaussJacksonIntegratorBaseSecond	60
7.2.3	Function Documentation	60
7.2.3.1	cast_to_gj_controls()	60
7.2.3.2	operator<<()	61
7.2.3.3	set_default_config_values()	62
7.2.3.4	validate_config()	62

8 Data Structure Documentation	63
8.1 <code>er7_utils::DoubleTwoDArray</code> Class Reference	63
8.1.1 Detailed Description	63
8.1.2 Friends And Related Function Documentation	63
8.1.2.1 <code>init_attrer7_utils__DoubleTwoDArray</code>	64
8.1.2.2 <code>InputProcessor</code>	64
8.2 <code>jeod::GaussJacksonCoefficientsPair</code> Class Reference	64
8.2.1 Detailed Description	65
8.2.2 Constructor & Destructor Documentation	65
8.2.2.1 <code>GaussJacksonCoefficientsPair()</code> [1/2]	65
8.2.2.2 <code>~GaussJacksonCoefficientsPair()</code>	65
8.2.2.3 <code>GaussJacksonCoefficientsPair()</code> [2/2]	65
8.2.3 Member Function Documentation	65
8.2.3.1 <code>allocate_arrays()</code>	65
8.2.3.2 <code>apply()</code> [1/2]	66
8.2.3.3 <code>apply()</code> [2/2]	66
8.2.3.4 <code>configure()</code>	67
8.2.3.5 <code>deallocate_arrays()</code>	67
8.2.3.6 <code>operator=()</code>	67
8.2.3.7 <code>printOut()</code>	67
8.2.3.8 <code>swap()</code>	68
8.2.4 Friends And Related Function Documentation	68
8.2.4.1 <code>init_attrjeod__GaussJacksonCoefficientsPair</code>	68
8.2.4.2 <code>InputProcessor</code>	68
8.2.5 Field Documentation	69
8.2.5.1 <code>gj_coefs</code>	69
8.2.5.2 <code>sa_coefs</code>	69
8.3 <code>jeod::GaussJacksonCoeffs</code> Class Reference	69
8.3.1 Detailed Description	70
8.3.2 Constructor & Destructor Documentation	70

8.3.2.1	GaussJacksonCoeffs() [1/2]	71
8.3.2.2	GaussJacksonCoeffs() [2/2]	71
8.3.2.3	~GaussJacksonCoeffs()	71
8.3.3	Member Function Documentation	71
8.3.3.1	compute_coeffs()	71
8.3.3.2	configure()	72
8.3.3.3	operator=()	72
8.3.3.4	swap()	73
8.3.4	Friends And Related Function Documentation	73
8.3.4.1	init_attrjeod__GaussJacksonCoeffs	73
8.3.4.2	InputProcessor	73
8.3.4.3	operator<<	73
8.3.5	Field Documentation	74
8.3.5.1	corrector	74
8.3.5.2	max_order	74
8.3.5.3	order	74
8.3.5.4	predictor	75
8.4	jeod::GaussJacksonConfig Class Reference	75
8.4.1	Detailed Description	76
8.4.2	Constructor & Destructor Documentation	76
8.4.2.1	GaussJacksonConfig()	76
8.4.3	Member Function Documentation	76
8.4.3.1	default_configuration()	76
8.4.3.2	standard_configuration()	77
8.4.3.3	validate_configuration()	77
8.4.4	Friends And Related Function Documentation	77
8.4.4.1	init_attrjeod__GaussJacksonConfig	77
8.4.4.2	InputProcessor	77
8.4.5	Field Documentation	77
8.4.5.1	absolute_tolerance	78

8.4.5.2	final_order	78
8.4.5.3	initial_order	78
8.4.5.4	max_correction_iterations	78
8.4.5.5	ndoubling_steps	79
8.4.5.6	priming_technique	79
8.4.5.7	relative_tolerance	79
8.5	jeod::GaussJacksonFirstOrderODEIntegrator Class Reference	80
8.5.1	Detailed Description	80
8.5.2	Constructor & Destructor Documentation	81
8.5.2.1	~GaussJacksonFirstOrderODEIntegrator()	81
8.5.2.2	GaussJacksonFirstOrderODEIntegrator() [1/2]	81
8.5.2.3	GaussJacksonFirstOrderODEIntegrator() [2/2]	81
8.5.3	Member Function Documentation	81
8.5.3.1	create_copy()	82
8.5.3.2	integrate()	82
8.5.3.3	operator=()	82
8.5.3.4	reset_integrator()	82
8.5.3.5	swap() [1/2]	83
8.5.3.6	swap() [2/2]	83
8.6	jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator Class Reference	83
8.6.1	Detailed Description	84
8.6.2	Constructor & Destructor Documentation	85
8.6.2.1	GaussJacksonGeneralizedDerivSecondOrderODEIntegrator() [1/3]	85
8.6.2.2	GaussJacksonGeneralizedDerivSecondOrderODEIntegrator() [2/3]	85
8.6.2.3	GaussJacksonGeneralizedDerivSecondOrderODEIntegrator() [3/3]	85
8.6.2.4	~GaussJacksonGeneralizedDerivSecondOrderODEIntegrator()	86
8.6.3	Member Function Documentation	86
8.6.3.1	create_copy()	86
8.6.3.2	integrate()	86
8.6.3.3	operator=()	86

8.6.3.4	reset_integrator()	87
8.6.3.5	swap()	87
8.6.4	Friends And Related Function Documentation	87
8.6.4.1	init_attrjeod__GaussJacksonGeneralizedDerivSecondOrderODEIntegrator	87
8.6.4.2	InputProcessor	87
8.6.5	Field Documentation	87
8.6.5.1	pos_integrator	88
8.6.5.2	posdot	88
8.6.5.3	posdotdot	88
8.6.5.4	vel_integrator	89
8.7	jeod::GaussJacksonIntegrationControls Class Reference	89
8.7.1	Detailed Description	91
8.7.2	Constructor & Destructor Documentation	91
8.7.2.1	GaussJacksonIntegrationControls() [1/3]	91
8.7.2.2	GaussJacksonIntegrationControls() [2/3]	91
8.7.2.3	GaussJacksonIntegrationControls() [3/3]	92
8.7.2.4	~GaussJacksonIntegrationControls()	92
8.7.3	Member Function Documentation	92
8.7.3.1	create_copy()	92
8.7.3.2	get_coeff()	93
8.7.3.3	get_config()	93
8.7.3.4	get_priming_controls()	93
8.7.3.5	get_state_machine()	94
8.7.3.6	integrate()	94
8.7.3.7	integrate_edit()	95
8.7.3.8	integrate_gj()	95
8.7.3.9	operator=()	95
8.7.3.10	reset_integrator()	95
8.7.3.11	start_cycle()	96
8.7.3.12	swap()	96

8.7.4	Friends And Related Function Documentation	96
8.7.4.1	init_attrjeod__GaussJacksonIntegrationControls	97
8.7.4.2	InputProcessor	97
8.7.5	Field Documentation	97
8.7.5.1	at_end_of_tour	97
8.7.5.2	coeff	97
8.7.5.3	config	98
8.7.5.4	cycle_dyndt	98
8.7.5.5	cycle_simdt	98
8.7.5.6	cycle_starttime	98
8.7.5.7	edit_count	99
8.7.5.8	fsm_state	99
8.7.5.9	initial_order	99
8.7.5.10	max_correction_iterations	99
8.7.5.11	order	100
8.7.5.12	priming_controls	100
8.7.5.13	reset_time	100
8.7.5.14	state_machine	100
8.8	jeod::GaussJacksonIntegratorBase< State, Primer > Class Template Reference	101
8.8.1	Detailed Description	105
8.8.2	Constructor & Destructor Documentation	105
8.8.2.1	GaussJacksonIntegratorBase() [1/3]	106
8.8.2.2	GaussJacksonIntegratorBase() [2/3]	106
8.8.2.3	GaussJacksonIntegratorBase() [3/3]	106
8.8.2.4	~GaussJacksonIntegratorBase()	107
8.8.3	Member Function Documentation	107
8.8.3.1	advance_edit_integration_constants() [1/3]	107
8.8.3.2	advance_edit_integration_constants() [2/3]	107
8.8.3.3	advance_edit_integration_constants() [3/3]	107
8.8.3.4	advance_predictor_integration_constants() [1/3]	108

8.8.3.5	advance_predictor_integration_constants() [2/3]	108
8.8.3.6	advance_predictor_integration_constants() [3/3]	108
8.8.3.7	allocate_state_contents() [1/3]	109
8.8.3.8	allocate_state_contents() [2/3]	109
8.8.3.9	allocate_state_contents() [3/3]	109
8.8.3.10	base_integrate()	109
8.8.3.11	base_reset()	110
8.8.3.12	correct() [1/3]	111
8.8.3.13	correct() [2/3]	111
8.8.3.14	correct() [3/3]	111
8.8.3.15	create_primer() [1/3]	112
8.8.3.16	create_primer() [2/3]	112
8.8.3.17	create_primer() [3/3]	112
8.8.3.18	deallocate_state_contents() [1/3]	113
8.8.3.19	deallocate_state_contents() [2/3]	113
8.8.3.20	deallocate_state_contents() [3/3]	113
8.8.3.21	downsample_hist()	113
8.8.3.22	edit_point()	114
8.8.3.23	initialize_edit_integration_constants() [1/3]	114
8.8.3.24	initialize_edit_integration_constants() [2/3]	115
8.8.3.25	initialize_edit_integration_constants() [3/3]	115
8.8.3.26	initialize_predictor_integration_constants() [1/3]	115
8.8.3.27	initialize_predictor_integration_constants() [2/3]	115
8.8.3.28	initialize_predictor_integration_constants() [3/3]	116
8.8.3.29	integrate_gj()	116
8.8.3.30	integrate_primer() [1/3]	117
8.8.3.31	integrate_primer() [2/3]	117
8.8.3.32	integrate_primer() [3/3]	117
8.8.3.33	mid_correct() [1/3]	118
8.8.3.34	mid_correct() [2/3]	118

8.8.3.35	<code>mid_correct()</code> [3/3]	118
8.8.3.36	<code>operator=()</code>	119
8.8.3.37	<code>predict()</code> [1/3]	119
8.8.3.38	<code>predict()</code> [2/3]	119
8.8.3.39	<code>predict()</code> [3/3]	119
8.8.3.40	<code>replicate_primer()</code> [1/3]	120
8.8.3.41	<code>replicate_primer()</code> [2/3]	120
8.8.3.42	<code>replicate_primer()</code> [3/3]	120
8.8.3.43	<code>replicate_state()</code> [1/3]	121
8.8.3.44	<code>replicate_state()</code> [2/3]	121
8.8.3.45	<code>replicate_state()</code> [3/3]	121
8.8.3.46	<code>rotate_acc_hist()</code>	122
8.8.3.47	<code>save_comparison_data()</code> [1/3]	122
8.8.3.48	<code>save_comparison_data()</code> [2/3]	122
8.8.3.49	<code>save_comparison_data()</code> [3/3]	122
8.8.3.50	<code>save_epoch_data()</code> [1/3]	123
8.8.3.51	<code>save_epoch_data()</code> [2/3]	123
8.8.3.52	<code>save_epoch_data()</code> [3/3]	123
8.8.3.53	<code>start_cycle()</code>	124
8.8.3.54	<code>swap()</code>	124
8.8.3.55	<code>swap_state()</code> [1/3]	125
8.8.3.56	<code>swap_state()</code> [2/3]	125
8.8.3.57	<code>swap_state()</code> [3/3]	126
8.8.3.58	<code>test_for_convergence()</code> [1/3]	127
8.8.3.59	<code>test_for_convergence()</code> [2/3]	127
8.8.3.60	<code>test_for_convergence()</code> [3/3]	127
8.8.4	Field Documentation	128
8.8.4.1	<code>absolute_tolerance</code>	128
8.8.4.2	<code>acc_hist</code>	128
8.8.4.3	<code>coeff</code>	129

8.8.4.4	corrector_sum	129
8.8.4.5	delinv	129
8.8.4.6	fsm_state	130
8.8.4.7	history_length	130
8.8.4.8	init_state	130
8.8.4.9	initial_order	131
8.8.4.10	max_history_size	131
8.8.4.11	order	131
8.8.4.12	pos_hist	132
8.8.4.13	position_corrector	132
8.8.4.14	primer	132
8.8.4.15	relative_tolerance	133
8.8.4.16	size	133
8.8.4.17	state_machine	133
8.8.4.18	velocity_corrector	134
8.9	jeod::GaussJacksonIntegratorConstructor Class Reference	134
8.9.1	Detailed Description	135
8.9.2	Constructor & Destructor Documentation	136
8.9.2.1	GaussJacksonIntegratorConstructor() [1/2]	136
8.9.2.2	GaussJacksonIntegratorConstructor() [2/2]	136
8.9.2.3	~GaussJacksonIntegratorConstructor()	136
8.9.3	Member Function Documentation	136
8.9.3.1	configure() [1/2]	136
8.9.3.2	configure() [2/2]	137
8.9.3.3	create_constructor()	137
8.9.3.4	create_copy()	137
8.9.3.5	create_first_order_ode_integrator()	137
8.9.3.6	create_generalized_deriv_second_order_ode_integrator()	138
8.9.3.7	create_integration_controls()	138
8.9.3.8	create_second_order_ode_integrator()	139

8.9.3.9	get_buffer_size()	139
8.9.3.10	get_class_name()	140
8.9.3.11	get_transition_table_size()	140
8.9.3.12	implements()	140
8.9.3.13	operator=()	140
8.9.3.14	provides()	140
8.9.3.15	swap()	140
8.9.4	Friends And Related Function Documentation	141
8.9.4.1	init_attrjeod__GaussJacksonIntegratorConstructor	141
8.9.4.2	InputProcessor	141
8.9.5	Field Documentation	141
8.9.5.1	config	141
8.9.5.2	priming_constructor	142
8.10	jeod::GaussJacksonOneState Class Reference	142
8.10.1	Detailed Description	142
8.10.2	Constructor & Destructor Documentation	143
8.10.2.1	GaussJacksonOneState() [1/2]	143
8.10.2.2	GaussJacksonOneState() [2/2]	143
8.10.3	Friends And Related Function Documentation	143
8.10.3.1	init_attrjeod__GaussJacksonOneState	143
8.10.3.2	InputProcessor	143
8.10.4	Field Documentation	143
8.10.4.1	first	144
8.11	jeod::GaussJacksonRationalCoefficients Class Reference	144
8.11.1	Detailed Description	145
8.11.2	Constructor & Destructor Documentation	145
8.11.2.1	GaussJacksonRationalCoefficients()	145
8.11.3	Member Function Documentation	145
8.11.3.1	configure_adams_corrector()	145
8.11.3.2	construct_predictor()	146

8.11.3.3	construct_stormer_cowell_corrector()	146
8.11.3.4	convert_to_ordinate_form()	146
8.11.3.5	discard_extra_terms()	147
8.11.3.6	displace_back()	147
8.11.4	Friends And Related Function Documentation	147
8.11.4.1	init_attrjeod__GaussJacksonRationalCoefficients	148
8.11.4.2	InputProcessor	148
8.11.5	Field Documentation	148
8.11.5.1	coefficients	148
8.12	jeod::GaussJacksonSimpleSecondOrderODEIntegrator Class Reference	148
8.12.1	Detailed Description	149
8.12.2	Constructor & Destructor Documentation	149
8.12.2.1	GaussJacksonSimpleSecondOrderODEIntegrator() [1/3]	150
8.12.2.2	~GaussJacksonSimpleSecondOrderODEIntegrator()	150
8.12.2.3	GaussJacksonSimpleSecondOrderODEIntegrator() [2/3]	150
8.12.2.4	GaussJacksonSimpleSecondOrderODEIntegrator() [3/3]	150
8.12.3	Member Function Documentation	151
8.12.3.1	create_copy()	151
8.12.3.2	integrate()	151
8.12.3.3	operator=()	151
8.12.3.4	reset_integrator()	153
8.12.3.5	swap() [1/2]	153
8.12.3.6	swap() [2/2]	153
8.12.4	Friends And Related Function Documentation	154
8.12.4.1	init_attrjeod__GaussJacksonSimpleSecondOrderODEIntegrator	154
8.12.4.2	InputProcessor	154
8.13	jeod::GaussJacksonStateMachine Class Reference	154
8.13.1	Detailed Description	157
8.13.2	Member Enumeration Documentation	157
8.13.2.1	FsmState	157

8.13.3	Constructor & Destructor Documentation	157
8.13.3.1	GaussJacksonStateMachine()	157
8.13.4	Member Function Documentation	158
8.13.4.1	configure()	158
8.13.4.2	exit_bootstrap_edit()	158
8.13.4.3	exit_bootstrap_step()	158
8.13.4.4	exit_priming()	159
8.13.4.5	get_at_downsample()	159
8.13.4.6	get_at_end_of_tour()	159
8.13.4.7	get_at_order_change()	159
8.13.4.8	get_at_reinitialize()	160
8.13.4.9	get_current_order()	160
8.13.4.10	get_cycle_scale()	160
8.13.4.11	get_cycle_start_time()	160
8.13.4.12	get_fsm_state()	161
8.13.4.13	get_history_length()	161
8.13.4.14	get_max_history_size()	161
8.13.4.15	perform_step()	161
8.13.4.16	reset()	162
8.13.4.17	set_bootstrap_edit_redo_needed()	162
8.13.4.18	state_name()	162
8.13.4.19	transition_state()	162
8.13.5	Friends And Related Function Documentation	163
8.13.5.1	init_attrjeod__GaussJacksonStateMachine	163
8.13.5.2	InputProcessor	163
8.13.6	Field Documentation	163
8.13.6.1	at_downsample	163
8.13.6.2	at_end_of_tour	163
8.13.6.3	at_order_change	164
8.13.6.4	at_reinitialize	164

8.13.6.5	bootstrap_edit_redo_needed	164
8.13.6.6	correction_iterations	164
8.13.6.7	current_order	165
8.13.6.8	cycle_scale	165
8.13.6.9	cycle_start_time	165
8.13.6.10	final_order	165
8.13.6.11	fsm_state	166
8.13.6.12	history_length	166
8.13.6.13	history_size	166
8.13.6.14	initial_order	166
8.13.6.15	max_correction_iterations	167
8.13.6.16	max_history_size	167
8.13.6.17	ndoubling_steps	167
8.13.6.18	scale_factor	167
8.13.6.19	step_increment	168
8.13.6.20	steps_since_reset	168
8.13.6.21	tour_count	168
8.14	jeod::GaussJacksonTwoState Class Reference	168
8.14.1	Detailed Description	169
8.14.2	Constructor & Destructor Documentation	169
8.14.2.1	GaussJacksonTwoState() [1/2]	169
8.14.2.2	GaussJacksonTwoState() [2/2]	169
8.14.3	Friends And Related Function Documentation	170
8.14.3.1	init_attrjeod__GaussJacksonTwoState	170
8.14.3.2	InputProcessor	170
8.14.4	Field Documentation	170
8.14.4.1	first	170
8.14.4.2	second	171
8.15	jeod::GeneralizedSecondOrderODETechnique Class Reference	171
8.15.1	Detailed Description	172

8.15.2	Member Enumeration Documentation	172
8.15.2.1	TechniqueType	172
8.15.3	Constructor & Destructor Documentation	172
8.15.3.1	GeneralizedSecondOrderODETechnique() [1/2]	172
8.15.3.2	GeneralizedSecondOrderODETechnique() [2/2]	172
8.15.3.3	~GeneralizedSecondOrderODETechnique()	173
8.15.4	Member Function Documentation	173
8.15.4.1	is_provided_by()	173
8.15.4.2	operator=()	173
8.15.4.3	validate_technique()	174
8.16	jeod::IntegrationMessages Class Reference	174
8.16.1	Detailed Description	175
8.16.2	Constructor & Destructor Documentation	175
8.16.2.1	IntegrationMessages() [1/2]	175
8.16.2.2	IntegrationMessages() [2/2]	175
8.16.3	Member Function Documentation	176
8.16.3.1	operator=()	176
8.16.4	Friends And Related Function Documentation	176
8.16.4.1	init_attrjeod__IntegrationMessages	176
8.16.4.2	InputProcessor	176
8.16.5	Field Documentation	176
8.16.5.1	information	176
8.16.5.2	internal_error	177
8.16.5.3	invalid_item	177
8.16.5.4	invalid_request	177
8.16.5.5	unsupported_option	178
8.17	jeod::JeodIntegrationGroup Class Reference	178
8.17.1	Detailed Description	180
8.17.2	Constructor & Destructor Documentation	180
8.17.2.1	JeodIntegrationGroup() [1/3]	180

8.17.2.2	JeodIntegrationGroup() [2/3]	180
8.17.2.3	~JeodIntegrationGroup()	181
8.17.2.4	JeodIntegrationGroup() [3/3]	181
8.17.3	Member Function Documentation	181
8.17.3.1	add_integrable_object()	181
8.17.3.2	initialize_group()	181
8.17.3.3	integrate_bodies()	182
8.17.3.4	integrate_container()	182
8.17.3.5	merge_integrator_result()	183
8.17.3.6	need_first_step_derivatives()	183
8.17.3.7	operator=()	183
8.17.3.8	register_classes()	183
8.17.3.9	remove_integrable_object()	184
8.17.3.10	reset_body_integrators()	184
8.17.3.11	reset_container()	184
8.17.3.12	respond_to_time_change()	185
8.17.3.13	update_from_owner()	185
8.17.4	Friends And Related Function Documentation	185
8.17.4.1	init_attrjeod__JeodIntegrationGroup	185
8.17.4.2	InputProcessor	185
8.17.5	Field Documentation	185
8.17.5.1	group_owner	185
8.17.5.2	integ_merger	186
8.17.5.3	integrable_objects	186
8.17.5.4	jeod_integ_interface	186
8.17.5.5	jeod_time_manager	186
8.18	jeod::JeodIntegrationGroupOwner Class Reference	187
8.18.1	Detailed Description	187
8.18.2	Constructor & Destructor Documentation	187
8.18.2.1	~JeodIntegrationGroupOwner()	187

8.18.3	Member Function Documentation	187
8.18.3.1	update_integration_group()	187
8.19	jeod::JeodIntegrationTime Class Reference	188
8.19.1	Detailed Description	189
8.19.2	Constructor & Destructor Documentation	189
8.19.2.1	JeodIntegrationTime() [1/2]	189
8.19.2.2	~JeodIntegrationTime()	189
8.19.2.3	JeodIntegrationTime() [2/2]	189
8.19.3	Member Function Documentation	189
8.19.3.1	add_time_change_subscriber()	189
8.19.3.2	get_timestamp_time()	190
8.19.3.3	notify_time_change_subscribers()	190
8.19.3.4	operator=()	190
8.19.3.5	remove_time_change_subscriber()	190
8.19.4	Friends And Related Function Documentation	191
8.19.4.1	init_attrjeod__JeodIntegrationTime	191
8.19.4.2	InputProcessor	191
8.19.5	Field Documentation	191
8.19.5.1	time_change_subscribers	191
8.20	jeod::LsodeControlDataInterface Class Reference	192
8.20.1	Detailed Description	193
8.20.2	Member Enumeration Documentation	193
8.20.2.1	CorrectorMethod	193
8.20.2.2	ErrorControlIndicator	194
8.20.2.3	IntegrationMethod	194
8.20.3	Constructor & Destructor Documentation	194
8.20.3.1	~LsodeControlDataInterface()	194
8.20.3.2	LsodeControlDataInterface()	194
8.20.4	Member Function Documentation	195
8.20.4.1	is_corrector_method_functional_iteration()	195

8.20.4.2	operator=()	195
8.20.5	Friends And Related Function Documentation	195
8.20.5.1	init_attrjeod__LsodeControlDataInterface	195
8.20.5.2	InputProcessor	195
8.20.6	Field Documentation	195
8.20.6.1	abs_tolerance_error_control	196
8.20.6.2	abs_tolerance_error_control_vec	196
8.20.6.3	corrector_method	196
8.20.6.4	error_control_indicator	196
8.20.6.5	error_control_vector_copied_over	197
8.20.6.6	initial_step_size	197
8.20.6.7	integration_method	197
8.20.6.8	max_correction_iters	197
8.20.6.9	max_num_conv_failure	198
8.20.6.10	max_num_small_step_warnings	198
8.20.6.11	max_num_steps	198
8.20.6.12	max_num_steps_jacobian	198
8.20.6.13	max_order	199
8.20.6.14	max_step_size	199
8.20.6.15	min_step_size	199
8.20.6.16	num_odes	200
8.20.6.17	num_odes_at_alloc	200
8.20.6.18	rel_tolerance_error_control	200
8.20.6.19	rel_tolerance_error_control_vec	201
8.21	jeod::LsodeDataArrays Class Reference	201
8.21.1	Detailed Description	202
8.21.2	Constructor & Destructor Documentation	202
8.21.2.1	LsodeDataArrays() [1/2]	202
8.21.2.2	~LsodeDataArrays()	202
8.21.2.3	LsodeDataArrays() [2/2]	202

8.21.3	Member Function Documentation	202
8.21.3.1	operator=()	202
8.21.4	Friends And Related Function Documentation	203
8.21.4.1	init_attrjeod__LsodeDataArrays	203
8.21.4.2	InputProcessor	203
8.21.5	Field Documentation	203
8.21.5.1	accum_correction	203
8.21.5.2	allocated	203
8.21.5.3	error_weight	204
8.21.5.4	history	204
8.21.5.5	lin_alg	204
8.21.5.6	lin_alg_1	205
8.21.5.7	lin_alg_2	205
8.21.5.8	lin_alg_index1	205
8.21.5.9	num_odes	205
8.21.5.10	pivots	206
8.21.5.11	save	206
8.22	jeod::LsodeDataJacobianPrep Class Reference	206
8.22.1	Detailed Description	207
8.22.2	Constructor & Destructor Documentation	207
8.22.2.1	LsodeDataJacobianPrep() [1/2]	207
8.22.2.2	~LsodeDataJacobianPrep()	207
8.22.2.3	LsodeDataJacobianPrep() [2/2]	207
8.22.3	Member Function Documentation	207
8.22.3.1	operator=()	208
8.22.4	Friends And Related Function Documentation	208
8.22.4.1	init_attrjeod__LsodeDataJacobianPrep	208
8.22.4.2	InputProcessor	208
8.22.5	Field Documentation	208
8.22.5.1	fac	208

8.22.5.2	hl0	208
8.22.5.3	index	209
8.22.5.4	index_max	209
8.22.5.5	r0	209
8.22.5.6	yj	209
8.23	jeod::LsodeDataStode Class Reference	209
8.23.1	Detailed Description	210
8.23.2	Constructor & Destructor Documentation	210
8.23.2.1	LsodeDataStode() [1/2]	210
8.23.2.2	~LsodeDataStode()	210
8.23.2.3	LsodeDataStode() [2/2]	211
8.23.3	Member Function Documentation	211
8.23.3.1	operator=()	211
8.23.4	Friends And Related Function Documentation	211
8.23.4.1	init_attrjeod__LsodeDataStode	211
8.23.4.2	InputProcessor	211
8.23.5	Field Documentation	211
8.23.5.1	dsm	211
8.23.5.2	iredo	212
8.23.5.3	iret	212
8.23.5.4	ncf	212
8.23.5.5	new_method_order	212
8.23.5.6	step_ratio	213
8.23.5.7	step_ratio_order_inc	213
8.23.5.8	told	213
8.24	jeod::LsodeFirstOrderODEIntegrator Class Reference	213
8.24.1	Detailed Description	218
8.24.2	Member Enumeration Documentation	218
8.24.2.1	CalculationTask	218
8.24.2.2	EntryPoint	219

8.24.3	Constructor & Destructor Documentation	219
8.24.3.1	LsodeFirstOrderODEIntegrator()	219
8.24.4	Member Function Documentation	219
8.24.4.1	get_re_entry_point()	219
8.24.4.2	operator=()	219
8.24.5	Friends And Related Function Documentation	220
8.24.5.1	init_attrjeod__LsodeFirstOrderODEIntegrator	220
8.24.5.2	InputProcessor	220
8.24.6	Field Documentation	220
8.24.6.1	arrays	220
8.24.6.2	calculation_task	220
8.24.6.3	control_data	221
8.24.6.4	convergence_factor	221
8.24.6.5	convergence_jacobian_flag	221
8.24.6.6	convergence_rate	222
8.24.6.7	cycle_target_time	222
8.24.6.8	data_prepj	222
8.24.6.9	data_stode	222
8.24.6.10	epsilon	223
8.24.6.11	first_pass	223
8.24.6.12	initialized	223
8.24.6.13	internal_state	223
8.24.6.14	iter_delta	224
8.24.6.15	iteration_count	224
8.24.6.16	iteration_matrix_singular	224
8.24.6.17	jacobian_current	224
8.24.6.18	max_history_size	225
8.24.6.19	max_order_internal	225
8.24.6.20	max_rel_change_without_jacobian	225
8.24.6.21	max_step_increase_ratio	225

8.24.6.22 max_step_size_inv	226
8.24.6.23 method_coeff_first	226
8.24.6.24 method_coeffs_complete	226
8.24.6.25 method_coeffs_current	226
8.24.6.26 method_order_current	227
8.24.6.27 modified_iteration_matrix_singular	227
8.24.6.28 num_equations	227
8.24.6.29 num_jacobian_evals	228
8.24.6.30 num_nordsiek_cols	228
8.24.6.31 num_predictor_elements	228
8.24.6.32 num_small_step_warnings	228
8.24.6.33 num_steps_taken	229
8.24.6.34 order_select_para	229
8.24.6.35 prev_good_step_size	229
8.24.6.36 prev_integration_method	230
8.24.6.37 prev_iter_delta	230
8.24.6.38 prev_method_order	230
8.24.6.39 prev_step_size	231
8.24.6.40 prior_num_steps	231
8.24.6.41 re_entry_point	231
8.24.6.42 rel_change_since_jacobian	231
8.24.6.43 sqrt_epsilon	232
8.24.6.44 stage_target_time	232
8.24.6.45 step_at_last_jacobian_update	232
8.24.6.46 step_error	233
8.24.6.47 step_size	233
8.24.6.48 test_coeffs_complete	233
8.24.6.49 update_jacobian	234
8.24.6.50 y	234
8.24.6.51 y_dot	234

8.25	jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator Class Reference	235
8.25.1	Detailed Description	236
8.25.2	Constructor & Destructor Documentation	236
8.25.2.1	LsodeGeneralizedDerivSecondOrderODEIntegrator()	236
8.25.3	Member Function Documentation	236
8.25.3.1	operator=()	236
8.25.4	Friends And Related Function Documentation	236
8.25.4.1	init_attrjeod__LsodeGeneralizedDerivSecondOrderODEIntegrator	236
8.25.4.2	InputProcessor	236
8.25.5	Field Documentation	237
8.25.5.1	posdot	237
8.26	jeod::LsodeIntegrationControls Class Reference	237
8.26.1	Detailed Description	238
8.26.2	Constructor & Destructor Documentation	238
8.26.2.1	LsodeIntegrationControls() [1/2]	238
8.26.2.2	~LsodeIntegrationControls()	238
8.26.2.3	LsodeIntegrationControls() [2/2]	238
8.26.3	Member Function Documentation	238
8.26.3.1	operator=()	238
8.26.4	Friends And Related Function Documentation	238
8.26.4.1	init_attrjeod__LsodeIntegrationControls	239
8.26.4.2	InputProcessor	239
8.27	jeod::LsodeIntegratorConstructor Class Reference	239
8.27.1	Detailed Description	240
8.27.2	Constructor & Destructor Documentation	240
8.27.2.1	LsodeIntegratorConstructor()	240
8.27.3	Member Function Documentation	240
8.27.3.1	get_class_name()	241
8.27.3.2	get_transition_table_size()	241
8.27.3.3	implements()	241

8.27.3.4	operator=()	241
8.27.3.5	provides()	241
8.27.4	Friends And Related Function Documentation	242
8.27.4.1	init_attrjeod__LsodeIntegratorConstructor	242
8.27.4.2	InputProcessor	242
8.27.5	Field Documentation	242
8.27.5.1	data_interface	242
8.28	jeod::LsodeSecondOrderODEIntegrator Class Reference	242
8.28.1	Detailed Description	243
8.28.2	Constructor & Destructor Documentation	243
8.28.2.1	LsodeSecondOrderODEIntegrator() [1/2]	244
8.28.2.2	LsodeSecondOrderODEIntegrator() [2/2]	244
8.28.3	Member Function Documentation	244
8.28.3.1	get_re_entry_point()	244
8.28.3.2	operator=()	244
8.28.3.3	reset_integrator()	244
8.28.4	Friends And Related Function Documentation	244
8.28.4.1	init_attrjeod__LsodeSecondOrderODEIntegrator	245
8.28.4.2	InputProcessor	245
8.28.5	Field Documentation	245
8.28.5.1	arrays_allocated	245
8.28.5.2	first_derivative_size	245
8.28.5.3	first_order_integrator	245
8.28.5.4	y	246
8.28.5.5	y_dot	246
8.28.5.6	zeroth_derivative_size	246
8.29	jeod::LsodeSimpleSecondOrderODEIntegrator Class Reference	247
8.29.1	Detailed Description	247
8.29.2	Constructor & Destructor Documentation	247
8.29.2.1	LsodeSimpleSecondOrderODEIntegrator() [1/2]	248

8.29.2.2	~LsodeSimpleSecondOrderODEIntegrator()	248
8.29.2.3	LsodeSimpleSecondOrderODEIntegrator() [2/2]	248
8.29.3	Member Function Documentation	248
8.29.3.1	operator=()	248
8.29.4	Friends And Related Function Documentation	248
8.29.4.1	init_attrjeod__LsodeSimpleSecondOrderODEIntegrator	248
8.29.4.2	InputProcessor	248
8.30	jeod::Restartable2DSecondOrderIntegrator Class Reference	249
8.30.1	Detailed Description	250
8.30.2	Constructor & Destructor Documentation	250
8.30.2.1	Restartable2DSecondOrderIntegrator() [1/2]	250
8.30.2.2	~Restartable2DSecondOrderIntegrator()	250
8.30.2.3	Restartable2DSecondOrderIntegrator() [2/2]	250
8.30.3	Member Function Documentation	250
8.30.3.1	create_integrator()	250
8.30.3.2	destroy_integrator()	251
8.30.3.3	integrate()	251
8.30.3.4	operator=()	252
8.30.3.5	reset_integrator()	252
8.30.3.6	simple_restore()	252
8.30.4	Friends And Related Function Documentation	252
8.30.4.1	init_attrjeod__Restartable2DSecondOrderIntegrator	252
8.30.4.2	InputProcessor	252
8.30.5	Field Documentation	252
8.30.5.1	integrator	253
8.30.5.2	integrator_manager	253
8.31	jeod::RestartableFirstOrderODEIntegrator< size > Class Template Reference	253
8.31.1	Detailed Description	254
8.31.2	Constructor & Destructor Documentation	254
8.31.2.1	RestartableFirstOrderODEIntegrator() [1/3]	254

8.31.2.2	~RestartableFirstOrderODEIntegrator()	254
8.31.2.3	RestartableFirstOrderODEIntegrator() [2/3]	254
8.31.2.4	RestartableFirstOrderODEIntegrator() [3/3]	255
8.31.3	Member Function Documentation	255
8.31.3.1	create_integrator_internal()	255
8.31.3.2	operator=()	255
8.32	jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions > Class Template Reference	256
8.32.1	Detailed Description	256
8.32.2	Constructor & Destructor Documentation	257
8.32.2.1	RestartableGeneralizedDerivSecondOrderODEIntegrator() [1/3]	257
8.32.2.2	~RestartableGeneralizedDerivSecondOrderODEIntegrator()	257
8.32.2.3	RestartableGeneralizedDerivSecondOrderODEIntegrator() [2/3]	257
8.32.2.4	RestartableGeneralizedDerivSecondOrderODEIntegrator() [3/3]	257
8.32.3	Member Function Documentation	258
8.32.3.1	create_integrator_internal()	258
8.32.3.2	operator=()	258
8.32.3.3	simple_restore_internal()	258
8.33	jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, StepFunctions > Class Template Reference	260
8.33.1	Detailed Description	261
8.33.2	Constructor & Destructor Documentation	261
8.33.2.1	RestartableGeneralizedStepSecondOrderODEIntegrator() [1/3]	261
8.33.2.2	~RestartableGeneralizedStepSecondOrderODEIntegrator()	261
8.33.2.3	RestartableGeneralizedStepSecondOrderODEIntegrator() [2/3]	261
8.33.2.4	RestartableGeneralizedStepSecondOrderODEIntegrator() [3/3]	262
8.33.3	Member Function Documentation	262
8.33.3.1	create_integrator_internal()	262
8.33.3.2	operator=()	262
8.33.3.3	simple_restore_internal()	263
8.34	jeod::RestartableScalarFirstOrderODEIntegrator Class Reference	263

8.34.1 Detailed Description	264
8.34.2 Constructor & Destructor Documentation	264
8.34.2.1 RestartableScalarFirstOrderODEIntegrator() [1/2]	264
8.34.2.2 ~RestartableScalarFirstOrderODEIntegrator()	264
8.34.2.3 RestartableScalarFirstOrderODEIntegrator() [2/2]	265
8.34.3 Member Function Documentation	265
8.34.3.1 create_integrator()	265
8.34.3.2 destroy_integrator()	265
8.34.3.3 integrate()	265
8.34.3.4 operator=()	266
8.34.3.5 reset_integrator()	266
8.34.3.6 simple_restore()	266
8.34.4 Friends And Related Function Documentation	266
8.34.4.1 init_attrjeod__RestartableScalarFirstOrderODEIntegrator	267
8.34.4.2 InputProcessor	267
8.34.5 Field Documentation	267
8.34.5.1 integrator	267
8.34.5.2 integrator_manager	267
8.35 jeod::RestartableSecondOrderODEIntegrator Class Reference	268
8.35.1 Detailed Description	268
8.35.2 Constructor & Destructor Documentation	268
8.35.2.1 ~RestartableSecondOrderODEIntegrator()	268
8.35.2.2 RestartableSecondOrderODEIntegrator() [1/3]	268
8.35.2.3 RestartableSecondOrderODEIntegrator() [2/3]	269
8.35.2.4 RestartableSecondOrderODEIntegrator() [3/3]	269
8.35.3 Member Function Documentation	269
8.35.3.1 operator=()	269
8.36 jeod::RestartableSimpleSecondOrderODEIntegrator< size > Class Template Reference	269
8.36.1 Detailed Description	270
8.36.2 Constructor & Destructor Documentation	270

8.36.2.1	RestartableSimpleSecondOrderODEIntegrator() [1/3]	270
8.36.2.2	~RestartableSimpleSecondOrderODEIntegrator()	271
8.36.2.3	RestartableSimpleSecondOrderODEIntegrator() [2/3]	271
8.36.2.4	RestartableSimpleSecondOrderODEIntegrator() [3/3]	271
8.36.3	Member Function Documentation	271
8.36.3.1	create_integrator_internal()	271
8.36.3.2	operator=()	272
8.37	jeod::RestartableSO3SecondOrderODEIntegrator Class Reference	272
8.37.1	Detailed Description	273
8.37.2	Constructor & Destructor Documentation	273
8.37.2.1	RestartableSO3SecondOrderODEIntegrator() [1/2]	273
8.37.2.2	~RestartableSO3SecondOrderODEIntegrator()	274
8.37.2.3	RestartableSO3SecondOrderODEIntegrator() [2/2]	274
8.37.3	Member Function Documentation	274
8.37.3.1	create_integrator()	274
8.37.3.2	destroy_integrator()	274
8.37.3.3	integrate()	275
8.37.3.4	operator=()	275
8.37.3.5	reset_integrator()	275
8.37.3.6	simple_restore()	276
8.37.4	Friends And Related Function Documentation	276
8.37.4.1	init_attrjeod__RestartableSO3SecondOrderODEIntegrator	276
8.37.4.2	InputProcessor	276
8.37.5	Field Documentation	276
8.37.5.1	generalized_deriv_integrator_manager	276
8.37.5.2	generalized_step_integrator_manager	277
8.37.5.3	integrator	277
8.37.5.4	technique	277
8.38	jeod::RestartableStateIntegrator< IntegratorType > Class Template Reference	278
8.38.1	Detailed Description	279

8.38.2	Constructor & Destructor Documentation	280
8.38.2.1	~RestartableStateIntegrator()	280
8.38.2.2	RestartableStateIntegrator() [1/3]	281
8.38.2.3	RestartableStateIntegrator() [2/3]	281
8.38.2.4	RestartableStateIntegrator() [3/3]	281
8.38.3	Member Function Documentation	281
8.38.3.1	clear_integrator_reference()	281
8.38.3.2	create_integrator()	282
8.38.3.3	create_integrator_internal()	283
8.38.3.4	destroy_integrator()	283
8.38.3.5	operator=()	284
8.38.3.6	set_integrator_reference()	284
8.38.3.7	simple_restore()	284
8.38.3.8	simple_restore_internal()	284
8.38.4	Field Documentation	285
8.38.4.1	integrator_handle	285
8.39	jeod::RestartableT3SecondOrderODEIntegrator Class Reference	285
8.39.1	Detailed Description	286
8.39.2	Constructor & Destructor Documentation	286
8.39.2.1	RestartableT3SecondOrderODEIntegrator() [1/2]	287
8.39.2.2	~RestartableT3SecondOrderODEIntegrator()	287
8.39.2.3	RestartableT3SecondOrderODEIntegrator() [2/2]	287
8.39.3	Member Function Documentation	287
8.39.3.1	create_integrator()	287
8.39.3.2	destroy_integrator()	287
8.39.3.3	integrate()	288
8.39.3.4	operator=()	288
8.39.3.5	reset_integrator()	289
8.39.3.6	simple_restore()	289
8.39.4	Friends And Related Function Documentation	289

8.39.4.1	init_attrjeod__RestartableT3SecondOrderODEIntegrator	289
8.39.4.2	InputProcessor	289
8.39.5	Field Documentation	289
8.39.5.1	integrator	290
8.39.5.2	integrator_manager	290
8.40	jeod::TimeChangeSubscriber Class Reference	290
8.40.1	Detailed Description	291
8.40.2	Constructor & Destructor Documentation	291
8.40.2.1	TimeChangeSubscriber() [1/2]	291
8.40.2.2	~TimeChangeSubscriber()	291
8.40.2.3	TimeChangeSubscriber() [2/2]	291
8.40.3	Member Function Documentation	291
8.40.3.1	operator=()	291
8.40.3.2	respond_to_time_change()	292
8.40.4	Friends And Related Function Documentation	292
8.40.4.1	init_attrjeod__TimeChangeSubscriber	292
8.40.4.2	InputProcessor	292
8.41	er7_utils::TwoDArray< T > Class Template Reference	292
8.41.1	Detailed Description	294
8.41.2	Constructor & Destructor Documentation	294
8.41.2.1	TwoDArray() [1/2]	294
8.41.2.2	TwoDArray() [2/2]	294
8.41.2.3	~TwoDArray()	295
8.41.3	Member Function Documentation	295
8.41.3.1	allocate()	295
8.41.3.2	allocate_internal()	296
8.41.3.3	at() [1/4]	296
8.41.3.4	at() [2/4]	296
8.41.3.5	at() [3/4]	297
8.41.3.6	at() [4/4]	297

8.41.3.7	<code>deallocate_internal()</code>	298
8.41.3.8	<code>downsample()</code>	298
8.41.3.9	<code>operator const T *const *()</code>	299
8.41.3.10	<code>operator T*const *()</code>	299
8.41.3.11	<code>operator>()</code> [1/2]	299
8.41.3.12	<code>operator>()</code> [2/2]	300
8.41.3.13	<code>operator=()</code>	300
8.41.3.14	<code>operator[]()</code> [1/2]	300
8.41.3.15	<code>operator[]()</code> [2/2]	301
8.41.3.16	<code>rotate_down()</code>	301
8.41.3.17	<code>rotate_up()</code>	302
8.41.3.18	<code>swap()</code>	302
8.41.4	Friends And Related Function Documentation	302
8.41.4.1	<code>swap</code>	303
8.41.5	Field Documentation	303
8.41.5.1	<code>data_array</code>	303
8.41.5.2	<code>m</code>	303
8.41.5.3	<code>n</code>	304
8.41.5.4	<code>row_array</code>	304
9	File Documentation	305
9.1	<code>gauss_jackson_coefficients_pair.cc</code> File Reference	305
9.1.1	Detailed Description	305
9.2	<code>gauss_jackson_coefficients_pair.hh</code> File Reference	305
9.2.1	Detailed Description	306
9.3	<code>gauss_jackson_coeffs.cc</code> File Reference	306
9.3.1	Detailed Description	306
9.4	<code>gauss_jackson_coeffs.hh</code> File Reference	307
9.4.1	Detailed Description	307
9.5	<code>gauss_jackson_config.cc</code> File Reference	307
9.5.1	Detailed Description	308

9.6	gauss_jackson_config.hh File Reference	308
9.6.1	Detailed Description	308
9.7	gauss_jackson_first_order_ode_integrator.hh File Reference	308
9.7.1	Detailed Description	309
9.8	gauss_jackson_generalized_second_order_ode_integrator.cc File Reference	309
9.8.1	Detailed Description	309
9.9	gauss_jackson_generalized_second_order_ode_integrator.hh File Reference	309
9.9.1	Detailed Description	310
9.10	gauss_jackson_integration_controls.cc File Reference	310
9.10.1	Detailed Description	310
9.11	gauss_jackson_integration_controls.hh File Reference	310
9.11.1	Detailed Description	311
9.12	gauss_jackson_integrator_base.hh File Reference	311
9.12.1	Detailed Description	311
9.13	gauss_jackson_integrator_base_first.hh File Reference	312
9.13.1	Detailed Description	312
9.14	gauss_jackson_integrator_base_second.hh File Reference	312
9.14.1	Detailed Description	313
9.15	gauss_jackson_integrator_constructor.cc File Reference	313
9.15.1	Detailed Description	313
9.16	gauss_jackson_integrator_constructor.hh File Reference	314
9.16.1	Detailed Description	314
9.17	gauss_jackson_one_state.hh File Reference	314
9.17.1	Detailed Description	315
9.18	gauss_jackson_rational_coeffs.cc File Reference	315
9.18.1	Detailed Description	315
9.19	gauss_jackson_rational_coeffs.hh File Reference	315
9.19.1	Detailed Description	315
9.20	gauss_jackson_simple_second_order_ode_integrator.hh File Reference	316
9.20.1	Detailed Description	316

9.21	gauss_jackson_state_machine.cc File Reference	316
9.21.1	Detailed Description	316
9.22	gauss_jackson_state_machine.hh File Reference	317
9.22.1	Detailed Description	317
9.23	gauss_jackson_two_state.hh File Reference	317
9.23.1	Detailed Description	317
9.24	generalized_second_order_ode_technique.cc File Reference	318
9.24.1	Detailed Description	318
9.25	generalized_second_order_ode_technique.hh File Reference	318
9.25.1	Detailed Description	318
9.26	integration_messages.cc File Reference	319
9.26.1	Detailed Description	319
9.26.2	Macro Definition Documentation	319
9.26.2.1	MAKE_INTEGRATION_MESSAGE_CODE	319
9.27	integration_messages.hh File Reference	319
9.27.1	Detailed Description	320
9.28	jeod_integration_group.cc File Reference	320
9.28.1	Detailed Description	320
9.29	jeod_integration_group.hh File Reference	320
9.29.1	Detailed Description	321
9.30	jeod_integration_time.cc File Reference	321
9.30.1	Detailed Description	321
9.31	jeod_integration_time.hh File Reference	321
9.31.1	Detailed Description	322
9.32	Isode_control_data_interface.cc File Reference	322
9.32.1	Detailed Description	322
9.33	Isode_control_data_interface.hh File Reference	322
9.33.1	Detailed Description	323
9.34	Isode_data_classes.cc File Reference	323
9.34.1	Detailed Description	323

9.35	Isode_data_classes.hh File Reference	323
9.35.1	Detailed Description	324
9.36	Isode_first_order_ode_integrator.hh File Reference	324
9.36.1	Detailed Description	324
9.37	Isode_first_order_ode_integrator__integrator.cc File Reference	324
9.37.1	Detailed Description	325
9.38	Isode_first_order_ode_integrator__manager.cc File Reference	325
9.38.1	Detailed Description	325
9.39	Isode_first_order_ode_integrator__support.cc File Reference	325
9.39.1	Detailed Description	325
9.40	Isode_first_order_ode_integrator__utility.cc File Reference	325
9.40.1	Detailed Description	326
9.41	Isode_generalized_second_order_ode_integrator.cc File Reference	326
9.41.1	Detailed Description	326
9.42	Isode_generalized_second_order_ode_integrator.hh File Reference	326
9.42.1	Detailed Description	326
9.43	Isode_integration_controls.cc File Reference	327
9.43.1	Detailed Description	327
9.44	Isode_integration_controls.hh File Reference	327
9.44.1	Detailed Description	327
9.45	Isode_integrator_constructor.cc File Reference	328
9.45.1	Detailed Description	328
9.46	Isode_integrator_constructor.hh File Reference	328
9.46.1	Detailed Description	328
9.47	Isode_second_order_ode_integrator.cc File Reference	329
9.47.1	Detailed Description	329
9.48	Isode_second_order_ode_integrator.hh File Reference	329
9.48.1	Detailed Description	329
9.49	Isode_simple_second_order_ode_integrator.cc File Reference	330
9.49.1	Detailed Description	330
9.50	Isode_simple_second_order_ode_integrator.hh File Reference	330
9.50.1	Detailed Description	330
9.51	restartable_2d_second_order_integrator.hh File Reference	331
9.51.1	Detailed Description	331
9.52	restartable_state_integrator.hh File Reference	331
9.52.1	Detailed Description	332
9.53	restartable_state_integrator_templates.hh File Reference	332
9.53.1	Detailed Description	333
9.54	time_change_subscriber.hh File Reference	333
9.54.1	Detailed Description	333
9.55	two_d_array.hh File Reference	334
9.55.1	Detailed Description	334

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Models	15
Utils	16
Integration	17
GaussJackson	18
Lsode	20

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

er7_utils	57
jeod	57
Namespace jeod	57

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

BaseIntegrationGroup	
jeod::JeodIntegrationGroup	178
FirstOrderODEIntegrator	
jeod::GaussJacksonFirstOrderODEIntegrator	80
jeod::LsodeFirstOrderODEIntegrator	213
jeod::GaussJacksonCoefficientsPair	64
jeod::GaussJacksonCoeffs	69
jeod::GaussJacksonConfig	75
jeod::GaussJacksonIntegratorBase< State, Primer >	101
jeod::GaussJacksonFirstOrderODEIntegrator	80
jeod::GaussJacksonSimpleSecondOrderODEIntegrator	148
jeod::GaussJacksonOneState	142
jeod::GaussJacksonRationalCoefficients	144
jeod::GaussJacksonStateMachine	154
jeod::GaussJacksonTwoState	168
jeod::GeneralizedSecondOrderODETechnique	171
IntegrationControls	
jeod::GaussJacksonIntegrationControls	89
jeod::IntegrationMessages	174
IntegratorConstructor	
jeod::GaussJacksonIntegratorConstructor	134
jeod::LsodeIntegratorConstructor	239
jeod::JeodIntegrationGroupOwner	187
jeod::LsodeControlDataInterface	192
jeod::LsodeDataArrays	201
jeod::LsodeDataJacobianPrep	206
jeod::LsodeDataStode	209
SecondOrderODEIntegrator	
jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator	83
jeod::GaussJacksonSimpleSecondOrderODEIntegrator	148
jeod::LsodeSecondOrderODEIntegrator	242
jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator	235
jeod::LsodeSimpleSecondOrderODEIntegrator	247
SimpleCheckpointable	
jeod::Restartable2DSecondOrderIntegrator	249

jeod::RestartableScalarFirstOrderODEIntegrator	263
jeod::RestartableSO3SecondOrderODEIntegrator	272
jeod::RestartableStateIntegrator< IntegratorType >	278
jeod::RestartableT3SecondOrderODEIntegrator	285
jeod::RestartableStateIntegrator< er7_utils::FirstOrderODEIntegrator >	278
jeod::RestartableFirstOrderODEIntegrator< size >	253
jeod::RestartableFirstOrderODEIntegrator< 1 >	253
jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >	278
jeod::RestartableSecondOrderODEIntegrator	268
jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions >	256
jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, StepFunctions >	260
jeod::RestartableSimpleSecondOrderODEIntegrator< size >	269
jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< 4, 3, er7_utils::Left←QuaternionGeneralizedPositionFunctions >	256
jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< 4, 3, er7_utils::Left←QuaternionGeneralizedPositionFunctions >	260
jeod::RestartableSimpleSecondOrderODEIntegrator< 2 >	269
jeod::RestartableSimpleSecondOrderODEIntegrator< 3 >	269
StandardIntegrationControls	
jeod::LsodeIntegrationControls	237
jeod::TimeChangeSubscriber	290
jeod::JeodIntegrationGroup	178
TimeInterface	
jeod::JeodIntegrationTime	188
er7_utils::TwoDArray< T >	292
er7_utils::TwoDArray< double >	292
er7_utils::DoubleTwoDArray	63

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

er7_utils::DoubleTwoDArray	2D array, specialized for doubles	63
jeod::GaussJacksonCoefficientsPair	Contains a summed Adams and Gauss-Jackson coefficient pair	64
jeod::GaussJacksonCoeffs	Contains the Gauss-Jackson predictor and corrector coefficients	69
jeod::GaussJacksonConfig	Contains Gauss-Jackson configuration data	75
jeod::GaussJacksonFirstOrderODEIntegrator	Integrates a first order ODE using the summed Adams technique that is embedded within the Gauss-Jackson technique	80
jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator	Integrates a generalized derivative second order ODE using Gauss-Jackson	83
jeod::GaussJacksonIntegrationControls	IntegrationControls specialized for Gauss-Jackson integration	89
jeod::GaussJacksonIntegratorBase< State, Primer >	Base template class for integrating state via the Gauss-Jackson technique	101
jeod::GaussJacksonIntegratorConstructor	Create state and time integrators that propagate using Gauss-Jackson	134
jeod::GaussJacksonOneState	Essentially just a double*	142
jeod::GaussJacksonRationalCoefficients	Contains a set of Adams or Stormer-Cowell coefficients	144
jeod::GaussJacksonSimpleSecondOrderODEIntegrator	Integrates a simple second order ODE using the Gauss-Jackson technique	148
jeod::GaussJacksonStateMachine	Guides the behavior of the Gauss-Jackson integration process via a finite state machine	154
jeod::GaussJacksonTwoState	Essentially just std::pair<double*>	168
jeod::GeneralizedSecondOrderODETechnique	Enumerates the integration schemes, generalized Cartesian or Lie group integration, and provides simple utilities that work with this	171
jeod::IntegrationMessages	Declares messages associated with the integration test model	174
jeod::JeodIntegrationGroup	A JeodIntegrationGroup integrates the state of a set of objects over time	178

jeod::JeodIntegrationGroupOwner	
The abstract class <code>IntegrationGroupOwner</code> contains an <code>IntegrationGroup</code>	187
jeod::JeodIntegrationTime	
The class <code>JeodIntegrationTime</code> adds the concepts of timestamping and notification of changes in the nature of time to the ER7 numerical utilities <code>TimeInterface</code> class	188
jeod::LsodeControlDataInterface	
Specifies controls for an LSODE integrator	192
jeod::LsodeDataArrays	
The data arrays	201
jeod::LsodeDataJacobianPrep	
Data associated with the method DPREPJ	206
jeod::LsodeDataStode	
The data associated with method Dstode	209
jeod::LsodeFirstOrderODEIntegrator	
Jeod-compatible version of the Livermore ODE solver, LSODE	213
jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator	
JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs	235
jeod::LsodeIntegrationControls	
Contains controls for an LSODE integrator	237
jeod::LsodeIntegratorConstructor	
Create state and time integrators that propagate using standard Lsode	239
jeod::LsodeSecondOrderODEIntegrator	
JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs	242
jeod::LsodeSimpleSecondOrderODEIntegrator	
JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs	247
jeod::Restartable2DSecondOrderIntegrator	
Integrates a second order ODE in two dimensional space, $d^2x/dt^2 = a(x,t)$, where x is a two-vector	249
jeod::RestartableFirstOrderODEIntegrator< size >	
A <code>RestartableFirstOrderODEIntegrator</code> is-a <code>RestartableStateIntegrator</code> that manages an <code>er7_↔utils::FirstOrderODEIntegrator</code>	253
jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions >	
A <code>RestartableGeneralizedDerivSecondOrderODEIntegrator</code> is-a <code>RestartableStateIntegrator</code> that manages the integrator for a generalized second order ODE problem, one in which the time derivative of the generalized position is some function of the generalized position and the generalized velocity	256
jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, StepFunctions >	
A <code>RestartableGeneralizedStepSecondOrderODEIntegrator</code> is-a <code>RestartableStateIntegrator</code> that manages the integrator for a generalized second order ODE problem, one in which the time derivative of the generalized position is some function of the generalized position and the generalized velocity	260
jeod::RestartableScalarFirstOrderODEIntegrator	
A <code>RestartableScalarFirstOrderODEIntegrator</code> integrates a first order ODE, $dx/dt = v(x,t)$, where x is a scalar	263
jeod::RestartableSecondOrderODEIntegrator	
A <code>RestartableSecondOrderODEIntegrator</code> is-a <code>RestartableStateIntegrator</code> that manages the integrator for a second order ODE problem	268
jeod::RestartableSimpleSecondOrderODEIntegrator< size >	
A <code>RestartableSimpleSecondOrderODEIntegrator</code> is-a <code>RestartableSecondOrderODEIntegrator</code> that manages the integrator for a simple second order ODE problem, one in which the time derivative of the generalized position is the generalized velocity	269
jeod::RestartableSO3SecondOrderODEIntegrator	
A <code>RestartableSO3SecondOrderODEIntegrator</code> integrates a generalized second order ODE that describes rotation in three space	272

jeod::RestartableStateIntegrator< IntegratorType >	
A RestartableStateIntegrator establishes the basic capabilities needed to make a state integrator a managed resource	278
jeod::RestartableT3SecondOrderODEIntegrator	
A RestartableT3SecondOrderODEIntegrator integrates a second order ODE in three space, $d^2x/dt^2 = a(x,t)$, where x is a three-vector	285
jeod::TimeChangeSubscriber	
A TimeChangeSubscriber is some object that wants to be notified of changes in the nature of time	290
er7_utils::TwoDArray< T >	
RAII template class that implements a rectangular two dimensional array	292

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

gauss_jackson_coefficients_pair.cc	Defines member functions for the class GaussJacksonCoefficientsPair	305
gauss_jackson_coefficients_pair.hh	Defines the class GaussJacksonCoefficientsPair, which contains summed Adams and Gauss-Jackson coefficient pair	305
gauss_jackson_coeffs.cc	Defines member functions for the class GaussJacksonCoeffs	306
gauss_jackson_coeffs.hh	Defines the class GaussJacksonCoeffs, which contains the Gauss-Jackson predictor and corrector coefficients	307
gauss_jackson_config.cc	Defines member functions for the class GaussJacksonIntegratorConstructor	307
gauss_jackson_config.hh	Defines the class GaussJacksonConfig, which specifies Gauss-Jackson configuration data . .	308
gauss_jackson_first_order_ode_integrator.hh	Defines the class GaussJacksonFirstOrderODEIntegrator, which integrates a first order ODE using the summed Adams technique	308
gauss_jackson_generalized_second_order_ode_integrator.cc	Defines member functions for the class GaussJacksonGeneralizedDerivSecondOrderODEIntegrator	309
gauss_jackson_generalized_second_order_ode_integrator.hh	Defines the class GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, which integrates a generalized derivative second order ODE using Gauss-Jackson	309
gauss_jackson_integration_controls.cc	Defines member functions for the class GaussJacksonIntegrationControls	310
gauss_jackson_integration_controls.hh	Defines the class GaussJacksonIntegrationControls, which controls Gauss-Jackson integration process	310
gauss_jackson_integrator_base.hh	Defines the template class GaussJacksonIntegratorBase, which provides the basis for Gauss-Jackson integration	311
gauss_jackson_integrator_base_first.hh	Defines the template specializations of GaussJacksonIntegratorBase for a first order ODE . . .	312
gauss_jackson_integrator_base_second.hh	Defines the template specializations of GaussJacksonIntegratorBase for a first order ODE . . .	312

gauss_jackson_integrator_constructor.cc	Defines member functions for the class GaussJacksonIntegratorConstructor	313
gauss_jackson_integrator_constructor.hh	Defines the class GaussJacksonIntegratorConstructor, which constructs integrators that use Gauss-Jackson integration	314
gauss_jackson_one_state.hh	Defines the class GaussJacksonOneState, which contains a double* pointer	314
gauss_jackson_rational_coeffs.cc	Defines member functions for the class GaussJacksonRationalCoefficients	315
gauss_jackson_rational_coeffs.hh	Defines the class GaussJacksonRationalCoefficients, which contains a set of Adams or Stormer-Cowell coefficients	315
gauss_jackson_simple_second_order_ode_integrator.hh	Defines the class GaussJacksonSimpleSecondOrderODEIntegrator, which integrates a simple second order ODE using Gauss-Jackson	316
gauss_jackson_state_machine.cc	Defines member functions for the class GaussJacksonStateMachine	316
gauss_jackson_state_machine.hh	Defines the class GaussJacksonStateMachine, which guides the Gauss-Jackson integration process	317
gauss_jackson_two_state.hh	Defines the class GaussJacksonTwoState, which contains a pair of double* pointers	317
generalized_second_order_ode_technique.cc	Define class GeneralizedSecondOrderODETechnique methods	318
generalized_second_order_ode_technique.hh	Define the static class GeneralizedSecondOrderODETechnique	318
integration_messages.cc	Implement the class IntegrationMessages	319
integration_messages.hh	Define the class IntegrationMessages, the class that specifies the message IDs used in the integration model	319
jeod_integration_group.cc	Define JeodIntegrationGroup methods	320
jeod_integration_group.hh	Define the extensible class IntegrationGroup, an instance of which is responsible for integrating the states of a set of DynBody objects	320
jeod_integration_time.cc	Define JeodIntegrationTime methods	321
jeod_integration_time.hh	Define the class JeodIntegrationTime	321
lode_control_data_interface.cc	Define member functions for the class LodeControlDataInterface	322
lode_control_data_interface.hh	Define the class LodeControlDataInterface	322
lode_data_classes.cc	Define member functions for the data-grouping classes specified in lode_data_classes	323
lode_data_classes.hh	Define LODE classes that contain just data members	323
lode_first_order_ode_integrator.hh	Define the class LodeFirstOrderODEIntegrator, the Jeod-compatible version of the Livermore ODE solver, LODE	324
lode_first_order_ode_integrator__integrator.cc	Define member functions for the class LodeFirstOrderODEIntegrator	324
lode_first_order_ode_integrator__manager.cc	Define member functions for the class LodeFirstOrderODEIntegrator	325
lode_first_order_ode_integrator__support.cc	Define member functions for the class LodeFirstOrderODEIntegrator	325

lsode_first_order_ode_integrator__utility.cc	
Define member functions for the class LsodeFirstOrderODEIntegrator	325
lsode_generalized_second_order_ode_integrator.cc	
Define member functions for the class LsodeGeneralizedDerivSecondOrderODEIntegrator . . .	326
lsode_generalized_second_order_ode_integrator.hh	
Define the class LsodeGeneralizedDerivSecondOrderODEIntegrator	326
lsode_integration_controls.cc	
Define the methods for the class LsodeIntegrationControls	327
lsode_integration_controls.hh	
Define the class LsodeIntegrationControls	327
lsode_integrator_constructor.cc	
Define the methods in the class LsodeIntegratorConstructor	328
lsode_integrator_constructor.hh	
Define the class LsodeIntegratorConstructor, the class that constructs the integration controls and the integrators for the LSODE method	328
lsode_second_order_ode_integrator.cc	
Define member functions for the class LsodeSecondOrderODEIntegrator	329
lsode_second_order_ode_integrator.hh	
Define the class LsodeSecondOrderODEIntegrator	329
lsode_simple_second_order_ode_integrator.cc	
Define member functions for the class LsodeSimpleSecondOrderODEIntegrator	330
lsode_simple_second_order_ode_integrator.hh	
Define the class LsodeSimpleSecondOrderODEIntegrator	330
restartable_2d_second_order_integrator.hh	
Defines the class Restartable2DSecondOrderODEIntegrator	331
restartable_state_integrator.hh	
Define classes that encapsulate the construction, destruction, checkpointing, and restarting of state integrators	331
restartable_state_integrator_templates.hh	
Define template classes that encapsulate the construction, destruction, checkpointing, and restarting of state integrators	332
time_change_subscriber.hh	
Define the class TimeChangeSubscriber	333
two_d_array.hh	
Defines the template class <code>er7_utils::TwoDArray</code> , which implements an RAll rectangular 2D array	334

Chapter 6

Module Documentation

6.1 Models

Modules

- [Utils](#)

6.1.1 Detailed Description

6.2 Utils

Modules

- [Integration](#)

6.2.1 Detailed Description

6.3 Integration

Modules

- [GaussJackson](#)
- [Lsode](#)

Files

- file [generalized_second_order_ode_technique.hh](#)
Define the static class GeneralizedSecondOrderODETechnique.
- file [integration_messages.hh](#)
Define the class IntegrationMessages, the class that specifies the message IDs used in the integration model.
- file [jeod_integration_group.hh](#)
Define the extensible class IntegrationGroup, an instance of which is responsible for integrating the states of a set of DynBody objects.
- file [jeod_integration_time.hh](#)
Define the class JeodIntegrationTime.
- file [restartable_2d_second_order_integrator.hh](#)
Defines the class Restartable2DSecondOrderODEIntegrator.
- file [restartable_state_integrator.hh](#)
Define classes that encapsulate the construction, destruction, checkpointing, and restarting of state integrators.
- file [restartable_state_integrator_templates.hh](#)
Define template classes that encapsulate the construction, destruction, checkpointing, and restarting of state integrators.
- file [time_change_subscriber.hh](#)
Define the class TimeChangeSubscriber.
- file [generalized_second_order_ode_technique.cc](#)
Define class GeneralizedSecondOrderODETechnique methods.
- file [integration_messages.cc](#)
Implement the class IntegrationMessages.
- file [jeod_integration_group.cc](#)
Define JeodIntegrationGroup methods.
- file [jeod_integration_time.cc](#)
Define JeodIntegrationTime methods.

Namespaces

- [jeod](#)
Namespace jeod.

6.3.1 Detailed Description

6.4 GaussJackson

Files

- file [gauss_jackson_coefficients_pair.hh](#)
Defines the class GaussJacksonCoefficientsPair, which contains summed Adams and Gauss-Jackson coefficient pair.
- file [gauss_jackson_coefs.hh](#)
Defines the class GaussJacksonCoeffs, which contains the Gauss-Jackson predictor and corrector coefficients.
- file [gauss_jackson_config.hh](#)
Defines the class GaussJacksonConfig, which specifies Gauss-Jackson configuration data.
- file [gauss_jackson_first_order_ode_integrator.hh](#)
Defines the class GaussJacksonFirstOrderODEIntegrator, which integrates a first order ODE using the summed Adams technique.
- file [gauss_jackson_generalized_second_order_ode_integrator.hh](#)
Defines the class GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, which integrates a generalized derivative second order ODE using Gauss-Jackson.
- file [gauss_jackson_integration_controls.hh](#)
Defines the class GaussJacksonIntegrationControls, which controls Gauss-Jackson integration process.
- file [gauss_jackson_integrator_base.hh](#)
Defines the template class GaussJacksonIntegratorBase, which provides the basis for Gauss-Jackson integration.
- file [gauss_jackson_integrator_base_first.hh](#)
Defines the template specializations of GaussJacksonIntegratorBase for a first order ODE.
- file [gauss_jackson_integrator_base_second.hh](#)
Defines the template specializations of GaussJacksonIntegratorBase for a first order ODE.
- file [gauss_jackson_integrator_constructor.hh](#)
Defines the class GaussJacksonIntegratorConstructor, which constructs integrators that use Gauss-Jackson integration.
- file [gauss_jackson_one_state.hh](#)
Defines the class GaussJacksonOneState, which contains a double pointer.*
- file [gauss_jackson_rational_coefs.hh](#)
Defines the class GaussJacksonRationalCoefficients, which contains a set of Adams or Stormer-Cowell coefficients.
- file [gauss_jackson_simple_second_order_ode_integrator.hh](#)
Defines the class GaussJacksonSimpleSecondOrderODEIntegrator, which integrates a simple second order ODE using Gauss-Jackson.
- file [gauss_jackson_state_machine.hh](#)
Defines the class GaussJacksonStateMachine, which guides the Gauss-Jackson integration process.
- file [gauss_jackson_two_state.hh](#)
Defines the class GaussJacksonTwoState, which contains a pair of double pointers.*
- file [two_d_array.hh](#)
Defines the template class `er7_utils::TwoDArray`, which implements an RAII rectangular 2D array.
- file [gauss_jackson_coefficients_pair.cc](#)
Defines member functions for the class GaussJacksonCoefficientsPair.
- file [gauss_jackson_coefs.cc](#)
Defines member functions for the class GaussJacksonCoeffs.
- file [gauss_jackson_config.cc](#)
Defines member functions for the class GaussJacksonConfig.
- file [gauss_jackson_generalized_second_order_ode_integrator.cc](#)
Defines member functions for the class GaussJacksonGeneralizedDerivSecondOrderODEIntegrator.
- file [gauss_jackson_integration_controls.cc](#)
Defines member functions for the class GaussJacksonIntegrationControls.
- file [gauss_jackson_integrator_constructor.cc](#)

Defines member functions for the class GaussJacksonIntegratorConstructor.

- file [gauss_jackson_rational_coeffs.cc](#)

Defines member functions for the class GaussJacksonRationalCoefficients.

- file [gauss_jackson_state_machine.cc](#)

Defines member functions for the class GaussJacksonStateMachine.

Namespaces

- [jeod](#)

Namespace jeod.

- [er7_utils](#)

6.4.1 Detailed Description

6.5 Lsode

Files

- file [lsode_control_data_interface.hh](#)
Define the class LsodeControlDataInterface.
- file [lsode_data_classes.hh](#)
Define LSODE classes that contain just data members.
- file [lsode_first_order_ode_integrator.hh](#)
Define the class LsodeFirstOrderODEIntegrator, the Jeod-compatible version of the Livermore ODE solver, LSODE.
- file [lsode_generalized_second_order_ode_integrator.hh](#)
Define the class LsodeGeneralizedDerivSecondOrderODEIntegrator.
- file [lsode_integration_controls.hh](#)
Define the class LsodeIntegrationControls.
- file [lsode_integrator_constructor.hh](#)
Define the class LsodeIntegratorConstructor, the class that constructs the integration controls and the integrators for the LSODE method.
- file [lsode_second_order_ode_integrator.hh](#)
Define the class LsodeSecondOrderODEIntegrator.
- file [lsode_simple_second_order_ode_integrator.hh](#)
Define the class LsodeSimpleSecondOrderODEIntegrator.
- file [lsode_control_data_interface.cc](#)
Define member functions for the class LsodeControlDataInterface.
- file [lsode_data_classes.cc](#)
Define member functions for the data-grouping classes specified in lsode_data_classes.
- file [lsode_first_order_ode_integrator__integrator.cc](#)
Define member functions for the class LsodeFirstOrderODEIntegrator.
- file [lsode_first_order_ode_integrator__manager.cc](#)
Define member functions for the class LsodeFirstOrderODEIntegrator.
- file [lsode_first_order_ode_integrator__support.cc](#)
Define member functions for the class LsodeFirstOrderODEIntegrator.
- file [lsode_first_order_ode_integrator__utility.cc](#)
Define member functions for the class LsodeFirstOrderODEIntegrator.
- file [lsode_generalized_second_order_ode_integrator.cc](#)
Define member functions for the class LsodeGeneralizedDerivSecondOrderODEIntegrator.
- file [lsode_integration_controls.cc](#)
Define the methods for the class LsodeIntegrationControls.
- file [lsode_integrator_constructor.cc](#)
Define the methods in the class LsodeIntegratorConstructor.
- file [lsode_second_order_ode_integrator.cc](#)
Define member functions for the class LsodeSecondOrderODEIntegrator.
- file [lsode_simple_second_order_ode_integrator.cc](#)
Define member functions for the class LsodeSimpleSecondOrderODEIntegrator.

Namespaces

- [jeod](#)
Namespace jeod.

Functions

- `jeod::LsodeControlDataInterface::LsodeControlDataInterface ()`
constructor
- `void jeod::LsodeControlDataInterface::check_interface_data ()`
verifies that the input data has legal values.
- `void jeod::LsodeControlDataInterface::allocate_arrays ()`
allocates space for vector-populated data to allow for restart
- `void jeod::LsodeControlDataInterface::destroy_allocated_arrays ()`
De-allocates allocated array.
- `void jeod::LsodeControlDataInterface::set_rel_tol (int index, double value)`
set values from external
- `void jeod::LsodeControlDataInterface::set_abs_tol (int index, double value)`
- `void jeod::LsodeDataArrays::allocate_arrays (unsigned int num_odes, LsodeControlDataInterface::CorrectorMethod corrector_method)`
Allocates memory for the variable size arrays.
- `void jeod::LsodeDataArrays::destroy_allocated_arrays ()`
Allows for refactoring and reallocation of newly sized arrays.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_core ()`
*integrator_core provides the front-end to all of the integrator_*i methods, which together perform one step of the integration of an initial value problem for a system of ordinary differential equations.*
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_reset_method_coeffs ()`
Sets/resets the method_coeffs_current array.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_test_stepsize_change ()`
Tests h against old h.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_reset_yh ()`
Resets history arrays and time-step.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_predict ()`
This section computes the predicted values by multiplying the history array by the Pascal Triangle matrix.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_reset_iteration_loop_part1 ()`
This method resets the iteration loop to the values generated by the integrator_predict method, which populated history[][0].*
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_reset_iteration_loop_part2 ()`
This code follows part 1 after the break-out to get to the external calls, and completes the reset of the integration iteration loop.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration ()`
Keeps looping through the iterations until convergence or failure.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part1 ()`
The corrector iteration failed to converge.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2 ()`
Retracts the history array in the case that the correction iteration failed to converge with either functional iteration or with an up-to-date Jacobian.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_converged ()`
Starts the processing of a converged iteration.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_error_test_failed ()`
Restores the history array following the failure of the corrector for exceeding local error bounds.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_prep ()`
The first steps in computing whether the order of the integrator should be changed.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order ()`
Computes the step-size scaling factors that will result once it is determined what happens to the order at the next step.
- `void jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_check_step_error ()`

- void `jeod::LsodeFirstOrderODEIntegrator::integrator_set_new_order ()`
Sets the new order and the step-ratio for the next step - or the current step if redoing it.
- void `jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part1 ()`
Repopulates the y-array from history, recomputes the original first derivatives, sets the order back to 1, and reduces the step size by a factor of 10.
- void `jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part2 ()`
Continue reset, with derivatives now at hand.
- void `jeod::LsodeFirstOrderODEIntegrator::integrator_wrapup ()`
Wraps up the completion of the integrator.
- void `jeod::LsodeFirstOrderODEIntegrator::integrator_terminate ()`
this is the only succesful path back from integrator to manager.
- `er7_utils::IntegratorResult jeod::LsodeFirstOrderODEIntegrator::integrate` (double dyn_dt, unsigned int target_stage, const double *y_dot, double *y) override
Propagate state via Lsode's method.
- void `jeod::LsodeFirstOrderODEIntegrator::process_entry_point_cycle_start ()`
The code block from the main integrate routine for re_entry_point=CycleStartFinish.
- void `jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part1 ()`
Sets the values for the case with calculation_phase = 1.
- void `jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part2 ()`
- int `jeod::LsodeFirstOrderODEIntegrator::manager_check_stop_conditions ()`
verifies whether the convergence conditions have been met to end the cycle.
- void `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part1 ()`
The iteration loop for the integration process.
- void `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part2 ()`
- void `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part3 ()`
- void `jeod::LsodeFirstOrderODEIntegrator::reset_integrator ()` override
Resets the integrator when the timestep changes or when identified as needing a reset.
- void `jeod::LsodeFirstOrderODEIntegrator::manager_set_calculation_phase_eq_2_reload ()`
- void `jeod::LsodeFirstOrderODEIntegrator::calculate_epsilon ()`
Identify the smallest double precision value, epsilon, such that the computer can distinguish (1+epsilon) from 1.
- void `jeod::LsodeFirstOrderODEIntegrator::calculate_integration_coefficients ()`
Modified from original DCFODE subroutine.
- void `jeod::LsodeFirstOrderODEIntegrator::interpolate_y ()`
Interpolates the zeroth-derivative of y Adapted from subroutine DINTDY, which was a general method for interpolating the K-th derivative of the dependent variable vector, y.
- void `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init ()`
Modified from DPREPJ.
- bool `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop ()`
- bool `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_wrap_up ()`
- void `jeod::LsodeFirstOrderODEIntegrator::linear_chord_iteration ()`
Modified from DSOLSY.
- void `jeod::LsodeFirstOrderODEIntegrator::load_ew_values ()`
- `jeod::LsodeFirstOrderODEIntegrator::LsodeFirstOrderODEIntegrator ()`
LsodeFirstOrderODEIntegrator default constructor.
- `jeod::LsodeFirstOrderODEIntegrator::LsodeFirstOrderODEIntegrator` (const `LsodeControlDataInterface` &data_in, `er7_utils::IntegrationControls` &controls, unsigned int size)
LsodeFirstOrderODEIntegrator non-default constructor.
- `jeod::LsodeFirstOrderODEIntegrator::~~LsodeFirstOrderODEIntegrator ()` override
LsodeFirstOrderODEIntegrator destructor.
- void `jeod::LsodeFirstOrderODEIntegrator::update_control_data ()`
Gets the control data from where it can be populated in the constructor and verifies that the input control parameters are not out of sensible range.

- `LsodeFirstOrderODEIntegrator * jeod::LsodeFirstOrderODEIntegrator::create_copy ()` const override
Create a copy of 'this' LsodeFirstOrderODEIntegrator object.
- `double jeod::LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array (const double *v)`
returns the RMS value of $\{V \cdot W\}$, where V and W are N -vectors.
- `double jeod::LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array (unsigned int ix, double **v)`
returns RMS value of $v[][index]$*
- `int jeod::LsodeFirstOrderODEIntegrator::gauss_elim_factor ()`
Factors a double array (arrays.lin_alg) by Gaussian elimination.
- `void jeod::LsodeFirstOrderODEIntegrator::linear_solver ()`
Solves the equation $Y' = A Y$, with $A = \text{arrays.lin_alg}$.
- `unsigned int jeod::LsodeFirstOrderODEIntegrator::index_of_max_magnitude (unsigned int num_points, double **mx, int starting_ix)`
Modified version of IDAMAX.
- `void jeod::LsodeFirstOrderODEIntegrator::load_derivatives (double *derivs)`
Load the externally generated derivative values (incoming as $y_{\cdot}i$) into the array derivs.
- `jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::LsodeGeneralizedDerivSecondOrderODEIntegrator (const LsodeControlDataInterface &data_in, er7_utils::IntegrationControls &controls, const er7_utils::GeneralizedPositionDerivativeFunctions &deriv_funs, unsigned int position_size, unsigned int velocity_size)`
non-default constructor
- `jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::LsodeGeneralizedDerivSecondOrderODEIntegrator (const LsodeGeneralizedDerivSecondOrderODEIntegrator &src)`
LsodeGeneralizedDerivSecondOrderODEIntegrator copy constructor.
- `LsodeGeneralizedDerivSecondOrderODEIntegrator * jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::create_copy ()` const override
Clone a LsodeGeneralizedDerivSecondOrderODEIntegrator.
- `jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::~~LsodeGeneralizedDerivSecondOrderODEIntegrator ()` override
LsodeGeneralizedDerivSecondOrderODEIntegrator destructor.
- `er7_utils::IntegratorResult jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::integrate (double dyn_dt, unsigned int target_stage, const double *accel, double *velocity, double *position)` override
Propagate state via Lsode's method.
- `jeod::LsodeIntegrationControls::LsodeIntegrationControls (unsigned int num_stages)`
- `LsodeIntegrationControls * jeod::LsodeIntegrationControls::create_copy ()` const override
Create a copy of 'this' StandardIntegrationControls object.
- `unsigned int jeod::LsodeIntegrationControls::integrate (double start_time, double sim_dt, er7_utils::TimeInterface &time_interface, er7_utils::IntegratorInterface &integ_interface, er7_utils::BaseIntegrationGroup &integ_group)` override
Perform one step of the integration process.
- `jeod::LsodeIntegratorConstructor::LsodeIntegratorConstructor (const LsodeIntegratorConstructor &src)`
- `static er7_utils::IntegratorConstructor * jeod::LsodeIntegratorConstructor::create_constructor ()`
Named constructor; create an LsodeIntegratorConstructor instance.
- `er7_utils::IntegratorConstructor * jeod::LsodeIntegratorConstructor::create_copy ()` const override
Create a duplicate of the constructor.
- `er7_utils::IntegrationControls * jeod::LsodeIntegratorConstructor::create_integration_controls ()` const override
Create an integration controls that guides the Lsode integration process.
- `er7_utils::FirstOrderODEIntegrator * jeod::LsodeIntegratorConstructor::create_first_order_ode_integrator (unsigned int size, er7_utils::IntegrationControls &controls)` const override
Create an Lsode state integrator for a first order ODE.
- `er7_utils::SecondOrderODEIntegrator * jeod::LsodeIntegratorConstructor::create_second_order_ode_integrator (unsigned int size, er7_utils::IntegrationControls &controls)` const override
Create an Lsode state integrator for a simple second order ODE.

- `er7_utils::SecondOrderODEIntegrator * jeod::LsodeIntegratorConstructor::create_generalized_deriv_second_order_ode_integ`
(unsigned int position_size, unsigned int velocity_size, const er7_utils::GeneralizedPositionDerivative↵
Functions &deriv_funs, er7_utils::IntegrationControls &controls) const override

*Create an Lsode state integrator for a generalized second order ODE where generalized position is advanced with
the use of the position derivative function.*
- `jeod::LsodeSecondOrderODEIntegrator::LsodeSecondOrderODEIntegrator` (const LsodeControlDataInterface
&data_in, er7_utils::IntegrationControls &controls, unsigned int size)

LsodeSecondOrderODEIntegrator non-default constructor.
- `jeod::LsodeSecondOrderODEIntegrator::LsodeSecondOrderODEIntegrator` (const LsodeControlDataInterface
&data_in, er7_utils::IntegrationControls &controls, const er7_utils::GeneralizedPositionDerivativeFunctions
&deriv_funs, unsigned int position_size, unsigned int velocity_size)
- `jeod::LsodeSecondOrderODEIntegrator::~~LsodeSecondOrderODEIntegrator` () override

LsodeSecondOrderODEIntegrator destructor.
- `jeod::LsodeSimpleSecondOrderODEIntegrator::LsodeSimpleSecondOrderODEIntegrator` (const LsodeControlDataInterface
&data_in, er7_utils::IntegrationControls &controls, unsigned int size)

LsodeSimpleSecondOrderODEIntegrator non-default constructor.
- `LsodeSimpleSecondOrderODEIntegrator * jeod::LsodeSimpleSecondOrderODEIntegrator::create_copy` ()
const override
- `er7_utils::IntegratorResult jeod::LsodeSimpleSecondOrderODEIntegrator::integrate` (double dyn_dt, un-
signed int target_stage, const double *accel, double *velocity, double *position) override

Propagate state via Lsode's method.

6.5.1 Detailed Description

6.5.2 Function Documentation

6.5.2.1 `allocate_arrays()` [1/2]

```
void LsodeDataArrays::allocate_arrays (
    unsigned int num_odes,
    LsodeControlDataInterface::CorrectorMethod corrector_method )
```

Allocates memory for the variable size arrays.

Definition at line 52 of file `Lsode_data_classes.cc`.

References `jeod::LsodeDataArrays::accum_correction`, `jeod::LsodeDataArrays::allocated`, `jeod::LsodeData↵
Arrays::error_weight`, `jeod::LsodeControlDataInterface::FunctionalIteration`, `jeod::LsodeDataArrays::history`, `jeod↵
::LsodeControlDataInterface::JacobiNewtonInternalJac`, `jeod::LsodeDataArrays::lin_alg`, `jeod::LsodeDataArrays↵
::lin_alg_index1`, `jeod::LsodeControlDataInterface::NewtonIterInternalJac`, `jeod::LsodeControlDataInterface::↵
NewtonIterUserJac`, `jeod::LsodeDataArrays::num_odes`, `jeod::LsodeDataArrays::pivots`, and `jeod::LsodeData↵
Arrays::save`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part1()`.

6.5.2.2 allocate_arrays() [2/2]

```
void LsodeControlDataInterface::allocate_arrays ( )
```

allocates space for vector-populated data to allow for restart

Definition at line 260 of file lsode_control_data_interface.cc.

References jeod::LsodeControlDataInterface::abs_tolerance_error_control, jeod::LsodeControlDataInterface::abs_tolerance_error_control_vec, jeod::LsodeControlDataInterface::error_control_vector_copied_over, jeod::LsodeControlDataInterface::num_odes, jeod::LsodeControlDataInterface::num_odes_at_alloc, jeod::LsodeControlDataInterface::rel_tolerance_error_control, and jeod::LsodeControlDataInterface::rel_tolerance_error_control_vec.

Referenced by jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part1().

6.5.2.3 calculate_epsilon()

```
void LsodeFirstOrderODEIntegrator::calculate_epsilon ( ) [protected]
```

Identify the smallest double precision value, epsilon, such that the computer can distinguish (1+epsilon) from 1.

Definition at line 58 of file lsode_first_order_ode_integrator__support.cc.

References jeod::LsodeFirstOrderODEIntegrator::epsilon, and jeod::LsodeFirstOrderODEIntegrator::sqrt_epsilon.

Referenced by jeod::LsodeFirstOrderODEIntegrator::LsodeFirstOrderODEIntegrator().

6.5.2.4 calculate_integration_coefficients()

```
void LsodeFirstOrderODEIntegrator::calculate_integration_coefficients ( ) [protected]
```

Modified from original DCFODE subroutine.

calculate_integration_coefficients is called by dstode to set coefficients needed there. The coefficients for the current method, as given by the value of integration_method, are set for all orders and saved. The maximum order assumed here is 12 if integration_method = ImplicitAdamsNonStiff and 5 if integration_method = ImplicitBackDiffStiff.

NOTE - A smaller value of the maximum order is also allowed and may be set by the user with the value control_data.max_order, which gets copied to the protected value max_order_internal.

calculate_integration_coefficients is called once at the beginning of the problem, and again only if integration_method is changed.

The coefficients are stored in two arrays: method_coeffs_complete is a 13x12 array that contains a complete set of coefficients for the method test_coeffs_complete is a 3x12 array that contains the coefficients for local error tests and selection of the step size and/or order. The 1st set of 12 coeffs is for order method_order_current - 1 The 2nd set of 12 coeffs is for order method_order_current The 3rd set of 12 coeffs is for order method_order_current + 1

The coefficients in `method_coeffs_complete` are computed by a generating polynomial. For a given order (note that order changes during the integration process up to the maximum allowable, and is identified in the integrator as the variable `method_order_current`), abbreviate `method_coeffs_complete[i][order-1]` to `mcc[i]`

Then $l(x) = \text{mcc}[0] + (\text{mcc}[1] * x) + (\text{mcc}[2] * x^2) + \dots + \text{mcc}[\text{order}] * (x^{\text{order}})$ For the implicit Adams methods, $l(x)$ is given by $dl/dx = (x+1)*(x+2)*\dots*(x+\text{order}-1)/(\text{order}-1)!$, $l(-1) = 0$.; For the BDF methods, $l(x)$ is given by $l(x) = (x+1)*(x+2)*\dots*(x+\text{order}) / ((\text{order})! * (1 + 1/2 + \dots + 1/\text{order}))$

(! represents factorial)

Note that while `method_coeffs_compelte` is a rectangular array for convenience, it is effectively a triangular array since `method_coeffs_complete[i][order-1]` has no meaning for $i \geq (\text{order} + 2)$ `order=1: method_coeffs_complete[0-1][0]` `order=2: method_coeffs_complete[0-2][1]` ... `order=12: method_coeffs_complete[0-12][11]` Hence a 13x12 array.

Definition at line 119 of file `lsode_first_order_ode_integrator__support.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeControlDataInterface::ImplicitAdamsNonStiff`, `jeod::LsodeControlDataInterface::ImplicitBackDiffStiff`, `jeod::LsodeControlDataInterface::integration_method`, `jeod::LsodeFirstOrderODEIntegrator::method_coeffs_complete`, and `jeod::LsodeFirstOrderODEIntegrator::test_coeffs_complete`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_core()`.

6.5.2.5 check_interface_data()

```
void LsodeControlDataInterface::check_interface_data ( )
```

verifies that the input data has legal values.

Definition at line 59 of file `lsode_control_data_interface.cc`.

References `jeod::LsodeControlDataInterface::abs_tolerance_error_control_vec`, `jeod::LsodeControlDataInterface::CommonAbsSpecificRel`, `jeod::LsodeControlDataInterface::corrector_method`, `jeod::LsodeControlDataInterface::error_control_indicator`, `jeod::LsodeControlDataInterface::error_control_vector_copied_over`, `jeod::LsodeControlDataInterface::integration_method`, `jeod::LsodeControlDataInterface::max_num_small_step_warnings`, `jeod::LsodeControlDataInterface::max_num_steps`, `jeod::LsodeControlDataInterface::max_order`, `jeod::LsodeControlDataInterface::max_step_size`, `jeod::LsodeControlDataInterface::min_step_size`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeControlDataInterface::rel_tolerance_error_control_vec`, `jeod::LsodeControlDataInterface::SpecificAbsCommonRel`, and `jeod::LsodeControlDataInterface::SpecificAbsSpecificRel`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::update_control_data()`.

6.5.2.6 create_constructor()

```
er7_utils::IntegratorConstructor * LsodeIntegratorConstructor::create_constructor ( ) [static]
```

Named constructor; create an [LsodeIntegratorConstructor](#) instance.

The caller is responsible for deleting the returned object.

Returns

Newly created [LsodeIntegratorConstructor](#) instance.

Definition at line 64 of file `lsode_integrator_constructor.cc`.

6.5.2.7 create_copy() [1/5]

```
LsodeIntegrationControls * LsodeIntegrationControls::create_copy ( ) const [override]
```

Create a copy of 'this' StandardIntegrationControls object.

Returns

Clone of 'this'.

Definition at line 56 of file lsode_integration_controls.cc.

6.5.2.8 create_copy() [2/5]

```
LsodeSimpleSecondOrderODEIntegrator * LsodeSimpleSecondOrderODEIntegrator::create_copy ( )
const [override]
```

Definition at line 57 of file lsode_simple_second_order_ode_integrator.cc.

6.5.2.9 create_copy() [3/5]

```
LsodeGeneralizedDerivSecondOrderODEIntegrator * LsodeGeneralizedDerivSecondOrderODEIntegrator↔
::create_copy ( ) const [override]
```

Clone a [LsodeGeneralizedDerivSecondOrderODEIntegrator](#).

Definition at line 81 of file lsode_generalized_second_order_ode_integrator.cc.

6.5.2.10 create_copy() [4/5]

```
er7_utils::IntegratorConstructor * LsodeIntegratorConstructor::create_copy ( ) const [override]
```

Create a duplicate of the constructor.

The caller is responsible for deleting the returned object.

Returns

Duplicated constructor.

Definition at line 70 of file lsode_integrator_constructor.cc.

6.5.2.11 create_copy() [5/5]

```
LsodeFirstOrderODEIntegrator * LsodeFirstOrderODEIntegrator::create_copy ( ) const [override]
```

Create a copy of 'this' [LsodeFirstOrderODEIntegrator](#) object.

Returns

Clone of 'this'.

Definition at line 119 of file `lsode_first_order_ode_integrator__utility.cc`.

6.5.2.12 create_first_order_ode_integrator()

```
er7_utils::FirstOrderODEIntegrator * LsodeIntegratorConstructor::create_first_order_ode_↵
integrator (
    unsigned int size,
    er7_utils::IntegrationControls & controls ) const [override]
```

Create an Lsode state integrator for a first order ODE.

The caller is responsible for deleting the created object.

Returns

State integrator

Parameters

in	<i>size</i>	State size
in, out	<i>controls</i>	Integration controls

Definition at line 82 of file `lsode_integrator_constructor.cc`.

References `jeod::LsodeIntegratorConstructor::data_interface`.

6.5.2.13 create_generalized_deriv_second_order_ode_integrator()

```
er7_utils::SecondOrderODEIntegrator * LsodeIntegratorConstructor::create_generalized_deriv_↵
second_order_ode_integrator (
    unsigned int position_size,
    unsigned int velocity_size,
    const er7_utils::GeneralizedPositionDerivativeFunctions & deriv_funs,
    er7_utils::IntegrationControls & controls ) const [override]
```

Create an Lsode state integrator for a generalized second order ODE where generalized position is advanced with the use of the position derivative function.

The caller is responsible for deleting the created object.

Returns

State integrator

Parameters

in	<i>position_size</i>	Size of the generalized position
in	<i>velocity_size</i>	Size of the generalized velocity
in	<i>deriv_funs</i>	Position derivative functions container
in, out	<i>controls</i>	Integration controls

Definition at line 102 of file `lsode_integrator_constructor.cc`.

References `jeod::LsodeIntegratorConstructor::data_interface`.

6.5.2.14 create_integration_controls()

```
er7_utils::IntegrationControls * LsodeIntegratorConstructor::create_integration_controls ( )
const [override]
```

Create an integration controls that guides the Lsode integration process.

The caller is responsible for deleting the created object.

Returns

Integration controls object

Definition at line 76 of file `lsode_integrator_constructor.cc`.

6.5.2.15 create_second_order_ode_integrator()

```
er7_utils::SecondOrderODEIntegrator * LsodeIntegratorConstructor::create_second_order_ode_↵
integrator (
    unsigned int size,
    er7_utils::IntegrationControls & controls ) const [override]
```

Create an Lsode state integrator for a simple second order ODE.

The caller is responsible for deleting the created object.

Returns

State integrator

Parameters

<i>in</i>	<i>size</i>	State size
<i>in, out</i>	<i>controls</i>	Integration controls

Definition at line 92 of file `lsode_integrator_constructor.cc`.

References `jeod::LsodeIntegratorConstructor::data_interface`.

6.5.2.16 `destroy_allocated_arrays()` [1/2]

```
void LsodeDataArrays::destroy_allocated_arrays ( )
```

Allows for refactoring and reallocation of newly sized arrays.

Definition at line 148 of file `lsode_data_classes.cc`.

References `jeod::LsodeDataArrays::accum_correction`, `jeod::LsodeDataArrays::allocated`, `jeod::LsodeDataArrays::error_weight`, `jeod::LsodeDataArrays::history`, `jeod::LsodeDataArrays::lin_alg`, `jeod::LsodeDataArrays::lin_alg_index1`, `jeod::LsodeDataArrays::num_odes`, `jeod::LsodeDataArrays::pivots`, and `jeod::LsodeDataArrays::save`.

Referenced by `jeod::LsodeDataArrays::~~LsodeDataArrays()`, and `jeod::LsodeFirstOrderODEIntegrator::~~LsodeFirstOrderODEIntegrator()`.

6.5.2.17 `destroy_allocated_arrays()` [2/2]

```
void LsodeControlDataInterface::destroy_allocated_arrays ( )
```

De-allocates allocated array.

Definition at line 312 of file `lsode_control_data_interface.cc`.

References `jeod::LsodeControlDataInterface::abs_tolerance_error_control`, `jeod::LsodeControlDataInterface::error_control_vector_copied_over`, and `jeod::LsodeControlDataInterface::rel_tolerance_error_control`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::~~LsodeFirstOrderODEIntegrator()`.

6.5.2.18 `gauss_elim_factor()`

```
int LsodeFirstOrderODEIntegrator::gauss_elim_factor ( ) [protected]
```

Factors a double array (`arrays.lin_alg`) by Gaussian elimination.

Modified version of DGEFA.

Definition at line 175 of file `lsode_first_order_ode_integrator_utility.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::index_of_max_magnitude()`, `jeod::LsodeDataArrays::lin_alg`, `jeod::LsodeControlDataInterface::num_odes`, and `jeod::LsodeDataArrays::pivots`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_wrap_up()`.

6.5.2.19 `index_of_max_magnitude()`

```
unsigned int LsodeFirstOrderODEIntegrator::index_of_max_magnitude (
    unsigned int num_points,
    double ** array,
    int start_ix ) [protected]
```

Modified version of IDAMAX.

IDAMAX has 2 operations, one for situations in which the index increments by 1, and another for the converse. Since all instances in LSODE use the unit-increment method, that is the only one represented here. Search through matrix "array", starting at array[start_ix_1][start_ix_2] for the next "num_points" elements. The boolean search_ix_1 controls whether to increment index#1 (true), or index#2 (false). Method returns the searched index that corresponds to the largest magnitude.

Note

The only call to this method passed "k" in for both indices, so I stripped the second argument. If DGBFA gets implemented, it will have to be added back in; the call from DGBFA is for array starting at (M,K)

Definition at line 338 of file `lsode_first_order_ode_integrator__utility.cc`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::gauss_elim_factor()`.

6.5.2.20 `integrate()` [1/4]

```
unsigned int LsodeIntegrationControls::integrate (
    double start_time,
    double sim_dt,
    er7_utils::TimeInterface & time_interface,
    er7_utils::IntegratorInterface & integ_interface,
    er7_utils::BaseIntegrationGroup & integ_group ) [override]
```

Perform one step of the integration process.

Definition at line 69 of file `lsode_integration_controls.cc`.

6.5.2.21 `integrate()` [2/4]

```
er7_utils::IntegratorResult LsodeSimpleSecondOrderODEIntegrator::integrate (
    double dyn_dt,
    unsigned int target_stage,
    const double * accel,
    double * velocity,
    double * position ) [override]
```

Propagate state via Lsode's method.

Parameters

in	<i>dyn_dt</i>	Integration interval step, dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in	<i>accel</i>	Generalized acceleration vector.
in, out	<i>velocity</i>	Generalized velocity vector.
in, out	<i>position</i>	Generalized position vector.

Returns

The status (time advance, pass/fail status) of the integration.

Definition at line 67 of file `lsode_simple_second_order_ode_integrator.cc`.

References `jeod::LsodeSecondOrderODEIntegrator::first_order_integrator`, `jeod::LsodeFirstOrderODEIntegrator::integrate()`, `jeod::LsodeSecondOrderODEIntegrator::y`, `jeod::LsodeSecondOrderODEIntegrator::y_dot`, and `jeod::LsodeSecondOrderODEIntegrator::zeroth_derivative_size`.

6.5.2.22 integrate() [3/4]

```
er7_utils::IntegratorResult LsodeGeneralizedDerivSecondOrderODEIntegrator::integrate (
    double dyn_dt,
    unsigned int target_stage,
    const double * accel,
    double * velocity,
    double * position ) [override]
```

Propagate state via `Lsode`'s method.

Parameters

in	<i>dyn_dt</i>	Integration interval step, dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in	<i>accel</i>	Generalized acceleration vector.
in, out	<i>velocity</i>	Generalized velocity vector.
in, out	<i>position</i>	Generalized position vector.

Returns

The status (time advance, pass/fail status) of the integration.

Definition at line 101 of file `lsode_generalized_second_order_ode_integrator.cc`.

References `jeod::LsodeSecondOrderODEIntegrator::first_derivative_size`, `jeod::LsodeSecondOrderODEIntegrator::first_order_integrator`, `jeod::LsodeFirstOrderODEIntegrator::integrate()`, `jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::posdot`, `jeod::LsodeSecondOrderODEIntegrator::y`, `jeod::LsodeSecondOrderODEIntegrator::y_dot`, and `jeod::LsodeSecondOrderODEIntegrator::zeroth_derivative_size`.

6.5.2.23 `integrate()` [4/4]

```
er7_utils::IntegratorResult LsodeFirstOrderODEIntegrator::integrate (
    double dyn_dt,
    unsigned int target_stage,
    const double * y_dot,
    double * y ) [override]
```

Propagate state via Lsode's method.

Propagate state via the LSODE method.

Parameters

in	<i>dyn_dt</i>	Integration interval step, dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in, out	<i>y_dot</i>	Generalized velocity vector.
in, out	<i>y</i>	Generalized position vector.

Returns

The status (time advance, pass/fail status) of the integration.

Definition at line 60 of file `lsode_first_order_ode_integrator__manager.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::cycle_target_time`, `jeod::LsodeFirstOrderODEIntegrator::CycleStartFinish`, `jeod::LsodeFirstOrderODEIntegrator::DstodeResetStep`, `jeod::LsodeFirstOrderODEIntegrator::InitCalc`, `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part2()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_iteration_loop_part2()`, `jeod::LsodeFirstOrderODEIntegrator::IterationLoop`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init()`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop()`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_wrap_up()`, `jeod::LsodeFirstOrderODEIntegrator::JacobianPrep`, `jeod::LsodeFirstOrderODEIntegrator::load_derivatives()`, `jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part2()`, `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part2()`, `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part3()`, `jeod::LsodeFirstOrderODEIntegrator::process_entry_point_cycle_start()`, `jeod::LsodeFirstOrderODEIntegrator::re_entry_point`, `jeod::LsodeFirstOrderODEIntegrator::ResetIterLoop`, `jeod::LsodeDataArrays::save`, `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`, `jeod::LsodeFirstOrderODEIntegrator::step_error`, `jeod::LsodeFirstOrderODEIntegrator::update_jacobian`, `jeod::LsodeFirstOrderODEIntegrator::y`, and `jeod::LsodeFirstOrderODEIntegrator::y_dot`.

Referenced by `jeod::LsodeSimpleSecondOrderODEIntegrator::integrate()`, and `jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::integrate()`.

6.5.2.24 `integrator_compute_new_order()`

```
void LsodeFirstOrderODEIntegrator::integrator_compute_new_order ( ) [protected]
```

Computes the step-size scaling factors that will result once it is determined what happens to the order at the next step.

Definition at line 747 of file `lsode_first_order_ode_integrator__integrator.cc`.

References jeod::LsodeDataArrays::accum_correction, jeod::LsodeFirstOrderODEIntegrator::arrays, jeod::LsodeFirstOrderODEIntegrator::control_data, jeod::LsodeFirstOrderODEIntegrator::data_stode, jeod::LsodeDataStode::dsm, jeod::LsodeDataArrays::history, jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_check_step_error(), jeod::LsodeFirstOrderODEIntegrator::integrator_set_new_order(), jeod::LsodeFirstOrderODEIntegrator::integrator_wrapup(), jeod::LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array(), jeod::LsodeFirstOrderODEIntegrator::method_coeffs_current, jeod::LsodeFirstOrderODEIntegrator::method_order_current, jeod::LsodeDataStode::new_method_order, jeod::LsodeFirstOrderODEIntegrator::num_nordsiek_cols, jeod::LsodeControlDataInterface::num_odes, jeod::LsodeFirstOrderODEIntegrator::order_select_para, jeod::LsodeDataStode::step_ratio, jeod::LsodeDataStode::step_ratio_order_inc, and jeod::LsodeFirstOrderODEIntegrator::test_coeffs_complete.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_prep(), and jeod::LsodeFirstOrderODEIntegrator::integrator_error_test_failed().

6.5.2.25 integrator_compute_new_order_check_step_error()

```
void LsodeFirstOrderODEIntegrator::integrator_compute_new_order_check_step_error ( ) [protected]
```

Definition at line 817 of file lsode_first_order_ode_integrator__integrator.cc.

References jeod::LsodeFirstOrderODEIntegrator::data_stode, jeod::LsodeFirstOrderODEIntegrator::integrator_set_new_order(), jeod::LsodeFirstOrderODEIntegrator::integrator_wrapup(), jeod::LsodeFirstOrderODEIntegrator::order_select_para, jeod::LsodeFirstOrderODEIntegrator::step_error, and jeod::LsodeDataStode::step_ratio.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order().

6.5.2.26 integrator_compute_new_order_prep()

```
void LsodeFirstOrderODEIntegrator::integrator_compute_new_order_prep ( ) [protected]
```

The first steps in computing whether the order of the integrator should be changed.

Regardless of the success or failure of the step, the step-ratio factors for an increase, decrease, or retention of the integrator order are computed. In the case of failure, the increase ratio (data_stode.step_ratio_order_inc) has already been set to 0.0 to prevent an order increase. The largest of these factors is determined and the new order chosen accordingly. In the unusual case of equality, the priority is given to:

1. retain the order
2. increase the order (if inc = dec > same)

If the order is to be increased, we compute one additional scaled derivative.

This process is spread over four methods - integrator_compute_new_order_prep integrator_compute_new_order integrator_compute_new_order_check_step_error integrator_set_new_order

extracted from lines 520-540

Definition at line 725 of file lsode_first_order_ode_integrator__integrator.cc.

References jeod::LsodeDataArrays::accum_correction, jeod::LsodeFirstOrderODEIntegrator::arrays, jeod::LsodeFirstOrderODEIntegrator::control_data, jeod::LsodeFirstOrderODEIntegrator::data_stode, jeod::LsodeDataArrays::history, jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order(), jeod::LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array(), jeod::LsodeFirstOrderODEIntegrator::max_history_size, jeod::LsodeFirstOrderODEIntegrator::method_order_current, jeod::LsodeFirstOrderODEIntegrator::num_nordsiek_cols, jeod::LsodeControlDataInterface::num_odes, jeod::LsodeDataArrays::save, jeod::LsodeDataStode::step_ratio_order_inc, and jeod::LsodeFirstOrderODEIntegrator::test_coeffs_complete.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_converged().

6.5.2.27 integrator_core()

```
void LsodeFirstOrderODEIntegrator::integrator_core ( ) [protected]
```

integrator_core provides the front-end to all of the integrator_*i methods, which together perform one step of the integration of an initial value problem for a system of ordinary differential equations.

Modified from DSTODE

NOTES: The entire integrator_* suite is independent of the value of the iteration method indicator, corrector_ method, when said is != 0, and hence is independent of the type of chord method used, or the Jacobian structure.

The value internal_state (JSTART) controls the direction that this method takes.

By commenting out substantial parts of the package that are not useful to the ER7 / JEOD / Trick implementation, the only viable values for internal_state are now 0 or 1. internal_state = 0 : take the first step. internal_state = 1 : take another step, continuing from the last. internal_state = -1 was associated with externally driven changes to the input aparameters, something we do not allow. internal_state = -2 is associated with the critical / singularity time avoidance, something we have not implemented.

Definition at line 78 of file lsode_first_order_ode_integrator__integrator.cc.

References jeod::LsodeFirstOrderODEIntegrator::arrays, jeod::LsodeFirstOrderODEIntegrator::calculate_ integration_coefficients(), jeod::LsodeFirstOrderODEIntegrator::control_data, jeod::LsodeFirstOrderODE Integrator::convergence_factor, jeod::LsodeFirstOrderODEIntegrator::convergence_jacobian_flag, jeod::Lsode FirstOrderODEIntegrator::convergence_rate, jeod::LsodeFirstOrderODEIntegrator::data_stode, jeod::Lsode ControlDataInterface::integration_method, jeod::LsodeFirstOrderODEIntegrator::integrator_predict(), jeod::Lsode FirstOrderODEIntegrator::integrator_reset_method_coeffs(), jeod::LsodeFirstOrderODEIntegrator::integrator_ _reset_yh(), jeod::LsodeFirstOrderODEIntegrator::integrator_test_stepsize_change(), jeod::LsodeFirstOrder ODEIntegrator::internal_state, jeod::LsodeDataStode::iredo, jeod::LsodeDataStode::iret, jeod::LsodeControl DataInterface::is_corrector_method_functional_iteration(), jeod::LsodeFirstOrderODEIntegrator::iteration_ matrix_singular, jeod::LsodeFirstOrderODEIntegrator::jacobian_current, jeod::LsodeFirstOrderODEIntegrator_ ::magnitude_of_weighted_array(), jeod::LsodeFirstOrderODEIntegrator::max_history_size, jeod::LsodeFirst OrderODEIntegrator::max_order_internal, jeod::LsodeFirstOrderODEIntegrator::max_step_increase_ratio, jeod_ ::LsodeFirstOrderODEIntegrator::method_coeff_first, jeod::LsodeFirstOrderODEIntegrator::method_coeffs_ complete, jeod::LsodeFirstOrderODEIntegrator::method_coeffs_current, jeod::LsodeFirstOrderODEIntegrator_ ::method_order_current, jeod::LsodeControlDataInterface::min_step_size, jeod::LsodeFirstOrderODEIntegrator_ ::modified_iteration_matrix_singular, jeod::LsodeDataStode::ncf, jeod::LsodeFirstOrderODEIntegrator::num_ equations, jeod::LsodeFirstOrderODEIntegrator::num_nordsiek_cols, jeod::LsodeFirstOrderODEIntegrator_ ::num_predictor_elements, jeod::LsodeFirstOrderODEIntegrator::order_select_para, jeod::LsodeFirstOrderOD Integrator::prev_integration_method, jeod::LsodeFirstOrderODEIntegrator::prev_step_size, jeod::LsodeFirst OrderODEIntegrator::rel_change_since_jacobian, jeod::LsodeDataArrays::save, jeod::LsodeFirstOrderODE Integrator::stage_target_time, jeod::LsodeFirstOrderODEIntegrator::step_at_last_jacobian_update, jeod::Lsode FirstOrderODEIntegrator::step_error, jeod::LsodeDataStode::step_ratio, jeod::LsodeFirstOrderODEIntegrator_ ::step_size, jeod::LsodeFirstOrderODEIntegrator::test_coeffs_complete, jeod::LsodeDataStode::told, and jeod:: LsodeFirstOrderODEIntegrator::update_jacobian.

Referenced by jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part2().

6.5.2.28 integrator_corrector_converged()

```
void LsodeFirstOrderODEIntegrator::integrator_corrector_converged ( ) [protected]
```

Starts the processing of a converged iteration.

Definition at line 588 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeDataArrays::accum_correction`, `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::data_stode`, `jeod::LsodeDataStode::dsm`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_prep()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_error_test_failed()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_wrapup()`, `jeod::LsodeDataStode::iredo`, `jeod::LsodeFirstOrderODEIntegrator::iteration_delta`, `jeod::LsodeFirstOrderODEIntegrator::iteration_count`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_current`, `jeod::LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array()`, `jeod::LsodeFirstOrderODEIntegrator::max_history_size`, `jeod::LsodeFirstOrderODEIntegrator::method_coeffs_current`, `jeod::LsodeFirstOrderODEIntegrator::method_order_current`, `jeod::LsodeFirstOrderODEIntegrator::num_nordsiek_cols`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeFirstOrderODEIntegrator::num_steps_taken`, `jeod::LsodeFirstOrderODEIntegrator::order_select_para`, `jeod::LsodeFirstOrderODEIntegrator::prev_good_step_size`, `jeod::LsodeFirstOrderODEIntegrator::prev_method_order`, `jeod::LsodeFirstOrderODEIntegrator::step_error`, `jeod::LsodeFirstOrderODEIntegrator::step_size`, and `jeod::LsodeFirstOrderODEIntegrator::test_coeffs_complete`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration()`.

6.5.2.29 integrator_corrector_failed_part1()

```
void LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part1 ( ) [protected]
```

The corrector iteration failed to converge.

If `corrector_method != FunctionalIteration` and the Jacobian is out of date, exit so that the Jacobian method can be called (externally) for the next try. Otherwise, try changing the step-size in part 2 of the failure recovery.

Extracted from DSTODE lines 410-430

Definition at line 516 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::convergence_jacobian_flag`, `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_iteration_loop_part1()`, `jeod::LsodeControlDataInterface::is_corrector_method_functional_iteration()`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_current`, and `jeod::LsodeFirstOrderODEIntegrator::update_jacobian`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration()`.

6.5.2.30 integrator_corrector_failed_part2()

```
void LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2 ( ) [protected]
```

Retracts the history array in the case that the correction iteration failed to converge with either functional iteration or with an up-to-date Jacobian.

If there are problems, the associated flags are set.

Definition at line 542 of file `Lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::convergence_jacobian_flag`, `jeod::LsodeFirstOrderODEIntegrator::data_`↵
`stode`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_yh()`, `jeod::Lsode`↵
`FirstOrderODEIntegrator::integrator_terminate()`, `jeod::LsodeDataStode::iredo`, `jeod::LsodeControlDataInterface`↵
`::is_corrector_method_functional_iteration()`, `jeod::LsodeControlDataInterface::max_num_conv_failure`, `jeod::`↵
`LsodeFirstOrderODEIntegrator::max_step_increase_ratio`, `jeod::LsodeFirstOrderODEIntegrator::method_order`↵
`_current`, `jeod::LsodeControlDataInterface::min_step_size`, `jeod::LsodeDataStode::ncf`, `jeod::LsodeFirstOrderO`↵
`DEIntegrator::num_equations`, `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`, `jeod::LsodeFirstOrder`↵
`ODEIntegrator::step_error`, `jeod::LsodeDataStode::step_ratio`, `jeod::LsodeFirstOrderODEIntegrator::step_size`,
`jeod::LsodeDataStode::told`, and `jeod::LsodeFirstOrderODEIntegrator::update_jacobian`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part1()`, and `jeod::LsodeFirst`↵
`OrderODEIntegrator::jacobian_prep_wrap_up()`.

6.5.2.31 integrator_corrector_iteration()

```
void LsodeFirstOrderODEIntegrator::integrator_corrector_iteration ( ) [protected]
```

Keeps looping through the iterations until convergence or failure.

Definition at line 411 of file `Lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeDataArrays::accum_correction`, `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::`↵
`LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::convergence_factor`, `jeod`↵
`::LsodeFirstOrderODEIntegrator::convergence_rate`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrder`↵
`ODEIntegrator::integrator_corrector_converged()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_`↵
`failed_part1()`, `jeod::LsodeControlDataInterface::is_corrector_method_functional_iteration()`, `jeod::LsodeFirst`↵
`OrderODEIntegrator::iter_delta`, `jeod::LsodeFirstOrderODEIntegrator::iteration_count`, `jeod::LsodeFirstOrder`↵
`ODEIntegrator::iterationLoop`, `jeod::LsodeFirstOrderODEIntegrator::linear_chord_iteration()`, `jeod::LsodeFirst`↵
`OrderODEIntegrator::magnitude_of_weighted_array()`, `jeod::LsodeControlDataInterface::max_correction_iters`,
`jeod::LsodeFirstOrderODEIntegrator::method_coeffs_current`, `jeod::LsodeFirstOrderODEIntegrator::method_`↵
`order_current`, `jeod::LsodeFirstOrderODEIntegrator::modified_iteration_matrix_singular`, `jeod::LsodeControlData`↵
`Interface::num_odes`, `jeod::LsodeFirstOrderODEIntegrator::prev_iter_delta`, `jeod::LsodeFirstOrderODEIntegrator`↵
`::re_entry_point`, `jeod::LsodeDataArrays::save`, `jeod::LsodeFirstOrderODEIntegrator::step_size`, `jeod::LsodeFirst`↵
`OrderODEIntegrator::test_coeffs_complete`, and `jeod::LsodeFirstOrderODEIntegrator::y`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrate()`.

6.5.2.32 integrator_error_test_failed()

```
void LsodeFirstOrderODEIntegrator::integrator_error_test_failed ( ) [protected]
```

Restores the history array following the failure of the corrector for exceeding local error bounds.

Definition at line 663 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::data_stode`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part1()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_terminate()`, `jeod::LsodeDataStode::iredo`, `jeod::LsodeFirstOrderODEIntegrator::max_step_increase_ratio`, `jeod::LsodeFirstOrderODEIntegrator::method_order_current`, `jeod::LsodeControlDataInterface::min_step_size`, `jeod::LsodeFirstOrderODEIntegrator::num_equations`, `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`, `jeod::LsodeFirstOrderODEIntegrator::step_error`, `jeod::LsodeDataStode::step_ratio_order_inc`, `jeod::LsodeFirstOrderODEIntegrator::step_size`, and `jeod::LsodeDataStode::told`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_converged()`.

6.5.2.33 integrator_fail_reset_order_1_part1()

```
void LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part1 ( ) [protected]
```

Repopulates the y-array from history, recomputes the original first derivatives, sets the order back to 1, and reduces the step size by a factor of 10.

Called when 3 or more failures have occurred. It is assumed that the derivatives that have accumulated in the history array have errors of the wrong order. Hence the first derivative is recomputed, and the order is set to 1. Then the step-size is reduced by a factor of 10, and the step is retried. Repeat until successful, or the step reaches the minimum step-size.

If 10 failures occur, exit with `step_error = -1`.

This method is divided in two by a call to calculate the derivatives. Part1 precedes that call, the execution exits from the integrator back to the sim control engine; then on return to the integrator, execution immediately proceeds with part2.

extracted from lines 640-

Definition at line 878 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::data_stode`, `jeod::LsodeFirstOrderODEIntegrator::dstodeResetStep`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::integrator_terminate()`, `jeod::LsodeControlDataInterface::min_step_size`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeFirstOrderODEIntegrator::re_entry_point`, `jeod::LsodeFirstOrderODEIntegrator::step_error`, `jeod::LsodeDataStode::step_ratio`, `jeod::LsodeFirstOrderODEIntegrator::step_size`, and `jeod::LsodeFirstOrderODEIntegrator::y`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_error_test_failed()`.

6.5.2.34 integrator_fail_reset_order_1_part2()

```
void LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part2 ( ) [protected]
```

Continue reset, with derivatives now at hand.

See `integrator_fail_reset_order_1_part1` for details.

Definition at line 902 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::data_stode`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::integrator_predict()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_method_coeffs()`, `jeod::LsodeDataStode::iret`, `jeod::LsodeControlDataInterface::is_corrector_method_functional_iteration()`, `jeod::LsodeFirstOrderODEIntegrator::method_order_current`, `jeod::LsodeFirstOrderODEIntegrator::num_nordsiek_cols`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeFirstOrderODEIntegrator::order_select_para`, `jeod::LsodeDataArrays::save`, `jeod::LsodeFirstOrderODEIntegrator::step_size`, and `jeod::LsodeFirstOrderODEIntegrator::update_jacobian`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrate()`.

6.5.2.35 integrator_predict()

```
void LsodeFirstOrderODEIntegrator::integrator_predict ( ) [protected]
```

This section computes the predicted values by multiplying the history array by the Pascal Triangle matrix.

Extracted from DSTODE lines 200-215

Definition at line 343 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_iteration_loop_part1()`, `jeod::LsodeControlDataInterface::is_corrector_method_functional_iteration()`, `jeod::LsodeControlDataInterface::max_num_steps_jacobian`, `jeod::LsodeFirstOrderODEIntegrator::max_rel_change_without_jacobian`, `jeod::LsodeFirstOrderODEIntegrator::method_order_current`, `jeod::LsodeFirstOrderODEIntegrator::num_equations`, `jeod::LsodeFirstOrderODEIntegrator::num_steps_taken`, `jeod::LsodeFirstOrderODEIntegrator::rel_change_since_jacobian`, `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`, `jeod::LsodeFirstOrderODEIntegrator::step_at_last_jacobian_update`, `jeod::LsodeFirstOrderODEIntegrator::step_size`, and `jeod::LsodeFirstOrderODEIntegrator::update_jacobian`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_core()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part2()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_method_coeffs()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_yh()`, and `jeod::LsodeFirstOrderODEIntegrator::integrator_test_stepsize_change()`.

6.5.2.36 integrator_reset_iteration_loop_part1()

```
void LsodeFirstOrderODEIntegrator::integrator_reset_iteration_loop_part1 ( ) [protected]
```

This method resets the iteration loop to the values generated by the `integrator_predict` method, which populated `history[*][0]`.

Definition at line 378 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::iteration_count`, `jeod::LsodeControlData↵Interface::num_odes`, `jeod::LsodeFirstOrderODEIntegrator::re_entry_point`, `jeod::LsodeFirstOrderODEIntegrator↵::ResetIterLoop`, and `jeod::LsodeFirstOrderODEIntegrator::y`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part1()`, and `jeod::LsodeFirst↵OrderODEIntegrator::integrator_predict()`.

6.5.2.37 integrator_reset_iteration_loop_part2()

```
void LsodeFirstOrderODEIntegrator::integrator_reset_iteration_loop_part2 ( ) [protected]
```

This code follows part 1 after the break-out to get to the external calls, and completes the reset of the integration iteration loop.

Definition at line 395 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeDataArrays::accum_correction`, `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::Lsode↵FirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::iter_delta`, `jeod::LsodeControlData↵Interface::num_odes`, and `jeod::LsodeFirstOrderODEIntegrator::prev_iter_delta`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrate()`.

6.5.2.38 integrator_reset_method_coeffs()

```
void LsodeFirstOrderODEIntegrator::integrator_reset_method_coeffs ( ) [protected]
```

Sets/resets the `method_coeffs_current` array.

Definition at line 238 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::convergence↵_factor`, `jeod::LsodeFirstOrderODEIntegrator::data_stode`, `jeod::LsodeFirstOrderODEIntegrator::integrator↵predict()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_yh()`, `jeod::LsodeFirstOrderODEIntegrator↵::integrator_test_stepsize_change()`, `jeod::LsodeDataStode::iret`, `jeod::LsodeFirstOrderODEIntegrator::method↵coeff_first`, `jeod::LsodeFirstOrderODEIntegrator::method_coeffs_complete`, `jeod::LsodeFirstOrderODEIntegrator↵::method_coeffs_current`, `jeod::LsodeFirstOrderODEIntegrator::method_order_current`, `jeod::LsodeControlData↵Interface::min_step_size`, `jeod::LsodeFirstOrderODEIntegrator::num_equations`, `jeod::LsodeFirstOrderODE↵Integrator::num_nordsiek_cols`, `jeod::LsodeFirstOrderODEIntegrator::num_predictor_elements`, `jeod::LsodeFirst↵OrderODEIntegrator::rel_change_since_jacobian`, `jeod::LsodeDataStode::step_ratio`, and `jeod::LsodeFirstOrder↵ODEIntegrator::step_size`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_core()`, `jeod::LsodeFirstOrderODEIntegrator↵::integrator_fail_reset_order_1_part2()`, and `jeod::LsodeFirstOrderODEIntegrator::integrator_set_new_order()`.

6.5.2.39 integrator_reset_yh()

```
void LsodeFirstOrderODEIntegrator::integrator_reset_yh ( ) [protected]
```

Resets history arrays and time-step.

Definition at line 303 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::data_stode`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::integrator_predict()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_wrapup()`, `jeod::LsodeDataStode::iredo`, `jeod::LsodeFirstOrderODEIntegrator::max_step_increase_ratio`, `jeod::LsodeFirstOrderODEIntegrator::max_step_size_inv`, `jeod::LsodeFirstOrderODEIntegrator::num_nordsiek_cols`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeFirstOrderODEIntegrator::order_select_para`, `jeod::LsodeFirstOrderODEIntegrator::rel_change_since_jacobian`, `jeod::LsodeDataStode::step_ratio`, and `jeod::LsodeFirstOrderODEIntegrator::step_size`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_core()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_method_coeffs()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_set_new_order()`, and `jeod::LsodeFirstOrderODEIntegrator::integrator_test_stepsize_change()`.

6.5.2.40 integrator_set_new_order()

```
void LsodeFirstOrderODEIntegrator::integrator_set_new_order ( ) [protected]
```

Sets the new order and the step-ratio for the next step - or the current step if redoing it.

Definition at line 840 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::data_stode`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_method_coeffs()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_yh()`, `jeod::LsodeDataStode::iret`, `jeod::LsodeFirstOrderODEIntegrator::method_order_current`, `jeod::LsodeControlDataInterface::min_step_size`, `jeod::LsodeDataStode::new_method_order`, `jeod::LsodeFirstOrderODEIntegrator::num_nordsiek_cols`, `jeod::LsodeDataStode::step_ratio`, and `jeod::LsodeFirstOrderODEIntegrator::step_size`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order()`, and `jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_check_step_error()`.

6.5.2.41 integrator_terminate()

```
void LsodeFirstOrderODEIntegrator::integrator_terminate ( ) [protected]
```

this is the only succesful path back from integrator to manager.

All other returns from `integrator_*` back to `manager_*` are in response to a need for new derivatives and carry with them a modified `re_entry_point` to provide access back to the integrator on reentry. All returns with `re_entry_point = CycleStartFinish` should go through this method.

Definition at line 945 of file `lsode_first_order_ode_integrator__integrator.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::internal_state`, `jeod::LsodeFirstOrderODEIntegrator::prev_step_size`, and `jeod::LsodeFirstOrderODEIntegrator::step_size`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_error_test_failed()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part1()`, and `jeod::LsodeFirstOrderODEIntegrator::integrator_wrapup()`.

6.5.2.42 integrator_test_stepsize_change()

```
void LsodeFirstOrderODEIntegrator::integrator_test_stepsize_change ( ) [protected]
```

Tests h against old h.

Definition at line 280 of file lsode_first_order_ode_integrator__integrator.cc.

References jeod::LsodeFirstOrderODEIntegrator::data_stode, jeod::LsodeFirstOrderODEIntegrator::integrator_↵
predict(), jeod::LsodeFirstOrderODEIntegrator::integrator_reset_yh(), jeod::LsodeDataStode::iredo, jeod::Lsode↵
FirstOrderODEIntegrator::prev_step_size, jeod::LsodeDataStode::step_ratio, and jeod::LsodeFirstOrderODE↵
Integrator::step_size.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_core(), and jeod::LsodeFirstOrderODEIntegrator↵
::integrator_reset_method_coeffs().

6.5.2.43 integrator_wrapup()

```
void LsodeFirstOrderODEIntegrator::integrator_wrapup ( ) [protected]
```

Wraps up the completion of the integrator.

Definition at line 925 of file lsode_first_order_ode_integrator__integrator.cc.

References jeod::LsodeDataArrays::accum_correction, jeod::LsodeFirstOrderODEIntegrator::arrays, jeod::Lsode↵
FirstOrderODEIntegrator::control_data, jeod::LsodeFirstOrderODEIntegrator::integrator_terminate(), jeod::Lsode↵
ControlDataInterface::num_odes, jeod::LsodeFirstOrderODEIntegrator::prev_method_order, and jeod::Lsode↵
FirstOrderODEIntegrator::test_coeffs_complete.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order(), jeod::LsodeFirstOrder↵
ODEIntegrator::integrator_compute_new_order_check_step_error(), jeod::LsodeFirstOrderODEIntegrator↵
::integrator_corrector_converged(), and jeod::LsodeFirstOrderODEIntegrator::integrator_reset_yh().

6.5.2.44 interpolate_y()

```
void LsodeFirstOrderODEIntegrator::interpolate_y ( ) [protected]
```

Interpolates the zeroth-derivative of y Adapted from subroutine DINTDY, which was a general method for interpolat-
ing the K-th derivative of the dependent variable vector, y.

Implementation notes - DINTDY was called exclusively with three input values that always matched.

1. Argument #1 T (time). Calls to DINTDY always passed TOUT (now cycle_target_time) in for T
2. Argument #2 K (order). Calls to DINTDY always passed 0 in for K
3. Argument #5 DKY (value). DKY is the value that DINTDY interpolates. i Calls to DINTDY always passed Y in
for DKY. This routine uses y directly in place of DKY, so can only interpolate the 0-th derivative (since that is
y), and always evaluates at cycle_target_time. replaced accordingly.

The computed values are gotten by interpolation using the Nordsieck history array, `arrays.history`. The formula for Y is:

$$Y[i] = \sum_{j=0}^{\text{method_order_current}} \{ (\text{cycle_target_time} - \text{stage_target_time})^j * \text{arrays.history}[i-1][j] / h^j \}$$

Definition at line 263 of file `lsode_first_order_ode_integrator__support.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::cycle_target_time`, `jeod::LsodeFirstOrderODEIntegrator::epsilon`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::method_order_current`, `jeod::LsodeFirstOrderODEIntegrator::num_nordsiek_cols`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeFirstOrderODEIntegrator::prev_good_step_size`, `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`, `jeod::LsodeFirstOrderODEIntegrator::step_size`, and `jeod::LsodeFirstOrderODEIntegrator::y`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::manager_check_stop_conditions()`, and `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part3()`.

6.5.2.45 jacobian_prep_init()

```
void LsodeFirstOrderODEIntegrator::jacobian_prep_init ( ) [protected]
```

Modified from DPREPJ.

DPREPJ was called by DSTODE to compute and process the matrix $P = I - h * el(1) * J$, where J is an approximation to the Jacobian.

NOTES

DPREPJ has been split into 3 parts, book-ended by the external-calls. `jacobian_prep_init` contains the code that precedes the first external-call `jacobian_prep_loop` contains the code that continues to loop according to the limits as written is DPREPJ. `jacobian_prep_wrap_up` contains the code that follows successful completion of the looping section of DPREPJ.

Note that the division in this implementation is not linear with that in the original Fortran. The external calls within the original fortran are embedded within switch-blocks and for loops; the return points - to go to the next routine will pick up from one of several locations in the Fortran code, depending on the configuration at the time the external call was made.

For the ER7_Utils / JEOD / Trick implementation, the derivative/jac calls must be external to the integrate call, so we must fully back out and then reenter. Rentry goes to `jacobian_prep_loop`.

Some variables that were local have been moved to the class so that their value is not lost in going from `jacobian_prep_init` to `jacobian_prep_loop`. These are typically identified with `***_dprepj` to indicate that their sole purpose is within `dprepj` (the original name of the `jacobian_prep_*` routines).

Here the jacobian is computed by the user-supplied routine JAC if `corrector_method = NewtonIterUserJac` or `NewtonIterUserBandJac`, or by finite differencing if `corrector_method = NewtonIterInternalJac`, `JacobiNewtonInternalJac`, or `NewtonIterInternalBandJac`.

If `corrector_method = JacobiNewtonInternalJac`, a diagonal approximation to the Jacobian is used. The Jacobian is stored in `arrays.lin_alg`

If `corrector_method != JacobiNewtonInternalJac`, P is subjected to LU decomposition in preparation for later solution of linear systems with P as coefficient matrix. This is done by `gauss_elim_factor` (DGEFA) if `corrector_method =`

NewtonIterUserJac or NewtonIterInternalJac, and by linear_solver (DGBFA) if corrector_method = NewtonIterUserBandJac or NewtonIterInternalBandJac.

Note that the corrector_method using the banded Jacobians is not supported in this release, so linear_solver is not used.

FTEM and ACOR were effectively the same, now arrays.accum_correction. SAVF is now arrays.save. WM is now arrays.lin_alg

Definition at line 374 of file lsode_first_order_ode_integrator__support.cc.

References jeod::LsodeFirstOrderODEIntegrator::arrays, jeod::LsodeFirstOrderODEIntegrator::control_data, jeod::LsodeControlDataInterface::corrector_method, jeod::LsodeFirstOrderODEIntegrator::data_prepj, jeod::LsodeFirstOrderODEIntegrator::epsilon, jeod::LsodeDataArrays::error_weight, jeod::LsodeDataJacobiPrep::fac, jeod::LsodeControlDataInterface::FunctionalIteration, jeod::LsodeDataArrays::history, jeod::LsodeDataJacobiPrep::hl0, jeod::LsodeDataJacobiPrep::index, jeod::LsodeDataJacobiPrep::index_max, jeod::LsodeFirstOrderODEIntegrator::iteration_matrix_singular, jeod::LsodeFirstOrderODEIntegrator::jacobian_current, jeod::LsodeFirstOrderODEIntegrator::JacobiNewtonInternalJac, jeod::LsodeDataArrays::lin_alg, jeod::LsodeDataArrays::lin_alg_1, jeod::LsodeDataArrays::lin_alg_2, jeod::LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array(), jeod::LsodeFirstOrderODEIntegrator::method_coeff_first, jeod::LsodeControlDataInterface::NewtonIterInternalBandJac, jeod::LsodeControlDataInterface::NewtonIterInternalJac, jeod::LsodeControlDataInterface::NewtonIterUserBandJac, jeod::LsodeControlDataInterface::NewtonIterUserJac, jeod::LsodeFirstOrderODEIntegrator::num_jacobian_evals, jeod::LsodeControlDataInterface::num_odes, jeod::LsodeDataJacobiPrep::r0, jeod::LsodeFirstOrderODEIntegrator::re_entry_point, jeod::LsodeDataArrays::save, jeod::LsodeFirstOrderODEIntegrator::step_size, jeod::LsodeFirstOrderODEIntegrator::y, and jeod::LsodeDataJacobiPrep::yj.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrate().

6.5.2.46 jacobian_prep_loop()

```
bool LsodeFirstOrderODEIntegrator::jacobian_prep_loop ( ) [protected]
```

Definition at line 515 of file lsode_first_order_ode_integrator__support.cc.

References jeod::LsodeDataArrays::accum_correction, jeod::LsodeFirstOrderODEIntegrator::arrays, jeod::LsodeFirstOrderODEIntegrator::control_data, jeod::LsodeControlDataInterface::corrector_method, jeod::LsodeFirstOrderODEIntegrator::data_prepj, jeod::LsodeDataArrays::error_weight, jeod::LsodeDataJacobiPrep::fac, jeod::LsodeControlDataInterface::FunctionalIteration, jeod::LsodeDataJacobiPrep::hl0, jeod::LsodeDataJacobiPrep::index, jeod::LsodeDataJacobiPrep::index_max, jeod::LsodeControlDataInterface::JacobiNewtonInternalJac, jeod::LsodeDataArrays::lin_alg, jeod::LsodeDataArrays::lin_alg_1, jeod::LsodeFirstOrderODEIntegrator::load_derivatives(), jeod::LsodeControlDataInterface::NewtonIterInternalBandJac, jeod::LsodeControlDataInterface::NewtonIterInternalJac, jeod::LsodeControlDataInterface::NewtonIterUserBandJac, jeod::LsodeControlDataInterface::NewtonIterUserJac, jeod::LsodeControlDataInterface::num_odes, jeod::LsodeDataJacobiPrep::r0, jeod::LsodeDataArrays::save, jeod::LsodeFirstOrderODEIntegrator::y, and jeod::LsodeDataJacobiPrep::yj.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrate().

6.5.2.47 jacobian_prep_wrap_up()

```
bool LsodeFirstOrderODEIntegrator::jacobian_prep_wrap_up ( ) [protected]
```

Definition at line 612 of file lsode_first_order_ode_integrator__support.cc.

References jeod::LsodeFirstOrderODEIntegrator::arrays, jeod::LsodeFirstOrderODEIntegrator::control_data, jeod::LsodeFirstOrderODEIntegrator::convergence_rate, jeod::LsodeControlDataInterface::corrector_method, jeod::LsodeFirstOrderODEIntegrator::data_prepj, jeod::LsodeFirstOrderODEIntegrator::epsilon, jeod::LsodeDataArrays::error_weight, jeod::LsodeControlDataInterface::FunctionalIteration, jeod::LsodeFirstOrderODEIntegrator::gauss_elim_factor(), jeod::LsodeDataArrays::history, jeod::LsodeDataJacobiPrep::hl0, jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2(), jeod::LsodeFirstOrderODEIntegrator::iteration_matrix_singular, jeod::LsodeControlDataInterface::JacobiNewtonInternalJac, jeod::LsodeDataArrays::lin_alg, jeod::LsodeFirstOrderODEIntegrator::load_derivatives(), jeod::LsodeControlDataInterface::NewtonIterInternalBandJac, jeod::LsodeControlDataInterface::NewtonIterInternalJac, jeod::LsodeControlDataInterface::NewtonIterUserBandJac, jeod::LsodeControlDataInterface::NewtonIterUserJac, jeod::LsodeControlDataInterface::num_odes, jeod::LsodeFirstOrderODEIntegrator::num_steps_taken, jeod::LsodeFirstOrderODEIntegrator::rel_change_since_jacobian, jeod::LsodeDataArrays::save, jeod::LsodeFirstOrderODEIntegrator::step_at_last_jacobian_update, jeod::LsodeFirstOrderODEIntegrator::step_size, and jeod::LsodeFirstOrderODEIntegrator::update_jacobian.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrate().

6.5.2.48 linear_chord_iteration()

```
void LsodeFirstOrderODEIntegrator::linear_chord_iteration ( ) [protected]
```

Modified from DSOLSY.

This routine manages the solution of the linear system arising from a chord iteration. It is called if corrector_method != FunctionalIteration.

If corrector_method == NewtonIterUserJac || NewtonIterInternalJac, it calls linear_solver (was DGESE). If corrector_method = JacobiNewtonInternalJac it updates the coefficient hl0 = step_size * method_coeff_first (previously H*EL0) in the diagonal matrix, and then computes the solution.

Definition at line 736 of file lsode_first_order_ode_integrator__support.cc.

References jeod::LsodeFirstOrderODEIntegrator::arrays, jeod::LsodeFirstOrderODEIntegrator::control_data, jeod::LsodeControlDataInterface::corrector_method, jeod::LsodeControlDataInterface::FunctionalIteration, jeod::LsodeControlDataInterface::JacobiNewtonInternalJac, jeod::LsodeDataArrays::lin_alg, jeod::LsodeDataArrays::lin_alg_2, jeod::LsodeFirstOrderODEIntegrator::linear_solver(), jeod::LsodeFirstOrderODEIntegrator::method_coeff_first, jeod::LsodeFirstOrderODEIntegrator::modified_iteration_matrix_singular, jeod::LsodeControlDataInterface::NewtonIterInternalBandJac, jeod::LsodeControlDataInterface::NewtonIterInternalJac, jeod::LsodeControlDataInterface::NewtonIterUserBandJac, jeod::LsodeControlDataInterface::NewtonIterUserJac, jeod::LsodeControlDataInterface::num_odes, jeod::LsodeFirstOrderODEIntegrator::step_size, and jeod::LsodeFirstOrderODEIntegrator::y.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration().

6.5.2.49 linear_solver()

```
void LsodeFirstOrderODEIntegrator::linear_solver ( ) [protected]
```

Solves the equation $Y' = A Y$, with $A = \text{arrays.lin_alg}$.

Definition at line 254 of file `lsode_first_order_ode_integrator__utility.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeDataArrays::lin_alg`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeDataArrays::pivots`, and `jeod::LsodeFirstOrderODEIntegrator::y`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::linear_chord_iteration()`.

6.5.2.50 load_derivatives()

```
void LsodeFirstOrderODEIntegrator::load_derivatives (
    double * derivs ) [protected]
```

Load the externally generated derivative values (incoming as `y_dot`)_i into the array `derivs`.

Definition at line 361 of file `lsode_first_order_ode_integrator__utility.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeControlDataInterface::num_odes`, and `jeod::LsodeFirstOrderODEIntegrator::y_dot`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrate()`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_↔
prep_loop()`, and `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_wrap_up()`.

6.5.2.51 load_ew_values()

```
void LsodeFirstOrderODEIntegrator::load_ew_values ( ) [protected]
```

Definition at line 800 of file `lsode_first_order_ode_integrator__support.cc`.

References `jeod::LsodeControlDataInterface::abs_tolerance_error_control`, `jeod::LsodeFirstOrderODEIntegrator↔
::arrays`, `jeod::LsodeControlDataInterface::CommonAbsCommonRel`, `jeod::LsodeControlDataInterface::Common↔
AbsSpecificRel`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeControlDataInterface::error_↔
control_indicator`, `jeod::LsodeDataArrays::error_weight`, `jeod::LsodeDataArrays::history`, `jeod::LsodeControlData↔
Interface::num_odes`, `jeod::LsodeControlDataInterface::rel_tolerance_error_control`, `jeod::LsodeControlData↔
Interface::SpecificAbsCommonRel`, and `jeod::LsodeControlDataInterface::SpecificAbsSpecificRel`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part2()`, and `jeod::Lsode↔
FirstOrderODEIntegrator::manager_integration_loop_part1()`.

6.5.2.52 LsodeControlDataInterface()

```
LsodeControlDataInterface::LsodeControlDataInterface ( )
```

constructor

Definition at line 50 of file `lsode_control_data_interface.cc`.

References `jeod::LsodeControlDataInterface::abs_tolerance_error_control_vec`, and `jeod::LsodeControlDataInterface::rel_tolerance_error_control_vec`.

6.5.2.53 LsodeFirstOrderODEIntegrator() [1/2]

```
LsodeFirstOrderODEIntegrator::LsodeFirstOrderODEIntegrator ( )
```

[LsodeFirstOrderODEIntegrator](#) default constructor.

Definition at line 52 of file `lsode_first_order_ode_integrator__utility.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::calculate_epsilon()`.

6.5.2.54 LsodeFirstOrderODEIntegrator() [2/2]

```
LsodeFirstOrderODEIntegrator::LsodeFirstOrderODEIntegrator (
    const LsodeControlDataInterface & data_in,
    er7_utils::IntegrationControls & controls,
    unsigned int size )
```

[LsodeFirstOrderODEIntegrator](#) non-default constructor.

Parameters

<code>in</code>	<code>data_in</code>	state variable data
<code>in</code>	<code>size</code>	State size
<code>in, out</code>	<code>controls</code>	Integration controls

Definition at line 59 of file `lsode_first_order_ode_integrator__utility.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::calculate_epsilon()`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeControlDataInterface::num_odes`, and `jeod::LsodeFirstOrderODEIntegrator::update_control_data()`.

6.5.2.55 LsodeGeneralizedDerivSecondOrderODEIntegrator() [1/2]

```
LsodeGeneralizedDerivSecondOrderODEIntegrator::LsodeGeneralizedDerivSecondOrderODEIntegrator (
    const LsodeGeneralizedDerivSecondOrderODEIntegrator & src )
```

[LsodeGeneralizedDerivSecondOrderODEIntegrator](#) copy constructor.

Copy Constructor.

Parameters

in	src	Item to be copied.
----	-----	--------------------

Definition at line 67 of file `lsode_generalized_second_order_ode_integrator.cc`.

6.5.2.56 LsodeGeneralizedDerivSecondOrderODEIntegrator() [2/2]

```
LsodeGeneralizedDerivSecondOrderODEIntegrator::LsodeGeneralizedDerivSecondOrderODEIntegrator (
    const LsodeControlDataInterface & data_in,
    er7_utils::IntegrationControls & controls,
    const er7_utils::GeneralizedPositionDerivativeFunctions & deriv_funs,
    unsigned int position_size,
    unsigned int velocity_size )
```

non-default constructor

Definition at line 53 of file `lsode_generalized_second_order_ode_integrator.cc`.

References `jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::posdot`.

6.5.2.57 LsodeIntegrationControls()

```
LsodeIntegrationControls::LsodeIntegrationControls (
    unsigned int num_stages ) [explicit]
```

Definition at line 51 of file `lsode_integration_controls.cc`.

6.5.2.58 LsodeIntegratorConstructor()

```
LsodeIntegratorConstructor::LsodeIntegratorConstructor (
    const LsodeIntegratorConstructor & src )
```

Definition at line 57 of file `lsode_integrator_constructor.cc`.

6.5.2.59 LsodeSecondOrderODEIntegrator() [1/2]

```
LsodeSecondOrderODEIntegrator::LsodeSecondOrderODEIntegrator (
    const LsodeControlDataInterface & data_in,
    er7_utils::IntegrationControls & controls,
    unsigned int size ) [protected]
```

[LsodeSecondOrderODEIntegrator](#) non-default constructor.

Parameters

in	<i>data_in</i>	LSODE-specific control data.
in, out	<i>controls</i>	Integration controls.
in	<i>size</i>	State size.

Definition at line 52 of file `lsode_second_order_ode_integrator.cc`.

References `jeod::LsodeSecondOrderODEIntegrator::arrays_allocated`, `jeod::LsodeSecondOrderODEIntegrator::y`, and `jeod::LsodeSecondOrderODEIntegrator::y_dot`.

6.5.2.60 `LsodeSecondOrderODEIntegrator()` [2/2]

```
LsodeSecondOrderODEIntegrator::LsodeSecondOrderODEIntegrator (
    const LsodeControlDataInterface & data_in,
    er7_utils::IntegrationControls & controls,
    const er7_utils::GeneralizedPositionDerivativeFunctions & deriv_funs,
    unsigned int position_size,
    unsigned int velocity_size ) [protected]
```

Definition at line 68 of file `lsode_second_order_ode_integrator.cc`.

References `jeod::LsodeSecondOrderODEIntegrator::arrays_allocated`, `jeod::LsodeSecondOrderODEIntegrator::y`, and `jeod::LsodeSecondOrderODEIntegrator::y_dot`.

6.5.2.61 `LsodeSimpleSecondOrderODEIntegrator()`

```
LsodeSimpleSecondOrderODEIntegrator::LsodeSimpleSecondOrderODEIntegrator (
    const LsodeControlDataInterface & data_in,
    er7_utils::IntegrationControls & controls,
    unsigned int size )
```

[LsodeSimpleSecondOrderODEIntegrator](#) non-default constructor.

Parameters

in	<i>data_in</i>	State Variable Data
in	<i>size</i>	State size
in, out	<i>controls</i>	Integration controls

Definition at line 49 of file `lsode_simple_second_order_ode_integrator.cc`.

6.5.2.62 `magnitude_of_weighted_array()` [1/2]

```
double LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array (
    const double * v ) [protected]
```

returns the RMS value of $\{V \text{ dot } W\}$, where V and W are N -vectors.

Modified version of DVNORM

The only places DVNORM is used, it is multiplying some array by the `error_weight` array (`arrays.error_weight`) across `control_data.num_odes` terms. These values are fixed for our application, and do not need to be passed in.

We provide two implementations, one for a one-dimensional array, and one for a two-dimensional array in which the first index is the variable.

Parameters

<code>v</code>	array
----------------	-------

Definition at line 141 of file `lsode_first_order_ode_integrator__utility.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeDataArrays::error_weight`, and `jeod::LsodeControlDataInterface::num_odes`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_prep()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_core()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_converged()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration()`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init()`, `jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part2()`, and `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part2()`.

6.5.2.63 `magnitude_of_weighted_array()` [2/2]

```
double LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array (
    unsigned int index,
    double ** v ) [protected]
```

returns RMS value of $v[*][\text{index}]$

Modified version of DVNORM, second implementation.

Parameters

<i>index</i>	use this index
<code>v</code>	array

Definition at line 159 of file `lsode_first_order_ode_integrator__utility.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeDataArrays::error_weight`, and `jeod::LsodeControlDataInterface::num_odes`.

6.5.2.64 manager_check_stop_conditions()

```
int LsodeFirstOrderODEIntegrator::manager_check_stop_conditions ( ) [protected]
```

verifies whether the convergence conditions have been met to end the cycle.

Definition at line 528 of file `lsode_first_order_ode_integrator__manager.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::calculation_task`, `jeod::LsodeFirstOrderODEIntegrator::CompleteCycle`, `jeod::LsodeFirstOrderODEIntegrator::cycle_target_time`, `jeod::LsodeFirstOrderODEIntegrator::interpolate_y()`, `jeod::LsodeFirstOrderODEIntegrator::Normal`, `jeod::LsodeFirstOrderODEIntegrator::NormalWithSingularity`, `jeod::LsodeFirstOrderODEIntegrator::num_steps_taken`, `jeod::LsodeFirstOrderODEIntegrator::OneStep`, `jeod::LsodeFirstOrderODEIntegrator::OneStepWithSingularity`, `jeod::LsodeFirstOrderODEIntegrator::prior_num_steps`, `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`, and `jeod::LsodeFirstOrderODEIntegrator::step_size`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::process_entry_point_cycle_start()`.

6.5.2.65 manager_initialize_calculation_part1()

```
void LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part1 ( ) [protected]
```

Sets the values for the case with `calculation_phase = 1`.

Definition at line 342 of file `lsode_first_order_ode_integrator__manager.cc`.

References `jeod::LsodeDataArrays::allocate_arrays()`, `jeod::LsodeControlDataInterface::allocate_arrays()`, `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeControlDataInterface::corrector_method`, `jeod::LsodeFirstOrderODEIntegrator::InitCalc`, `jeod::LsodeFirstOrderODEIntegrator::internal_state`, `jeod::LsodeControlDataInterface::is_corrector_method_functional_iteration()`, `jeod::LsodeDataArrays::lin_alg_1`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeFirstOrderODEIntegrator::re_entry_point`, and `jeod::LsodeFirstOrderODEIntegrator::sqrt_epsilon`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::process_entry_point_cycle_start()`.

6.5.2.66 manager_initialize_calculation_part2()

```
void LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part2 ( ) [protected]
```

Definition at line 389 of file `lsode_first_order_ode_integrator__manager.cc`.

References `jeod::LsodeControlDataInterface::abs_tolerance_error_control`, `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeControlDataInterface::CommonAbsSpecificRel`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::cycle_target_time`, `jeod::LsodeFirstOrderODEIntegrator::epsilon`, `jeod::LsodeControlDataInterface::error_control_indicator`, `jeod::LsodeDataArrays::error_weight`, `jeod::LsodeDataArrays::history`, `jeod::LsodeControlDataInterface::initial_step_size`, `jeod::LsodeFirstOrderODEIntegrator::load_ew_values()`, `jeod::LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array()`, `jeod::LsodeFirstOrderODEIntegrator::max_step_size_inv`, `jeod::LsodeFirstOrderODEIntegrator::method_order_current`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeControlDataInterface::rel_tolerance_error_control`, `jeod::LsodeControlDataInterface::SpecificAbsCommonRel`, `jeod::LsodeControlDataInterface::SpecificAbsSpecificRel`, `jeod::LsodeFirstOrderODEIntegrator::step_size`, `jeod::LsodeFirstOrderODEIntegrator::y`, and `jeod::LsodeFirstOrderODEIntegrator::y_dot`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrate()`.

6.5.2.67 manager_integration_loop_part1()

```
void LsodeFirstOrderODEIntegrator::manager_integration_loop_part1 ( ) [protected]
```

The iteration loop for the integration process.

Definition at line 644 of file `lsode_first_order_ode_integrator__manager.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeDataArrays::error_weight`, `jeod::LsodeFirstOrderODEIntegrator::load_ew_values()`, `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part2()`, `jeod::LsodeControlDataInterface::max_num_steps`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeFirstOrderODEIntegrator::num_steps_taken`, `jeod::LsodeFirstOrderODEIntegrator::prior_num_steps`, and `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part3()`, and `jeod::LsodeFirstOrderODEIntegrator::process_entry_point_cycle_start()`.

6.5.2.68 manager_integration_loop_part2()

```
void LsodeFirstOrderODEIntegrator::manager_integration_loop_part2 ( ) [protected]
```

Definition at line 700 of file `lsode_first_order_ode_integrator__manager.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::CycleStartFinish`, `jeod::LsodeFirstOrderODEIntegrator::epsilon`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::integrator_core()`, `jeod::LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array()`, `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part3()`, `jeod::LsodeControlDataInterface::max_num_small_step_warnings`, `jeod::LsodeFirstOrderODEIntegrator::num_small_step_warnings`, `jeod::LsodeFirstOrderODEIntegrator::re_entry_point`, `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`, and `jeod::LsodeFirstOrderODEIntegrator::step_size`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrate()`, and `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part1()`.

6.5.2.69 manager_integration_loop_part3()

```
void LsodeFirstOrderODEIntegrator::manager_integration_loop_part3 ( ) [protected]
```

Definition at line 765 of file `lsode_first_order_ode_integrator__manager.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::calculation_task`, `jeod::LsodeFirstOrderODEIntegrator::CompleteCycle`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::cycle_target_time`, `jeod::LsodeFirstOrderODEIntegrator::initialized`, `jeod::LsodeFirstOrderODEIntegrator::interpolate_y()`, `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part1()`, `jeod::LsodeControlDataInterface::min_step_size`, `jeod::LsodeFirstOrderODEIntegrator::Normal`, `jeod::LsodeFirstOrderODEIntegrator::NormalWithSingularity`, `jeod::LsodeFirstOrderODEIntegrator::OneStep`, `jeod::LsodeFirstOrderODEIntegrator::OneStepWithSingularity`, `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`, `jeod::LsodeFirstOrderODEIntegrator::step_error`, and `jeod::LsodeFirstOrderODEIntegrator::step_size`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrate()`, and `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part2()`.

6.5.2.70 manager_set_calculation_phase_eq_2_reload()

```
void LsodeFirstOrderODEIntegrator::manager_set_calculation_phase_eq_2_reload ( ) [protected]
```

Definition at line 917 of file `lsode_first_order_ode_integrator__manager.cc`.

6.5.2.71 process_entry_point_cycle_start()

```
void LsodeFirstOrderODEIntegrator::process_entry_point_cycle_start ( ) [protected]
```

The code block from the main integrate routine for `re_entry_point=CycleStartFinish`.

Definition at line 269 of file `lsode_first_order_ode_integrator__manager.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::cycle_target_time`, `jeod::LsodeFirstOrderODEIntegrator::first_pass`, `jeod::LsodeControlDataInterface::initial_step_size`, `jeod::LsodeFirstOrderODEIntegrator::initialized`, `jeod::LsodeFirstOrderODEIntegrator::manager_check_stop_conditions()`, `jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part1()`, `jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part1()`, `jeod::LsodeFirstOrderODEIntegrator::num_equations`, `jeod::LsodeControlDataInterface::num_odes`, and `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrate()`.

6.5.2.72 reset_integrator()

```
void LsodeFirstOrderODEIntegrator::reset_integrator ( ) [override]
```

Resets the integrator when the timestep changes or when identified as needing a reset.

Definition at line 893 of file `lsode_first_order_ode_integrator__manager.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeFirstOrderODEIntegrator::cycle_target_time`, `jeod::LsodeFirstOrderODEIntegrator::first_pass`, `jeod::LsodeDataArrays::history`, `jeod::LsodeFirstOrderODEIntegrator::InitCalc`, `jeod::LsodeControlDataInterface::initial_step_size`, `jeod::LsodeFirstOrderODEIntegrator::method_order_current`, `jeod::LsodeControlDataInterface::num_odes`, `jeod::LsodeFirstOrderODEIntegrator::prev_good_step_size`, `jeod::LsodeFirstOrderODEIntegrator::re_entry_point`, and `jeod::LsodeFirstOrderODEIntegrator::stage_target_time`.

6.5.2.73 set_abs_tol()

```
void LsodeControlDataInterface::set_abs_tol (
    int index,
    double value )
```

Definition at line 382 of file `lsode_control_data_interface.cc`.

References `jeod::LsodeControlDataInterface::abs_tolerance_error_control`, `jeod::LsodeControlDataInterface::abs_tolerance_error_control_vec`, `jeod::LsodeControlDataInterface::error_control_vector_copied_over`, and `jeod::LsodeControlDataInterface::num_odes_at_alloc`.

6.5.2.74 set_rel_tol()

```
void LsodeControlDataInterface::set_rel_tol (
    int index,
    double value )
```

set values from external

Definition at line 325 of file `lsode_control_data_interface.cc`.

References `jeod::LsodeControlDataInterface::error_control_vector_copied_over`, `jeod::LsodeControlDataInterface::num_odes_at_alloc`, `jeod::LsodeControlDataInterface::rel_tolerance_error_control`, and `jeod::LsodeControlDataInterface::rel_tolerance_error_control_vec`.

6.5.2.75 update_control_data()

```
void LsodeFirstOrderODEIntegrator::update_control_data ( )
```

Gets the control data from where it can be populated in the constructor and verifies that the input control parameters are not out of sensible range.

Definition at line 88 of file `lsode_first_order_ode_integrator__utility.cc`.

References `jeod::LsodeControlDataInterface::check_interface_data()`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeControlDataInterface::ImplicitAdamsNonStiff`, `jeod::LsodeControlDataInterface::integration_method`, `jeod::LsodeControlDataInterface::max_order`, `jeod::LsodeFirstOrderODEIntegrator::max_order_internal`, `jeod::LsodeControlDataInterface::max_step_size`, and `jeod::LsodeFirstOrderODEIntegrator::max_step_size_inv`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::LsodeFirstOrderODEIntegrator()`.

6.5.2.76 ~LsodeFirstOrderODEIntegrator()

```
LsodeFirstOrderODEIntegrator::~LsodeFirstOrderODEIntegrator ( ) [override]
```

[LsodeFirstOrderODEIntegrator](#) destructor.

Definition at line 74 of file `lsode_first_order_ode_integrator__utility.cc`.

References `jeod::LsodeFirstOrderODEIntegrator::arrays`, `jeod::LsodeFirstOrderODEIntegrator::control_data`, `jeod::LsodeDataArrays::destroy_allocated_arrays()`, `jeod::LsodeControlDataInterface::destroy_allocated_arrays()`, and `jeod::LsodeFirstOrderODEIntegrator::first_pass`.

6.5.2.77 `~LsodeGeneralizedDerivSecondOrderODEIntegrator()`

```
LsodeGeneralizedDerivSecondOrderODEIntegrator::~~LsodeGeneralizedDerivSecondOrderODEIntegrator  
( ) [override]
```

[LsodeGeneralizedDerivSecondOrderODEIntegrator](#) destructor.

Destructor.

Definition at line 95 of file `lsode_generalized_second_order_ode_integrator.cc`.

References `jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::posdot`.

6.5.2.78 `~LsodeSecondOrderODEIntegrator()`

```
LsodeSecondOrderODEIntegrator::~~LsodeSecondOrderODEIntegrator ( ) [override]
```

[LsodeSecondOrderODEIntegrator](#) destructor.

Definition at line 86 of file `lsode_second_order_ode_integrator.cc`.

References `jeod::LsodeSecondOrderODEIntegrator::arrays_allocated`, `jeod::LsodeSecondOrderODEIntegrator::y`, and `jeod::LsodeSecondOrderODEIntegrator::y_dot`.

Chapter 7

Namespace Documentation

7.1 er7_utils Namespace Reference

Data Structures

- class [DoubleTwoDArray](#)
2D array, specialized for doubles.
- class [TwoDArray](#)
RAII template class that implements a rectangular two dimensional array.

7.2 jeod Namespace Reference

Namespace jeod.

Data Structures

- class [GaussJacksonCoefficientsPair](#)
Contains a summed Adams and Gauss-Jackson coefficient pair.
- class [GaussJacksonCoeffs](#)
Contains the Gauss-Jackson predictor and corrector coefficients.
- class [GaussJacksonConfig](#)
Contains Gauss-Jackson configuration data.
- class [GaussJacksonFirstOrderODEIntegrator](#)
Integrates a first order ODE using the summed Adams technique that is embedded within the Gauss-Jackson technique.
- class [GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#)
Integrates a generalized derivative second order ODE using Gauss-Jackson.
- class [GaussJacksonIntegrationControls](#)
IntegrationControls specialized for Gauss-Jackson integration.
- class [GaussJacksonIntegratorBase](#)
Base template class for integrating state via the Gauss-Jackson technique.
- class [GaussJacksonIntegratorConstructor](#)
Create state and time integrators that propagate using Gauss-Jackson.

- class [GaussJacksonOneState](#)
Essentially just a double.*
- class [GaussJacksonRationalCoefficients](#)
Contains a set of Adams or Stormer-Cowell coefficients.
- class [GaussJacksonSimpleSecondOrderODEIntegrator](#)
Integrates a simple second order ODE using the Gauss-Jackson technique.
- class [GaussJacksonStateMachine](#)
Guides the behavior of the Gauss-Jackson integration process via a finite state machine.
- class [GaussJacksonTwoState](#)
Essentially just std::pair<double>.*
- class [GeneralizedSecondOrderODETechnique](#)
Enumerates the integration schemes, generalized Cartesian or Lie group integration, and provides simple utilities that work with this.
- class [IntegrationMessages](#)
Declares messages associated with the integration test model.
- class [JeodIntegrationGroup](#)
A [JeodIntegrationGroup](#) integrates the state of a set of objects over time.
- class [JeodIntegrationGroupOwner](#)
The abstract class [IntegrationGroupOwner](#) contains an [IntegrationGroup](#).
- class [JeodIntegrationTime](#)
The class [JeodIntegrationTime](#) adds the concepts of timestamping and notification of changes in the nature of time to the ER7 numerical utilities [TimeInterface](#) class.
- class [LsodeControlDataInterface](#)
Specifies controls for an LSODE integrator.
- class [LsodeDataArrays](#)
The data arrays.
- class [LsodeDataJacobianPrep](#)
Data associated with the method DPREPJ.
- class [LsodeDataStode](#)
The data associated with method Dstode.
- class [LsodeFirstOrderODEIntegrator](#)
Jeod-compatible version of the Livermore ODE solver, LSODE.
- class [LsodeGeneralizedDerivSecondOrderODEIntegrator](#)
JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.
- class [LsodeIntegrationControls](#)
Contains controls for an LSODE integrator.
- class [LsodeIntegratorConstructor](#)
Create state and time integrators that propagate using standard Lsode.
- class [LsodeSecondOrderODEIntegrator](#)
JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.
- class [LsodeSimpleSecondOrderODEIntegrator](#)
JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.
- class [Restartable2DSecondOrderIntegrator](#)
Integrates a second order ODE in two dimensional space, $d^2x/dt^2 = a(x,t)$, where x is a two-vector.
- class [RestartableFirstOrderODEIntegrator](#)
A [RestartableFirstOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages an `er7_utils::FirstOrderODEIntegrator`.
- class [RestartableGeneralizedDerivSecondOrderODEIntegrator](#)
A [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a generalized second order ODE problem, one in which the time derivative of the generalized position is some function of the generalized position and the generalized velocity.
- class [RestartableGeneralizedStepSecondOrderODEIntegrator](#)

A [RestartableGeneralizedStepSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a generalized second order ODE problem, one in which the time derivative of the generalized position is some function of the generalized position and the generalized velocity.

- class [RestartableScalarFirstOrderODEIntegrator](#)

A [RestartableScalarFirstOrderODEIntegrator](#) integrates a first order ODE, $dx/dt = v(x,t)$, where x is a scalar.

- class [RestartableSecondOrderODEIntegrator](#)

A [RestartableSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a second order ODE problem.

- class [RestartableSimpleSecondOrderODEIntegrator](#)

A [RestartableSimpleSecondOrderODEIntegrator](#) is-a [RestartableSecondOrderODEIntegrator](#) that manages the integrator for a simple second order ODE problem, one in which the time derivative of the generalized position is the generalized velocity.

- class [RestartableSO3SecondOrderODEIntegrator](#)

A [RestartableSO3SecondOrderODEIntegrator](#) integrates a generalized second order ODE that describes rotation in three space.

- class [RestartableStateIntegrator](#)

A [RestartableStateIntegrator](#) establishes the basic capabilities needed to make a state integrator a managed resource.

- class [RestartableT3SecondOrderODEIntegrator](#)

A [RestartableT3SecondOrderODEIntegrator](#) integrates a second order ODE in three space, $d^2x/dt^2 = a(x,t)$, where x is a three-vector.

- class [TimeChangeSubscriber](#)

A [TimeChangeSubscriber](#) is some object that wants to be notified of changes in the nature of time.

Typedefs

- using [GaussJacksonIntegratorBaseFirst](#) = [GaussJacksonIntegratorBase](#)< [GaussJacksonOneState](#), [er7_utils::FirstOrderODEIntegrator](#) >

Alias for a first order Gauss Jackson integrator.

- using [GaussJacksonIntegratorBaseSecond](#) = [GaussJacksonIntegratorBase](#)< [GaussJacksonTwoState](#), [er7_utils::SecondOrderODEIntegrator](#) >

Alias for a second order Gauss Jackson integrator.

Functions

- [std::ostream & operator<<](#) ([std::ostream &stream](#), const [GaussJacksonCoeffs](#) &coeff)
- static [GaussJacksonConfig set_default_config_values](#) (const [GaussJacksonConfig](#) &config)
Swap the negative ones in the supplied config with the default values, some of which are computed.
- static unsigned int [validate_config](#) (const [GaussJacksonConfig](#) &config)
Check for invalid values in the supplied config.
- static [GaussJacksonIntegrationControls](#) * [cast_to_gj_controls](#) ([er7_utils::IntegrationControls](#) &controls)
Cast the provided integration controls to a [GaussJacksonIntegrationControls](#).

7.2.1 Detailed Description

Namespace jeod.

7.2.2 Typedef Documentation

7.2.2.1 GaussJacksonIntegratorBaseFirst

```
using jeod::GaussJacksonIntegratorBaseFirst = typedef GaussJacksonIntegratorBase<GaussJacksonOneState,
er7_utils::FirstOrderODEIntegrator>
```

Alias for a first order Gauss Jackson integrator.

Definition at line 81 of file gauss_jackson_integrator_base_first.hh.

7.2.2.2 GaussJacksonIntegratorBaseSecond

```
using jeod::GaussJacksonIntegratorBaseSecond = typedef GaussJacksonIntegratorBase<GaussJacksonTwoState,
er7_utils::SecondOrderODEIntegrator>
```

Alias for a second order Gauss Jackson integrator.

Definition at line 81 of file gauss_jackson_integrator_base_second.hh.

7.2.3 Function Documentation

7.2.3.1 cast_to_gj_controls()

```
static GaussJacksonIntegrationControls* jeod::cast_to_gj_controls (
    er7_utils::IntegrationControls & controls ) [static]
```

Cast the provided integration controls to a [GaussJacksonIntegrationControls](#).

Parameters

<i>controls</i>	Generic controls to be cast.
-----------------	------------------------------

Returns

[GaussJacksonIntegrationControls](#) pointer, guaranteed to be non-null.

Definition at line 47 of file gauss_jackson_integrator_constructor.cc.

Referenced by `jeod::GaussJacksonIntegratorConstructor::create_first_order_ode_integrator()`, `jeod::GaussJacksonIntegratorConstructor::create_generalized_deriv_second_order_ode_integrator()`, and `jeod::GaussJacksonIntegratorConstructor::create_second_order_ode_integrator()`.

7.2.3.2 operator<<()

```
std::ostream& jeod::operator<< (
    std::ostream & stream,
    const GaussJacksonCoeffs & coeff )
```

Parameters

<i>stream</i>	The stream to be printed to.
<i>coeff</i>	The coefficients to be printed.

Definition at line 130 of file gauss_jackson_coeffs.cc.

References jeod::GaussJacksonCoeffs::corrector, jeod::GaussJacksonCoeffs::order, jeod::GaussJacksonCoeffs↵
::predictor, and jeod::GaussJacksonCoefficientsPair::printOut().

7.2.3.3 set_default_config_values()

```
static GaussJacksonConfig jeod::set_default_config_values (
    const GaussJacksonConfig & config ) [static]
```

Swap the negative ones in the supplied config with the default values, some of which are computed.

Definition at line 75 of file gauss_jackson_config.cc.

References jeod::GaussJacksonConfig::absolute_tolerance, jeod::GaussJacksonConfig::final_order, jeod::Gauss↵
JacksonConfig::initial_order, jeod::GaussJacksonConfig::max_correction_iterations, jeod::GaussJacksonConfig↵
::ndoubling_steps, jeod::GaussJacksonConfig::priming_technique, and jeod::GaussJacksonConfig::relative_↵
tolerance.

Referenced by jeod::GaussJacksonConfig::validate_configuration().

7.2.3.4 validate_config()

```
static unsigned int jeod::validate_config (
    const GaussJacksonConfig & config ) [static]
```

Check for invalid values in the supplied config.

Definition at line 167 of file gauss_jackson_config.cc.

References jeod::GaussJacksonConfig::absolute_tolerance, jeod::GaussJacksonConfig::final_order, jeod::↵
GaussJacksonConfig::initial_order, jeod::GaussJacksonConfig::ndoubling_steps, and jeod::GaussJacksonConfig↵
::relative_tolerance.

Referenced by jeod::GaussJacksonConfig::validate_configuration().

Chapter 8

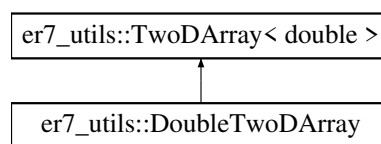
Data Structure Documentation

8.1 er7_utils::DoubleTwoDArray Class Reference

2D array, specialized for doubles.

```
#include <two_d_array.hh>
```

Inheritance diagram for er7_utils::DoubleTwoDArray:



Friends

- class [InputProcessor](#)
- void [init_attrer7_utils__DoubleTwoDArray](#) ()

Additional Inherited Members

8.1.1 Detailed Description

2D array, specialized for doubles.

Definition at line 403 of file two_d_array.hh.

8.1.2 Friends And Related Function Documentation

8.1.2.1 init_attrer7_utils__DoubleTwoDArray

```
void init_attrer7_utils__DoubleTwoDArray ( ) [friend]
```

8.1.2.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 405 of file two_d_array.hh.

The documentation for this class was generated from the following file:

- [two_d_array.hh](#)

8.2 jeod::GaussJacksonCoefficientsPair Class Reference

Contains a summed Adams and Gauss-Jackson coefficient pair.

```
#include <gauss_jackson_coefficients_pair.hh>
```

Public Member Functions

- [GaussJacksonCoefficientsPair](#) ()=default
Default constructor.
- [~GaussJacksonCoefficientsPair](#) ()
Destructor.
- void [configure](#) (int max_order)

Allocate (re-allocate) memory for the coefficients.
- void [swap](#) ([GaussJacksonCoefficientsPair](#) &other)
Non-throwing swap.
- void [allocate_arrays](#) (int size)
Allocate space for the coefficients.
- void [deallocate_arrays](#) ()
Release allocated memory.
- void [apply](#) (int nelelem, int ncoeff, const double *const *acc_hist, const [GaussJacksonTwoState](#) &state_sum) const
Apply both sets of coefficients to the supplied history data.
- void [apply](#) (int nelelem, int ncoeff, const double *const *acc_hist, const [GaussJacksonOneState](#) &state_sum) const
Apply just the Adams coefficients to the supplied history data.
- void [printOut](#) (int order, std::ostream &stream) const
Print the coefficients.
- [GaussJacksonCoefficientsPair](#) (const [GaussJacksonCoefficientsPair](#) &)=delete
- [GaussJacksonCoefficientsPair](#) & [operator=](#) (const [GaussJacksonCoefficientsPair](#) &)=delete

Data Fields

- double * [sa_coefs](#) {}
Summed Adams coefficients, in ordinate form.
- double * [gj_coefs](#) {}
Gauss Jackson coefficients, in ordinate form.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__GaussJacksonCoefficientsPair](#) ()

8.2.1 Detailed Description

Contains a summed Adams and Gauss-Jackson coefficient pair.

Definition at line 82 of file `gauss_jackson_coefficients_pair.hh`.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 GaussJacksonCoefficientsPair() [1/2]

```
jeod::GaussJacksonCoefficientsPair::GaussJacksonCoefficientsPair ( ) [default]
```

Default constructor.

8.2.2.2 ~GaussJacksonCoefficientsPair()

```
jeod::GaussJacksonCoefficientsPair::~~GaussJacksonCoefficientsPair ( ) [inline]
```

Destructor.

Definition at line 107 of file `gauss_jackson_coefficients_pair.hh`.

8.2.2.3 GaussJacksonCoefficientsPair() [2/2]

```
jeod::GaussJacksonCoefficientsPair::GaussJacksonCoefficientsPair (
    const GaussJacksonCoefficientsPair & ) [delete]
```

8.2.3 Member Function Documentation

8.2.3.1 allocate_arrays()

```
void jeod::GaussJacksonCoefficientsPair::allocate_arrays (
    int size )
```

Allocate space for the coefficients.

Parameters

<i>size</i>	Array size.
-------------	-------------

Definition at line 34 of file gauss_jackson_coefficients_pair.cc.

References `gj_coefs`, and `sa_coefs`.

8.2.3.2 `apply()` [1/2]

```
void jeod::GaussJacksonCoefficientsPair::apply (
    int nelem,
    int ncoeff,
    const double *const * acc_hist,
    const GaussJacksonTwoState & state_sum ) const
```

Apply both sets of coefficients to the supplied history data.

The first element of the output `state_sum` is calculated as the inner products of the acceleration history with the summed Adams coefficients. The second element is calculated as the inner product with the Gauss-Jackson coefficients. (First = first integral; second = second integral.)

Parameters

<i>nelem</i>	Dimensionality of each acceleration history element
<i>ncoeff</i>	Number of elements in the acceleration history
<i>acc_hist</i>	Acceleration history
<i>state_sum</i>	Output inner products

Definition at line 52 of file gauss_jackson_coefficients_pair.cc.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj()`.

8.2.3.3 `apply()` [2/2]

```
void jeod::GaussJacksonCoefficientsPair::apply (
    int nelem,
    int ncoeff,
    const double *const * acc_hist,
    const GaussJacksonOneState & state_sum ) const
```

Apply just the Adams coefficients to the supplied history data.

Parameters

<i>nelem</i>	Dimensionality of each acceleration history element
<i>ncoeff</i>	Number of elements in the acceleration history
<i>acc_hist</i>	Acceleration history
<i>state_sum</i>	Output inner products

Definition at line 84 of file gauss_jackson_coefficients_pair.cc.

8.2.3.4 configure()

```
void jeod::GaussJacksonCoefficientsPair::configure (
    int max_order ) [inline]
```

Allocate (re-allocate) memory for the coefficients.

Arrays are size & to contain max_order+1 elements.

Parameters

<i>max_order</i>	Maximum order that will be used.
------------------	----------------------------------

Definition at line 117 of file gauss_jackson_coefficients_pair.hh.

Referenced by jeod::GaussJacksonCoeffs::configure().

8.2.3.5 deallocate_arrays()

```
void jeod::GaussJacksonCoefficientsPair::deallocate_arrays ( )
```

Release allocated memory.

Definition at line 40 of file gauss_jackson_coefficients_pair.cc.

References `gj_coefs`, and `sa_coefs`.

8.2.3.6 operator=()

```
GaussJacksonCoefficientsPair& jeod::GaussJacksonCoefficientsPair::operator= (
    const GaussJacksonCoefficientsPair & ) [delete]
```

8.2.3.7 printOut()

```
void jeod::GaussJacksonCoefficientsPair::printOut (
    int order,
    std::ostream & stream ) const
```

Print the coefficients.

Parameters

<i>order</i>	Coefficients order
<i>stream</i>	Output stream

Definition at line 110 of file gauss_jackson_coefficients_pair.cc.

References `gj_coefs`, and `sa_coefs`.

Referenced by `jeod::operator<<()`.

8.2.3.8 swap()

```
void jeod::GaussJacksonCoefficientsPair::swap (
    GaussJacksonCoefficientsPair & other )
```

Non-throwing swap.

Parameters

<i>other</i>	Coeffs pair with which contents are to be swapped.
--------------	--

Definition at line 46 of file gauss_jackson_coefficients_pair.cc.

References `gj_coefs`, and `sa_coefs`.

Referenced by `jeod::GaussJacksonCoeffs::swap()`.

8.2.4 Friends And Related Function Documentation**8.2.4.1 init_attrjeod__GaussJacksonCoefficientsPair**

```
void init_attrjeod__GaussJacksonCoefficientsPair ( ) [friend]
```

8.2.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 84 of file gauss_jackson_coefficients_pair.hh.

8.2.5 Field Documentation

8.2.5.1 gj_coefs

```
double* jeod::GaussJacksonCoefficientsPair::gj_coefs {}
```

Gauss Jackson coefficients, in ordinate form.

trick_units(—)

Definition at line 95 of file gauss_jackson_coefficients_pair.hh.

Referenced by `allocate_arrays()`, `jeod::GaussJacksonCoeffs::compute_coefs()`, `deallocate_arrays()`, `printOut()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle()`, and `swap()`.

8.2.5.2 sa_coefs

```
double* jeod::GaussJacksonCoefficientsPair::sa_coefs {}
```

Summed Adams coefficients, in ordinate form.

trick_units(—)

Definition at line 90 of file gauss_jackson_coefficients_pair.hh.

Referenced by `allocate_arrays()`, `jeod::GaussJacksonCoeffs::compute_coefs()`, `deallocate_arrays()`, `printOut()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle()`, and `swap()`.

The documentation for this class was generated from the following files:

- [gauss_jackson_coefficients_pair.hh](#)
- [gauss_jackson_coefficients_pair.cc](#)

8.3 jeod::GaussJacksonCoeffs Class Reference

Contains the Gauss-Jackson predictor and corrector coefficients.

```
#include <gauss_jackson_coeffs.hh>
```

Public Member Functions

- [GaussJacksonCoeffs](#) ()=default
Default constructor.
- [GaussJacksonCoeffs](#) (const [GaussJacksonCoeffs](#) &src)
Copy constructor.
- [~GaussJacksonCoeffs](#) ()
Destructor.
- [GaussJacksonCoeffs](#) & [operator=](#) ([GaussJacksonCoeffs](#) src)
Copy-and-swap assignment operator.
- void [swap](#) ([GaussJacksonCoeffs](#) &src)
Non-throwing swap.
- void [configure](#) (unsigned int max_order_in)
Configure to enable coefficients up to the specified maximum order.
- void [compute_coeffs](#) (unsigned int order_in)
Compute coefficients for the specified order.

Data Fields

- [GaussJacksonCoefficientsPair](#) predictor
Summed Adams and Gauss-Jackson predictor coefficients.
- [GaussJacksonCoefficientsPair](#) * [corrector](#) {}
Summed Adams and Gauss-Jackson corrector coefficients.
- unsigned int [max_order](#) {}
Maximum order; used for sizing.
- unsigned int [order](#) {}
Current order; dictates the coefficient values.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__GaussJacksonCoeffs](#) ()
- std::ostream & [operator<<](#) (std::ostream &stream, const [GaussJacksonCoeffs](#) &coeff)
Print the coefficients.

8.3.1 Detailed Description

Contains the Gauss-Jackson predictor and corrector coefficients.

Definition at line 81 of file gauss_jackson_coeffs.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 GaussJacksonCoeffs() [1/2]

```
jeod::GaussJacksonCoeffs::GaussJacksonCoeffs ( ) [default]
```

Default constructor.

8.3.2.2 GaussJacksonCoeffs() [2/2]

```
jeod::GaussJacksonCoeffs::GaussJacksonCoeffs (
    const GaussJacksonCoeffs & src ) [inline]
```

Copy constructor.

Note that this doesn't copy; it recomputes. The end result is as if a copy had been made.

Parameters

<code>src</code>	Object to be copied.
------------------	----------------------

Definition at line 119 of file gauss_jackson_coeffs.hh.

8.3.2.3 ~GaussJacksonCoeffs()

```
jeod::GaussJacksonCoeffs::~~GaussJacksonCoeffs ( )
```

Destructor.

Definition at line 43 of file gauss_jackson_coeffs.cc.

References corrector.

8.3.3 Member Function Documentation

8.3.3.1 compute_coeffs()

```
void jeod::GaussJacksonCoeffs::compute_coeffs (
    unsigned int order_in )
```

Compute coefficients for the specified order.

Parameters

<i>order</i> ↔ _in	The current order.
-----------------------	--------------------

Definition at line 75 of file gauss_jackson_coeffs.cc.

References jeod::GaussJacksonRationalCoefficients::configure_adams_corrector(), jeod::GaussJacksonRationalCoefficients::construct_predictor(), jeod::GaussJacksonRationalCoefficients::construct_stormer_cowell_corrector(), jeod::GaussJacksonRationalCoefficients::convert_to_ordinate_form(), corrector, jeod::GaussJacksonRationalCoefficients::discard_extra_terms(), jeod::GaussJacksonRationalCoefficients::displace_back(), jeod::GaussJacksonCoefficientsPair::gj_coefs, max_order, order, predictor, and jeod::GaussJacksonCoefficientsPair::sa_coefs.

Referenced by jeod::GaussJacksonIntegrationControls::GaussJacksonIntegrationControls(), and jeod::GaussJacksonIntegrationControls::start_cycle().

8.3.3.2 configure()

```
void jeod::GaussJacksonCoeffs::configure (
    unsigned int max_order_in )
```

Configure to enable coefficients up to the specified maximum order.

Parameters

<i>max_order</i> ↔ _in	The maximum order to be used.
---------------------------	-------------------------------

Definition at line 58 of file gauss_jackson_coeffs.cc.

References jeod::GaussJacksonCoefficientsPair::configure(), corrector, max_order, order, and predictor.

Referenced by jeod::GaussJacksonIntegrationControls::GaussJacksonIntegrationControls().

8.3.3.3 operator=()

```
GaussJacksonCoeffs& jeod::GaussJacksonCoeffs::operator= (
    GaussJacksonCoeffs src ) [inline]
```

Copy-and-swap assignment operator.

Parameters

<i>src</i>	Object to be copied.
------------	----------------------

Definition at line 141 of file gauss_jackson_coeffs.hh.

8.3.3.4 swap()

```
void jeod::GaussJacksonCoeffs::swap (
    GaussJacksonCoeffs & src )
```

Non-throwing swap.

Parameters

<i>src</i>	Object to swap contents with.
------------	-------------------------------

Definition at line 49 of file gauss_jackson_coeffs.cc.

References corrector, max_order, order, predictor, and jeod::GaussJacksonCoefficientsPair::swap().

8.3.4 Friends And Related Function Documentation

8.3.4.1 init_attrjeod__GaussJacksonCoeffs

```
void init_attrjeod__GaussJacksonCoeffs ( ) [friend]
```

8.3.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 83 of file gauss_jackson_coeffs.hh.

8.3.4.3 operator<<

```
std::ostream& operator<< (
    std::ostream & stream,
    const GaussJacksonCoeffs & coeff ) [friend]
```

Print the coefficients.

Parameters

<i>stream</i>	The stream to be printed to.
<i>coeff</i>	The coefficients to be printed.

Definition at line 130 of file gauss_jackson_coeffs.cc.

8.3.5 Field Documentation**8.3.5.1 corrector**

```
GaussJacksonCoefficientsPair* jeod::GaussJacksonCoeffs::corrector {}
```

Summed Adams and Gauss-Jackson corrector coefficients.

trick_units(-)

Definition at line 94 of file gauss_jackson_coeffs.hh.

Referenced by compute_coeffs(), configure(), jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj(), jeod::operator<<(), jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle(), swap(), and ~GaussJacksonCoeffs().

8.3.5.2 max_order

```
unsigned int jeod::GaussJacksonCoeffs::max_order {}
```

Maximum order; used for sizing.

trick_units(-)

Definition at line 99 of file gauss_jackson_coeffs.hh.

Referenced by compute_coeffs(), configure(), and swap().

8.3.5.3 order

```
unsigned int jeod::GaussJacksonCoeffs::order {}
```

Current order; dictates the coefficient values.

trick_units(-)

Definition at line 104 of file gauss_jackson_coeffs.hh.

Referenced by compute_coeffs(), configure(), jeod::operator<<(), and swap().

8.3.5.4 predictor

`GaussJacksonCoefficientsPair` jeod::GaussJacksonCoeffs::predictor

Summed Adams and Gauss-Jackson predictor coefficients.

trick_units(-)

Definition at line 89 of file gauss_jackson_coeffs.hh.

Referenced by compute_coeffs(), configure(), jeod::operator<<(), and swap().

The documentation for this class was generated from the following files:

- [gauss_jackson_coeffs.hh](#)
- [gauss_jackson_coeffs.cc](#)

8.4 jeod::GaussJacksonConfig Class Reference

Contains Gauss-Jackson configuration data.

```
#include <gauss_jackson_config.hh>
```

Public Member Functions

- [GaussJacksonConfig](#) ()

Static Public Member Functions

- static [GaussJacksonConfig default_configuration](#) ()
Creates a [GaussJacksonConfig](#) with all members set to -1.
- static [GaussJacksonConfig standard_configuration](#) ()
Creates a [GaussJacksonConfig](#) with all members set to their defaults.
- static [GaussJacksonConfig validate_configuration](#) (const [GaussJacksonConfig](#) &config)
Creates a [GaussJacksonConfig](#) based on the supplied configuration.

Data Fields

- er7_utils::Integration::Technique [priming_technique](#) {er7_utils::Integration::Unspecified}
The integration technique to be used to prime the Gauss-Jackson process.
- unsigned int [initial_order](#) {4}
The order of the Gauss Jackson integrator immediately after priming.
- unsigned int [final_order](#) {12}
The order of the Gauss Jackson integrator once it's operational.
- unsigned int [ndoubling_steps](#)
The number of time doubling steps involved in the bootstrap operation.
- unsigned int [max_correction_iterations](#) {10}
Maximum number of correction steps allowed before the integrator is deemed to be not converging.
- double [relative_tolerance](#) {1E-14}
Number that indicates the allowable relative difference for two states to be considered converged.
- double [absolute_tolerance](#) {1E-10}
Number that indicates the allowable absolute difference for two states to be considered converged.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__GaussJacksonConfig](#) ()

8.4.1 Detailed Description

Contains Gauss-Jackson configuration data.

All member data are public; this is essentially a struct.

Definition at line 75 of file `gauss_jackson_config.hh`.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 GaussJacksonConfig()

```
jeod::GaussJacksonConfig::GaussJacksonConfig ( ) [inline]
```

Definition at line 78 of file `gauss_jackson_config.hh`.

8.4.3 Member Function Documentation

8.4.3.1 default_configuration()

```
GaussJacksonConfig jeod::GaussJacksonConfig::default_configuration ( ) [static]
```

Creates a [GaussJacksonConfig](#) with all members set to -1.

This otherwise invalid value has a special meaning to the validation function. When encountered, the item is silently replaced with the default for that item.

Definition at line 39 of file `gauss_jackson_config.cc`.

References `priming_technique`.

8.4.3.2 standard_configuration()

```
GaussJacksonConfig jeod::GaussJacksonConfig::standard_configuration ( ) [static]
```

Creates a [GaussJacksonConfig](#) with all members set to their defaults.

Definition at line 55 of file gauss_jackson_config.cc.

References [priming_technique](#).

Referenced by [jeod::GaussJacksonIntegratorConstructor::create_integration_controls\(\)](#), and [validate_configuration\(\)](#).

8.4.3.3 validate_configuration()

```
GaussJacksonConfig jeod::GaussJacksonConfig::validate_configuration (
    const GaussJacksonConfig & config ) [static]
```

Creates a [GaussJacksonConfig](#) based on the supplied configuration.

Values of -1 are replaced with their defaults. The standard configuration is used if any invalid item is invalid.

Definition at line 236 of file gauss_jackson_config.cc.

References [jeod::set_default_config_values\(\)](#), [standard_configuration\(\)](#), and [jeod::validate_config\(\)](#).

Referenced by [jeod::GaussJacksonIntegratorConstructor::configure\(\)](#).

8.4.4 Friends And Related Function Documentation

8.4.4.1 init_attrjeod__GaussJacksonConfig

```
void init_attrjeod__GaussJacksonConfig ( ) [friend]
```

8.4.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 77 of file gauss_jackson_config.hh.

8.4.5 Field Documentation

8.4.5.1 absolute_tolerance

```
double jeod::GaussJacksonConfig::absolute_tolerance {1E-10}
```

Number that indicates the allowable absolute difference for two states to be considered converged.

Defaults to 1e-10.trick_units(-)

Definition at line 157 of file gauss_jackson_config.hh.

Referenced by jeod::set_default_config_values(), and jeod::validate_config().

8.4.5.2 final_order

```
unsigned int jeod::GaussJacksonConfig::final_order {12}
```

The order of the Gauss Jackson integrator once it's operational.

This must be an even number between initial_order and 14, inclusive. Defaults to 12.trick_units(-)

Definition at line 126 of file gauss_jackson_config.hh.

Referenced by jeod::GaussJacksonStateMachine::configure(), jeod::GaussJacksonIntegrationControls::GaussJacksonIntegrationControls(), jeod::set_default_config_values(), and jeod::validate_config().

8.4.5.3 initial_order

```
unsigned int jeod::GaussJacksonConfig::initial_order {4}
```

The order of the Gauss Jackson integrator immediately after priming.

This must be an even number and must be 14 or less. Defaults to 4.trick_units(-)

Definition at line 119 of file gauss_jackson_config.hh.

Referenced by jeod::GaussJacksonStateMachine::configure(), jeod::set_default_config_values(), and jeod::validate_config().

8.4.5.4 max_correction_iterations

```
unsigned int jeod::GaussJacksonConfig::max_correction_iterations {10}
```

Maximum number of correction steps allowed before the integrator is deemed to be not converging.

The algorithm is run in predict-only mode if this limit is zero. The corrector is applied but once with the limit is one. A one-time warning is issued if the limit is 2 or more and if the the algorithm would make more corrections were it not for this limit. Defaults to 10.trick_units(-)

Definition at line 143 of file gauss_jackson_config.hh.

Referenced by jeod::GaussJacksonStateMachine::configure(), and jeod::set_default_config_values().

8.4.5.5 ndoubling_steps

```
unsigned int jeod::GaussJacksonConfig::ndoubling_steps
```

The number of time doubling steps involved in the bootstrap operation.

Defaults to (final_order - initial_order)/2.trick_units(-)

Definition at line 132 of file gauss_jackson_config.hh.

Referenced by jeod::GaussJacksonStateMachine::configure(), jeod::set_default_config_values(), and jeod::validate_config().

8.4.5.6 priming_technique

```
er7_utils::Integration::Technique jeod::GaussJacksonConfig::priming_technique {er7_utils::Integration::Unspecified}
```

The integration technique to be used to prime the Gauss-Jackson process.

Defaults to er7_utils::Integration::Unspecified, the interpretation of which depends on the initial order.trick_units(-)

Definition at line 112 of file gauss_jackson_config.hh.

Referenced by jeod::GaussJacksonIntegratorConstructor::configure(), default_configuration(), jeod::set_default_config_values(), and standard_configuration().

8.4.5.7 relative_tolerance

```
double jeod::GaussJacksonConfig::relative_tolerance {1E-14}
```

Number that indicates the allowable relative difference for two states to be considered converged.

Defaults to 1e-14.trick_units(-)

Definition at line 150 of file gauss_jackson_config.hh.

Referenced by jeod::set_default_config_values(), and jeod::validate_config().

The documentation for this class was generated from the following files:

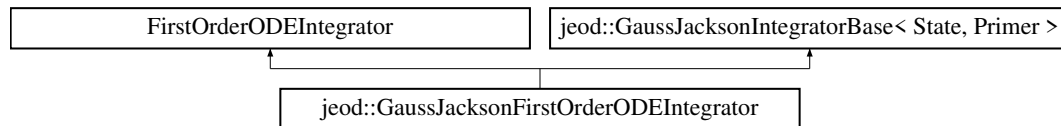
- [gauss_jackson_config.hh](#)
- [gauss_jackson_config.cc](#)

8.5 jeod::GaussJacksonFirstOrderODEIntegrator Class Reference

Integrates a first order ODE using the summed Adams technique that is embedded within the Gauss-Jackson technique.

```
#include <gauss_jackson_first_order_ode_integrator.hh>
```

Inheritance diagram for jeod::GaussJacksonFirstOrderODEIntegrator:



Private Member Functions

- JEOD_MAKE_SIM_INTERFACES(jeod, [GaussJacksonFirstOrderODEIntegrator](#)) public [~GaussJacksonFirstOrderODEIntegrator](#) () override=default
- [GaussJacksonFirstOrderODEIntegrator](#) (const [er7_utils::IntegratorConstructor](#) &priming_constructor, [GaussJacksonIntegrationControls](#) &controls, unsigned int size_in, [er7_utils::IntegrationControls](#) &priming_controls)
 - Non-default constructor.*
- [GaussJacksonFirstOrderODEIntegrator](#) (const [GaussJacksonFirstOrderODEIntegrator](#) &src)
 - Copy constructor.*
- [GaussJacksonFirstOrderODEIntegrator](#) & operator= ([GaussJacksonFirstOrderODEIntegrator](#) src)
 - Assignment operator.*
- void [swap](#) ([GaussJacksonFirstOrderODEIntegrator](#) &other)
 - Non-throwing swap.*
- [er7_utils::FirstOrderODEIntegrator](#) * [create_copy](#) () const override
 - Replicate this.*
- void [reset_integrator](#) () override
 - Reset the integrator.*
- [er7_utils::IntegratorResult](#) [integrate](#) (double dyn_dt, unsigned int target_stage, const double *ER7_UTILS_RESTRICT deriv, double *ER7_UTILS_RESTRICT state) override
 - Integrate.*
- void [swap](#) ([GaussJacksonIntegratorBase](#) &other)
 - Non-throwing swap.*

Additional Inherited Members

8.5.1 Detailed Description

Integrates a first order ODE using the summed Adams technique that is embedded within the Gauss-Jackson technique.

Definition at line 79 of file `gauss_jackson_first_order_ode_integrator.hh`.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 ~GaussJacksonFirstOrderODEIntegrator()

```
JEOD_MAKE_SIM_INTERFACES (jeod, GaussJacksonFirstOrderODEIntegrator) public jeod::Gauss←
JacksonFirstOrderODEIntegrator::~~GaussJacksonFirstOrderODEIntegrator ( ) [override], [private],
[default]
```

8.5.2.2 GaussJacksonFirstOrderODEIntegrator() [1/2]

```
jeod::GaussJacksonFirstOrderODEIntegrator::GaussJacksonFirstOrderODEIntegrator (
    const er7_utils::IntegratorConstructor & priming_constructor,
    GaussJacksonIntegrationControls & controls,
    unsigned int size_in,
    er7_utils::IntegrationControls & priming_controls ) [inline], [private]
```

Non-default constructor.

Parameters

<i>priming_constructor</i>	Integrator constructor for the technique used during priming.
<i>controls</i>	The Gauss-Jackson integration controls that drives this state integrator.
<i>size_in</i>	State size.
<i>priming_controls</i>	Integration controls used during priming.

Definition at line 98 of file gauss_jackson_first_order_ode_integrator.hh.

8.5.2.3 GaussJacksonFirstOrderODEIntegrator() [2/2]

```
jeod::GaussJacksonFirstOrderODEIntegrator::GaussJacksonFirstOrderODEIntegrator (
    const GaussJacksonFirstOrderODEIntegrator & src ) [inline], [private]
```

Copy constructor.

Definition at line 110 of file gauss_jackson_first_order_ode_integrator.hh.

8.5.3 Member Function Documentation

8.5.3.1 create_copy()

```
er7_utils::FirstOrderODEIntegrator* jeod::GaussJacksonFirstOrderODEIntegrator::create_copy ( )
const [inline], [override], [private]
```

Replicate this.

Definition at line 137 of file gauss_jackson_first_order_ode_integrator.hh.

8.5.3.2 integrate()

```
er7_utils::IntegratorResult jeod::GaussJacksonFirstOrderODEIntegrator::integrate (
    double dyn_dt,
    unsigned int target_stage,
    const double *ER7_UTILS_RESTRICT deriv,
    double *ER7_UTILS_RESTRICT state ) [inline], [override], [private]
```

Integrate.

Definition at line 153 of file gauss_jackson_first_order_ode_integrator.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate().

8.5.3.3 operator=()

```
GaussJacksonFirstOrderODEIntegrator& jeod::GaussJacksonFirstOrderODEIntegrator::operator= (
    GaussJacksonFirstOrderODEIntegrator src ) [inline], [private]
```

Assignment operator.

Definition at line 119 of file gauss_jackson_first_order_ode_integrator.hh.

References swap().

8.5.3.4 reset_integrator()

```
void jeod::GaussJacksonFirstOrderODEIntegrator::reset_integrator ( ) [inline], [override],
[private]
```

Reset the integrator.

Definition at line 145 of file gauss_jackson_first_order_ode_integrator.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::base_reset().

8.5.3.5 swap() [1/2]

```
void jeod::GaussJacksonFirstOrderODEIntegrator::swap (
    GaussJacksonFirstOrderODEIntegrator & other ) [inline], [private]
```

Non-throwing swap.

Definition at line 128 of file gauss_jackson_first_order_ode_integrator.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

Referenced by operator=(), and jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::swap().

8.5.3.6 swap() [2/2]

```
void jeod::GaussJacksonIntegratorBase< State, Primer >::swap [inline], [private]
```

Non-throwing swap.

Parameters

<i>other</i>	Item whose contents are to be swapped with this.
--------------	--

Definition at line 397 of file gauss_jackson_integrator_base.hh.

The documentation for this class was generated from the following file:

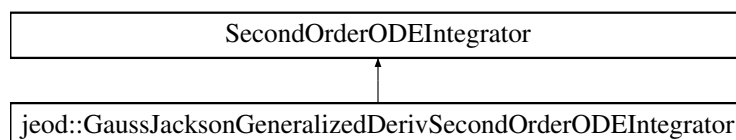
- [gauss_jackson_first_order_ode_integrator.hh](#)

8.6 jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator Class Reference

Integrates a generalized derivative second order ODE using Gauss-Jackson.

```
#include <gauss_jackson_generalized_second_order_ode_integrator.hh>
```

Inheritance diagram for jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator:



Public Member Functions

- [GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#) ()=default
Default constructor.
- [GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#) (const er7_utils::IntegratorConstructor &priming_↵_constructor, [GaussJacksonIntegrationControls](#) &controls, unsigned int position_size, unsigned int velocity_↵_size, const er7_utils::GeneralizedPositionDerivativeFunctions &deriv_funs, er7_utils::IntegrationControls &priming_controls)
Non-default constructor.
- [GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#) (const [GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#) &src)
Copy constructor.
- [~GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#) () override
Destructor.
- [GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#) & operator= ([GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#) &src)
Assignment operator.
- void [swap](#) ([GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#) &other)
Non-throwing swap.
- er7_utils::SecondOrderODEIntegrator * [create_copy](#) () const override
Replicate this.
- void [reset_integrator](#) () override
Reset the integrator.
- er7_utils::IntegratorResult [integrate](#) (double dyn_dt, unsigned int target_stage, const double *acc, double *vel, double *pos) override
Integrate state.

Private Attributes

- [GaussJacksonFirstOrderODEIntegrator](#) vel_integrator
Integrator for the generalized velocity.
- [GaussJacksonSimpleSecondOrderODEIntegrator](#) pos_integrator
Integrator for the generalized position.
- double * [posdot](#) {}
Generalized position time derivative.
- double * [posdotdot](#) {}
Generalized position second time derivative.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#) ()

8.6.1 Detailed Description

Integrates a generalized derivative second order ODE using Gauss-Jackson.

Generalized position is integrated via a simple second order Gauss-Jackson integrator. Generalized velocity is integrated via a first order summed Adams integrator.

Definition at line 82 of file `gauss_jackson_generalized_second_order_ode_integrator.hh`.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 GaussJacksonGeneralizedDerivSecondOrderODEIntegrator() [1/3]

```
jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator ( ) [default]
```

Default constructor.

8.6.2.2 GaussJacksonGeneralizedDerivSecondOrderODEIntegrator() [2/3]

```
jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator (
    const er7_utils::IntegratorConstructor & priming_constructor,
    GaussJacksonIntegrationControls & controls,
    unsigned int position_size,
    unsigned int velocity_size,
    const er7_utils::GeneralizedPositionDerivativeFunctions & deriv_funs,
    er7_utils::IntegrationControls & priming_controls )
```

Non-default constructor.

Parameters

<i>priming_constructor</i>	Integrator constructor for the technique used during priming.
<i>controls</i>	The Gauss-Jackson integration controls that drives this state integrator.
<i>position_size</i>	Generalized position vector size.
<i>velocity_size</i>	Generalized velocity vector size.
<i>deriv_funs</i>	Position vector time deriv functions.
<i>priming_controls</i>	Integration controls used during priming.

Definition at line 34 of file gauss_jackson_generalized_second_order_ode_integrator.cc.

References posdot, and posdotdot.

8.6.2.3 GaussJacksonGeneralizedDerivSecondOrderODEIntegrator() [3/3]

```
jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator (
    const GaussJacksonGeneralizedDerivSecondOrderODEIntegrator & src )
```

Copy constructor.

Definition at line 50 of file gauss_jackson_generalized_second_order_ode_integrator.cc.

References posdot, and posdotdot.

8.6.2.4 `~GaussJacksonGeneralizedDerivSecondOrderODEIntegrator()`

```
jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::~~GaussJacksonGeneralizedDerivSecondOrderODEIntegrator ( ) [override]
```

Destructor.

Definition at line 61 of file `gauss_jackson_generalized_second_order_ode_integrator.cc`.

References `posdot`, and `posdotdot`.

8.6.3 Member Function Documentation

8.6.3.1 `create_copy()`

```
er7_utils::SecondOrderODEIntegrator * jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::create_copy ( ) const [override]
```

Replicate this.

Definition at line 81 of file `gauss_jackson_generalized_second_order_ode_integrator.cc`.

8.6.3.2 `integrate()`

```
er7_utils::IntegratorResult jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::integrate (
    double dyn_dt,
    unsigned int target_stage,
    const double * acc,
    double * vel,
    double * pos ) [inline], [override]
```

Integrate state.

Definition at line 154 of file `gauss_jackson_generalized_second_order_ode_integrator.hh`.

8.6.3.3 `operator=()`

```
GaussJacksonGeneralizedDerivSecondOrderODEIntegrator& jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::operator= (
    GaussJacksonGeneralizedDerivSecondOrderODEIntegrator src ) [inline]
```

Assignment operator.

Definition at line 123 of file `gauss_jackson_generalized_second_order_ode_integrator.hh`.

8.6.3.4 reset_integrator()

```
void jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::reset_integrator ( ) [inline],  
[override]
```

Reset the integrator.

Definition at line 145 of file gauss_jackson_generalized_second_order_ode_integrator.hh.

8.6.3.5 swap()

```
void jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::swap (  
    GaussJacksonGeneralizedDerivSecondOrderODEIntegrator & other )
```

Non-throwing swap.

Definition at line 68 of file gauss_jackson_generalized_second_order_ode_integrator.cc.

References `pos_integrator`, `posdot`, `posdotdot`, `jeod::GaussJacksonFirstOrderODEIntegrator::swap()`, `jeod::GaussJacksonSimpleSecondOrderODEIntegrator::swap()`, and `vel_integrator`.

8.6.4 Friends And Related Function Documentation

8.6.4.1 init_attrjeod__GaussJacksonGeneralizedDerivSecondOrderODEIntegrator

```
void init_attrjeod__GaussJacksonGeneralizedDerivSecondOrderODEIntegrator ( ) [friend]
```

8.6.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 84 of file gauss_jackson_generalized_second_order_ode_integrator.hh.

8.6.5 Field Documentation

8.6.5.1 pos_integrator

```
GaussJacksonSimpleSecondOrderODEIntegrator jeod::GaussJacksonGeneralizedDerivSecondOrderODE↔
Integrator::pos_integrator [private]
```

Integrator for the generalized position.

trick_units(-)

Definition at line 176 of file gauss_jackson_generalized_second_order_ode_integrator.hh.

Referenced by swap().

8.6.5.2 posdot

```
double* jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::posdot {} [private]
```

Generalized position time derivative.

trick_units(-)

Definition at line 181 of file gauss_jackson_generalized_second_order_ode_integrator.hh.

Referenced by GaussJacksonGeneralizedDerivSecondOrderODEIntegrator(), swap(), and ~GaussJackson↔GeneralizedDerivSecondOrderODEIntegrator().

8.6.5.3 posdotdot

```
double* jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::posdotdot {} [private]
```

Generalized position second time derivative.

trick_units(-)

Definition at line 186 of file gauss_jackson_generalized_second_order_ode_integrator.hh.

Referenced by GaussJacksonGeneralizedDerivSecondOrderODEIntegrator(), swap(), and ~GaussJackson↔GeneralizedDerivSecondOrderODEIntegrator().

8.6.5.4 vel_integrator

```
GaussJacksonFirstOrderODEIntegrator jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator↔
::vel_integrator [private]
```

Integrator for the generalized velocity.

trick_units(—)

Definition at line 171 of file gauss_jackson_generalized_second_order_ode_integrator.hh.

Referenced by swap().

The documentation for this class was generated from the following files:

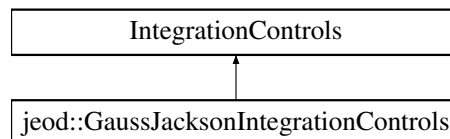
- [gauss_jackson_generalized_second_order_ode_integrator.hh](#)
- [gauss_jackson_generalized_second_order_ode_integrator.cc](#)

8.7 jeod::GaussJacksonIntegrationControls Class Reference

IntegrationControls specialized for Gauss-Jackson integration.

```
#include <gauss_jackson_integration_controls.hh>
```

Inheritance diagram for jeod::GaussJacksonIntegrationControls:



Public Member Functions

- [GaussJacksonIntegrationControls](#) ()
Default constructor.
- [GaussJacksonIntegrationControls](#) (const [er7_utils::IntegratorConstructor](#) &priming_constructor, const [GaussJacksonConfig](#) &config_in)
Non-default constructor.
- [GaussJacksonIntegrationControls](#) (const [GaussJacksonIntegrationControls](#) &src)
Copy constructor.
- [~GaussJacksonIntegrationControls](#) () override
Destructor.
- [GaussJacksonIntegrationControls](#) & operator= ([GaussJacksonIntegrationControls](#) src)
Copy and swap assignment operator.
- [er7_utils::IntegrationControls](#) * [create_copy](#) () const override
Create a duplicate of this object.
- [er7_utils::IntegrationControls](#) & [get_priming_controls](#) () const
Getter for the priming_controls data member.
- const [GaussJacksonCoeffs](#) & [get_coeff](#) () const

- *Getter for the coeff data member.*
- const [GaussJacksonConfig](#) & [get_config](#) () const
Getter for the config data member.
- const [GaussJacksonStateMachine](#) & [get_state_machine](#) () const
Getter for the state_machine data member.
- void [reset_integrator](#) () override
Reset the integration controls object.
- unsigned int [integrate](#) (double start_time, double sim_dt, er7_utils::TimeInterface &time_interface, er7_utils::IntegratorInterface &integ_interface, er7_utils::BaseIntegrationGroup &integ_group) override
Make one step in the process that eventually integrates state from the start_time to start_time+sim_dt.

Protected Member Functions

- virtual void [swap](#) ([GaussJacksonIntegrationControls](#) &other)
Non-throwing swap function.

Private Member Functions

- void [start_cycle](#) (double sim_dt)
Perform start of integration cycle actions.
- void [integrate_edit](#) (er7_utils::TimeInterface &time_interface, er7_utils::BaseIntegrationGroup &integ_group)
Guide integration while in BootstrapEdit mode.
- void [integrate_gj](#) (er7_utils::TimeInterface &time_interface, er7_utils::BaseIntegrationGroup &integ_group)
Guide integration while in BootstrapStep or Operational mode.

Private Attributes

- er7_utils::IntegrationControls * [priming_controls](#) {}
The integration controls object used to prime the Gauss-Jackson integration process.
- double [cycle_starttime](#) {}
The simulation time of the start of the current integration cycle.
- double [cycle_simdt](#) {}
The simulation time span of the current integration cycle.
- double [cycle_dyndt](#) {}
The dynamic time span corresponding to cycle_simdt.
- double [reset_time](#) {}
The simulation time of the most recent reset.
- [GaussJacksonCoeffs](#) [coeff](#)
The Gauss-Jackson corrector and predictor coefficients.
- [GaussJacksonConfig](#) [config](#)
The Gauss-Jackson configuration data.
- [GaussJacksonStateMachine](#) [state_machine](#)
The Gauss-Jackson state machine.
- [GaussJacksonStateMachine::FsmState](#) [fsm_state](#) {[GaussJacksonStateMachine::Reset](#)}
The state machine's finite state.
- unsigned int [max_correction_iterations](#) {}
Maximum number of correction iterations allowed.
- unsigned int [initial_order](#) {}
The order of the Gauss Jackson integrator immediately after priming.

- unsigned int [order](#) {}
The current order of the Gauss Jackson integrator.
- unsigned int [edit_count](#) {}
Number of times that the current set of history have been edited.
- bool [at_end_of_tour](#) {}
Flag indicating that the current integration cycle is the last one in an integration tour (i.e., that a major time step will be completed).

Friends

- class [InputProcessor](#)
- void [init_attrjeod__GaussJacksonIntegrationControls](#) ()

8.7.1 Detailed Description

IntegrationControls specialized for Gauss-Jackson integration.

Definition at line 80 of file `gauss_jackson_integration_controls.hh`.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 GaussJacksonIntegrationControls() [1/3]

```
jeod::GaussJacksonIntegrationControls::GaussJacksonIntegrationControls ( )
```

Default constructor.

Definition at line 40 of file `gauss_jackson_integration_controls.cc`.

8.7.2.2 GaussJacksonIntegrationControls() [2/3]

```
jeod::GaussJacksonIntegrationControls::GaussJacksonIntegrationControls (
    const er7_utils::IntegratorConstructor & priming_constructor,
    const GaussJacksonConfig & config_in )
```

Non-default constructor.

This is the constructor invoked by the [GaussJacksonIntegratorConstructor](#).

Parameters

<i>priming_constructor</i>	Integrator constructor for the technique used during priming.
<i>config_in</i>	Gauss-Jackson configuration data.

Definition at line 46 of file gauss_jackson_integration_controls.cc.

References `coeff`, `jeod::GaussJacksonCoeffs::compute_coeffs()`, `config`, `jeod::GaussJacksonCoeffs::configure()`, `jeod::GaussJacksonStateMachine::configure()`, `jeod::GaussJacksonConfig::final_order`, `initial_order`, `priming_↵` controls, and `state_machine`.

8.7.2.3 GaussJacksonIntegrationControls() [3/3]

```
jeod::GaussJacksonIntegrationControls::GaussJacksonIntegrationControls (
    const GaussJacksonIntegrationControls & src )
```

Copy constructor.

Parameters

<i>src</i>	Object to be copied.
------------	----------------------

Definition at line 62 of file gauss_jackson_integration_controls.cc.

References `priming_controls`.

8.7.2.4 ~GaussJacksonIntegrationControls()

```
jeod::GaussJacksonIntegrationControls::~~GaussJacksonIntegrationControls ( ) [override]
```

Destructor.

Definition at line 85 of file gauss_jackson_integration_controls.cc.

References `priming_controls`.

8.7.3 Member Function Documentation

8.7.3.1 create_copy()

```
er7_utils::IntegrationControls * jeod::GaussJacksonIntegrationControls::create_copy ( ) const
[override]
```

Create a duplicate of this object.

Returns

Replicated [GaussJacksonIntegrationControls](#).

Definition at line 90 of file gauss_jackson_integration_controls.cc.

8.7.3.2 get_coeff()

```
const GaussJacksonCoeffs& jeod::GaussJacksonIntegrationControls::get_coeff ( ) const [inline]
```

Getter for the coeff data member.

Returns

Reference to the Gauss-Jackson coefficients object.

Definition at line 138 of file gauss_jackson_integration_controls.hh.

8.7.3.3 get_config()

```
const GaussJacksonConfig& jeod::GaussJacksonIntegrationControls::get_config ( ) const [inline]
```

Getter for the config data member.

Returns

Reference to the Gauss-Jackson configuration object.

Definition at line 147 of file gauss_jackson_integration_controls.hh.

8.7.3.4 get_priming_controls()

```
er7_utils::IntegrationControls& jeod::GaussJacksonIntegrationControls::get_priming_controls (
) const [inline]
```

Getter for the priming_controls data member.

Returns

Reference to the integration controls used during priming.

Definition at line 129 of file gauss_jackson_integration_controls.hh.

Referenced by `jeod::GaussJacksonIntegratorConstructor::create_first_order_ode_integrator()`, `jeod::GaussJacksonIntegratorConstructor::create_generalized_deriv_second_order_ode_integrator()`, and `jeod::GaussJacksonIntegratorConstructor::create_second_order_ode_integrator()`.

8.7.3.5 get_state_machine()

```
const GaussJacksonStateMachine& jeod::GaussJacksonIntegrationControls::get_state_machine ( )
const [inline]
```

Getter for the state_machine data member.

Returns

Reference to the Gauss-Jackson state_machine object.

Definition at line 156 of file gauss_jackson_integration_controls.hh.

8.7.3.6 integrate()

```
unsigned int jeod::GaussJacksonIntegrationControls::integrate (
    double start_time,
    double sim_dt,
    er7_utils::TimeInterface & time_interface,
    er7_utils::IntegratorInterface & integ_interface,
    er7_utils::BaseIntegrationGroup & integ_group ) [override]
```

Make one step in the process that eventually integrates state from the start_time to start_time+sim_dt.

Returns

Step number; zero when finished.

Parameters

in	<i>start_time</i>	The simulation engine time at the start of the integration tour.
in	<i>sim_dt</i>	The difference between the simulation time at the end and start of the integration tour.
in, out	<i>time_interface</i>	Object external to the ER7 utilities suite that represents time.
in, out	<i>integ_interface</i>	Interface with the simulation engine for this integration controls.
in, out	<i>integ_group</i>	The integration group that contains this integration controls.

Definition at line 126 of file gauss_jackson_integration_controls.cc.

References at_end_of_tour, jeod::GaussJacksonStateMachine::BootstrapEdit, jeod::GaussJacksonState↵
Machine::BootstrapStep, cycle_dyndt, cycle_simdt, cycle_starttime, fsm_state, jeod::GaussJacksonState↵
Machine::get_cycle_scale(), integrate_edit(), integrate_gj(), jeod::GaussJacksonStateMachine::Operational, jeod↵
::GaussJacksonStateMachine::Priming, priming_controls, jeod::GaussJacksonStateMachine::Reset, reset↵
integrator(), reset_time, start_cycle(), and state_machine.

8.7.3.7 integrate_edit()

```
void jeod::GaussJacksonIntegrationControls::integrate_edit (
    er7_utils::TimeInterface & time_interface,
    er7_utils::BaseIntegrationGroup & integ_group ) [private]
```

Guide integration while in BootstrapEdit mode.

Definition at line 222 of file gauss_jackson_integration_controls.cc.

References `cycle_dyndt`, `cycle_starttime`, `edit_count`, `jeod::GaussJacksonStateMachine::get_history_length()`, `max_correction_iterations`, `order`, `jeod::GaussJacksonStateMachine::set_bootstrap_edit_redo_needed()`, and `state_machine`.

Referenced by `integrate()`.

8.7.3.8 integrate_gj()

```
void jeod::GaussJacksonIntegrationControls::integrate_gj (
    er7_utils::TimeInterface & time_interface,
    er7_utils::BaseIntegrationGroup & integ_group ) [private]
```

Guide integration while in BootstrapStep or Operational mode.

Definition at line 255 of file gauss_jackson_integration_controls.cc.

References `cycle_dyndt`, `cycle_simdt`, `cycle_starttime`, `edit_count`, and `max_correction_iterations`.

Referenced by `integrate()`.

8.7.3.9 operator=()

```
GaussJacksonIntegrationControls& jeod::GaussJacksonIntegrationControls::operator= (
    GaussJacksonIntegrationControls src ) [inline]
```

Copy and swap assignment operator.

Parameters

<i>src</i>	Object to be copied.
------------	----------------------

Definition at line 113 of file gauss_jackson_integration_controls.hh.

8.7.3.10 reset_integrator()

```
void jeod::GaussJacksonIntegrationControls::reset_integrator ( ) [override]
```

Reset the integration controls object.

Definition at line 116 of file gauss_jackson_integration_controls.cc.

References `at_end_of_tour`, `edit_count`, `fsm_state`, `initial_order`, `order`, `jeod::GaussJacksonStateMachine::Reset`, `jeod::GaussJacksonStateMachine::reset()`, and `state_machine`.

Referenced by `integrate()`.

8.7.3.11 start_cycle()

```
void jeod::GaussJacksonIntegrationControls::start_cycle (
    double sim_dt ) [private]
```

Perform start of integration cycle actions.

Definition at line 288 of file gauss_jackson_integration_controls.cc.

References `at_end_of_tour`, `jeod::GaussJacksonStateMachine::BootstrapEdit`, `coeff`, `jeod::GaussJacksonCoeffs::compute_coeffs()`, `cycle_dyndt`, `cycle_simdt`, `cycle_starttime`, `edit_count`, `fsm_state`, `jeod::GaussJacksonStateMachine::get_at_downsample()`, `jeod::GaussJacksonStateMachine::get_at_end_of_tour()`, `jeod::GaussJacksonStateMachine::get_at_order_change()`, `jeod::GaussJacksonStateMachine::get_at_reinitialize()`, `jeod::GaussJacksonStateMachine::get_current_order()`, `jeod::GaussJacksonStateMachine::get_cycle_scale()`, `jeod::GaussJacksonStateMachine::get_cycle_start_time()`, `jeod::GaussJacksonStateMachine::get_fsm_state()`, `order`, `jeod::GaussJacksonStateMachine::perform_step()`, `reset_time`, and `state_machine`.

Referenced by `integrate()`.

8.7.3.12 swap()

```
void jeod::GaussJacksonIntegrationControls::swap (
    GaussJacksonIntegrationControls & other ) [protected], [virtual]
```

Non-throwing swap function.

Swap contents of 'this' with that of the other.

Parameters

<code>in</code>	<code>other</code>	Item with which contents are to be swapped.
-----------------	--------------------	---

Definition at line 96 of file gauss_jackson_integration_controls.cc.

References `at_end_of_tour`, `coeff`, `config`, `cycle_dyndt`, `cycle_simdt`, `cycle_starttime`, `edit_count`, `fsm_state`, `initial_order`, `max_correction_iterations`, `order`, `priming_controls`, `reset_time`, and `state_machine`.

8.7.4 Friends And Related Function Documentation

8.7.4.1 init_attrjeod__GaussJacksonIntegrationControls

```
void init_attrjeod__GaussJacksonIntegrationControls ( ) [friend]
```

8.7.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 82 of file gauss_jackson_integration_controls.hh.

8.7.5 Field Documentation

8.7.5.1 at_end_of_tour

```
bool jeod::GaussJacksonIntegrationControls::at_end_of_tour {} [private]
```

Flag indicating that the current integration cycle is the last one in an integration tour (i.e., that a major time step will be completed).

trick_units(–)

Definition at line 271 of file gauss_jackson_integration_controls.hh.

Referenced by integrate(), reset_integrator(), start_cycle(), and swap().

8.7.5.2 coeff

```
GaussJacksonCoeffs jeod::GaussJacksonIntegrationControls::coeff [private]
```

The Gauss-Jackson corrector and predictor coefficients.

trick_units(–)

Definition at line 230 of file gauss_jackson_integration_controls.hh.

Referenced by GaussJacksonIntegrationControls(), start_cycle(), and swap().

8.7.5.3 config

```
GaussJacksonConfig jeod::GaussJacksonIntegrationControls::config [private]
```

The Gauss-Jackson configuration data.

trick_units(-)

Definition at line 235 of file gauss_jackson_integration_controls.hh.

Referenced by GaussJacksonIntegrationControls(), and swap().

8.7.5.4 cycle_dyndt

```
double jeod::GaussJacksonIntegrationControls::cycle_dyndt {} [private]
```

The dynamic time span corresponding to cycle_simdt.

trick_units(s)

Definition at line 220 of file gauss_jackson_integration_controls.hh.

Referenced by integrate(), integrate_edit(), integrate_gj(), start_cycle(), and swap().

8.7.5.5 cycle_simdt

```
double jeod::GaussJacksonIntegrationControls::cycle_simdt {} [private]
```

The simulation time span of the current integration cycle.

trick_units(-)

Definition at line 215 of file gauss_jackson_integration_controls.hh.

Referenced by integrate(), integrate_gj(), start_cycle(), and swap().

8.7.5.6 cycle_starttime

```
double jeod::GaussJacksonIntegrationControls::cycle_starttime {} [private]
```

The simulation time of the start of the current integration cycle.

An integration cycle starts when cycle_stage is zero and ends when it reaches zero once again. trick_units(-)

Definition at line 210 of file gauss_jackson_integration_controls.hh.

Referenced by integrate(), integrate_edit(), integrate_gj(), start_cycle(), and swap().

8.7.5.7 edit_count

```
unsigned int jeod::GaussJacksonIntegrationControls::edit_count {} [private]
```

Number of times that the current set of history have been edited.

trick_units(-)

Definition at line 265 of file gauss_jackson_integration_controls.hh.

Referenced by integrate_edit(), integrate_gj(), reset_integrator(), start_cycle(), and swap().

8.7.5.8 fsm_state

```
GaussJacksonStateMachine::FsmState jeod::GaussJacksonIntegrationControls::fsm_state {GaussJacksonStateMachine::FsmState} [private]
```

The state machine's finite state.

trick_units(-)

Definition at line 245 of file gauss_jackson_integration_controls.hh.

Referenced by integrate(), reset_integrator(), start_cycle(), and swap().

8.7.5.9 initial_order

```
unsigned int jeod::GaussJacksonIntegrationControls::initial_order {} [private]
```

The order of the Gauss Jackson integrator immediately after priming.

trick_units(-)

Definition at line 255 of file gauss_jackson_integration_controls.hh.

Referenced by GaussJacksonIntegrationControls(), reset_integrator(), and swap().

8.7.5.10 max_correction_iterations

```
unsigned int jeod::GaussJacksonIntegrationControls::max_correction_iterations {} [private]
```

Maximum number of correction iterations allowed.

trick_units(-)

Definition at line 250 of file gauss_jackson_integration_controls.hh.

Referenced by integrate_edit(), integrate_gj(), and swap().

8.7.5.11 order

```
unsigned int jeod::GaussJacksonIntegrationControls::order {} [private]
```

The current order of the Gauss Jackson integrator.

trick_units(–)

Definition at line 260 of file gauss_jackson_integration_controls.hh.

Referenced by integrate_edit(), reset_integrator(), start_cycle(), and swap().

8.7.5.12 priming_controls

```
er7_utils::IntegrationControls* jeod::GaussJacksonIntegrationControls::priming_controls {}  
[private]
```

The integration controls object used to prime the Gauss-Jackson integration process.

trick_units(–)

Definition at line 203 of file gauss_jackson_integration_controls.hh.

Referenced by GaussJacksonIntegrationControls(), integrate(), swap(), and ~GaussJacksonIntegrationControls().

8.7.5.13 reset_time

```
double jeod::GaussJacksonIntegrationControls::reset_time {} [private]
```

The simulation time of the most recent reset.

trick_units(–)

Definition at line 225 of file gauss_jackson_integration_controls.hh.

Referenced by integrate(), start_cycle(), and swap().

8.7.5.14 state_machine

```
GaussJacksonStateMachine jeod::GaussJacksonIntegrationControls::state_machine [private]
```

The Gauss-Jackson state machine.

trick_units(–)

Definition at line 240 of file gauss_jackson_integration_controls.hh.

Referenced by GaussJacksonIntegrationControls(), integrate(), integrate_edit(), reset_integrator(), start_cycle(), and swap().

The documentation for this class was generated from the following files:

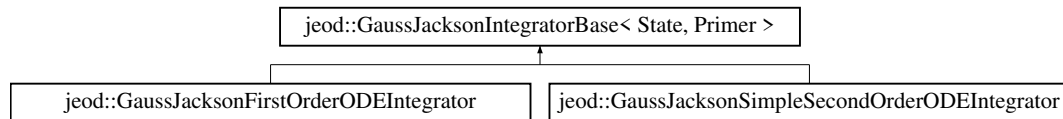
- [gauss_jackson_integration_controls.hh](#)
- [gauss_jackson_integration_controls.cc](#)

8.8 jeod::GaussJacksonIntegratorBase< State, Primer > Class Template Reference

Base template class for integrating state via the Gauss-Jackson technique.

```
#include <gauss_jackson_integrator_base.hh>
```

Inheritance diagram for jeod::GaussJacksonIntegratorBase< State, Primer >:



Public Member Functions

- [GaussJacksonIntegratorBase](#) ()=default
Default constructor.
- [GaussJacksonIntegratorBase](#) (const er7_utils::IntegratorConstructor &priming_constructor, const [GaussJacksonIntegrationCon](#) &controls, unsigned int size_in, er7_utils::IntegrationControls &priming_controls)
Non-default constructor.
- [GaussJacksonIntegratorBase](#) (const [GaussJacksonIntegratorBase](#) &src)
Copy constructor.
- [~GaussJacksonIntegratorBase](#) ()
Destructor.
- [GaussJacksonIntegratorBase](#) & operator= (const [GaussJacksonIntegratorBase](#) &)=delete

Data Fields

- const [GaussJacksonCoeffs](#) * coeff {}
The summed Adams and Gauss-Jackson coefficients, in ordinate form.
- const [GaussJacksonStateMachine](#) * state_machine {}
The Gauss-Jackson state machine.
- Primer * primer {}
The integrator used to prime the Gauss-Jackson integration process.
- State [init_state](#)
The state at the time of the last reset.
- State [delinv](#)
Inverse backward differences.
- State [corrector_sum](#)
Speed hack for the corrector.
- er7_utils::DoubleTwoDArray [acc_hist](#)
Acceleration history.
- er7_utils::DoubleTwoDArray [pos_hist](#)
Position history (or velocity history in case of a first order ODE).
- double [relative_tolerance](#) {}
Number that indicates the allowable relative difference for two states to be considered converged.
- double [absolute_tolerance](#) {}
Number that indicates the allowable absolute difference for two states to be considered converged.
- double [velocity_corrector](#) {}

- *Correction coefficient for the first integral (velocity).*
- double `position_corrector` {}
- *Correction coefficient for the second integral (position).*
- `GaussJacksonStateMachine::FsmState fsm_state {GaussJacksonStateMachine::Reset}`
- *Finite state machine state.*
- unsigned int `max_history_size` {}
- *Maximum history size.*
- unsigned int `initial_order` {}
- *Initial order.*
- unsigned int `order` {}
- *Current order.*
- unsigned int `size` {}
- *State size.*
- unsigned int `history_length` {}
- *Current history length.*

Protected Member Functions

- void `base_reset` ()
- *Reset the integrator.*
- `er7_utils::IntegratorResult base_integrate` (double dyn_dt, unsigned int target_stage, const double *deriv, State state)
- *Propagate state to the specified target_stage.*
- void `swap` (`GaussJacksonIntegratorBase` &other)
- *Non-throwing swap.*

Private Member Functions

- void `start_cycle` (double dt, const double *acc, const State &state)
- *Start an integration cycle.*
- bool `edit_point` (double dt, const double *acc, State &state)
- *Edit the specified point using the mid-corrector that pertains to the point being edited, which is that at history_length.*
- bool `integrate_gj` (double dt, unsigned int target_stage, int advance_index, int target_index, const double *acc, const double *const *ahist, State &state)
- *Integrate using the Gauss-Jackson predictor and corrector.*
- void `downsample_hist` ()
- *Downsample the acceleration and position histories.*
- void `rotate_acc_hist` ()
- *Rotate the acceleration history.*
- `er7_utils::IntegratorResult integrate_primer` (double dyn_dt, unsigned int target_stage, const double *deriv, State &state)
- *Integrate state using the primer.*
- void `save_epoch_data` (const double *acc, const State &state)
- *Save epoch data.*
- void `save_comparison_data` (const State &state, double *pos_hist_elem)
- *Save comparison data.*
- void `initialize_edit_integration_constants` (double dt)
- *Initialize the integration constants (i.e., delinv).*
- void `initialize_predictor_integration_constants` (double dt)
- *Initialize the integration constants (i.e., delinv).*

- void [advance_edit_integration_constants](#) (unsigned int index)
Advance the integration constants by one cycle.
- void [advance_predictor_integration_constants](#) (unsigned int index)
Advance the integration constants by one cycle.
- void [mid_correct](#) (unsigned int coeff_idx, double dt, State &state)
Apply a mid-corrector.
- void [predict](#) (double dt, const double *const *ahist, State &state)
Apply the predictor.
- void [correct](#) (double dt, const double *acc, State &state)
Apply the corrector.
- bool [test_for_convergence](#) (const State &state, double *hist_data)
Test for convergence.
- void [swap_state](#) (State &item, State &other_item)
Swap state data with another of the same.
- void [replicate_state](#) (const State &source, State &target)
Replicate state data.
- void [allocate_state_contents](#) (State &item)
Allocate memory for a state item.
- void [deallocate_state_contents](#) (State &item)
Deallocate state item memory.
- template<>
er7_utils::FirstOrderODEIntegrator * [create_primer](#) (const er7_utils::IntegratorConstructor &priming_↵
constructor, unsigned int [size](#), er7_utils::IntegrationControls &priming_controls)
Create the priming integrator.
- template<>
er7_utils::FirstOrderODEIntegrator * [replicate_primer](#) (const er7_utils::FirstOrderODEIntegrator *src)
Replicate the priming integrator.
- template<>
er7_utils::IntegratorResult [integrate_primer](#) (double dyn_dt, unsigned int target_stage, const double *deriv,
[GaussJacksonOneState](#) &state)
Integrate with the primer.
- template<>
void [save_epoch_data](#) (const double *acc, const [GaussJacksonOneState](#) &state)
Save epoch data.
- template<>
void [save_comparison_data](#) (const [GaussJacksonOneState](#) &state, double *pos_hist_elem)
Save comparison data.
- template<>
void [initialize_edit_integration_constants](#) (double dt)
Initialize the integration constants (i.e., delinv).
- template<>
void [advance_edit_integration_constants](#) (unsigned int index)
Advance the integration constants by one cycle.
- template<>
void [initialize_predictor_integration_constants](#) (double dt)
Initialize the integration constants (i.e., delinv).
- template<>
void [advance_predictor_integration_constants](#) (unsigned int index)
Advance the integration constants by one cycle.
- template<>
void [mid_correct](#) (unsigned int coeff_idx, double dt, [GaussJacksonOneState](#) &state)
Apply a mid-corrector.

- `template<>`
`void predict (double dt, const double *const *ahist, GaussJacksonOneState &state)`
Apply the predictor.
- `template<>`
`void correct (double dt, const double *acc, GaussJacksonOneState &state)`
Apply the corrector.
- `template<>`
`bool test_for_convergence (const GaussJacksonOneState &state, double *hist_data)`
Test for convergence.
- `template<>`
`void swap_state (GaussJacksonOneState &item, GaussJacksonOneState &other_item)`
Swap state data with another of the same.
- `template<>`
`void replicate_state (const GaussJacksonOneState &source, GaussJacksonOneState &target)`
Replicate state data.
- `template<>`
`void allocate_state_contents (GaussJacksonOneState &item)`
Allocate memory for a state item.
- `template<>`
`void deallocate_state_contents (GaussJacksonOneState &item)`
Deallocate state item memory.
- `template<>`
`er7_utils::SecondOrderODEIntegrator * create_primer (const er7_utils::IntegratorConstructor &priming_↵
constructor, unsigned int size, er7_utils::IntegrationControls &priming_controls)`
- `template<>`
`er7_utils::SecondOrderODEIntegrator * replicate_primer (const er7_utils::SecondOrderODEIntegrator *src)`
- `template<>`
`er7_utils::IntegratorResult integrate_primer (double dyn_dt, unsigned int target_stage, const double *deriv,
GaussJacksonTwoState &state)`
Integrate with the primer.
- `template<>`
`void save_epoch_data (const double *acc, const GaussJacksonTwoState &state)`
Save epoch data.
- `template<>`
`void save_comparison_data (const GaussJacksonTwoState &state, double *pos_hist_elem)`
Save comparison data.
- `template<>`
`void initialize_edit_integration_constants (double dt)`
Initialize the integration constants (i.e., delinv).
- `template<>`
`void advance_edit_integration_constants (unsigned int index)`
Advance the integration constants by one cycle.
- `template<>`
`void initialize_predictor_integration_constants (double dt)`
Initialize the integration constants (i.e., delinv).
- `template<>`
`void advance_predictor_integration_constants (unsigned int index)`
Advance the integration constants by one cycle.
- `template<>`
`void mid_correct (unsigned int coeff_idx, double dt, GaussJacksonTwoState &state)`
Apply a mid-corrector.
- `template<>`
`void predict (double dt, const double *const *ahist, GaussJacksonTwoState &state)`

- Apply the predictor.*

 - `template<>`
`void correct` (double dt, const double *acc, `GaussJacksonTwoState` &state)
- Apply the corrector.*

 - `template<>`
`bool test_for_convergence` (const `GaussJacksonTwoState` &state, double *hist_data)
- Test for convergence.*

 - `template<>`
`void swap_state` (`GaussJacksonTwoState` &item, `GaussJacksonTwoState` &other_item)
- Swap state data with another of the same.*

 - `template<>`
`void replicate_state` (const `GaussJacksonTwoState` &source, `GaussJacksonTwoState` &target)
- Replicate state data.*

 - `template<>`
`void allocate_state_contents` (`GaussJacksonTwoState` &item)
- Allocate memory for a state item.*

 - `template<>`
`void deallocate_state_contents` (`GaussJacksonTwoState` &item)
- Deallocate state item memory.*

Static Private Member Functions

- static `Primer * create_primer` (const `er7_utils::IntegratorConstructor` &priming_constructor, unsigned int size_in, `er7_utils::IntegrationControls` &priming_controls)

Create the integrator to be used during priming.
- static `Primer * replicate_primer` (const `Primer *src`)

Create a replica of the provided primer.

8.8.1 Detailed Description

```
template<typename State, typename Primer>
class jeod::GaussJacksonIntegratorBase< State, Primer >
```

Base template class for integrating state via the Gauss-Jackson technique.

Template Parameters

<i>State</i>	Structure that contains the state.
<i>Primer</i>	Class for priming the Gauss-Jackson integrator.

Definition at line 87 of file `gauss_jackson_integrator_base.hh`.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 GaussJacksonIntegratorBase() [1/3]

```
template<typename State , typename Primer >
jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase ( ) [default]
```

Default constructor.

8.8.2.2 GaussJacksonIntegratorBase() [2/3]

```
template<typename State , typename Primer >
jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase (
    const er7_utils::IntegratorConstructor & priming_constructor,
    const GaussJacksonIntegrationControls & controls,
    unsigned int size_in,
    er7_utils::IntegrationControls & priming_controls ) [inline]
```

Non-default constructor.

Parameters

<i>priming_constructor</i>	Integrator constructor for the technique used during priming.
<i>controls</i>	The Gauss-Jackson integration controls that drives this state integrator.
<i>size_in</i>	State size.
<i>priming_controls</i>	Integration controls used during priming.

Definition at line 201 of file `gauss_jackson_integrator_base.hh`.

References `jeod::GaussJacksonIntegratorBase< State, Primer >::acc_hist`, `er7_utils::TwoDArray< T >::allocate()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::allocate_state_contents()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::corrector_sum`, `jeod::GaussJacksonIntegratorBase< State, Primer >::create_primer()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::delinv`, `jeod::GaussJacksonIntegratorBase< State, Primer >::init_state`, `jeod::GaussJacksonIntegratorBase< State, Primer >::max_history_size`, `jeod::GaussJacksonIntegratorBase< State, Primer >::pos_hist`, `jeod::GaussJacksonIntegratorBase< State, Primer >::primer`, and `jeod::GaussJacksonIntegratorBase< State, Primer >::size`.

8.8.2.3 GaussJacksonIntegratorBase() [3/3]

```
template<typename State , typename Primer >
jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase (
    const GaussJacksonIntegratorBase< State, Primer > & src ) [inline]
```

Copy constructor.

Parameters

<i>src</i>	Item to be copied.
------------	--------------------

Definition at line 230 of file gauss_jackson_integrator_base.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::corrector_sum, jeod::GaussJacksonIntegratorBase< State, Primer >::delinv, jeod::GaussJacksonIntegratorBase< State, Primer >::init_state, jeod::GaussJacksonIntegratorBase< State, Primer >::primer, jeod::GaussJacksonIntegratorBase< State, Primer >::replicate_primer(), and jeod::GaussJacksonIntegratorBase< State, Primer >::replicate_state().

8.8.2.4 ~GaussJacksonIntegratorBase()

```
template<typename State , typename Primer >
jeod::GaussJacksonIntegratorBase< State, Primer >::~~GaussJacksonIntegratorBase ( ) [inline]
```

Destructor.

Definition at line 256 of file gauss_jackson_integrator_base.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::corrector_sum, jeod::GaussJacksonIntegratorBase< State, Primer >::deallocate_state_contents(), jeod::GaussJacksonIntegratorBase< State, Primer >::delinv, jeod::GaussJacksonIntegratorBase< State, Primer >::init_state, and jeod::GaussJacksonIntegratorBase< State, Primer >::primer.

8.8.3 Member Function Documentation

8.8.3.1 advance_edit_integration_constants() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator >::advance_edit_integration_constants (
    unsigned int index ) [inline], [private]
```

Advance the integration constants by one cycle.

Definition at line 170 of file gauss_jackson_integrator_base_first.hh.

8.8.3.2 advance_edit_integration_constants() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator >::advance_edit_integration_constants (
    unsigned int index ) [inline], [private]
```

Advance the integration constants by one cycle.

Definition at line 171 of file gauss_jackson_integrator_base_second.hh.

8.8.3.3 advance_edit_integration_constants() [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::advance_edit_integration_constants (
    unsigned int index ) [private]
```

Advance the integration constants by one cycle.

Parameters

<i>index</i>	Coefficient index.
--------------	--------------------

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::edit_point()`.

8.8.3.4 `advance_predictor_integration_constants()` [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::advance_predictor_integration_constants (
    unsigned int index ) [inline], [private]
```

Advance the integration constants by one cycle.

Definition at line 201 of file `gauss_jackson_integrator_base_first.hh`.

8.8.3.5 `advance_predictor_integration_constants()` [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::advance_predictor_integration_constants (
    unsigned int index ) [inline], [private]
```

Advance the integration constants by one cycle.

Definition at line 205 of file `gauss_jackson_integrator_base_second.hh`.

8.8.3.6 `advance_predictor_integration_constants()` [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::advance_predictor_integration_constants
(
    unsigned int index ) [private]
```

Advance the integration constants by one cycle.

Parameters

<i>index</i>	Coefficient index.
--------------	--------------------

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj()`.

8.8.3.7 allocate_state_contents() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::allocate_state_contents (
    GaussJacksonOneState & item ) [inline], [private]
```

Allocate memory for a state item.

Definition at line 314 of file gauss_jackson_integrator_base_first.hh.

References jeod::GaussJacksonOneState::first.

8.8.3.8 allocate_state_contents() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::allocate_state_contents (
    GaussJacksonTwoState & item ) [inline], [private]
```

Allocate memory for a state item.

Definition at line 329 of file gauss_jackson_integrator_base_second.hh.

References jeod::GaussJacksonTwoState::first, and jeod::GaussJacksonTwoState::second.

8.8.3.9 allocate_state_contents() [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::allocate_state_contents (
    State & item ) [private]
```

Allocate memory for a state item.

Parameters

<i>item</i>	State item to be allocated.
-------------	-----------------------------

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase().

8.8.3.10 base_integrate()

```
template<typename State , typename Primer >
er7_utils::IntegratorResult jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate
```

```
(
    double dyn_dt,
    unsigned int target_stage,
    const double * deriv,
    State state ) [inline], [protected]
```

Propagate state to the specified target_stage.

Parameters

in	<i>dyn_dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in	<i>deriv</i>	Acceleration vector.
in, out	<i>state</i>	State vector(s).

Returns

The status (time advance, pass/fail status) of the integration.

Definition at line 288 of file gauss_jackson_integrator_base.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::acc_hist, jeod::GaussJacksonStateMachine::BootstrapEdit, jeod::GaussJacksonStateMachine::BootstrapStep, jeod::GaussJacksonIntegratorBase< State, Primer >::edit_point(), jeod::GaussJacksonIntegratorBase< State, Primer >::fsm_state, jeod::GaussJacksonIntegratorBase< State, Primer >::history_length, jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj(), jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_primer(), jeod::GaussJacksonStateMachine::Operational, jeod::GaussJacksonIntegratorBase< State, Primer >::order, jeod::GaussJacksonIntegratorBase< State, Primer >::pos_hist, jeod::GaussJacksonStateMachine::Priming, jeod::GaussJacksonStateMachine::Reset, jeod::GaussJacksonIntegratorBase< State, Primer >::rotate_acc_hist(), jeod::GaussJacksonIntegratorBase< State, Primer >::save_comparison_data(), jeod::GaussJacksonIntegratorBase< State, Primer >::size, and jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle().

Referenced by jeod::GaussJacksonFirstOrderODEIntegrator::integrate().

8.8.3.11 base_reset()

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::base_reset ( ) [inline], [protected]
```

Reset the integrator.

Definition at line 271 of file gauss_jackson_integrator_base.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::fsm_state, jeod::GaussJacksonIntegratorBase< State, Primer >::history_length, jeod::GaussJacksonIntegratorBase< State, Primer >::initial_order, jeod::GaussJacksonIntegratorBase< State, Primer >::order, and jeod::GaussJacksonStateMachine::Reset.

Referenced by jeod::GaussJacksonFirstOrderODEIntegrator::reset_integrator().

8.8.3.12 correct() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::correct (
    double dt,
    const double * acc,
    GaussJacksonOneState & state ) [inline], [private]
```

Apply the corrector.

Definition at line 252 of file gauss_jackson_integrator_base_first.hh.

References jeod::GaussJacksonOneState::first.

8.8.3.13 correct() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::correct (
    double dt,
    const double * acc,
    GaussJacksonTwoState & state ) [inline], [private]
```

Apply the corrector.

Definition at line 258 of file gauss_jackson_integrator_base_second.hh.

References jeod::GaussJacksonTwoState::first, and jeod::GaussJacksonTwoState::second.

8.8.3.14 correct() [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::correct (
    double dt,
    const double * acc,
    State & state ) [private]
```

Apply the corrector.

Parameters

<i>dt</i>	Dynamic time step.
<i>acc</i>	Acceleration data.
<i>state</i>	Corrected state.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj().

8.8.3.15 create_primer() [1/3]

```
template<>
er7_utils::SecondOrderODEIntegrator * jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState,
er7_utils::SecondOrderODEIntegrator >::create_primer (
    const er7_utils::IntegratorConstructor & priming_constructor,
    unsigned int size,
    er7_utils::IntegrationControls & priming_controls ) [inline], [private]
```

Definition at line 91 of file gauss_jackson_integrator_base_second.hh.

8.8.3.16 create_primer() [2/3]

```
template<>
er7_utils::FirstOrderODEIntegrator * jeod::GaussJacksonIntegratorBase< GaussJacksonOneState,
er7_utils::FirstOrderODEIntegrator >::create_primer (
    const er7_utils::IntegratorConstructor & priming_constructor,
    unsigned int size,
    er7_utils::IntegrationControls & priming_controls ) [inline], [private]
```

Create the priming integrator.

Definition at line 92 of file gauss_jackson_integrator_base_first.hh.

8.8.3.17 create_primer() [3/3]

```
template<typename State , typename Primer >
static Primer* jeod::GaussJacksonIntegratorBase< State, Primer >::create_primer (
    const er7_utils::IntegratorConstructor & priming_constructor,
    unsigned int size_in,
    er7_utils::IntegrationControls & priming_controls ) [static], [private]
```

Create the integrator to be used during priming.

Parameters

<i>priming_constructor</i>	Integrator constructor for the technique used during priming.
<i>size_in</i>	State size.
<i>priming_controls</i>	Integration controls used during priming.

Returns

Constructed primer.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase().

8.8.3.18 deallocate_state_contents() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::deallocate_state_contents (
    GaussJacksonOneState & item ) [inline], [private]
```

Deallocate state item memory.

Definition at line 325 of file gauss_jackson_integrator_base_first.hh.

References jeod::GaussJacksonOneState::first.

8.8.3.19 deallocate_state_contents() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::deallocate_state_contents (
    GaussJacksonTwoState & item ) [inline], [private]
```

Deallocate state item memory.

Definition at line 341 of file gauss_jackson_integrator_base_second.hh.

References jeod::GaussJacksonTwoState::first, and jeod::GaussJacksonTwoState::second.

8.8.3.20 deallocate_state_contents() [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::deallocate_state_contents (
    State & item ) [private]
```

Deallocate state item memory.

Parameters

<i>item</i>	State item to be deallocated.
-------------	-------------------------------

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::~GaussJacksonIntegratorBase().

8.8.3.21 downsample_hist()

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::downsample_hist ( ) [inline], [private]
```

Downsample the acceleration and position histories.

Definition at line 559 of file gauss_jackson_integrator_base.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::acc_hist, er7_utils::TwoDArray< T >::downsample(), jeod::GaussJacksonIntegratorBase< State, Primer >::history_length, and jeod::GaussJacksonIntegratorBase< State, Primer >::pos_hist.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle().

8.8.3.22 edit_point()

```
template<typename State , typename Primer >
bool jeod::GaussJacksonIntegratorBase< State, Primer >::edit_point (
    double dt,
    const double * acc,
    State & state ) [inline], [private]
```

Edit the specified point using the mid-corrector that pertains to the point being edited, which is that at history_length.

Parameters

in	<i>dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>acc</i>	Acceleration vector.
out	<i>state</i>	State vector(s).

Definition at line 511 of file gauss_jackson_integrator_base.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::advance_edit_integration_constants(), jeod::GaussJacksonIntegratorBase< State, Primer >::history_length, jeod::GaussJacksonIntegratorBase< State, Primer >::mid_correct(), jeod::GaussJacksonIntegratorBase< State, Primer >::order, jeod::GaussJacksonIntegratorBase< State, Primer >::pos_hist, and jeod::GaussJacksonIntegratorBase< State, Primer >::test_for_convergence().

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate().

8.8.3.23 initialize_edit_integration_constants() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator >::initialize_edit_integration_constants (
    double dt ) [inline], [private]
```

Initialize the integration constants (i.e., delinv).

Definition at line 153 of file gauss_jackson_integrator_base_second.hh.

8.8.3.24 initialize_edit_integration_constants() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::initialize_edit_integration_constants (
    double dt ) [inline], [private]
```

Initialize the integration constants (i.e., delinv).

Definition at line 155 of file gauss_jackson_integrator_base_first.hh.

8.8.3.25 initialize_edit_integration_constants() [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::initialize_edit_integration_constants
(
    double dt ) [private]
```

Initialize the integration constants (i.e., delinv).

Parameters

<i>dt</i>	Dynamic time step.
-----------	--------------------

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle().

8.8.3.26 initialize_predictor_integration_constants() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::initialize_predictor_integration_constants (
    double dt ) [inline], [private]
```

Initialize the integration constants (i.e., delinv).

Definition at line 185 of file gauss_jackson_integrator_base_second.hh.

8.8.3.27 initialize_predictor_integration_constants() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::initialize_predictor_integration_constants (
    double dt ) [inline], [private]
```

Initialize the integration constants (i.e., delinv).

Definition at line 186 of file gauss_jackson_integrator_base_first.hh.

8.8.3.28 initialize_predictor_integration_constants() [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::initialize_predictor_integration_↵
constants (
    double dt ) [private]
```

Initialize the integration constants (i.e., delinv).

Parameters

<i>dt</i>	Dynamic time step.
-----------	--------------------

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle().

8.8.3.29 integrate_gj()

```
template<typename State , typename Primer >
bool jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj (
    double dt,
    unsigned int target_stage,
    int advance_index,
    int target_index,
    const double * acc,
    const double *const * ahist,
    State & state ) [inline], [private]
```

Integrate using the Gauss-Jackson predictor and corrector.

Parameters

in	<i>dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in	<i>advance_index</i>	Acceleration history index.
in	<i>target_index</i>	Position history index.
in	<i>acc</i>	Acceleration vector.
in	<i>ahist</i>	Acceleration vector history.
out	<i>state</i>	State vector(s).

Returns

True if step was successful, false otherwise.

Definition at line 532 of file gauss_jackson_integrator_base.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::advance_predictor_integration_constants(), jeod::GaussJacksonCoefficientsPair::apply(), jeod::GaussJacksonIntegratorBase< State, Primer >::coeff, jeod↵::GaussJacksonIntegratorBase< State, Primer >::correct(), jeod::GaussJacksonCoeffs::corrector, jeod::Gauss↵JacksonIntegratorBase< State, Primer >::corrector_sum, jeod::GaussJacksonIntegratorBase< State, Primer

>::order, jeod::GaussJacksonIntegratorBase< State, Primer >::pos_hist, jeod::GaussJacksonIntegratorBase< State, Primer >::predict(), jeod::GaussJacksonIntegratorBase< State, Primer >::save_comparison_data(), jeod::GaussJacksonIntegratorBase< State, Primer >::size, and jeod::GaussJacksonIntegratorBase< State, Primer >::test_for_convergence().

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate().

8.8.3.30 integrate_primer() [1/3]

```
template<>
er7_utils::IntegratorResult jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator >::integrate_primer (
    double dyn_dt,
    unsigned int target_stage,
    const double * deriv,
    GaussJacksonTwoState & state ) [inline], [private]
```

Integrate with the primer.

Definition at line 121 of file gauss_jackson_integrator_base_second.hh.

References jeod::GaussJacksonTwoState::first, and jeod::GaussJacksonTwoState::second.

8.8.3.31 integrate_primer() [2/3]

```
template<>
er7_utils::IntegratorResult jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator >::integrate_primer (
    double dyn_dt,
    unsigned int target_stage,
    const double * deriv,
    GaussJacksonOneState & state ) [inline], [private]
```

Integrate with the primer.

Definition at line 123 of file gauss_jackson_integrator_base_first.hh.

References jeod::GaussJacksonOneState::first.

8.8.3.32 integrate_primer() [3/3]

```
template<typename State , typename Primer >
er7_utils::IntegratorResult jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_primer (
    double dyn_dt,
    unsigned int target_stage,
    const double * deriv,
    State & state ) [private]
```

Integrate state using the primer.

Parameters

in	<i>dyn_dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in	<i>deriv</i>	Acceleration vector.
in, out	<i>state</i>	State vector(s).

Returns

The status (time advance, pass/fail status) of the integration.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate()`.

8.8.3.33 mid_correct() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::mid_correct (
    unsigned int coeff_idx,
    double dt,
    GaussJacksonOneState & state ) [inline], [private]
```

Apply a mid-corrector.

Definition at line 216 of file `gauss_jackson_integrator_base_first.hh`.

References `jeod::GaussJacksonOneState::first`.

8.8.3.34 mid_correct() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::mid_correct (
    unsigned int coeff_idx,
    double dt,
    GaussJacksonTwoState & state ) [inline], [private]
```

Apply a mid-corrector.

Definition at line 218 of file `gauss_jackson_integrator_base_second.hh`.

References `jeod::GaussJacksonTwoState::first`, and `jeod::GaussJacksonTwoState::second`.

8.8.3.35 mid_correct() [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::mid_correct (
    unsigned int coeff_idx,
    double dt,
    State & state ) [private]
```

Apply a mid-corrector.

Parameters

<i>coeff_idx</i>	Coefficient index; item to be corrected.
<i>dt</i>	Dynamic time step.
<i>state</i>	Corrected state.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::edit_point()`.

8.8.3.36 operator=()

```
template<typename State , typename Primer >
GaussJacksonIntegratorBase& jeod::GaussJacksonIntegratorBase< State, Primer >::operator= (
    const GaussJacksonIntegratorBase< State, Primer > & ) [delete]
```

8.8.3.37 predict() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::predict (
    double dt,
    const double *const * ahist,
    GaussJacksonOneState & state ) [inline], [private]
```

Apply the predictor.

Definition at line 234 of file `gauss_jackson_integrator_base_first.hh`.

References `jeod::GaussJacksonOneState::first`.

8.8.3.38 predict() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::predict (
    double dt,
    const double *const * ahist,
    GaussJacksonTwoState & state ) [inline], [private]
```

Apply the predictor.

Definition at line 236 of file `gauss_jackson_integrator_base_second.hh`.

References `jeod::GaussJacksonTwoState::first`, and `jeod::GaussJacksonTwoState::second`.

8.8.3.39 predict() [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::predict (
    double dt,
    const double *const * ahist,
    State & state ) [private]
```

Apply the predictor.

Parameters

<i>dt</i>	Dynamic time step.
<i>ahist</i>	Acceleration history.
<i>state</i>	Corrected state.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj()`.

8.8.3.40 replicate_primer() [1/3]

```
template<>
er7_utils::SecondOrderODEIntegrator * jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState,
er7_utils::SecondOrderODEIntegrator >::replicate_primer (
    const er7_utils::SecondOrderODEIntegrator * src ) [inline], [private]
```

Definition at line 103 of file `gauss_jackson_integrator_base_second.hh`.

8.8.3.41 replicate_primer() [2/3]

```
template<>
er7_utils::FirstOrderODEIntegrator * jeod::GaussJacksonIntegratorBase< GaussJacksonOneState,
er7_utils::FirstOrderODEIntegrator >::replicate_primer (
    const er7_utils::FirstOrderODEIntegrator * src ) [inline], [private]
```

Replicate the priming integrator.

Definition at line 105 of file `gauss_jackson_integrator_base_first.hh`.

8.8.3.42 replicate_primer() [3/3]

```
template<typename State , typename Primer >
static Primer* jeod::GaussJacksonIntegratorBase< State, Primer >::replicate_primer (
    const Primer * src ) [static], [private]
```

Create a replica of the provided primer.

Parameters

<i>src</i>	Primer to be replicated.
------------	--------------------------

Returns

Constructed primer.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase().

8.8.3.43 replicate_state() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::replicate_state (
    const GaussJacksonOneState & source,
    GaussJacksonOneState & target ) [inline], [private]
```

Replicate state data.

Definition at line 303 of file gauss_jackson_integrator_base_first.hh.

References jeod::GaussJacksonOneState::first.

8.8.3.44 replicate_state() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::replicate_state (
    const GaussJacksonTwoState & source,
    GaussJacksonTwoState & target ) [inline], [private]
```

Replicate state data.

Definition at line 317 of file gauss_jackson_integrator_base_second.hh.

References jeod::GaussJacksonTwoState::first, and jeod::GaussJacksonTwoState::second.

8.8.3.45 replicate_state() [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::replicate_state (
    const State & source,
    State & target ) [private]
```

Replicate state data.

Parameters

<i>source</i>	State item to be copied.
<i>target</i>	Replicated state item.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase().

8.8.3.46 rotate_acc_hist()

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::rotate_acc_hist ( ) [inline], [private]
```

Rotate the acceleration history.

Definition at line 571 of file gauss_jackson_integrator_base.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::acc_hist, jeod::GaussJacksonIntegratorBase< State, Primer >::order, and er7_utils::TwoDArray< T >::rotate_down().

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate().

8.8.3.47 save_comparison_data() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::save_comparison_data (
    const GaussJacksonTwoState & state,
    double * pos_hist_elem ) [inline], [private]
```

Save comparison data.

Definition at line 142 of file gauss_jackson_integrator_base_second.hh.

References jeod::GaussJacksonTwoState::second.

8.8.3.48 save_comparison_data() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::save_comparison_data (
    const GaussJacksonOneState & state,
    double * pos_hist_elem ) [inline], [private]
```

Save comparison data.

Definition at line 144 of file gauss_jackson_integrator_base_first.hh.

References jeod::GaussJacksonOneState::first.

8.8.3.49 save_comparison_data() [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::save_comparison_data (
    const State & state,
    double * pos_hist_elem ) [private]
```

Save comparison data.

Parameters

<i>state</i>	State to be saved.
<i>pos_hist_elem</i>	Element of the position history to be updated.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate()`, and `jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj()`.

8.8.3.50 `save_epoch_data()` [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::save_epoch_data (
    const double * acc,
    const GaussJacksonTwoState & state ) [inline], [private]
```

Save epoch data.

Definition at line 131 of file `gauss_jackson_integrator_base_second.hh`.

References `jeod::GaussJacksonTwoState::first`, and `jeod::GaussJacksonTwoState::second`.

8.8.3.51 `save_epoch_data()` [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::save_epoch_data (
    const double * acc,
    const GaussJacksonOneState & state ) [inline], [private]
```

Save epoch data.

Definition at line 133 of file `gauss_jackson_integrator_base_first.hh`.

References `jeod::GaussJacksonOneState::first`.

8.8.3.52 `save_epoch_data()` [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::save_epoch_data (
    const double * acc,
    const State & state ) [private]
```

Save epoch data.

Parameters

<i>acc</i>	Acceleration to be saved.
<i>state</i>	State to be saved.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle()`.

8.8.3.53 start_cycle()

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle (
    double dt,
    const double * acc,
    const State & state ) [inline], [private]
```

Start an integration cycle.

Parameters

in	<i>dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>acc</i>	Acceleration vector.
in	<i>state</i>	State vector(s).

Definition at line 456 of file `gauss_jackson_integrator_base.hh`.

References `jeod::GaussJacksonIntegratorBase< State, Primer >::acc_hist`, `jeod::GaussJacksonStateMachine::BootstrapEdit`, `jeod::GaussJacksonIntegratorBase< State, Primer >::coeff`, `jeod::GaussJacksonCoeffs::corrector`, `jeod::GaussJacksonIntegratorBase< State, Primer >::downsample_hist()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::fsm_state`, `jeod::GaussJacksonStateMachine::get_at_downsample()`, `jeod::GaussJacksonStateMachine::get_at_order_change()`, `jeod::GaussJacksonStateMachine::get_at_reinitialize()`, `jeod::GaussJacksonStateMachine::get_current_order()`, `jeod::GaussJacksonStateMachine::get_fsm_state()`, `jeod::GaussJacksonCoefficientsPair::gj_coefs`, `jeod::GaussJacksonIntegratorBase< State, Primer >::history_length`, `jeod::GaussJacksonIntegratorBase< State, Primer >::initialize_edit_integration_constants()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::initialize_predictor_integration_constants()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::order`, `jeod::GaussJacksonIntegratorBase< State, Primer >::position_corrector`, `jeod::GaussJacksonStateMachine::Reset`, `jeod::GaussJacksonCoefficientsPair::sa_coefs`, `jeod::GaussJacksonIntegratorBase< State, Primer >::save_epoch_data()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::size`, `jeod::GaussJacksonIntegratorBase< State, Primer >::state_machine`, and `jeod::GaussJacksonIntegratorBase< State, Primer >::velocity_corrector`.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate()`.

8.8.3.54 swap()

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::swap (
    GaussJacksonIntegratorBase< State, Primer > & other ) [inline], [protected]
```

Non-throwing swap.

Parameters

<i>other</i>	Item whose contents are to be swapped with this.
--------------	--

Definition at line 397 of file gauss_jackson_integrator_base.hh.

References jeod::GaussJacksonIntegratorBase< State, Primer >::absolute_tolerance, jeod::GaussJacksonIntegratorBase< State, Primer >::acc_hist, jeod::GaussJacksonIntegratorBase< State, Primer >::coeff, jeod::GaussJacksonIntegratorBase< State, Primer >::corrector_sum, jeod::GaussJacksonIntegratorBase< State, Primer >::delinv, jeod::GaussJacksonIntegratorBase< State, Primer >::fsm_state, jeod::GaussJacksonIntegratorBase< State, Primer >::history_length, jeod::GaussJacksonIntegratorBase< State, Primer >::init_state, jeod::GaussJacksonIntegratorBase< State, Primer >::initial_order, jeod::GaussJacksonIntegratorBase< State, Primer >::max_history_size, jeod::GaussJacksonIntegratorBase< State, Primer >::order, jeod::GaussJacksonIntegratorBase< State, Primer >::pos_hist, jeod::GaussJacksonIntegratorBase< State, Primer >::position_corrector, jeod::GaussJacksonIntegratorBase< State, Primer >::primer, jeod::GaussJacksonIntegratorBase< State, Primer >::relative_tolerance, jeod::GaussJacksonIntegratorBase< State, Primer >::size, jeod::GaussJacksonIntegratorBase< State, Primer >::state_machine, er7_utils::TwoDArray< T >::swap(), jeod::GaussJacksonIntegratorBase< State, Primer >::swap_state(), and jeod::GaussJacksonIntegratorBase< State, Primer >::velocity_corrector.

Referenced by jeod::GaussJacksonFirstOrderODEIntegrator::swap().

8.8.3.55 swap_state() [1/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator >::swap_state (
    GaussJacksonOneState & item,
    GaussJacksonOneState & other_item ) [inline], [private]
```

Swap state data with another of the same.

Definition at line 293 of file gauss_jackson_integrator_base_first.hh.

References jeod::GaussJacksonOneState::first.

8.8.3.56 swap_state() [2/3]

```
template<>
void jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator >::swap_state (
    GaussJacksonTwoState & item,
    GaussJacksonTwoState & other_item ) [inline], [private]
```

Swap state data with another of the same.

Definition at line 306 of file gauss_jackson_integrator_base_second.hh.

References jeod::GaussJacksonTwoState::first, and jeod::GaussJacksonTwoState::second.

8.8.3.57 `swap_state()` [3/3]

```
template<typename State , typename Primer >
void jeod::GaussJacksonIntegratorBase< State, Primer >::swap_state (
    State & item,
    State & other_item ) [private]
```

Swap state data with another of the same.

Parameters

<i>item</i>	State item.
<i>other_item</i>	The other state item.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::swap()`.

8.8.3.58 test_for_convergence() [1/3]

```
template<>
bool jeod::GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator
>::test_for_convergence (
    const GaussJacksonOneState & state,
    double * hist_data ) [inline], [private]
```

Test for convergence.

Definition at line 271 of file `gauss_jackson_integrator_base_first.hh`.

References `jeod::GaussJacksonOneState::first`.

8.8.3.59 test_for_convergence() [2/3]

```
template<>
bool jeod::GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator
>::test_for_convergence (
    const GaussJacksonTwoState & state,
    double * hist_data ) [inline], [private]
```

Test for convergence.

Definition at line 285 of file `gauss_jackson_integrator_base_second.hh`.

References `jeod::GaussJacksonTwoState::second`.

8.8.3.60 test_for_convergence() [3/3]

```
template<typename State , typename Primer >
bool jeod::GaussJacksonIntegratorBase< State, Primer >::test_for_convergence (
    const State & state,
    double * hist_data ) [private]
```

Test for convergence.

Parameters

<i>state</i>	Item to be compared.
<i>hist_data</i>	Previous state value.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::edit_point()`, and `jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj()`.

8.8.4 Field Documentation**8.8.4.1 absolute_tolerance**

```
template<typename State , typename Primer >
double jeod::GaussJacksonIntegratorBase< State, Primer >::absolute_tolerance {}
```

Number that indicates the allowable absolute difference for two states to be considered converged.

`trick_units(-)`

Definition at line 143 of file `gauss_jackson_integrator_base.hh`.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::swap()`.

8.8.4.2 acc_hist

```
template<typename State , typename Primer >
er7_utils::DoubleTwoDArray jeod::GaussJacksonIntegratorBase< State, Primer >::acc_hist
```

Acceleration history.

`trick_units(-)`

Definition at line 126 of file `gauss_jackson_integrator_base.hh`.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::downsample_hist()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::rotate_acc_hist()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle()`, and `jeod::GaussJacksonIntegratorBase< State, Primer >::swap()`.

8.8.4.3 coeff

```
template<typename State , typename Primer >
const GaussJacksonCoeffs* jeod::GaussJacksonIntegratorBase< State, Primer >::coeff {}
```

The summed Adams and Gauss-Jackson coefficients, in ordinate form.

trick_units(—)

Definition at line 96 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj(), jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.4 corrector_sum

```
template<typename State , typename Primer >
State jeod::GaussJacksonIntegratorBase< State, Primer >::corrector_sum
```

Speed hack for the corrector.

trick_units(—)

Definition at line 121 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase(), jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj(), jeod::GaussJacksonIntegratorBase< State, Primer >::swap(), and jeod::GaussJacksonIntegratorBase< State, Primer >::~~GaussJacksonIntegratorBase().

8.8.4.5 delinv

```
template<typename State , typename Primer >
State jeod::GaussJacksonIntegratorBase< State, Primer >::delinv
```

Inverse backward differences.

trick_units(—)

Definition at line 116 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase(), jeod::GaussJacksonIntegratorBase< State, Primer >::swap(), and jeod::GaussJacksonIntegratorBase< State, Primer >::~~GaussJacksonIntegratorBase().

8.8.4.6 fsm_state

```
template<typename State , typename Primer >
GaussJacksonStateMachine::FsmState jeod::GaussJacksonIntegratorBase< State, Primer >::fsm_↵
state {GaussJacksonStateMachine::Reset}
```

Finite state machine state.

trick_units(-)

Definition at line 158 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate(), jeod::GaussJackson↵
IntegratorBase< State, Primer >::base_reset(), jeod::GaussJacksonIntegratorBase< State, Primer >::start_↵
cycle(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.7 history_length

```
template<typename State , typename Primer >
unsigned int jeod::GaussJacksonIntegratorBase< State, Primer >::history_length {}
```

Current history length.

trick_units(-)

Definition at line 183 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate(), jeod::GaussJackson↵
IntegratorBase< State, Primer >::base_reset(), jeod::GaussJacksonIntegratorBase< State, Primer >↵
::downsample_hist(), jeod::GaussJacksonIntegratorBase< State, Primer >::edit_point(), jeod::GaussJackson↵
IntegratorBase< State, Primer >::start_cycle(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.8 init_state

```
template<typename State , typename Primer >
State jeod::GaussJacksonIntegratorBase< State, Primer >::init_state
```

The state at the time of the last reset.

trick_units(-)

Definition at line 111 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase(), jeod::↵
GaussJacksonIntegratorBase< State, Primer >::swap(), and jeod::GaussJacksonIntegratorBase< State, Primer ↵
>::~~GaussJacksonIntegratorBase().

8.8.4.9 initial_order

```
template<typename State , typename Primer >
unsigned int jeod::GaussJacksonIntegratorBase< State, Primer >::initial_order {}
```

Initial order.

trick_units(-)

Definition at line 168 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::base_reset(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.10 max_history_size

```
template<typename State , typename Primer >
unsigned int jeod::GaussJacksonIntegratorBase< State, Primer >::max_history_size {}
```

Maximum history size.

trick_units(-)

Definition at line 163 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.11 order

```
template<typename State , typename Primer >
unsigned int jeod::GaussJacksonIntegratorBase< State, Primer >::order {}
```

Current order.

trick_units(-)

Definition at line 173 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate(), jeod::GaussJacksonIntegratorBase< State, Primer >::base_reset(), jeod::GaussJacksonIntegratorBase< State, Primer >::edit_point(), jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj(), jeod::GaussJacksonIntegratorBase< State, Primer >::rotate_acc_hist(), jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.12 pos_hist

```
template<typename State , typename Primer >
er7_utils::DoubleTwoDArray jeod::GaussJacksonIntegratorBase< State, Primer >::pos_hist
```

Position history (or velocity history in case of a first order ODE).

trick_units(-)

Definition at line 131 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate(), jeod::GaussJacksonIntegratorBase< State, Primer >::downsample_hist(), jeod::GaussJacksonIntegratorBase< State, Primer >::edit_point(), jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase(), jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.13 position_corrector

```
template<typename State , typename Primer >
double jeod::GaussJacksonIntegratorBase< State, Primer >::position_corrector {}
```

Correction coefficient for the second integral (position).

trick_units(-)

Definition at line 153 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.14 primer

```
template<typename State , typename Primer >
Primer* jeod::GaussJacksonIntegratorBase< State, Primer >::primer {}
```

The integrator used to prime the Gauss-Jackson integration process.

trick_units(-)

Definition at line 106 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase(), jeod::GaussJacksonIntegratorBase< State, Primer >::swap(), and jeod::GaussJacksonIntegratorBase< State, Primer >::~~GaussJacksonIntegratorBase().

8.8.4.15 relative_tolerance

```
template<typename State , typename Primer >
double jeod::GaussJacksonIntegratorBase< State, Primer >::relative_tolerance {}
```

Number that indicates the allowable relative difference for two states to be considered converged.

trick_units(-)

Definition at line 137 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.16 size

```
template<typename State , typename Primer >
unsigned int jeod::GaussJacksonIntegratorBase< State, Primer >::size {}
```

State size.

trick_units(-)

Definition at line 178 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::base_integrate(), jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase(), jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_gj(), jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.17 state_machine

```
template<typename State , typename Primer >
const GaussJacksonStateMachine* jeod::GaussJacksonIntegratorBase< State, Primer >::state_machine {}
```

The Gauss-Jackson state machine.

trick_units(-)

Definition at line 101 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.8.4.18 velocity_corrector

```
template<typename State , typename Primer >
double jeod::GaussJacksonIntegratorBase< State, Primer >::velocity_corrector {}
```

Correction coefficient for the first integral (velocity).

trick_units(-)

Definition at line 148 of file gauss_jackson_integrator_base.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::start_cycle(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

The documentation for this class was generated from the following file:

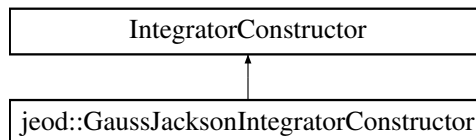
- [gauss_jackson_integrator_base.hh](#)

8.9 jeod::GaussJacksonIntegratorConstructor Class Reference

Create state and time integrators that propagate using Gauss-Jackson.

```
#include <gauss_jackson_integrator_constructor.hh>
```

Inheritance diagram for jeod::GaussJacksonIntegratorConstructor:



Public Member Functions

- [GaussJacksonIntegratorConstructor](#) ()=default
GaussJackson default constructor.
- [GaussJacksonIntegratorConstructor](#) (const [GaussJacksonIntegratorConstructor](#) &src)
GaussJacksonIntegratorConstructor copy constructor.
- [~GaussJacksonIntegratorConstructor](#) () override
GaussJacksonIntegratorConstructor destructor.
- [GaussJacksonIntegratorConstructor](#) & operator= ([GaussJacksonIntegratorConstructor](#) src)
GaussJacksonIntegratorConstructor assignment operator.
- void [configure](#) (const [GaussJacksonConfig](#) &config_in, er7_utils::Integration::Technique priming_↵ technique=er7_utils::Integration::Unspecified)
Configure the Gauss-Jackson integrator constructor.
- void [configure](#) (const [GaussJacksonConfig](#) &config_in, const er7_utils::IntegratorConstructor &priming_↵ cotr_in)
Configure the Gauss-Jackson integrator constructor.
- const char * [get_class_name](#) () const override
Return the class name.
- bool [implements](#) (er7_utils::Integration::ODEProblemType problem_type) const override

- GaussJackson does not implement a 2nd order generalized step integrator.*
- bool [provides](#) (er7_utils::Integration::ODEProblemType problem_type) const override
GaussJackson does not provide a 2nd order generalized step integrator.
- virtual void [swap](#) ([GaussJacksonIntegratorConstructor](#) &src)
Non-throwing swap.
- er7_utils::IntegratorConstructor * [create_copy](#) () const override
Create a duplicate of the constructor.
- er7_utils::IntegrationControls * [create_integration_controls](#) () const override
Create an integration controls that guides the GaussJackson integration process.
- er7_utils::FirstOrderODEIntegrator * [create_first_order_ode_integrator](#) (unsigned int size, er7_utils::IntegrationControls &controls) const override
Create a GaussJackson state integrator for a first order ODE.
- er7_utils::SecondOrderODEIntegrator * [create_second_order_ode_integrator](#) (unsigned int size, er7_utils::IntegrationControls &controls) const override
Create a GaussJackson state integrator for a simple second order ODE.
- er7_utils::SecondOrderODEIntegrator * [create_generalized_deriv_second_order_ode_integrator](#) (unsigned int position_size, unsigned int velocity_size, const er7_utils::GeneralizedPositionDerivativeFunctions &deriv_funcs, er7_utils::IntegrationControls &controls) const override
Create a GaussJackson state integrator for a generalized second order ODE where generalized position is advanced with the use of the position derivative function.
- unsigned int [get_buffer_size](#) () const override
GaussJackson can use a large number of steps per Trick cycle.
- unsigned int [get_transition_table_size](#) () const override
GaussJackson uses two steps per cycle once primed.

Static Public Member Functions

- static er7_utils::IntegratorConstructor * [create_constructor](#) ()
Named constructor; create an [GaussJacksonIntegratorConstructor](#) instance.

Private Attributes

- er7_utils::IntegratorConstructor * [priming_constructor](#) {}
The integrator constructor that creates the priming integrators.
- [GaussJacksonConfig](#) [config](#)
Data used to configure the Gauss-Jackson integration process.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__GaussJacksonIntegratorConstructor](#) ()

8.9.1 Detailed Description

Create state and time integrators that propagate using Gauss-Jackson.

Definition at line 80 of file `gauss_jackson_integrator_constructor.hh`.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 GaussJacksonIntegratorConstructor() [1/2]

```
jeod::GaussJacksonIntegratorConstructor::GaussJacksonIntegratorConstructor ( ) [default]
```

GaussJackson default constructor.

8.9.2.2 GaussJacksonIntegratorConstructor() [2/2]

```
jeod::GaussJacksonIntegratorConstructor::GaussJacksonIntegratorConstructor (
    const GaussJacksonIntegratorConstructor & src )
```

[GaussJacksonIntegratorConstructor](#) copy constructor.

Definition at line 69 of file gauss_jackson_integrator_constructor.cc.

References [priming_constructor](#).

8.9.2.3 ~GaussJacksonIntegratorConstructor()

```
jeod::GaussJacksonIntegratorConstructor::~~GaussJacksonIntegratorConstructor ( ) [override]
```

[GaussJacksonIntegratorConstructor](#) destructor.

Definition at line 80 of file gauss_jackson_integrator_constructor.cc.

References [priming_constructor](#).

8.9.3 Member Function Documentation

8.9.3.1 configure() [1/2]

```
void jeod::GaussJacksonIntegratorConstructor::configure (
    const GaussJacksonConfig & config_in,
    er7_utils::Integration::Technique priming_technique = er7_utils::Integration::↔
:Unspecified )
```

Configure the Gauss-Jackson integrator constructor.

Definition at line 93 of file gauss_jackson_integrator_constructor.cc.

References [config](#), [priming_constructor](#), [jeod::GaussJacksonConfig::priming_technique](#), and [jeod::GaussJacksonConfig::validate_configuration\(\)](#).

8.9.3.2 configure() [2/2]

```
void jeod::GaussJacksonIntegratorConstructor::configure (
    const GaussJacksonConfig & config_in,
    const er7_utils::IntegratorConstructor & priming_cotr_in )
```

Configure the Gauss-Jackson integrator constructor.

Definition at line 107 of file gauss_jackson_integrator_constructor.cc.

References `config`, `priming_constructor`, and `jeod::GaussJacksonConfig::validate_configuration()`.

8.9.3.3 create_constructor()

```
er7_utils::IntegratorConstructor * jeod::GaussJacksonIntegratorConstructor::create_constructor
( ) [static]
```

Named constructor; create an [GaussJacksonIntegratorConstructor](#) instance.

The caller is responsible for deleting the returned object.

Returns

Newly created [GaussJacksonIntegratorConstructor](#) instance.

Definition at line 63 of file gauss_jackson_integrator_constructor.cc.

8.9.3.4 create_copy()

```
er7_utils::IntegratorConstructor * jeod::GaussJacksonIntegratorConstructor::create_copy ( )
const [override]
```

Create a duplicate of the constructor.

The caller is responsible for deleting the returned object.

Returns

Duplicated constructor.

Definition at line 115 of file gauss_jackson_integrator_constructor.cc.

8.9.3.5 create_first_order_ode_integrator()

```
er7_utils::FirstOrderODEIntegrator * jeod::GaussJacksonIntegratorConstructor::create_first_order_ode_integrator (
    unsigned int size,
    er7_utils::IntegrationControls & controls ) const [override]
```

Create a GaussJackson state integrator for a first order ODE.

The caller is responsible for deleting the created object.

Returns

State integrator

Parameters

<i>in</i>	<i>size</i>	State size
<i>in, out</i>	<i>controls</i>	Integration controls

Definition at line 136 of file `gauss_jackson_integrator_constructor.cc`.

References `jeod::cast_to_gj_controls()`, `jeod::GaussJacksonIntegrationControls::get_priming_controls()`, and `priming_constructor`.

8.9.3.6 `create_generalized_deriv_second_order_ode_integrator()`

```
er7_utils::SecondOrderODEIntegrator * jeod::GaussJacksonIntegratorConstructor::create_generalized←
_deriv_second_order_ode_integrator (
    unsigned int position_size,
    unsigned int velocity_size,
    const er7_utils::GeneralizedPositionDerivativeFunctions & deriv_funs,
    er7_utils::IntegrationControls & controls ) const [override]
```

Create a GaussJackson state integrator for a generalized second order ODE where generalized position is advanced with the use of the position derivative function.

The caller is responsible for deleting the created object.

Returns

State integrator

Parameters

<i>in</i>	<i>position_size</i>	Size of the generalized position
<i>in</i>	<i>velocity_size</i>	Size of the generalized velocity
<i>in</i>	<i>deriv_funs</i>	Position derivative functions container
<i>in, out</i>	<i>controls</i>	Integration controls

Definition at line 169 of file `gauss_jackson_integrator_constructor.cc`.

References `jeod::cast_to_gj_controls()`, `jeod::GaussJacksonIntegrationControls::get_priming_controls()`, and `priming_constructor`.

8.9.3.7 `create_integration_controls()`

```
er7_utils::IntegrationControls * jeod::GaussJacksonIntegratorConstructor::create_integration←
_controls ( ) const [override]
```

Create an integration controls that guides the GaussJackson integration process.

The caller is responsible for deleting the created object.

Returns

Integration controls object

Definition at line 121 of file gauss_jackson_integrator_constructor.cc.

References `config`, `priming_constructor`, and `jeod::GaussJacksonConfig::standard_configuration()`.

8.9.3.8 create_second_order_ode_integrator()

```
er7_utils::SecondOrderODEIntegrator * jeod::GaussJacksonIntegratorConstructor::create_second_order_ode_integrator (
    unsigned int size,
    er7_utils::IntegrationControls & controls ) const [override]
```

Create a GaussJackson state integrator for a simple second order ODE.

The caller is responsible for deleting the created object.

Returns

State integrator

Parameters

<code>in</code>	<code>size</code>	State size
<code>in, out</code>	<code>controls</code>	Integration controls

Definition at line 152 of file gauss_jackson_integrator_constructor.cc.

References `jeod::cast_to_gj_controls()`, `jeod::GaussJacksonIntegrationControls::get_priming_controls()`, and `priming_constructor`.

8.9.3.9 get_buffer_size()

```
unsigned int jeod::GaussJacksonIntegratorConstructor::get_buffer_size ( ) const [inline],
[override]
```

GaussJackson can use a large number of steps per Trick cycle.

The magic number 192 is for order=16, ndboubling=6.

Returns

Always returns 192.

Definition at line 217 of file gauss_jackson_integrator_constructor.hh.

8.9.3.10 get_class_name()

```
const char* jeod::GaussJacksonIntegratorConstructor::get_class_name ( ) const [inline], [override]
```

Return the class name.

Definition at line 134 of file gauss_jackson_integrator_constructor.hh.

8.9.3.11 get_transition_table_size()

```
unsigned int jeod::GaussJacksonIntegratorConstructor::get_transition_table_size ( ) const [inline], [override]
```

GaussJackson uses two steps per cycle once primed.

Returns

Always returns 2.

Definition at line 226 of file gauss_jackson_integrator_constructor.hh.

8.9.3.12 implements()

```
bool jeod::GaussJacksonIntegratorConstructor::implements (
    er7_utils::Integration::ODEProblemType problem_type ) const [inline], [override]
```

GaussJackson does not implement a 2nd order generalized step integrator.

Definition at line 142 of file gauss_jackson_integrator_constructor.hh.

8.9.3.13 operator=()

```
GaussJacksonIntegratorConstructor& jeod::GaussJacksonIntegratorConstructor::operator= (
    GaussJacksonIntegratorConstructor src ) [inline]
```

[GaussJacksonIntegratorConstructor](#) assignment operator.

Definition at line 112 of file gauss_jackson_integrator_constructor.hh.

8.9.3.14 provides()

```
bool jeod::GaussJacksonIntegratorConstructor::provides (
    er7_utils::Integration::ODEProblemType problem_type ) const [inline], [override]
```

GaussJackson does not provide a 2nd order generalized step integrator.

Definition at line 150 of file gauss_jackson_integrator_constructor.hh.

8.9.3.15 swap()

```
void jeod::GaussJacksonIntegratorConstructor::swap (
    GaussJacksonIntegratorConstructor & src ) [virtual]
```

Non-throwing swap.

Parameters

<code>in, out</code>	<code>src</code>	Object with which contents are to be swapped.
----------------------	------------------	---

Definition at line 86 of file `gauss_jackson_integrator_constructor.cc`.

References `config`, and `priming_constructor`.

8.9.4 Friends And Related Function Documentation**8.9.4.1 `init_attrjeod__GaussJacksonIntegratorConstructor`**

```
void init_attrjeod__GaussJacksonIntegratorConstructor ( ) [friend]
```

8.9.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 82 of file `gauss_jackson_integrator_constructor.hh`.

8.9.5 Field Documentation**8.9.5.1 `config`**

```
GaussJacksonConfig jeod::GaussJacksonIntegratorConstructor::config [private]
```

Data used to configure the Gauss-Jackson integration process.

`trick_units(-)`

Definition at line 240 of file `gauss_jackson_integrator_constructor.hh`.

Referenced by `configure()`, `create_integration_controls()`, and `swap()`.

8.9.5.2 priming_constructor

```
er7_utils::IntegratorConstructor* jeod::GaussJacksonIntegratorConstructor::priming_constructor
{} [private]
```

The integrator constructor that creates the priming integrators.

trick_units(-)

Definition at line 235 of file gauss_jackson_integrator_constructor.hh.

Referenced by `configure()`, `create_first_order_ode_integrator()`, `create_generalized_deriv_second_order_ode_integrator()`, `create_integration_controls()`, `create_second_order_ode_integrator()`, `GaussJacksonIntegratorConstructor()`, `swap()`, and `~GaussJacksonIntegratorConstructor()`.

The documentation for this class was generated from the following files:

- [gauss_jackson_integrator_constructor.hh](#)
- [gauss_jackson_integrator_constructor.cc](#)

8.10 jeod::GaussJacksonOneState Class Reference

Essentially just a double*.

```
#include <gauss_jackson_one_state.hh>
```

Public Member Functions

- [GaussJacksonOneState](#) ()=default
Default constructor.
- [GaussJacksonOneState](#) (double *first_in)
Conversion constructor.

Data Fields

- double * [first](#) {}
The pointed-to data.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__GaussJacksonOneState](#) ()

8.10.1 Detailed Description

Essentially just a double*.

Definition at line 71 of file gauss_jackson_one_state.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 GaussJacksonOneState() [1/2]

```
jeod::GaussJacksonOneState::GaussJacksonOneState ( ) [default]
```

Default constructor.

8.10.2.2 GaussJacksonOneState() [2/2]

```
jeod::GaussJacksonOneState::GaussJacksonOneState (
    double * first_in ) [inline], [explicit]
```

Conversion constructor.

Parameters

<i>first_in</i>	The pointed-to data.
-----------------	----------------------

Definition at line 88 of file gauss_jackson_one_state.hh.

8.10.3 Friends And Related Function Documentation

8.10.3.1 init_attrjeod__GaussJacksonOneState

```
void init_attrjeod__GaussJacksonOneState ( ) [friend]
```

8.10.3.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 73 of file gauss_jackson_one_state.hh.

8.10.4 Field Documentation

8.10.4.1 first

```
double* jeod::GaussJacksonOneState::first {}
```

The pointed-to data.

```
trick_units(-)
```

Definition at line 77 of file gauss_jackson_one_state.hh.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::allocate_state_contents()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::correct()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::deallocate_state_contents()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_primer()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::mid_correct()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::predict()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::replicate_state()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::save_comparison_data()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::save_epoch_data()`, `jeod::GaussJacksonIntegratorBase< State, Primer >::swap_state()`, and `jeod::GaussJacksonIntegratorBase< State, Primer >::test_for_convergence()`.

The documentation for this class was generated from the following file:

- [gauss_jackson_one_state.hh](#)

8.11 jeod::GaussJacksonRationalCoefficients Class Reference

Contains a set of Adams or Stormer-Cowell coefficients.

```
#include <gauss_jackson_rational_coeffs.hh>
```

Public Member Functions

- [GaussJacksonRationalCoefficients](#) ()=default
Default constructor.
- void [configure_adams_corrector](#) (unsigned int nelelem)
Configure the coefficients as an Adams corrector in difference form.
- [GaussJacksonRationalCoefficients construct_stormer_cowell_corrector](#) () const
Construct a [GaussJacksonRationalCoefficients](#) that contains the Stormer-Cowell corrector coefficients.
- [GaussJacksonRationalCoefficients construct_predictor](#) () const
Construct a [GaussJacksonRationalCoefficients](#) that contains a set of predictor coefficients.
- void [convert_to_ordinate_form](#) (er7_utils::NChooseM &n_choose_m, double *result) const
Convert the coefficients to ordinate form.
- void [discard_extra_terms](#) (unsigned int nfront, unsigned int nback)
Discard the specified number of terms from the front and back of the coefficients array.
- void [displace_back](#) ()
Displace the corrector coefficients back one time step.

Data Fields

- `std::vector< er7_utils::Ratio128 >` [coefficients](#)
The coefficients.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__GaussJacksonRationalCoefficients](#) ()

8.11.1 Detailed Description

Contains a set of Adams or Stormer-Cowell coefficients.

Definition at line 78 of file `gauss_jackson_rational_coeffs.hh`.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 GaussJacksonRationalCoefficients()

```
jeod::GaussJacksonRationalCoefficients::GaussJacksonRationalCoefficients ( ) [default]
```

Default constructor.

8.11.3 Member Function Documentation

8.11.3.1 configure_adams_corrector()

```
void jeod::GaussJacksonRationalCoefficients::configure_adams_corrector (
    unsigned int nelem )
```

Configure the coefficients as an Adams corrector in difference form.

Parameters

<i>nelem</i>	The number of elements in the coefficients vector.
--------------	--

Definition at line 32 of file `gauss_jackson_rational_coeffs.cc`.

References `coefficients`.

Referenced by `jeod::GaussJacksonCoeffs::compute_coeffs()`.

8.11.3.2 `construct_predictor()`

```
GaussJacksonRationalCoefficients jeod::GaussJacksonRationalCoefficients::construct_predictor (
) const
```

Construct a [GaussJacksonRationalCoefficients](#) that contains a set of predictor coefficients.

The coefficients are assumed to be configured as either Adams or Stormer-Cowell corrector coefficients.

Returns

A [GaussJacksonRationalCoefficients](#) object with the coefficients configured as Adams or Stormer-Cowell predictor coefficients.

Definition at line 78 of file `gauss_jackson_rational_coeffs.cc`.

References coefficients.

Referenced by `jeod::GaussJacksonCoeffs::compute_coeffs()`.

8.11.3.3 `construct_stormer_cowell_corrector()`

```
GaussJacksonRationalCoefficients jeod::GaussJacksonRationalCoefficients::construct_stormer_↵
cowell_corrector ( ) const
```

Construct a [GaussJacksonRationalCoefficients](#) that contains the Stormer-Cowell corrector coefficients.

The coefficients are assumed to be configured as Adams coefficients in difference form.

Returns

A [GaussJacksonRationalCoefficients](#) object with the coefficients configured as Stormer-Cowell corrector coefficients.

Definition at line 54 of file `gauss_jackson_rational_coeffs.cc`.

References coefficients.

Referenced by `jeod::GaussJacksonCoeffs::compute_coeffs()`.

8.11.3.4 `convert_to_ordinate_form()`

```
void jeod::GaussJacksonRationalCoefficients::convert_to_ordinate_form (
    er7_utils::NChooseM & n_choose_m,
    double * result ) const
```

Convert the coefficients to ordinate form.

Parameters

<i>n_choose_m</i>	An NChooseM object that computes N choose M.
<i>result</i>	The output ordinate form coefficients.

Definition at line 101 of file gauss_jackson_rational_coeffs.cc.

References coefficients.

Referenced by jeod::GaussJacksonCoeffs::compute_coeffs().

8.11.3.5 discard_extra_terms()

```
void jeod::GaussJacksonRationalCoefficients::discard_extra_terms (
    unsigned int nfront,
    unsigned int nback )
```

Discard the specified number of terms from the front and back of the coefficients array.

Parameters

<i>nfront</i>	The number of terms to be discarded from the front of the coefficients vector.
<i>nback</i>	The number of terms to be discarded from the back of the coefficients vector.

Definition at line 133 of file gauss_jackson_rational_coeffs.cc.

References coefficients.

Referenced by jeod::GaussJacksonCoeffs::compute_coeffs().

8.11.3.6 displace_back()

```
void jeod::GaussJacksonRationalCoefficients::displace_back ( )
```

Displace the corrector coefficients back one time step.

Definition at line 146 of file gauss_jackson_rational_coeffs.cc.

References coefficients.

Referenced by jeod::GaussJacksonCoeffs::compute_coeffs().

8.11.4 Friends And Related Function Documentation

8.11.4.1 init_attrjeod__GaussJacksonRationalCoefficients

```
void init_attrjeod__GaussJacksonRationalCoefficients ( ) [friend]
```

8.11.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 80 of file gauss_jackson_rational_coeffs.hh.

8.11.5 Field Documentation

8.11.5.1 coefficients

```
std::vector<er7_utils::Ratiol28> jeod::GaussJacksonRationalCoefficients::coefficients
```

The coefficients.

trick_units(-)

Definition at line 84 of file gauss_jackson_rational_coeffs.hh.

Referenced by `configure_adams_corrector()`, `construct_predictor()`, `construct_stormer_cowell_corrector()`, `convert_to_ordinate_form()`, `discard_extra_terms()`, and `displace_back()`.

The documentation for this class was generated from the following files:

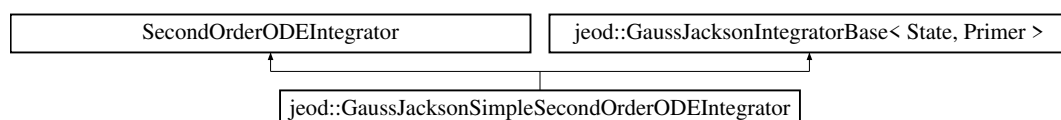
- [gauss_jackson_rational_coeffs.hh](#)
- [gauss_jackson_rational_coeffs.cc](#)

8.12 jeod::GaussJacksonSimpleSecondOrderODEIntegrator Class Reference

Integrates a simple second order ODE using the Gauss-Jackson technique.

```
#include <gauss_jackson_simple_second_order_ode_integrator.hh>
```

Inheritance diagram for jeod::GaussJacksonSimpleSecondOrderODEIntegrator:



Public Member Functions

- [GaussJacksonSimpleSecondOrderODEIntegrator](#) ()=default
- [~GaussJacksonSimpleSecondOrderODEIntegrator](#) () override=default
- [GaussJacksonSimpleSecondOrderODEIntegrator](#) (const [er7_utils::IntegratorConstructor](#) &priming_constructor, [GaussJacksonIntegrationControls](#) &controls, unsigned int size_in, [er7_utils::IntegrationControls](#) &priming_controls)
Non-default constructor.
- [GaussJacksonSimpleSecondOrderODEIntegrator](#) (const [GaussJacksonSimpleSecondOrderODEIntegrator](#) &src)
Copy constructor.
- [GaussJacksonSimpleSecondOrderODEIntegrator](#) & operator= ([GaussJacksonSimpleSecondOrderODEIntegrator](#) src)
Copy and swap assignment operator.
- void [swap](#) ([GaussJacksonSimpleSecondOrderODEIntegrator](#) &other)
Non-throwing swap.
- [er7_utils::SecondOrderODEIntegrator](#) * [create_copy](#) () const override
Replicate this.
- void [reset_integrator](#) () override
Reset the integrator.
- [er7_utils::IntegratorResult](#) [integrate](#) (double dyn_dt, unsigned int target_stage, const double *acc, double *vel, double *pos) override
Propagate state using Gauss-Jackson.

Private Member Functions

- void [swap](#) ([GaussJacksonIntegratorBase](#) &other)
Non-throwing swap.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__GaussJacksonSimpleSecondOrderODEIntegrator](#) ()

Additional Inherited Members

8.12.1 Detailed Description

Integrates a simple second order ODE using the Gauss-Jackson technique.

The class inherits from [er7_utils::SecondOrderODEIntegrator](#) as an is-a relationship (public inheritance) and from [GaussJacksonIntegratorBaseSecond](#) as an is-implemented-by relationship (private inheritance). Using composition instead of private inheritance would make Trick 13 checkpoint/restart a lot trickier to implement. With private inheritance, the Trick 13 `io_src` file contains all the necessary information.

Definition at line 83 of file `gauss_jackson_simple_second_order_ode_integrator.hh`.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 GaussJacksonSimpleSecondOrderODEIntegrator() [1/3]

```
jeod::GaussJacksonSimpleSecondOrderODEIntegrator::GaussJacksonSimpleSecondOrderODEIntegrator (
) [default]
```

8.12.2.2 ~GaussJacksonSimpleSecondOrderODEIntegrator()

```
jeod::GaussJacksonSimpleSecondOrderODEIntegrator::~~GaussJacksonSimpleSecondOrderODEIntegrator
( ) [override], [default]
```

8.12.2.3 GaussJacksonSimpleSecondOrderODEIntegrator() [2/3]

```
jeod::GaussJacksonSimpleSecondOrderODEIntegrator::GaussJacksonSimpleSecondOrderODEIntegrator (
    const er7_utils::IntegratorConstructor & priming_constructor,
    GaussJacksonIntegrationControls & controls,
    unsigned int size_in,
    er7_utils::IntegrationControls & priming_controls ) [inline]
```

Non-default constructor.

This is the constructor invoked by the [GaussJacksonIntegratorConstructor](#).

Parameters

<i>priming_constructor</i>	Integrator constructor for the technique used during priming.
<i>controls</i>	The Gauss-Jackson integration controls that drives this state integrator.
<i>size_in</i>	State size.
<i>priming_controls</i>	Integration controls used during priming.

Definition at line 100 of file `gauss_jackson_simple_second_order_ode_integrator.hh`.

8.12.2.4 GaussJacksonSimpleSecondOrderODEIntegrator() [3/3]

```
jeod::GaussJacksonSimpleSecondOrderODEIntegrator::GaussJacksonSimpleSecondOrderODEIntegrator (
    const GaussJacksonSimpleSecondOrderODEIntegrator & src ) [inline]
```

Copy constructor.

Parameters

<i>src</i>	Item to be copied.
------------	--------------------

Definition at line 113 of file `gauss_jackson_simple_second_order_ode_integrator.hh`.

8.12.3 Member Function Documentation

8.12.3.1 create_copy()

```
er7_utils::SecondOrderODEIntegrator* jeod::GaussJacksonSimpleSecondOrderODEIntegrator::create←
_copy ( ) const [inline], [override]
```

Replicate this.

Returns

Replicate of this.

Definition at line 143 of file gauss_jackson_simple_second_order_ode_integrator.hh.

8.12.3.2 integrate()

```
er7_utils::IntegratorResult jeod::GaussJacksonSimpleSecondOrderODEIntegrator::integrate (
    double dyn_dt,
    unsigned int target_stage,
    const double * acc,
    double * vel,
    double * pos ) [inline], [override]
```

Propagate state using Gauss-Jackson.

Parameters

in	<i>dyn_dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in	<i>acc</i>	Acceleration vector.
in, out	<i>vel</i>	Velocity vector.
in, out	<i>pos</i>	Position vector.

Returns

The status (time advance, pass/fail status) of the integration.

Definition at line 167 of file gauss_jackson_simple_second_order_ode_integrator.hh.

8.12.3.3 operator=()

```
GaussJacksonSimpleSecondOrderODEIntegrator& jeod::GaussJacksonSimpleSecondOrderODEIntegrator←
::operator= (
    GaussJacksonSimpleSecondOrderODEIntegrator src ) [inline]
```

Copy and swap assignment operator.

Parameters

<i>src</i>	Item to be copied.
------------	--------------------

Definition at line 123 of file gauss_jackson_simple_second_order_ode_integrator.hh.

8.12.3.4 reset_integrator()

```
void jeod::GaussJacksonSimpleSecondOrderODEIntegrator::reset_integrator ( ) [inline], [override]
```

Reset the integrator.

Definition at line 151 of file gauss_jackson_simple_second_order_ode_integrator.hh.

8.12.3.5 swap() [1/2]

```
void jeod::GaussJacksonSimpleSecondOrderODEIntegrator::swap (
    GaussJacksonSimpleSecondOrderODEIntegrator & other ) [inline]
```

Non-throwing swap.

Parameters

<i>other</i>	Item whose contents are to be swapped with this.
--------------	--

Definition at line 133 of file gauss_jackson_simple_second_order_ode_integrator.hh.

Referenced by jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator::swap().

8.12.3.6 swap() [2/2]

```
void jeod::GaussJacksonIntegratorBase< State, Primer >::swap [inline], [private]
```

Non-throwing swap.

Parameters

<i>other</i>	Item whose contents are to be swapped with this.
--------------	--

Definition at line 397 of file gauss_jackson_integrator_base.hh.

8.12.4 Friends And Related Function Documentation

8.12.4.1 init_attrjeod__GaussJacksonSimpleSecondOrderODEIntegrator

```
void init_attrjeod__GaussJacksonSimpleSecondOrderODEIntegrator ( ) [friend]
```

8.12.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 86 of file gauss_jackson_simple_second_order_ode_integrator.hh.

The documentation for this class was generated from the following file:

- [gauss_jackson_simple_second_order_ode_integrator.hh](#)

8.13 jeod::GaussJacksonStateMachine Class Reference

Guides the behavior of the Gauss-Jackson integration process via a finite state machine.

```
#include <gauss_jackson_state_machine.hh>
```

Public Types

- enum [FsmState](#) {
 [Reset](#), [Priming](#), [BootstrapEdit](#), [BootstrapStep](#),
 [Operational](#) }

Specifies the Gauss-Jackson finite state machine states.

Public Member Functions

- [GaussJacksonStateMachine](#) ()=default
Default constructor.
- [FsmState get_fsm_state](#) () const
Get the finite state machine state.
- unsigned int [get_max_history_size](#) () const
Get the maximum history size.
- unsigned int [get_current_order](#) () const
Get the current order.
- unsigned int [get_history_length](#) () const
Get the current history length.
- double [get_cycle_scale](#) () const
Get the current time scale factor.
- double [get_cycle_start_time](#) () const
Get the current cycle start time.
- bool [get_at_downsample](#) () const
Get the at_downsample flag.
- bool [get_at_reinitialize](#) () const
Get the at_reinitialize flag.
- bool [get_at_order_change](#) () const
Get the at_order_change flag.
- bool [get_at_end_of_tour](#) () const
Get the at_end_of_tour flag.
- void [set_bootstrap_edit_redo_needed](#) ()
Tell the state machine that the edit did not pass a convergence test.
- void [configure](#) (const [GaussJacksonConfig](#) &config)
Configure (or reconfigure) the Gauss-Jackson state machine.
- void [reset](#) ()
Reset the Gauss-Jackson state machine.
- void [perform_step](#) ()
Advance the state machine by one step.

Static Public Member Functions

- static std::string [state_name](#) ([FsmState](#) state)
Translates a finite state machine state value to a string.

Private Member Functions

- void [transition_state](#) ()
Make a state transition.
- void [exit_priming](#) ()
Make the transition out of Priming.
- void [exit_bootstrap_edit](#) ()
Make a transition out of BootstrapEdit.
- void [exit_bootstrap_step](#) ()
Make a transition out of BootstrapStep.

Private Attributes

- unsigned int `initial_order` {}
The order to be used immediately after priming is complete.
- unsigned int `final_order` {}
The order to be used in operational mode.
- unsigned int `ndoubling_steps` {}
The number of times the time step is to be doubled between priming and operational modes.
- unsigned int `max_correction_iterations` {}
The maximum number of corrections to be performed.
- unsigned int `max_history_size` {}
The maximum history size.
- unsigned int `tour_count` {}
The number of small steps that represent a step to the simulation engine, $2 \times n_doubling_steps$.
- `FsmState` `fsm_state` {Reset}
The finite state machine state.
- unsigned int `current_order` {}
The current order.
- unsigned int `history_size` {}
The current history size, the number of history elements that must be accumulated to transition to the next state.
- unsigned int `history_length` {}
The current history length, the number of history elements that have been accumulated so far.
- unsigned int `scale_factor` {}
A power of two that starts at $2 \times ndoubling_steps$ and is halved with each downsample.
- unsigned int `step_increment` {}
A power of two that starts at 1 and is doubled with each downsample.
- unsigned int `steps_since_reset` {}
The number of steps since the reset, measured in units of priming cycle steps.
- unsigned int `correction_iterations` {}
The number of correction iterations made during BootstrapEdit.
- double `cycle_scale` {}
The unitless time step size of the current integration cycle, measured in integration tour time step units.
- double `cycle_start_time` {}
The unitless start time of the current integration cycle, measured in integration tour time step units.
- bool `bootstrap_edit_redo_needed` {}
Flag indicating that the current edit sequence has failed to converge.
- bool `at_downsample` {}
Flag indicating that history data are to be downsampled and the time step is to be doubled.
- bool `at_reinitialize` {}
Flag indicating that the Gauss-Jackson integration constants are to be reinitialized.
- bool `at_order_change` {}
Flag indicating that the order is to be increased.
- bool `at_end_of_tour` {}
Flag indicating that the current integration cycle is the last one in an integration tour (i.e., that a major time step will be completed).

Friends

- class `InputProcessor`
- void `init_attrjeod__GaussJacksonStateMachine` ()

8.13.1 Detailed Description

Guides the behavior of the Gauss-Jackson integration process via a finite state machine.

The Gauss-Jackson integration process comprises four distinct modes:

- Priming: Using an alternate integrator, the primer, to build the requisite number of data points needed by the initial Gauss-Jackson algorithm.
- Editing: Using a Gauss-Jackson mid-corrector to make the collected data consistent with the Gauss-Jackson technique.
- Stepping: Using a Gauss-Jackson predictor/corrector to build the requisite number of data points needed by the next step of the Gauss-Jackson algorithm.
- Operational: Using the Gauss-Jackson predictor/corrector at the final user-specified time step and order.

Definition at line 89 of file gauss_jackson_state_machine.hh.

8.13.2 Member Enumeration Documentation

8.13.2.1 FsmState

```
enum jeod::GaussJacksonStateMachine::FsmState
```

Specifies the Gauss-Jackson finite state machine states.

Enumerator

Reset	Module was just commanded to reset itself.
Priming	Using primer to build initial set of data.
BootstrapEdit	Editing primer / lower-level Gauss-Jackson data.
BootstrapStep	Building toward downsample / change in order.
Operational	At desired rate and order.

Definition at line 95 of file gauss_jackson_state_machine.hh.

8.13.3 Constructor & Destructor Documentation

8.13.3.1 GaussJacksonStateMachine()

```
jeod::GaussJacksonStateMachine::GaussJacksonStateMachine ( ) [default]
```

Default constructor.

8.13.4 Member Function Documentation

8.13.4.1 `configure()`

```
void jeod::GaussJacksonStateMachine::configure (
    const GaussJacksonConfig & config )
```

Configure (or reconfigure) the Gauss-Jackson state machine.

Definition at line 55 of file `gauss_jackson_state_machine.cc`.

References `jeod::GaussJacksonConfig::final_order`, `final_order`, `jeod::GaussJacksonConfig::initial_order`, `initial_order`, `jeod::GaussJacksonConfig::max_correction_iterations`, `max_correction_iterations`, `max_history_size`, `jeod::GaussJacksonConfig::ndoubling_steps`, `ndoubling_steps`, and `tour_count`.

Referenced by `jeod::GaussJacksonIntegrationControls::GaussJacksonIntegrationControls()`.

8.13.4.2 `exit_bootstrap_edit()`

```
void jeod::GaussJacksonStateMachine::exit_bootstrap_edit ( ) [private]
```

Make a transition out of `BootstrapEdit`.

Definition at line 197 of file `gauss_jackson_state_machine.cc`.

References `at_reinitialize`, `bootstrap_edit_redo_needed`, `BootstrapEdit`, `BootstrapStep`, `correction_iterations`, `current_order`, `final_order`, `fsm_state`, `history_length`, `history_size`, `Operational`, `scale_factor`, `step_increment`, and `steps_since_reset`.

Referenced by `exit_bootstrap_step()`, `exit_priming()`, and `transition_state()`.

8.13.4.3 `exit_bootstrap_step()`

```
void jeod::GaussJacksonStateMachine::exit_bootstrap_step ( ) [private]
```

Make a transition out of `BootstrapStep`.

Definition at line 252 of file `gauss_jackson_state_machine.cc`.

References `at_downsample`, `at_order_change`, `at_reinitialize`, `bootstrap_edit_redo_needed`, `BootstrapEdit`, `correction_iterations`, `current_order`, `cycle_scale`, `exit_bootstrap_edit()`, `final_order`, `fsm_state`, `history_length`, `history_size`, `max_correction_iterations`, `scale_factor`, and `step_increment`.

Referenced by `transition_state()`.

8.13.4.4 exit_priming()

```
void jeod::GaussJacksonStateMachine::exit_priming ( ) [private]
```

Make the transition out of Priming.

Definition at line 174 of file gauss_jackson_state_machine.cc.

References `at_order_change`, `at_reinitialize`, `bootstrap_edit_redo_needed`, `BootstrapEdit`, `correction_iterations`, `current_order`, `exit_bootstrap_edit()`, `fsm_state`, `history_length`, `initial_order`, and `max_correction_iterations`.

Referenced by `transition_state()`.

8.13.4.5 get_at_downsample()

```
bool jeod::GaussJacksonStateMachine::get_at_downsample ( ) const [inline]
```

Get the `at_downsample` flag.

Definition at line 171 of file gauss_jackson_state_machine.hh.

Referenced by `jeod::GaussJacksonIntegrationControls::start_cycle()`, and `jeod::GaussJacksonIntegratorBase<State, Primer>::start_cycle()`.

8.13.4.6 get_at_end_of_tour()

```
bool jeod::GaussJacksonStateMachine::get_at_end_of_tour ( ) const [inline]
```

Get the `at_end_of_tour` flag.

Definition at line 195 of file gauss_jackson_state_machine.hh.

Referenced by `jeod::GaussJacksonIntegrationControls::start_cycle()`.

8.13.4.7 get_at_order_change()

```
bool jeod::GaussJacksonStateMachine::get_at_order_change ( ) const [inline]
```

Get the `at_order_change` flag.

Definition at line 187 of file gauss_jackson_state_machine.hh.

Referenced by `jeod::GaussJacksonIntegrationControls::start_cycle()`, and `jeod::GaussJacksonIntegratorBase<State, Primer>::start_cycle()`.

8.13.4.8 `get_at_reinitialize()`

```
bool jeod::GaussJacksonStateMachine::get_at_reinitialize ( ) const [inline]
```

Get the at_reinitialize flag.

Definition at line 179 of file gauss_jackson_state_machine.hh.

Referenced by `jeod::GaussJacksonIntegrationControls::start_cycle()`, and `jeod::GaussJacksonIntegratorBase<State, Primer>::start_cycle()`.

8.13.4.9 `get_current_order()`

```
unsigned int jeod::GaussJacksonStateMachine::get_current_order ( ) const [inline]
```

Get the current order.

Definition at line 139 of file gauss_jackson_state_machine.hh.

Referenced by `jeod::GaussJacksonIntegrationControls::start_cycle()`, and `jeod::GaussJacksonIntegratorBase<State, Primer>::start_cycle()`.

8.13.4.10 `get_cycle_scale()`

```
double jeod::GaussJacksonStateMachine::get_cycle_scale ( ) const [inline]
```

Get the current time scale factor.

Definition at line 155 of file gauss_jackson_state_machine.hh.

Referenced by `jeod::GaussJacksonIntegrationControls::integrate()`, and `jeod::GaussJacksonIntegrationControls::start_cycle()`.

8.13.4.11 `get_cycle_start_time()`

```
double jeod::GaussJacksonStateMachine::get_cycle_start_time ( ) const [inline]
```

Get the current cycle start time.

Definition at line 163 of file gauss_jackson_state_machine.hh.

Referenced by `jeod::GaussJacksonIntegrationControls::start_cycle()`.

8.13.4.12 get_fsm_state()

```
FsmState jeod::GaussJacksonStateMachine::get_fsm_state ( ) const [inline]
```

Get the finite state machine state.

Definition at line 123 of file gauss_jackson_state_machine.hh.

Referenced by jeod::GaussJacksonIntegrationControls::start_cycle(), and jeod::GaussJacksonIntegratorBase<State, Primer >::start_cycle().

8.13.4.13 get_history_length()

```
unsigned int jeod::GaussJacksonStateMachine::get_history_length ( ) const [inline]
```

Get the current history length.

Definition at line 147 of file gauss_jackson_state_machine.hh.

Referenced by jeod::GaussJacksonIntegrationControls::integrate_edit().

8.13.4.14 get_max_history_size()

```
unsigned int jeod::GaussJacksonStateMachine::get_max_history_size ( ) const [inline]
```

Get the maximum history size.

Definition at line 131 of file gauss_jackson_state_machine.hh.

8.13.4.15 perform_step()

```
void jeod::GaussJacksonStateMachine::perform_step ( )
```

Advance the state machine by one step.

Definition at line 103 of file gauss_jackson_state_machine.cc.

References at_downsample, at_end_of_tour, at_order_change, at_reinitialize, BootstrapEdit, cycle_start_time, fsm_state, history_length, history_size, step_increment, steps_since_reset, tour_count, and transition_state().

Referenced by jeod::GaussJacksonIntegrationControls::start_cycle().

8.13.4.16 reset()

```
void jeod::GaussJacksonStateMachine::reset ( )
```

Reset the Gauss-Jackson state machine.

Definition at line 71 of file gauss_jackson_state_machine.cc.

References at_downsample, at_end_of_tour, at_order_change, at_reinitialize, current_order, cycle_scale, cycle_↵
start_time, fsm_state, history_length, history_size, initial_order, Reset, scale_factor, step_increment, steps_since_↵
_reset, and tour_count.

Referenced by jeod::GaussJacksonIntegrationControls::reset_integrator().

8.13.4.17 set_bootstrap_edit_redo_needed()

```
void jeod::GaussJacksonStateMachine::set_bootstrap_edit_redo_needed ( )
```

Tell the state machine that the edit did not pass a convergence test.

Definition at line 96 of file gauss_jackson_state_machine.cc.

References bootstrap_edit_redo_needed, BootstrapEdit, correction_iterations, fsm_state, and max_correction_↵
iterations.

Referenced by jeod::GaussJacksonIntegrationControls::integrate_edit().

8.13.4.18 state_name()

```
std::string jeod::GaussJacksonStateMachine::state_name (
    FsmState state ) [static]
```

Translates a finite state machine state value to a string.

Definition at line 30 of file gauss_jackson_state_machine.cc.

References BootstrapEdit, BootstrapStep, Operational, Priming, and Reset.

8.13.4.19 transition_state()

```
void jeod::GaussJacksonStateMachine::transition_state ( ) [private]
```

Make a state transition.

Definition at line 131 of file gauss_jackson_state_machine.cc.

References BootstrapEdit, BootstrapStep, current_order, exit_bootstrap_edit(), exit_bootstrap_step(), exit_↵
priming(), fsm_state, history_size, initial_order, Operational, Priming, Reset, and steps_since_reset.

Referenced by perform_step().

8.13.5 Friends And Related Function Documentation

8.13.5.1 init_attrjeod__GaussJacksonStateMachine

```
void init_attrjeod__GaussJacksonStateMachine ( ) [friend]
```

8.13.5.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 91 of file gauss_jackson_state_machine.hh.

8.13.6 Field Documentation

8.13.6.1 at_downsample

```
bool jeod::GaussJacksonStateMachine::at_downsample {} [private]
```

Flag indicating that history data are to be downsampled and the time step is to be doubled.

The flag is set on transitions from BootstrapStep to BootstrapEdit when the step size has not yet reached the desired value, clear otherwise.trick_units(-)

Definition at line 330 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_step(), perform_step(), and reset().

8.13.6.2 at_end_of_tour

```
bool jeod::GaussJacksonStateMachine::at_end_of_tour {} [private]
```

Flag indicating that the current integration cycle is the last one in an integration tour (i.e., that a major time step will be completed).

The flag is set at the start of the cycle that completes the tour, clear otherwise. This flag is never set during BootstrapEdit.trick_units(-)

Definition at line 352 of file gauss_jackson_state_machine.hh.

Referenced by perform_step(), and reset().

8.13.6.3 at_order_change

```
bool jeod::GaussJacksonStateMachine::at_order_change {} [private]
```

Flag indicating that the order is to be increased.

The flag is set on on transitions from BootstrapStep to BootstrapEdit when the order has has not yet reached the desired value, clear otherwise.trick_units(-)

Definition at line 344 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_step(), exit_priming(), perform_step(), and reset().

8.13.6.4 at_reinitialize

```
bool jeod::GaussJacksonStateMachine::at_reinitialize {} [private]
```

Flag indicating that the Gauss-Jackson integration constants are to be reinitialized.

The flag is set on entry to any state except Reset and Priming, clear otherwise.trick_units(-)

Definition at line 337 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_edit(), exit_bootstrap_step(), exit_priming(), perform_step(), and reset().

8.13.6.5 bootstrap_edit_redo_needed

```
bool jeod::GaussJacksonStateMachine::bootstrap_edit_redo_needed {} [private]
```

Flag indicating that the current edit sequence has failed to converge.

This flag is set externally by the Gauss-Jackson integration controls.trick_units(-)

Definition at line 322 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_edit(), exit_bootstrap_step(), exit_priming(), and set_bootstrap_edit_redo_needed().

8.13.6.6 correction_iterations

```
unsigned int jeod::GaussJacksonStateMachine::correction_iterations {} [private]
```

The number of correction iterations made during BootstrapEdit.

trick_units(-)

Definition at line 303 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_edit(), exit_bootstrap_step(), exit_priming(), and set_bootstrap_edit_redo_needed().

8.13.6.7 current_order

```
unsigned int jeod::GaussJacksonStateMachine::current_order {} [private]
```

The current order.

This is incremented by two on transitions from BootstrapStep to BootstrapEdit until the final_order is reached. `trick_units(-)`

Definition at line 267 of file gauss_jackson_state_machine.hh.

Referenced by `exit_bootstrap_edit()`, `exit_bootstrap_step()`, `exit_priming()`, `reset()`, and `transition_state()`.

8.13.6.8 cycle_scale

```
double jeod::GaussJacksonStateMachine::cycle_scale {} [private]
```

The unitless time step size of the current integration cycle, measured in integration tour time step units.

This starts at $2^{**}(-ndoubling_steps)$ and doubles with each downsample. `trick_units(-)`

Definition at line 310 of file gauss_jackson_state_machine.hh.

Referenced by `exit_bootstrap_step()`, and `reset()`.

8.13.6.9 cycle_start_time

```
double jeod::GaussJacksonStateMachine::cycle_start_time {} [private]
```

The unitless start time of the current integration cycle, measured in integration tour time step units.

`trick_units(-)`

Definition at line 316 of file gauss_jackson_state_machine.hh.

Referenced by `perform_step()`, and `reset()`.

8.13.6.10 final_order

```
unsigned int jeod::GaussJacksonStateMachine::final_order {} [private]
```

The order to be used in operational mode.

This must be an even integer and must not be less than initial_order. `trick_units(-)`

Definition at line 233 of file gauss_jackson_state_machine.hh.

Referenced by `configure()`, `exit_bootstrap_edit()`, and `exit_bootstrap_step()`.

8.13.6.11 fsm_state

```
FsmState jeod::GaussJacksonStateMachine::fsm_state {Reset} [private]
```

The finite state machine state.

trick_units(-)

Definition at line 261 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_edit(), exit_bootstrap_step(), exit_priming(), perform_step(), reset(), set_bootstrap←_edit_redo_needed(), and transition_state().

8.13.6.12 history_length

```
unsigned int jeod::GaussJacksonStateMachine::history_length {} [private]
```

The current history length, the number of history elements that have been accumulated so far.

trick_units(-)

Definition at line 279 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_edit(), exit_bootstrap_step(), exit_priming(), perform_step(), and reset().

8.13.6.13 history_size

```
unsigned int jeod::GaussJacksonStateMachine::history_size {} [private]
```

The current history size, the number of history elements that must be accumulated to transition to the next state.

trick_units(-)

Definition at line 273 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_edit(), exit_bootstrap_step(), perform_step(), reset(), and transition_state().

8.13.6.14 initial_order

```
unsigned int jeod::GaussJacksonStateMachine::initial_order {} [private]
```

The order to be used immediately after priming is complete.

This must be an even, non-negative integer.trick_units(-)

Definition at line 227 of file gauss_jackson_state_machine.hh.

Referenced by configure(), exit_priming(), reset(), and transition_state().

8.13.6.15 max_correction_iterations

```
unsigned int jeod::GaussJacksonStateMachine::max_correction_iterations {} [private]
```

The maximum number of corrections to be performed.

trick_units(-)

Definition at line 244 of file gauss_jackson_state_machine.hh.

Referenced by configure(), exit_bootstrap_step(), exit_priming(), and set_bootstrap_edit_redo_needed().

8.13.6.16 max_history_size

```
unsigned int jeod::GaussJacksonStateMachine::max_history_size {} [private]
```

The maximum history size.

This is calculated for the benefit of state integrators.trick_units(-)

Definition at line 250 of file gauss_jackson_state_machine.hh.

Referenced by configure().

8.13.6.17 ndoubling_steps

```
unsigned int jeod::GaussJacksonStateMachine::ndoubling_steps {} [private]
```

The number of times the time step is to be doubled between priming and operational modes.

trick_units(-)

Definition at line 239 of file gauss_jackson_state_machine.hh.

Referenced by configure().

8.13.6.18 scale_factor

```
unsigned int jeod::GaussJacksonStateMachine::scale_factor {} [private]
```

A power of two that starts at $2^{ndoubling_steps}$ and is halved with each downsample.

When the scale_factor reaches 1 it is time to transition to operational mode.trick_units(-)

Definition at line 286 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_edit(), exit_bootstrap_step(), and reset().

8.13.6.19 step_increment

```
unsigned int jeod::GaussJacksonStateMachine::step_increment {} [private]
```

A power of two that starts at 1 and is doubled with each downsample.

trick_units(-)

Definition at line 291 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_edit(), exit_bootstrap_step(), perform_step(), and reset().

8.13.6.20 steps_since_reset

```
unsigned int jeod::GaussJacksonStateMachine::steps_since_reset {} [private]
```

The number of steps since the reset, measured in units of priming cycle steps.

The counter is incremented by the step_increment upon completion of a cycle and is reset to zero on entry into BootstrapEdit.trick_units(-)

Definition at line 298 of file gauss_jackson_state_machine.hh.

Referenced by exit_bootstrap_edit(), perform_step(), reset(), and transition_state().

8.13.6.21 tour_count

```
unsigned int jeod::GaussJacksonStateMachine::tour_count {} [private]
```

The number of small steps that represent a step to the simulation engine, $2^{**n_doubling_steps}$.

trick_units(-)

Definition at line 256 of file gauss_jackson_state_machine.hh.

Referenced by configure(), perform_step(), and reset().

The documentation for this class was generated from the following files:

- [gauss_jackson_state_machine.hh](#)
- [gauss_jackson_state_machine.cc](#)

8.14 jeod::GaussJacksonTwoState Class Reference

Essentially just `std::pair<double*>`.

```
#include <gauss_jackson_two_state.hh>
```


Public Member Functions

- [GaussJacksonTwoState](#) ()=default
Default constructor.
- [GaussJacksonTwoState](#) (double *first_in, double *second_in)
Non-default constructor.

Data Fields

- double * [first](#) {}
The first element of the pair.
- double * [second](#) {}
The second element of the pair.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__GaussJacksonTwoState](#) ()

8.14.1 Detailed Description

Essentially just `std::pair<double*>`.

Definition at line 72 of file `gauss_jackson_two_state.hh`.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 GaussJacksonTwoState() [1/2]

```
jeod::GaussJacksonTwoState::GaussJacksonTwoState ( ) [default]
```

Default constructor.

8.14.2.2 GaussJacksonTwoState() [2/2]

```
jeod::GaussJacksonTwoState::GaussJacksonTwoState (
    double * first_in,
    double * second_in ) [inline]
```

Non-default constructor.

Parameters

<i>first_in</i>	The first element of the pair.
<i>second_in</i>	The second element of the pair.

Definition at line 95 of file gauss_jackson_two_state.hh.

8.14.3 Friends And Related Function Documentation**8.14.3.1 init_attrjeod__GaussJacksonTwoState**

```
void init_attrjeod__GaussJacksonTwoState ( ) [friend]
```

8.14.3.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 74 of file gauss_jackson_two_state.hh.

8.14.4 Field Documentation**8.14.4.1 first**

```
double* jeod::GaussJacksonTwoState::first {}
```

The first element of the pair.

trick_units(-)

Definition at line 78 of file gauss_jackson_two_state.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::allocate_state_contents(), jeod::GaussJacksonIntegratorBase< State, Primer >::correct(), jeod::GaussJacksonIntegratorBase< State, Primer >::deallocate_state_contents(), jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_primer(), jeod::GaussJacksonIntegratorBase< State, Primer >::mid_correct(), jeod::GaussJacksonIntegratorBase< State, Primer >::predict(), jeod::GaussJacksonIntegratorBase< State, Primer >::replicate_state(), jeod::GaussJacksonIntegratorBase< State, Primer >::save_epoch_data(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap_state().

8.14.4.2 second

```
double* jeod::GaussJacksonTwoState::second {}
```

The second element of the pair.

```
trick_units(-)
```

Definition at line 83 of file gauss_jackson_two_state.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::allocate_state_contents(), jeod::GaussJacksonIntegratorBase< State, Primer >::correct(), jeod::GaussJacksonIntegratorBase< State, Primer >::deallocate_state_contents(), jeod::GaussJacksonIntegratorBase< State, Primer >::integrate_primer(), jeod::GaussJacksonIntegratorBase< State, Primer >::mid_correct(), jeod::GaussJacksonIntegratorBase< State, Primer >::predict(), jeod::GaussJacksonIntegratorBase< State, Primer >::replicate_state(), jeod::GaussJacksonIntegratorBase< State, Primer >::save_comparison_data(), jeod::GaussJacksonIntegratorBase< State, Primer >::save_epoch_data(), jeod::GaussJacksonIntegratorBase< State, Primer >::swap_state(), and jeod::GaussJacksonIntegratorBase< State, Primer >::test_for_convergence().

The documentation for this class was generated from the following file:

- [gauss_jackson_two_state.hh](#)

8.15 jeod::GeneralizedSecondOrderODETechnique Class Reference

Enumerates the integration schemes, generalized Cartesian or Lie group integration, and provides simple utilities that work with this.

```
#include <generalized_second_order_ode_technique.hh>
```

Public Types

- enum [TechniqueType](#) { [Unspecified](#), [Cartesian](#), [LieGroup](#) }

Enumerates the types of second order ODE solvers that can be used to integrate a generalized second order ODE.

Public Member Functions

- [GeneralizedSecondOrderODETechnique](#) ()=delete
- [GeneralizedSecondOrderODETechnique](#) (const [GeneralizedSecondOrderODETechnique](#) &)=delete
- [GeneralizedSecondOrderODETechnique](#) & operator= (const [GeneralizedSecondOrderODETechnique](#) &)=delete

Static Public Member Functions

- static bool [is_provided_by](#) (const er7_utils::IntegratorConstructor &generator, [TechniqueType](#) technique)
Test whether an integration method provides an integrator for the specified technique.
- static [TechniqueType](#) [validate_technique](#) (const er7_utils::IntegratorConstructor &generator, [TechniqueType](#) technique, const char *file, unsigned int line, const char *requester, const char *name)
Validate the specified technique with respect to the integration method.

Private Member Functions

- [~GeneralizedSecondOrderODETechnique\(\)](#)

8.15.1 Detailed Description

Enumerates the integration schemes, generalized Cartesian or Lie group integration, and provides simple utilities that work with this.

Definition at line 78 of file `generalized_second_order_ode_technique.hh`.

8.15.2 Member Enumeration Documentation

8.15.2.1 TechniqueType

```
enum jeod::GeneralizedSecondOrderODETechnique::TechniqueType
```

Enumerates the types of second order ODE solvers that can be used to integrate a generalized second order ODE.

Enumerator

Unspecified	No technique specified (an error).
Cartesian	Integrate using a generalized derivative scheme. The integrator treats generalized position as if it lives in some Cartesian space.
LieGroup	Integrate using a generalized step scheme. The integrator treats generalized position as if it lives in some Lie group.

Definition at line 91 of file `generalized_second_order_ode_technique.hh`.

8.15.3 Constructor & Destructor Documentation

8.15.3.1 GeneralizedSecondOrderODETechnique() [1/2]

```
jeod::GeneralizedSecondOrderODETechnique::GeneralizedSecondOrderODETechnique ( ) [delete]
```

8.15.3.2 GeneralizedSecondOrderODETechnique() [2/2]

```
jeod::GeneralizedSecondOrderODETechnique::GeneralizedSecondOrderODETechnique (
    const GeneralizedSecondOrderODETechnique & ) [delete]
```

8.15.3.3 ~GeneralizedSecondOrderODETechnique()

```
jeod::GeneralizedSecondOrderODETechnique::~~GeneralizedSecondOrderODETechnique ( ) [private]
```

8.15.4 Member Function Documentation

8.15.4.1 is_provided_by()

```
bool jeod::GeneralizedSecondOrderODETechnique::is_provided_by (
    const er7_utils::IntegratorConstructor & generator,
    TechniqueType technique ) [static]
```

Test whether an integration method provides an integrator for the specified technique.

Parameters

in	<i>generator</i>	Integrator constructor for the integration technique.
in	<i>technique</i>	Technique to be queried.

Returns

True if the constructor can create an integrator for the specified technique, false otherwise.

Definition at line 41 of file generalized_second_order_ode_technique.cc.

References Cartesian, jeod::IntegrationMessages::invalid_request, LieGroup, and Unspecified.

Referenced by validate_technique().

8.15.4.2 operator=()

```
GeneralizedSecondOrderODETechnique& jeod::GeneralizedSecondOrderODETechnique::operator= (
    const GeneralizedSecondOrderODETechnique & ) [delete]
```

8.15.4.3 validate_technique()

```
GeneralizedSecondOrderODETechnique::TechniqueType jeod::GeneralizedSecondOrderODETechnique↵
::validate_technique (
    const er7_utils::IntegratorConstructor & generator,
    TechniqueType technique,
    const char * file,
    unsigned int line,
    const char * requester,
    const char * name ) [static]
```

Validate the specified technique with respect to the integration method.

Possible outcomes are:

- Failure if the generator doesn't provide either of the generalized second order ODE integrators.
- Switch to plan B if the generator doesn't provide the requested integrator but does provide the alternate technique.
- Nothing happens if the generator does provide the requested integrator.

Parameters

in	<i>generator</i>	Integrator constructor for the integration technique.
in	<i>technique</i>	Technique to be queried.
in	<i>file</i>	Typically FILE
in	<i>line</i>	Typically LINE
in	<i>requester</i>	Something to identify the caller.
in	<i>name</i>	The name of the object associated with the caller.

Returns

Input technique if supported, alternate if not. The function does not return is neither of the options is supported.

Definition at line 65 of file generalized_second_order_ode_technique.cc.

References Cartesian, jeod::IntegrationMessages::invalid_request, is_provided_by(), LieGroup, and jeod::↵ IntegrationMessages::unsupported_option.

The documentation for this class was generated from the following files:

- [generalized_second_order_ode_technique.hh](#)
- [generalized_second_order_ode_technique.cc](#)

8.16 jeod::IntegrationMessages Class Reference

Declares messages associated with the integration test model.

```
#include <integration_messages.hh>
```

Public Member Functions

- [IntegrationMessages](#) ()=delete
- [IntegrationMessages](#) (const [IntegrationMessages](#) &)=delete
- [IntegrationMessages](#) & operator= (const [IntegrationMessages](#) &)=delete

Static Public Attributes

- static const char * [unsupported_option](#) = "utils/integration/" "unsupported_option"
Issued when some user input is invalid.
- static const char * [invalid_item](#) = "utils/integration/" "invalid_item"
Issued when an item is somehow invalid; a duplicate entry for example.
- static const char * [internal_error](#) = "utils/integration/" "internal_error"
Issued when the JEOD programmer messed up.
- static const char * [invalid_request](#) = "utils/integration/" "invalid_request"
Issued when a non-JEOD programmer messed up.
- static const char * [information](#) = "utils/integration/" "information"
Issued in non-error messages.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__IntegrationMessages](#) ()

8.16.1 Detailed Description

Declares messages associated with the integration test model.

Definition at line 82 of file `integration_messages.hh`.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 IntegrationMessages() [1/2]

```
jeod::IntegrationMessages::IntegrationMessages ( ) [delete]
```

8.16.2.2 IntegrationMessages() [2/2]

```
jeod::IntegrationMessages::IntegrationMessages (
    const IntegrationMessages & ) [delete]
```

8.16.3 Member Function Documentation

8.16.3.1 operator=()

```
IntegrationMessages& jeod::IntegrationMessages::operator= (
    const IntegrationMessages & ) [delete]
```

8.16.4 Friends And Related Function Documentation

8.16.4.1 init_attrjeod__IntegrationMessages

```
void init_attrjeod__IntegrationMessages ( ) [friend]
```

8.16.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 84 of file integration_messages.hh.

8.16.5 Field Documentation

8.16.5.1 information

```
char const * jeod::IntegrationMessages::information = "utils/integration/" "information" [static]
```

Issued in non-error messages.

```
trick_units(-)
```

Definition at line 110 of file integration_messages.hh.

8.16.5.2 internal_error

```
char const * jeod::IntegrationMessages::internal_error = "utils/integration/" "internal_error"  
[static]
```

Issued when the JEOD programmer messed up.

trick_units(-)

Definition at line 100 of file integration_messages.hh.

8.16.5.3 invalid_item

```
char const * jeod::IntegrationMessages::invalid_item = "utils/integration/" "invalid_item"  
[static]
```

Issued when an item is somehow invalid; a duplicate entry for example.

trick_units(-)

Definition at line 95 of file integration_messages.hh.

Referenced by jeod::JeodIntegrationGroup::remove_integrable_object().

8.16.5.4 invalid_request

```
char const * jeod::IntegrationMessages::invalid_request = "utils/integration/" "invalid_  
request" [static]
```

Issued when a non-JEOD programmer messed up.

trick_units(-)

Definition at line 105 of file integration_messages.hh.

Referenced by jeod::JeodIntegrationTime::add_time_change_subscriber(), jeod::RestartableStateIntegrator<er7_utils::SecondOrderODEIntegrator >::create_integrator(), jeod::GeneralizedSecondOrderODETechnique::is_←_provided_by(), jeod::JeodIntegrationTime::remove_time_change_subscriber(), and jeod::GeneralizedSecond←OrderODETechnique::validate_technique().

8.16.5.5 unsupported_option

```
char const * jeod::IntegrationMessages::unsupported_option = "utils/integration/" "unsupported←
_option" [static]
```

Issued when some user input is invalid.

trick_units(-)

Definition at line 90 of file integration_messages.hh.

Referenced by jeod::GeneralizedSecondOrderODETechnique::validate_technique().

The documentation for this class was generated from the following files:

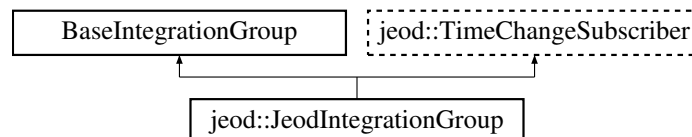
- [integration_messages.hh](#)
- [integration_messages.cc](#)

8.17 jeod::JeodIntegrationGroup Class Reference

A [JeodIntegrationGroup](#) integrates the state of a set of objects over time.

```
#include <jeod_integration_group.hh>
```

Inheritance diagram for jeod::JeodIntegrationGroup:



Public Member Functions

- [JeodIntegrationGroup](#) ()
JeodIntegrationGroup default constructor, needed for checkpoint/restart, and to support derived classes' default constructors.
- [JeodIntegrationGroup](#) ([JeodIntegrationGroupOwner](#) &owner, [er7_utils::IntegratorConstructor](#) &integ_ctr, [JeodIntegratorInterface](#) &integ_inter, [JeodIntegrationTime](#) &time_mgr)
JeodIntegrationGroup non-default constructor.
- [~JeodIntegrationGroup](#) () override
JeodIntegrationGroup destructor.
- [JeodIntegrationGroup](#) (const [JeodIntegrationGroup](#) &)=delete
- [JeodIntegrationGroup](#) & operator= (const [JeodIntegrationGroup](#) &)=delete
- bool [need_first_step_derivatives](#) () const
Indicate whether derivatives need to be calculated on the initial step of an integration cycle.
- void [update_from_owner](#) ()
Update the group via its owner.
- bool [merge_integrator_result](#) (const [er7_utils::IntegratorResult](#) &new_result, [er7_utils::IntegratorResult](#) &merged_result) const
Merge an *IntegratorResult* into another.

- void [respond_to_time_change](#) () override
Respond to a change in the nature of time.
- void [initialize_group](#) () override
Initialize the integration group.
- void [reset_body_integrators](#) () override
Reset the integrators for the integrable objects managed by this group.
- er7_utils::IntegratorResult [integrate_bodies](#) (double cycle_dyndt, unsigned int target_stage) override
Integrate the states of the integrable objects managed by this group.
- virtual void [add_integrable_object](#) (er7_utils::IntegrableObject &integrable_object)
Add an integrable object to the vector of such.
- virtual void [remove_integrable_object](#) (er7_utils::IntegrableObject &integrable_object)
Remove an integrable object from the vector of such.

Static Public Member Functions

- static void [register_classes](#) ()
Register classes associated with integration.

Protected Member Functions

- template<typename T >
void [reset_container](#) (const T &container)
Issue a reset to each member of a container.
- template<typename T >
er7_utils::IntegratorResult [integrate_container](#) (double dyn_dt, unsigned int target_stage, const T &container)
Integrate each member of a collection.

Protected Attributes

- [JeodIntegrationGroupOwner](#) *const [group_owner](#) {}
The object that owns this integration group, typically by containment.
- er7_utils::IntegratorResultMergerContainer [integ_merger](#)
The object that merges results from multiple integrators.
- JeodIntegratorInterface *const [jeod_integ_interface](#) {}
The interface between the integration module and the simulation engine's integration structure.
- [JeodIntegrationTime](#) *const [jeod_time_manager](#) {}
The interface between the integration module and the object that represents time.
- JeodPointerVector< er7_utils::IntegrableObject >::type [integrable_objects](#)
The objects whose states are integrated by this integration group.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__JeodIntegrationGroup](#) ()

8.17.1 Detailed Description

A [JeodIntegrationGroup](#) integrates the state of a set of objects over time.

This class is designed for extensibility. Authors of derived classes should follow the extension notes in the source file.

Definition at line 115 of file `jeod_integration_group.hh`.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 [JeodIntegrationGroup\(\)](#) [1/3]

```
jeod::JeodIntegrationGroup::JeodIntegrationGroup ( )
```

[JeodIntegrationGroup](#) default constructor, needed for checkpoint/restart, and to support derived classes' default constructors.

Definition at line 60 of file `jeod_integration_group.cc`.

References `integrable_objects`, and `register_classes()`.

8.17.2.2 [JeodIntegrationGroup\(\)](#) [2/3]

```
jeod::JeodIntegrationGroup::JeodIntegrationGroup (
    JeodIntegrationGroupOwner & owner,
    er7_utils::IntegratorConstructor & integ_cotr,
    JeodIntegratorInterface & integ_inter,
    JeodIntegrationTime & time_mgr )
```

[JeodIntegrationGroup](#) non-default constructor.

Parameters

in	<i>owner</i>	The object that contains this group.
in	<i>integ_cotr</i>	Integrator constructor
in	<i>integ_inter</i>	Integrator interface
in	<i>time_mgr</i>	Time manager

Definition at line 68 of file `jeod_integration_group.cc`.

References `jeod::JeodIntegrationTime::add_time_change_subscriber()`, `integ_merger`, `integrable_objects`, and `register_classes()`.

8.17.2.3 ~JeodIntegrationGroup()

```
jeod::JeodIntegrationGroup::~~JeodIntegrationGroup ( ) [override]
```

[JeodIntegrationGroup](#) destructor.

Definition at line 86 of file `jeod_integration_group.cc`.

References `integrable_objects`, `jeod_time_manager`, and `jeod::JeodIntegrationTime::remove_time_change_↔subscriber()`.

8.17.2.4 JeodIntegrationGroup() [3/3]

```
jeod::JeodIntegrationGroup::JeodIntegrationGroup (
    const JeodIntegrationGroup & ) [delete]
```

8.17.3 Member Function Documentation

8.17.3.1 add_integrable_object()

```
void jeod::JeodIntegrationGroup::add_integrable_object (
    er7_utils::IntegrableObject & integrable_object ) [virtual]
```

Add an integrable object to the vector of such.

Parameters

in	<i>integrable_object</i>	Object to be added.
----	--------------------------	---------------------

Definition at line 97 of file `jeod_integration_group.cc`.

References `integrable_objects`.

8.17.3.2 initialize_group()

```
void jeod::JeodIntegrationGroup::initialize_group ( ) [override]
```

Initialize the integration group.

Some integration techniques are configurable by user input, and thus the creation of the controls and integrators needs to be delayed a bit.

Definition at line 153 of file `jeod_integration_group.cc`.

References `integrable_objects`.

8.17.3.3 `integrate_bodies()`

```
er7_utils::IntegratorResult jeod::JeodIntegrationGroup::integrate_bodies (
    double cycle_dyndt,
    unsigned int target_stage ) [inline], [override]
```

Integrate the states of the integrable objects managed by this group.

This function should only be called by `IntegrationControls::integrate` or by an override of that function. Derived classes are free to override this default implementation. However, those derived class overrides either must call this method to integrate the states of the registered integrable bodies or must somehow take on the burden of integrating those states.

Parameters

in	<i>cycle_dyndt</i>	Dynamic time step, in dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.

Returns

The status (time advance, pass/fail status) of the integration.

Definition at line 234 of file `jeod_integration_group.hh`.

8.17.3.4 `integrate_container()`

```
template<typename T >
er7_utils::IntegratorResult jeod::JeodIntegrationGroup::integrate_container (
    double dyn_dt,
    unsigned int target_stage,
    const T & container ) [inline], [protected]
```

Integrate each member of a collection.

Template Parameters

<i>T</i>	The container type.
----------	---------------------

Parameters

in	<i>dyn_dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in, out	<i>container</i>	The container to be integrated.

Definition at line 277 of file `jeod_integration_group.hh`.

References `integ_merger`.

8.17.3.5 merge_integrator_result()

```
bool jeod::JeodIntegrationGroup::merge_integrator_result (
    const er7_utils::IntegratorResult & new_result,
    er7_utils::IntegratorResult & merged_result ) const [inline]
```

Merge an IntegratorResult into another.

Returns

True if merger was successful, false if some error occurred.

Parameters

<i>in</i>	<i>new_result</i>	Size of the generalized position vector
<i>in, out</i>	<i>merged_result</i>	Size of the generalized position vector

Definition at line 182 of file jeod_integration_group.hh.

8.17.3.6 need_first_step_derivatives()

```
bool jeod::JeodIntegrationGroup::need_first_step_derivatives ( ) const [inline]
```

Indicate whether derivatives need to be calculated on the initial step of an integration cycle.

Returns

Desired flag.

Definition at line 163 of file jeod_integration_group.hh.

8.17.3.7 operator=()

```
JeodIntegrationGroup& jeod::JeodIntegrationGroup::operator= (
    const JeodIntegrationGroup & ) [delete]
```

8.17.3.8 register_classes()

```
void jeod::JeodIntegrationGroup::register_classes ( ) [static]
```

Register classes associated with integration.

This is a static method, and is best called prior to initialization time.

Definition at line 51 of file jeod_integration_group.cc.

Referenced by JeodIntegrationGroup().

8.17.3.9 remove_integrable_object()

```
void jeod::JeodIntegrationGroup::remove_integrable_object (
    er7_utils::IntegrableObject & integrable_object ) [virtual]
```

Remove an integrable object from the vector of such.

Parameters

in	<i>integrable_object</i>	Object to be removed.
----	--------------------------	-----------------------

Definition at line 125 of file jeod_integration_group.cc.

References `integrable_objects`, and `jeod::IntegrationMessages::invalid_item`.

8.17.3.10 reset_body_integrators()

```
void jeod::JeodIntegrationGroup::reset_body_integrators ( ) [inline], [override]
```

Reset the integrators for the integrable objects managed by this group.

Resets can occur when time changes behavior (call is internal to the integration process) or when some external event would render an integrator's history invalid (call comes from outside). When either happens, integrators that depend on history need to reset their internal state to indicate that the saved data are invalid.)

Definition at line 216 of file jeod_integration_group.hh.

8.17.3.11 reset_container()

```
template<typename T >
void jeod::JeodIntegrationGroup::reset_container (
    const T & container ) [inline], [protected]
```

Issue a reset to each member of a container.

Template Parameters

<i>T</i>	The container type.
----------	---------------------

Parameters

in, out	<i>container</i>	The container to be reset.
---------	------------------	----------------------------

Definition at line 259 of file jeod_integration_group.hh.

8.17.3.12 respond_to_time_change()

```
void jeod::JeodIntegrationGroup::respond_to_time_change ( ) [inline], [override], [virtual]
```

Respond to a change in the nature of time.

Implements [jeod::TimeChangeSubscriber](#).

Definition at line 193 of file jeod_integration_group.hh.

8.17.3.13 update_from_owner()

```
void jeod::JeodIntegrationGroup::update_from_owner ( ) [inline]
```

Update the group via its owner.

Definition at line 171 of file jeod_integration_group.hh.

8.17.4 Friends And Related Function Documentation

8.17.4.1 init_attrjeod__JeodIntegrationGroup

```
void init_attrjeod__JeodIntegrationGroup ( ) [friend]
```

8.17.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 118 of file jeod_integration_group.hh.

8.17.5 Field Documentation

8.17.5.1 group_owner

```
JeodIntegrationGroupOwner* const jeod::JeodIntegrationGroup::group_owner {} [protected]
```

The object that owns this integration group, typically by containment.

trick_units(—)

Definition at line 298 of file jeod_integration_group.hh.

8.17.5.2 integ_merger

```
er7_utils::IntegratorResultMergerContainer jeod::JeodIntegrationGroup::integ_merger [protected]
```

The object that merges results from multiple integrators.

trick_units(-)

Definition at line 303 of file jeod_integration_group.hh.

Referenced by integrate_container(), and JeodIntegrationGroup().

8.17.5.3 integrable_objects

```
JeodPointerVector<er7_utils::IntegrableObject>::type jeod::JeodIntegrationGroup::integrable_↔  
objects [protected]
```

The objects whose states are integrated by this integration group.

trick_io(**)

Definition at line 320 of file jeod_integration_group.hh.

Referenced by add_integrable_object(), initialize_group(), JeodIntegrationGroup(), remove_integrable_object(), and ~JeodIntegrationGroup().

8.17.5.4 jeod_integ_interface

```
JeodIntegratorInterface* const jeod::JeodIntegrationGroup::jeod_integ_interface {} [protected]
```

The interface between the integration module and the simulation engine's integration structure.

trick_units(-)

Definition at line 309 of file jeod_integration_group.hh.

8.17.5.5 jeod_time_manager

```
JeodIntegrationTime* const jeod::JeodIntegrationGroup::jeod_time_manager {} [protected]
```

The interface between the integration module and the object that represents time.

trick_units(-)

Definition at line 315 of file jeod_integration_group.hh.

Referenced by ~JeodIntegrationGroup().

The documentation for this class was generated from the following files:

- [jeod_integration_group.hh](#)
- [jeod_integration_group.cc](#)

8.18 jeod::JeodIntegrationGroupOwner Class Reference

The abstract class IntegrationGroupOwner contains an IntegrationGroup.

```
#include <jeod_integration_group.hh>
```

Public Member Functions

- virtual [~JeodIntegrationGroupOwner](#) ()=default
Destructor.
- virtual void [update_integration_group](#) ([JeodIntegrationGroup](#) &group)=0
Somehow update the specified integration group.

8.18.1 Detailed Description

The abstract class IntegrationGroupOwner contains an IntegrationGroup.

This simple interface class has no data members.

Definition at line 95 of file jeod_integration_group.hh.

8.18.2 Constructor & Destructor Documentation

8.18.2.1 ~JeodIntegrationGroupOwner()

```
virtual jeod::JeodIntegrationGroupOwner::~~JeodIntegrationGroupOwner ( ) [virtual], [default]
```

Destructor.

8.18.3 Member Function Documentation

8.18.3.1 update_integration_group()

```
virtual void jeod::JeodIntegrationGroupOwner::update_integration_group (
    JeodIntegrationGroup & group ) [pure virtual]
```

Somehow update the specified integration group.

Parameters

in, out	<i>group</i>	Integration group to be updated.
---------	--------------	----------------------------------

The documentation for this class was generated from the following file:

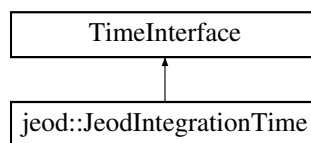
- [jeod_integration_group.hh](#)

8.19 jeod::JeodIntegrationTime Class Reference

The class [JeodIntegrationTime](#) adds the concepts of timestamping and notification of changes in the nature of time to the ER7 numerical utilities TimeInterface class.

```
#include <jeod_integration_time.hh>
```

Inheritance diagram for jeod::JeodIntegrationTime:



Public Member Functions

- [JeodIntegrationTime](#) ()
JeodIntegrationTime constructor.
- [~JeodIntegrationTime](#) () override
JeodIntegrationTime destructor.
- [JeodIntegrationTime](#) (const [JeodIntegrationTime](#) &)=delete
- [JeodIntegrationTime](#) & operator= (const [JeodIntegrationTime](#) &)=delete
- virtual double [get_timestamp_time](#) () const =0
Get the time used to timestamp some object.
- void [add_time_change_subscriber](#) ([TimeChangeSubscriber](#) &subscriber)
Add a time change subscriber.
- void [remove_time_change_subscriber](#) ([TimeChangeSubscriber](#) &subscriber)
Remove a time change subscriber.

Protected Member Functions

- void [notify_time_change_subscribers](#) ()
Notify subscribers that the nature of time has changed.

Private Attributes

- [JeodPointerVector](#)< [TimeChangeSubscriber](#) >::type [time_change_subscribers](#)
List of pointers to objects that wish to be notified of a change in the nature of time.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__JeodIntegrationTime](#) ()

8.19.1 Detailed Description

The class [JeodIntegrationTime](#) adds the concepts of timestamping and notification of changes in the nature of time to the ER7 numerical utilities TimeInterface class.

Definition at line 84 of file jeod_integration_time.hh.

8.19.2 Constructor & Destructor Documentation

8.19.2.1 JeodIntegrationTime() [1/2]

```
jeod::JeodIntegrationTime::JeodIntegrationTime ( )
```

[JeodIntegrationTime](#) constructor.

Definition at line 45 of file jeod_integration_time.cc.

References [time_change_subscribers](#).

8.19.2.2 ~JeodIntegrationTime()

```
jeod::JeodIntegrationTime::~~JeodIntegrationTime ( ) [override]
```

[JeodIntegrationTime](#) destructor.

Definition at line 57 of file jeod_integration_time.cc.

References [time_change_subscribers](#).

8.19.2.3 JeodIntegrationTime() [2/2]

```
jeod::JeodIntegrationTime::JeodIntegrationTime (
    const JeodIntegrationTime & ) [delete]
```

8.19.3 Member Function Documentation

8.19.3.1 add_time_change_subscriber()

```
void jeod::JeodIntegrationTime::add_time_change_subscriber (
    TimeChangeSubscriber & subscriber )
```

Add a time change subscriber.

Parameters

<i>subscriber</i>	Object to be added to list of subscribers.
-------------------	--

Definition at line 68 of file jeod_integration_time.cc.

References jeod::IntegrationMessages::invalid_request, and time_change_subscribers.

Referenced by jeod::JeodIntegrationGroup::JeodIntegrationGroup().

8.19.3.2 get_timestamp_time()

```
virtual double jeod::JeodIntegrationTime::get_timestamp_time ( ) const [pure virtual]
```

Get the time used to timestamp some object.

8.19.3.3 notify_time_change_subscribers()

```
void jeod::JeodIntegrationTime::notify_time_change_subscribers ( ) [protected]
```

Notify subscribers that the nature of time has changed.

Definition at line 108 of file jeod_integration_time.cc.

References time_change_subscribers.

8.19.3.4 operator=()

```
JeodIntegrationTime& jeod::JeodIntegrationTime::operator= (
    const JeodIntegrationTime & ) [delete]
```

8.19.3.5 remove_time_change_subscriber()

```
void jeod::JeodIntegrationTime::remove_time_change_subscriber (
    TimeChangeSubscriber & subscriber )
```

Remove a time change subscriber.

Parameters

<i>subscriber</i>	Object to be removed from list of subscribers.
-------------------	--

Definition at line 88 of file jeod_integration_time.cc.

References `jeod::IntegrationMessages::invalid_request`, and `time_change_subscribers`.

Referenced by `jeod::JeodIntegrationGroup::~~JeodIntegrationGroup()`.

8.19.4 Friends And Related Function Documentation

8.19.4.1 init_attrjeod__JeodIntegrationTime

```
void init_attrjeod__JeodIntegrationTime ( ) [friend]
```

8.19.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 86 of file jeod_integration_time.hh.

8.19.5 Field Documentation

8.19.5.1 time_change_subscribers

```
JeodPointerVector<TimeChangeSubscriber>::type jeod::JeodIntegrationTime::time_change_subscribers  
[private]
```

List of pointers to objects that wish to be notified of a change in the nature of time.

```
trick_io(**)
```

Definition at line 116 of file jeod_integration_time.hh.

Referenced by `add_time_change_subscriber()`, `JeodIntegrationTime()`, `notify_time_change_subscribers()`, `remove_time_change_subscriber()`, and `~JeodIntegrationTime()`.

The documentation for this class was generated from the following files:

- [jeod_integration_time.hh](#)
- [jeod_integration_time.cc](#)

8.20 jeod::LsodeControlDataInterface Class Reference

Specifies controls for an LSODE integrator.

```
#include <lsode_control_data_interface.hh>
```

Public Types

- enum [IntegrationMethod](#) { [ImplicitAdamsNonStiff](#) = 1, [ImplicitBackDiffStiff](#) = 2 }
- enum [CorrectorMethod](#) { [FunctionalIteration](#) = 0, [NewtonIterUserJac](#) = 1, [NewtonIterInternalJac](#) = 2, [JacobiNewtonInternalJac](#) = 3, [NewtonIterUserBandJac](#) = 4, [NewtonIterInternalBandJac](#) = 5 }
- enum [ErrorControlIndicator](#) { [CommonAbsCommonRel](#) = 1, [SpecificAbsCommonRel](#) = 2, [CommonAbsSpecificRel](#) = 3, [SpecificAbsSpecificRel](#) = 4 }

Public Member Functions

- [LsodeControlDataInterface](#) ()
constructor
- virtual [~LsodeControlDataInterface](#) ()
- [LsodeControlDataInterface](#) (const [LsodeControlDataInterface](#) &)=default
- [LsodeControlDataInterface](#) & [operator=](#) (const [LsodeControlDataInterface](#) &)=delete
- void [check_interface_data](#) ()
verifies that the input data has legal values.
- void [set_rel_tol](#) (int index, double value)
set values from external
- void [set_abs_tol](#) (int index, double value)
- void [allocate_arrays](#) ()
allocates space for vector-populated data to allow for restart
- void [destroy_allocated_arrays](#) ()
De-allocates allocated array.
- bool [is_corrector_method_functional_iteration](#) ()
Tests whether corrector is functional iteration.

Data Fields

- [ErrorControlIndicator](#) [error_control_indicator](#) {[CommonAbsCommonRel](#)}
Was ITOL.
- std::vector< double > [abs_tolerance_error_control_vec](#)
Temporary pre-initialized place to store loaded error values.
- std::vector< double > [rel_tolerance_error_control_vec](#)
Temporary pre-initialized place to store loaded error values.
- bool [error_control_vector_copied_over](#) {}
- unsigned int [num_odes_at_alloc](#) {}
- double * [abs_tolerance_error_control](#) {}
Was ATOL.
- double * [rel_tolerance_error_control](#) {}
Was RTOL.
- unsigned int [num_odes](#) {3}

- Was *N*, in DLS001 common block.
- [IntegrationMethod](#) `integration_method` {ImplicitAdamsNonStiff}
 - Was *METH*, in DLS001 common block.
- [CorrectorMethod](#) `corrector_method` {FunctionalIteration}
 - Was *MITER*, in DLS001 common block.
- double [min_step_size](#) {}
 - was *HMIN*, in DLS001 common block.
- double [max_step_size](#) {}
 - was *HMAX*.
- double [initial_step_size](#) {}
 - Was *H0*.
- unsigned int [max_order](#) {12}
 - Was *MAXORD*, in DLS001 common block.
- unsigned int [max_num_small_step_warnings](#) {10}
 - Was *MXHNILI*, in DLS001 common block.
- unsigned int [max_correction_iters](#) {3}
 - Was *MAXCOR*, in DLS001 common block Maximum number of corrector-iterations to attempt on any step.
- unsigned int [max_num_steps_jacobian](#) {20}
 - Was *MSBP*, in DLS001 common block.
- unsigned int [max_num_conv_failure](#) {10}
 - Was *MXNCF*, in DLS001 common block.
- unsigned int [max_num_steps](#) {500}
 - Was *MXSTEP*, in DLS001 common block.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__LsodeControlDataInterface](#) ()

8.20.1 Detailed Description

Specifies controls for an LSODE integrator.

Definition at line 82 of file `lsode_control_data_interface.hh`.

8.20.2 Member Enumeration Documentation

8.20.2.1 CorrectorMethod

```
enum jeod::LsodeControlDataInterface::CorrectorMethod
```

Enumerator

FunctionalIteration	Functional iteration.
NewtonIterUserJac	Modified Newton iteration with.
NewtonIterInternalJac	Modified Newton iteration with internally.
JacobiNewtonInternalJac	Modified Jacobi-Newton iteration with.
NewtonIterUserBandJac	Modified Newton iteration with.
NewtonIterInternalBandJac	Modified Newton iteration with internally.

Definition at line 100 of file `lsode_control_data_interface.hh`.

8.20.2.2 ErrorControlIndicator

```
enum jeod::LsodeControlDataInterface::ErrorControlIndicator
```

Enumerator

CommonAbsCommonRel	Use the same absolute and relative values.
SpecificAbsCommonRel	Use separate absolute values for each.
CommonAbsSpecificRel	Use a common absolute values and separate.
SpecificAbsSpecificRel	Use separate absolute and relative values.

Definition at line 120 of file `lsode_control_data_interface.hh`.

8.20.2.3 IntegrationMethod

```
enum jeod::LsodeControlDataInterface::IntegrationMethod
```

Enumerator

ImplicitAdamsNonStiff	Variable-step, variable-order, implicit Adams.
ImplicitBackDiffStiff	Variable-step, variable-order, implicit.

Definition at line 88 of file `lsode_control_data_interface.hh`.

8.20.3 Constructor & Destructor Documentation

8.20.3.1 ~LsodeControlDataInterface()

```
virtual jeod::LsodeControlDataInterface::~~LsodeControlDataInterface ( ) [inline], [virtual]
```

Definition at line 134 of file `lsode_control_data_interface.hh`.

8.20.3.2 LsodeControlDataInterface()

```
jeod::LsodeControlDataInterface::LsodeControlDataInterface (
    const LsodeControlDataInterface & ) [default]
```

8.20.4 Member Function Documentation

8.20.4.1 is_corrector_method_functional_iteration()

```
bool jeod::LsodeControlDataInterface::is_corrector_method_functional_iteration ( ) [inline]
```

Tests whether corrector is functional iteration.

Definition at line 151 of file lsode_control_data_interface.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_core(), jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part1(), jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2(), jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration(), jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part2(), jeod::LsodeFirstOrderODEIntegrator::integrator_predict(), and jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part1().

8.20.4.2 operator=()

```
LsodeControlDataInterface& jeod::LsodeControlDataInterface::operator= (
    const LsodeControlDataInterface & ) [delete]
```

8.20.5 Friends And Related Function Documentation

8.20.5.1 init_attrjeod__LsodeControlDataInterface

```
void init_attrjeod__LsodeControlDataInterface ( ) [friend]
```

8.20.5.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 84 of file lsode_control_data_interface.hh.

8.20.6 Field Documentation

8.20.6.1 abs_tolerance_error_control

```
double* jeod::LsodeControlDataInterface::abs_tolerance_error_control {}
```

Was ATOL.

Vector of the absolute error tolerances.`trick_units(-)`

Definition at line 177 of file `lsode_control_data_interface.hh`.

Referenced by `allocate_arrays()`, `destroy_allocated_arrays()`, `jeod::LsodeFirstOrderODEIntegrator::load_ew↵values()`, `jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part2()`, and `set_abs_tol()`.

8.20.6.2 abs_tolerance_error_control_vec

```
std::vector<double> jeod::LsodeControlDataInterface::abs_tolerance_error_control_vec
```

Temporary pre-initialized place to store loaded error values.

`trick_units(-)`

Definition at line 165 of file `lsode_control_data_interface.hh`.

Referenced by `allocate_arrays()`, `check_interface_data()`, `LsodeControlDataInterface()`, and `set_abs_tol()`.

8.20.6.3 corrector_method

```
CorrectorMethod jeod::LsodeControlDataInterface::corrector_method {FunctionalIteration}
```

Was MITER, in DLS001 common block.

`trick_units(-)`

Definition at line 200 of file `lsode_control_data_interface.hh`.

Referenced by `check_interface_data()`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init()`, `jeod::Lsode↵FirstOrderODEIntegrator::jacobian_prep_loop()`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_wrap_↵up()`, `jeod::LsodeFirstOrderODEIntegrator::linear_chord_iteration()`, and `jeod::LsodeFirstOrderODEIntegrator↵::manager_initialize_calculation_part1()`.

8.20.6.4 error_control_indicator

```
ErrorControlIndicator jeod::LsodeControlDataInterface::error_control_indicator {CommonAbsCommonRel}
```

Was ITOL.

`trick_units(-)`

Definition at line 160 of file `lsode_control_data_interface.hh`.

Referenced by `check_interface_data()`, `jeod::LsodeFirstOrderODEIntegrator::load_ew_values()`, and `jeod::Lsode↵FirstOrderODEIntegrator::manager_initialize_calculation_part2()`.

8.20.6.5 error_control_vector_copied_over

```
bool jeod::LsodeControlDataInterface::error_control_vector_copied_over {}
```

Definition at line 172 of file lsode_control_data_interface.hh.

Referenced by `allocate_arrays()`, `check_interface_data()`, `destroy_allocated_arrays()`, `set_abs_tol()`, and `set_rel_tol()`.

8.20.6.6 initial_step_size

```
double jeod::LsodeControlDataInterface::initial_step_size {}
```

Was H0.

Initial guess at the step size. May be input, will be calculated if not. Note - this is the actual step, not the magnitude of the step. whereas `max_step_size` and `min_step_size` are magnitudes.`trick_units(-)`

Definition at line 220 of file lsode_control_data_interface.hh.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part2()`, `jeod::LsodeFirstOrderODEIntegrator::process_entry_point_cycle_start()`, and `jeod::LsodeFirstOrderODEIntegrator::reset_integrator()`.

8.20.6.7 integration_method

```
IntegrationMethod jeod::LsodeControlDataInterface::integration_method {ImplicitAdamsNonStiff}
```

Was METH, in DLS001 common block.

`trick_units(-)`

Definition at line 195 of file lsode_control_data_interface.hh.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::calculate_integration_coefficients()`, `check_interface_data()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_core()`, and `jeod::LsodeFirstOrderODEIntegrator::update_control_data()`.

8.20.6.8 max_correction_iters

```
unsigned int jeod::LsodeControlDataInterface::max_correction_iters {3}
```

Was MAXCOR, in DLS001 common block Maximum number of corrector-iterations to attempt on any step.

`trick_units(-)`

Definition at line 236 of file lsode_control_data_interface.hh.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration()`.

8.20.6.9 max_num_conv_failure

```
unsigned int jeod::LsodeControlDataInterface::max_num_conv_failure {10}
```

Was MXNCF, in DLS001 common block.

Maximum number of convergence failures on one step.trick_units(-)

Definition at line 246 of file lsode_control_data_interface.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2().

8.20.6.10 max_num_small_step_warnings

```
unsigned int jeod::LsodeControlDataInterface::max_num_small_step_warnings {10}
```

Was MXHNILI, in DLS001 common block.

Populated from IWORK[7] Maximum number of small-step warnings that may be printed.trick_units(-)

Definition at line 231 of file lsode_control_data_interface.hh.

Referenced by check_interface_data(), and jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_↔ part2().

8.20.6.11 max_num_steps

```
unsigned int jeod::LsodeControlDataInterface::max_num_steps {500}
```

Was MXSTEP, in DLS001 common block.

Maximum number of steps that the integrator may take. Default = 500.trick_units(-)

Definition at line 251 of file lsode_control_data_interface.hh.

Referenced by check_interface_data(), and jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_↔ part1().

8.20.6.12 max_num_steps_jacobian

```
unsigned int jeod::LsodeControlDataInterface::max_num_steps_jacobian {20}
```

Was MSBP, in DLS001 common block.

Populated from IWORK[6] Maximum number of steps for which the same Jacobian can be used.trick_units(-)

Definition at line 241 of file lsode_control_data_interface.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_predict().

8.20.6.13 max_order

```
unsigned int jeod::LsodeControlDataInterface::max_order {12}
```

Was MAXORD, in DLS001 common block.

Populated from IWORK[5] Maximum order allowable.trick_units(-)

Definition at line 226 of file lsode_control_data_interface.hh.

Referenced by check_interface_data(), and jeod::LsodeFirstOrderODEIntegrator::update_control_data().

8.20.6.14 max_step_size

```
double jeod::LsodeControlDataInterface::max_step_size {}
```

was HMAX.

RWORK[6] Maximum absolute value of step size allowable. Default to 0.0, interpreted as infinity. user-specified otherwise.trick_units(-)

Definition at line 213 of file lsode_control_data_interface.hh.

Referenced by check_interface_data(), and jeod::LsodeFirstOrderODEIntegrator::update_control_data().

8.20.6.15 min_step_size

```
double jeod::LsodeControlDataInterface::min_step_size {}
```

was HMIN, in DLS001 common block.

Minimum absolute value of step size allowable. Default to 0.0, user-specified otherwise.trick_units(-)

Definition at line 207 of file lsode_control_data_interface.hh.

Referenced by check_interface_data(), jeod::LsodeFirstOrderODEIntegrator::integrator_core(), jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2(), jeod::LsodeFirstOrderODEIntegrator::integrator_error_test_failed(), jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part1(), jeod::LsodeFirstOrderODEIntegrator::integrator_reset_method_coeffs(), jeod::LsodeFirstOrderODEIntegrator::integrator_set_new_order(), and jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part3().

8.20.6.16 num_odes

```
unsigned int jeod::LsodeControlDataInterface::num_odes {3}
```

Was N, in DLS001 common block.

Number of ODEs to be solved at next step. In this implementation, num_odes = num_equations. In original implementation, num_odes (N) was set to NEQ at the start, the some subset could be identified, NYH and solved for.trick_units(-)

Definition at line 190 of file lsode_control_data_interface.hh.

Referenced by allocate_arrays(), check_interface_data(), jeod::LsodeFirstOrderODEIntegrator::gauss_elim_factor(), jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order(), jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_prep(), jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_converged(), jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration(), jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part1(), jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part2(), jeod::LsodeFirstOrderODEIntegrator::integrator_reset_iteration_loop_part1(), jeod::LsodeFirstOrderODEIntegrator::integrator_reset_iteration_loop_part2(), jeod::LsodeFirstOrderODEIntegrator::integrator_reset_yh(), jeod::LsodeFirstOrderODEIntegrator::integrator_wrapup(), jeod::LsodeFirstOrderODEIntegrator::interpolate_y(), jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init(), jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop(), jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_wrap_up(), jeod::LsodeFirstOrderODEIntegrator::linear_chord_iteration(), jeod::LsodeFirstOrderODEIntegrator::linear_solver(), jeod::LsodeFirstOrderODEIntegrator::load_derivatives(), jeod::LsodeFirstOrderODEIntegrator::load_ew_values(), jeod::LsodeFirstOrderODEIntegrator::LsodeFirstOrderODEIntegrator(), jeod::LsodeFirstOrderODEIntegrator::magnitude_of_weighted_array(), jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part1(), jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part2(), jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part1(), jeod::LsodeFirstOrderODEIntegrator::process_entry_point_cycle_start(), and jeod::LsodeFirstOrderODEIntegrator::reset_integrator().

8.20.6.17 num_odes_at_alloc

```
unsigned int jeod::LsodeControlDataInterface::num_odes_at_alloc {}
```

Definition at line 173 of file lsode_control_data_interface.hh.

Referenced by allocate_arrays(), set_abs_tol(), and set_rel_tol().

8.20.6.18 rel_tolerance_error_control

```
double* jeod::LsodeControlDataInterface::rel_tolerance_error_control {}
```

Was RTOL.

Vector of the relative error tolerances.trick_units(-)

Definition at line 181 of file lsode_control_data_interface.hh.

Referenced by allocate_arrays(), destroy_allocated_arrays(), jeod::LsodeFirstOrderODEIntegrator::load_ew_values(), jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part2(), and set_rel_tol().

8.20.6.19 rel_tolerance_error_control_vec

```
std::vector<double> jeod::LsodeControlDataInterface::rel_tolerance_error_control_vec
```

Temporary pre-initialized place to store loaded error values.

trick_units(-)

Definition at line 170 of file lsode_control_data_interface.hh.

Referenced by allocate_arrays(), check_interface_data(), LsodeControlDataInterface(), and set_rel_tol().

The documentation for this class was generated from the following files:

- [lsode_control_data_interface.hh](#)
- [lsode_control_data_interface.cc](#)

8.21 jeod::LsodeDataArrays Class Reference

The data arrays.

```
#include <lsode_data_classes.hh>
```

Public Member Functions

- [LsodeDataArrays](#) ()=default
- virtual [~LsodeDataArrays](#) ()
- [LsodeDataArrays](#) & [operator=](#) (const [LsodeDataArrays](#) &)=delete
- [LsodeDataArrays](#) (const [LsodeDataArrays](#) &)=delete
- void [allocate_arrays](#) (unsigned int [num_odes](#), [LsodeControlDataInterface::CorrectorMethod](#) [corrector_](#)↔
method)
Allocates memory for the variable size arrays.
- void [destroy_allocated_arrays](#) ()
Allows for refactoring and reallocation of newly sized arrays.

Data Fields

- int * [pivots](#) {}
Was IWM(21) or IPV T.
- double ** [history](#) {}
was RWORK[LYH:LYH+NYH(MAXORD+1)-1].*
- double [lin_alg_1](#) {}
- double [lin_alg_2](#) {}
- double ** [lin_alg](#) {}
was RWORK[LWM:LWM+LENWM-1].
- double * [error_weight](#) {}
was RWORK[LEWT:LEWT+N-1].
- double * [save](#) {}
was RWORK[LSAVF:LSAVF+N-1].
- double * [accum_correction](#) {}
was RWORK[LACOR:LACOR+N-1].
- unsigned int [lin_alg_index1](#) {}
Number of record, this is the value used for data allocation.
- unsigned int [num_odes](#) {3}
Number of record, this is the value used for data allocation.
- bool [allocated](#) {}
Indicator of whether the arrays have been allocated.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__LsodeDataArrays](#) ()

8.21.1 Detailed Description

The data arrays.

Definition at line 107 of file `lsode_data_classes.hh`.

8.21.2 Constructor & Destructor Documentation

8.21.2.1 `LsodeDataArrays()` [1/2]

```
jeod::LsodeDataArrays::LsodeDataArrays ( ) [default]
```

8.21.2.2 `~LsodeDataArrays()`

```
virtual jeod::LsodeDataArrays::~~LsodeDataArrays ( ) [inline], [virtual]
```

Definition at line 112 of file `lsode_data_classes.hh`.

References `destroy_allocated_arrays()`.

8.21.2.3 `LsodeDataArrays()` [2/2]

```
jeod::LsodeDataArrays::LsodeDataArrays (
    const LsodeDataArrays & ) [delete]
```

8.21.3 Member Function Documentation

8.21.3.1 `operator=()`

```
LsodeDataArrays& jeod::LsodeDataArrays::operator= (
    const LsodeDataArrays & ) [delete]
```

8.21.4 Friends And Related Function Documentation

8.21.4.1 init_attrjeod__LsodeDataArrays

```
void init_attrjeod__LsodeDataArrays ( ) [friend]
```

8.21.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 109 of file lside_data_classes.hh.

8.21.5 Field Documentation

8.21.5.1 accum_correction

```
double* jeod::LsodeDataArrays::accum_correction {}
```

was RWORK[LACOR:LACOR+N-1].

LACOR = LSAVF + N acum_correction[i] = rwork[lacor+i].trick_units(-)

Definition at line 173 of file lside_data_classes.hh.

Referenced by `allocate_arrays()`, `destroy_allocated_arrays()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_←_compute_new_order()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_prep()`, `jeod_←::LsodeFirstOrderODEIntegrator::integrator_corrector_converged()`, `jeod::LsodeFirstOrderODEIntegrator_←::integrator_corrector_iteration()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_reset_iteration_loop_part2()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_wrapup()`, and `jeod::LsodeFirstOrderODEIntegrator::jacobian_←_prep_loop()`.

8.21.5.2 allocated

```
bool jeod::LsodeDataArrays::allocated {}
```

Indicator of whether the arrays have been allocated.

`trick_units(-)`

Definition at line 187 of file lside_data_classes.hh.

Referenced by `allocate_arrays()`, and `destroy_allocated_arrays()`.

8.21.5.3 error_weight

```
double* jeod::LsodeDataArrays::error_weight {}
```

was RWORK[LEWT:LEWT+N-1].

LEWT = LWM + LENWM error_weight[i] = rwork[lewt+i].trick_units(-)

Definition at line 163 of file lsode_data_classes.hh.

Referenced by allocate_arrays(), destroy_allocated_arrays(), jeod::LsodeFirstOrderODEIntegrator::jacobian_↵
prep_init(), jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop(), jeod::LsodeFirstOrderODEIntegrator↵
::jacobian_prep_wrap_up(), jeod::LsodeFirstOrderODEIntegrator::load_ew_values(), jeod::LsodeFirstOrderODE↵
Integrator::magnitude_of_weighted_array(), jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation↵
_part2(), and jeod::LsodeFirstOrderODEIntegrator::manager_integration_loop_part1().

8.21.5.4 history

```
double** jeod::LsodeDataArrays::history {}
```

was RWORK[LYH:LYH+NYH*(MAXORD+1)-1].

LYH = 21 First index is to "i" in y_i, second index is to history order. history[i,j] = rwork[LYH + j*nyh + i], with lyh = 21 typically.trick_units(-)

Definition at line 142 of file lsode_data_classes.hh.

Referenced by allocate_arrays(), destroy_allocated_arrays(), jeod::LsodeFirstOrderODEIntegrator::integrator↵
_compute_new_order(), jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_prep(), jeod↵
::LsodeFirstOrderODEIntegrator::integrator_corrector_converged(), jeod::LsodeFirstOrderODEIntegrator↵
::integrator_corrector_failed_part2(), jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration(), jeod↵
::LsodeFirstOrderODEIntegrator::integrator_error_test_failed(), jeod::LsodeFirstOrderODEIntegrator::integrator↵
_fail_reset_order_1_part1(), jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part2(), jeod↵
::LsodeFirstOrderODEIntegrator::integrator_predict(), jeod::LsodeFirstOrderODEIntegrator::integrator_reset_↵
iteration_loop_part1(), jeod::LsodeFirstOrderODEIntegrator::integrator_reset_yh(), jeod::LsodeFirstOrderODE↵
Integrator::interpolate_y(), jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init(), jeod::LsodeFirstOrderO↵
DEIntegrator::jacobian_prep_wrap_up(), jeod::LsodeFirstOrderODEIntegrator::load_ew_values(), jeod::Lsode↵
FirstOrderODEIntegrator::manager_initialize_calculation_part2(), jeod::LsodeFirstOrderODEIntegrator::manager↵
_integration_loop_part2(), and jeod::LsodeFirstOrderODEIntegrator::reset_integrator().

8.21.5.5 lin_alg

```
double** jeod::LsodeDataArrays::lin_alg {}
```

was RWORK[LWM:LWM+LENWM-1].

LWM = LYH + (NYH*(MAXORD+1)) lin_alg_1 = rwork[lwm] lin_alg_2 = rwork[lwm + 1] lin_alg[i,j] = rwork[lwm+ j*n
+ i + 2]. The first two elements are treated differently, then it goes to an array that is sized based on the correction↵
_method. The array sizes are as follows, ordered by value of correction_method: 0: 0 1,2: n x n 3: 1 x n 4,5:
(2*ml+mu+1) x n.trick_units(-)

Definition at line 158 of file lsode_data_classes.hh.

Referenced by allocate_arrays(), destroy_allocated_arrays(), jeod::LsodeFirstOrderODEIntegrator::gauss_↵
elim_factor(), jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init(), jeod::LsodeFirstOrderODEIntegrator↵
::jacobian_prep_loop(), jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_wrap_up(), jeod::LsodeFirstOrderO↵
DEIntegrator::linear_chord_iteration(), and jeod::LsodeFirstOrderODEIntegrator::linear_solver().

8.21.5.6 lin_alg_1

```
double jeod::LsodeDataArrays::lin_alg_1 {}
```

Definition at line 143 of file lsode_data_classes.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init(), jeod::LsodeFirstOrderODEIntegrator↔::jacobian_prep_loop(), and jeod::LsodeFirstOrderODEIntegrator::manager_initialize_calculation_part1().

8.21.5.7 lin_alg_2

```
double jeod::LsodeDataArrays::lin_alg_2 {}
```

Definition at line 144 of file lsode_data_classes.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init(), and jeod::LsodeFirstOrderODE↔Integrator::linear_chord_iteration().

8.21.5.8 lin_alg_index1

```
unsigned int jeod::LsodeDataArrays::lin_alg_index1 {}
```

Number of record, this is the value used for data allocation.

trick_units(−)

Definition at line 178 of file lsode_data_classes.hh.

Referenced by allocate_arrays(), and destroy_allocated_arrays().

8.21.5.9 num_odes

```
unsigned int jeod::LsodeDataArrays::num_odes {3}
```

Number of record, this is the value used for data allocation.

trick_units(−)

Definition at line 182 of file lsode_data_classes.hh.

Referenced by allocate_arrays(), and destroy_allocated_arrays().

8.21.5.10 pivots

```
int* jeod::LsodeDataArrays::pivots {}
```

Was IWM(21) or IPVT.

Pivot vector generated in dgefa, and used in dgesl.trick_units(-)

Definition at line 127 of file lsode_data_classes.hh.

Referenced by `allocate_arrays()`, `destroy_allocated_arrays()`, `jeod::LsodeFirstOrderODEIntegrator::gauss_elim_factor()`, and `jeod::LsodeFirstOrderODEIntegrator::linear_solver()`.

8.21.5.11 save

```
double* jeod::LsodeDataArrays::save {}
```

was RWORK[LSAVF:LSAVF+N-1].

LSAVF = LEWT + N save[i] = rwork[lsavf+i].trick_units(-)

Definition at line 168 of file lsode_data_classes.hh.

Referenced by `allocate_arrays()`, `destroy_allocated_arrays()`, `jeod::LsodeFirstOrderODEIntegrator::integrate()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_prep()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_core()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_iteration()`, `jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part2()`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init()`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop()`, and `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_wrap_up()`.

The documentation for this class was generated from the following files:

- [lsode_data_classes.hh](#)
- [lsode_data_classes.cc](#)

8.22 jeod::LsodeDataJacobianPrep Class Reference

Data associated with the method DPREPJ.

```
#include <lsode_data_classes.hh>
```

Public Member Functions

- [LsodeDataJacobianPrep\(\)](#)=default
- virtual [~LsodeDataJacobianPrep\(\)](#)=default
- [LsodeDataJacobianPrep & operator=](#) (const [LsodeDataJacobianPrep](#) &)=delete
- [LsodeDataJacobianPrep](#) (const [LsodeDataJacobianPrep](#) &)=delete

Data Fields

- double [fac](#) {}
- double [hl0](#) {}
- int [index](#) {}
- int [index_max](#) {}
- double [r0](#) {}
- double [yj](#) {}

Friends

- class [InputProcessor](#)
- void [init_attrjeod__LsodeDataJacobianPrep](#) ()

8.22.1 Detailed Description

Data associated with the method DPREPJ.

Definition at line 86 of file `lsode_data_classes.hh`.

8.22.2 Constructor & Destructor Documentation

8.22.2.1 LsodeDataJacobianPrep() [1/2]

```
jeod::LsodeDataJacobianPrep::LsodeDataJacobianPrep ( ) [default]
```

8.22.2.2 ~LsodeDataJacobianPrep()

```
virtual jeod::LsodeDataJacobianPrep::~~LsodeDataJacobianPrep ( ) [virtual], [default]
```

8.22.2.3 LsodeDataJacobianPrep() [2/2]

```
jeod::LsodeDataJacobianPrep::LsodeDataJacobianPrep (
    const LsodeDataJacobianPrep & ) [delete]
```

8.22.3 Member Function Documentation

8.22.3.1 operator=()

```
LsodeDataJacobianPrep& jeod::LsodeDataJacobianPrep::operator= (
    const LsodeDataJacobianPrep & ) [delete]
```

8.22.4 Friends And Related Function Documentation

8.22.4.1 init_attrjeod__LsodeDataJacobianPrep

```
void init_attrjeod__LsodeDataJacobianPrep ( ) [friend]
```

8.22.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 88 of file `lsode_data_classes.hh`.

8.22.5 Field Documentation

8.22.5.1 fac

```
double jeod::LsodeDataJacobianPrep::fac {}
```

Definition at line 96 of file `lsode_data_classes.hh`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init()`, and `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop()`.

8.22.5.2 h10

```
double jeod::LsodeDataJacobianPrep::h10 {}
```

Definition at line 97 of file `lsode_data_classes.hh`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init()`, `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop()`, and `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_wrap_up()`.

8.22.5.3 index

```
int jeod::LsodeDataJacobianPrep::index {}
```

Definition at line 98 of file `lsode_data_classes.hh`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init()`, and `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop()`.

8.22.5.4 index_max

```
int jeod::LsodeDataJacobianPrep::index_max {}
```

Definition at line 99 of file `lsode_data_classes.hh`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init()`, and `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop()`.

8.22.5.5 r0

```
double jeod::LsodeDataJacobianPrep::r0 {}
```

Definition at line 100 of file `lsode_data_classes.hh`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init()`, and `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop()`.

8.22.5.6 yj

```
double jeod::LsodeDataJacobianPrep::yj {}
```

Definition at line 101 of file `lsode_data_classes.hh`.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_init()`, and `jeod::LsodeFirstOrderODEIntegrator::jacobian_prep_loop()`.

The documentation for this class was generated from the following file:

- [lsode_data_classes.hh](#)

8.23 jeod::LsodeDataStode Class Reference

The data associated with method `Dstode`.

```
#include <lsode_data_classes.hh>
```

Public Member Functions

- [LsodeDataStode](#) ()=default
- virtual [~LsodeDataStode](#) ()=default
- [LsodeDataStode](#) & [operator=](#) (const [LsodeDataStode](#) &)=delete
- [LsodeDataStode](#) (const [LsodeDataStode](#) &)=delete

Data Fields

- double [step_ratio](#) {}
- double [step_ratio_order_inc](#) {}
- double [told](#) {}
- double [dsm](#) {}
- int [iredo](#) {}
- int [iret](#) {}
- unsigned int [ncf](#) {}
- unsigned int [new_method_order](#) {}

Friends

- class [InputProcessor](#)
- void [init_attrjeod__LsodeDataStode](#) ()

8.23.1 Detailed Description

The data associated with method Dstode.

Definition at line 193 of file lsode_data_classes.hh.

8.23.2 Constructor & Destructor Documentation

8.23.2.1 [LsodeDataStode\(\)](#) [1/2]

```
jeod::LsodeDataStode::LsodeDataStode ( ) [default]
```

8.23.2.2 [~LsodeDataStode\(\)](#)

```
virtual jeod::LsodeDataStode::~~LsodeDataStode ( ) [virtual], [default]
```

8.23.2.3 LsodeDataStode() [2/2]

```
jeod::LsodeDataStode::LsodeDataStode (
    const LsodeDataStode & ) [delete]
```

8.23.3 Member Function Documentation**8.23.3.1 operator=()**

```
LsodeDataStode& jeod::LsodeDataStode::operator= (
    const LsodeDataStode & ) [delete]
```

8.23.4 Friends And Related Function Documentation**8.23.4.1 init_attrjeod__LsodeDataStode**

```
void init_attrjeod__LsodeDataStode ( ) [friend]
```

8.23.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 195 of file lsode_data_classes.hh.

8.23.5 Field Documentation**8.23.5.1 dsm**

```
double jeod::LsodeDataStode::dsm {}
```

Definition at line 207 of file lsode_data_classes.hh.

Referenced by `jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order()`, and `jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_converged()`.

8.23.5.2 iredo

```
int jeod::LsodeDataStode::iredo {}
```

Definition at line 208 of file lsode_data_classes.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_core(), jeod::LsodeFirstOrderODEIntegrator↔
::integrator_corrector_converged(), jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2(),
jeod::LsodeFirstOrderODEIntegrator::integrator_error_test_failed(), jeod::LsodeFirstOrderODEIntegrator↔
::integrator_reset_yh(), and jeod::LsodeFirstOrderODEIntegrator::integrator_test_stepsize_change().

8.23.5.3 iret

```
int jeod::LsodeDataStode::iret {}
```

Definition at line 209 of file lsode_data_classes.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_core(), jeod::LsodeFirstOrderODEIntegrator↔
::integrator_fail_reset_order_1_part2(), jeod::LsodeFirstOrderODEIntegrator::integrator_reset_method_coeffs(),
and jeod::LsodeFirstOrderODEIntegrator::integrator_set_new_order().

8.23.5.4 ncf

```
unsigned int jeod::LsodeDataStode::ncf {}
```

Definition at line 210 of file lsode_data_classes.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_core(), and jeod::LsodeFirstOrderODEIntegrator↔
::integrator_corrector_failed_part2().

8.23.5.5 new_method_order

```
unsigned int jeod::LsodeDataStode::new_method_order {}
```

Definition at line 211 of file lsode_data_classes.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order(), and jeod::LsodeFirst↔
OrderODEIntegrator::integrator_set_new_order().

8.23.5.6 step_ratio

```
double jeod::LsodeDataStode::step_ratio {}
```

Definition at line 204 of file lsode_data_classes.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order(), jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_check_step_error(), jeod::LsodeFirstOrderODEIntegrator::integrator_core(), jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2(), jeod::LsodeFirstOrderODEIntegrator::integrator_fail_reset_order_1_part1(), jeod::LsodeFirstOrderODEIntegrator::integrator_reset_method_coeffs(), jeod::LsodeFirstOrderODEIntegrator::integrator_reset_yh(), jeod::LsodeFirstOrderODEIntegrator::integrator_set_new_order(), and jeod::LsodeFirstOrderODEIntegrator::integrator_test_stepsize_change().

8.23.5.7 step_ratio_order_inc

```
double jeod::LsodeDataStode::step_ratio_order_inc {}
```

Definition at line 205 of file lsode_data_classes.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order(), jeod::LsodeFirstOrderODEIntegrator::integrator_compute_new_order_prep(), and jeod::LsodeFirstOrderODEIntegrator::integrator_error_test_failed().

8.23.5.8 told

```
double jeod::LsodeDataStode::told {}
```

Definition at line 206 of file lsode_data_classes.hh.

Referenced by jeod::LsodeFirstOrderODEIntegrator::integrator_core(), jeod::LsodeFirstOrderODEIntegrator::integrator_corrector_failed_part2(), and jeod::LsodeFirstOrderODEIntegrator::integrator_error_test_failed().

The documentation for this class was generated from the following file:

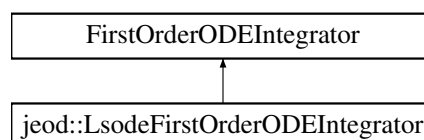
- [lsode_data_classes.hh](#)

8.24 jeod::LsodeFirstOrderODEIntegrator Class Reference

Jeod-compatible version of the Livermore ODE solver, LSODE.

```
#include <lsode_first_order_ode_integrator.hh>
```

Inheritance diagram for jeod::LsodeFirstOrderODEIntegrator:



Public Types

- enum `EntryPoint` {
`CycleStartFinish` = 0, `InitCalc` = 1, `JacobianPrep` = 2, `ResetIterLoop` = 3,
`IterationLoop` = 4, `DstodeResetStep` = 5 }
- enum `CalculationTask` {
`Normal` = 1, `OneStep` = 2, `CompleteCycle` = 3, `NormalWithSingularity` = 4,
`OneStepWithSingularity` = 5 }

Public Member Functions

- `LsodeFirstOrderODEIntegrator` ()
LsodeFirstOrderODEIntegrator default constructor.
- `LsodeFirstOrderODEIntegrator` (const `LsodeControlDataInterface` &data_in, `er7_utils::IntegrationControls` &controls, unsigned int size)
LsodeFirstOrderODEIntegrator non-default constructor.
- `~LsodeFirstOrderODEIntegrator` () override
LsodeFirstOrderODEIntegrator destructor.
- `LsodeFirstOrderODEIntegrator` & `operator=` (const `LsodeFirstOrderODEIntegrator` &)=delete
- `LsodeFirstOrderODEIntegrator` (const `LsodeFirstOrderODEIntegrator` &)=delete
- `LsodeFirstOrderODEIntegrator` * `create_copy` () const override
Create a copy of 'this' *LsodeFirstOrderODEIntegrator* object.
- `EntryPoint` `get_re_entry_point` ()
Get *re_entry_point* member.
- `er7_utils::IntegratorResult` `integrate` (double dyn_dt, unsigned int target_stage, const double *`y_dot`, double *`y`) override
Propagate state via *Lsode's* method.
- void `reset_integrator` () override
Resets the integrator when the timestep changes or when identified as needing a reset.
- void `update_control_data` ()
Gets the control data from where it can be populated in the constructor and verifies that the input control parameters are not out of sensible range.

Data Fields

- double * `y` {}
Was *Y*.
- const double * `y_dot` {}
Was *.*
- double `cycle_target_time` {}
Was *TOUT*.
- double `convergence_factor` {}
was *CONIT*, in *DLS001* common block.
- double `convergence_rate` {}
was *CRATE*, in *DLS001* common block.
- unsigned int `order_select_para` {}
Was *IALTH*, in *DLS001* common block.
- unsigned int `num_equations` {1}
Was *NYH*, in *DLS001* common block.
- unsigned int `num_nordsiek_cols` {1}
Was *L*, in *DLS001* common block Number of columns in *Nordsiek* array.

- unsigned int [max_history_size](#) {12}
Was LMAX, in DLS001 common block Maximum allowable number of histories.
- unsigned int [num_predictor_elements](#) {1}
Was NQNYH, in DLS001 common block.
- unsigned int [method_order_current](#) {}
Was NQ, in DLS001 common block.
- double [stage_target_time](#) {}
was TN, in DLS001 common block.
- double [max_step_increase_ratio](#) {1.0}
was RMAX, in DLS001 common block.
- double [max_rel_change_without_jacobian](#) {0.3}
was CCMAX, in DLS001 common block.

Protected Member Functions

- void [process_entry_point_cycle_start](#) ()
The code block from the main integrate routine for re_entry_point=CycleStartFinish.
- void [manager_initialize_calculation_part1](#) ()
Sets the values for the case with calculation_phase = 1.
- void [manager_initialize_calculation_part2](#) ()
- int [manager_check_stop_conditions](#) ()
verifies whether the convergence conditions have been met to end the cycle.
- void [manager_integration_loop_part1](#) ()
The iteration loop for the integration process.
- void [manager_integration_loop_part2](#) ()
- void [manager_integration_loop_part3](#) ()
- void [manager_set_calculation_phase_eq_2_reload](#) ()
- void [integrator_core](#) ()
*integrator_core provides the front-end to all of the integrator_*i methods, which together perform one step of the integration of an initial value problem for a system of ordinary differential equations.*
- void [integrator_reset_method_coeffs](#) ()
Sets/resets the method_coeffs_current array.
- void [integrator_test_stepsize_change](#) ()
Tests h against old h.
- void [integrator_reset_yh](#) ()
Resets history arrays and time-step.
- void [integrator_predict](#) ()
This section computes the predicted values by multiplying the history array by the Pascal Triangle matrix.
- void [integrator_reset_iteration_loop_part1](#) ()
This method resets the iteration loop to the values generated by the integrator_predict method, which populated history[][0].*
- void [integrator_reset_iteration_loop_part2](#) ()
This code follows part 1 after the break-out to get to the external calls, and completes the reset of the integration iteration loop.
- void [integrator_corrector_iteration](#) ()
Keeps looping through the iterations until convergence or failure.
- void [integrator_corrector_failed_part1](#) ()
The corrector iteration failed to converge.
- void [integrator_corrector_failed_part2](#) ()
Retracts the history array in the case that the correction iteration failed to converge with either functional iteration or with an up-to-date Jacobian.

- void [integrator_corrector_converged](#) ()
Starts the processing of a converged iteration.
- void [integrator_error_test_failed](#) ()
Restores the history array following the failure of the corrector for exceeding local error bounds.
- void [integrator_compute_new_order_prep](#) ()
The first steps in computing whether the order of the integrator should be changed.
- void [integrator_compute_new_order](#) ()
Computes the step-size scaling factors that will result once it is determined what happens to the order at the next step.
- void [integrator_compute_new_order_check_step_error](#) ()
- void [integrator_set_new_order](#) ()
Sets the new order and the step-ratio for the next step - or the current step if redoing it.
- void [integrator_fail_reset_order_1_part1](#) ()
Repopulates the y-array from history, recomputes the original first derivatives, sets the order back to 1, and reduces the step size by a factor of 10.
- void [integrator_fail_reset_order_1_part2](#) ()
Continue reset, with derivatives now at hand.
- void [integrator_wrapup](#) ()
Wraps up the completion of the integrator.
- void [integrator_terminate](#) ()
this is the only succesful path back from integrator to manager.
- void [calculate_epsilon](#) ()
Identify the smallest double precision value, epsilon, such that the computer can distinguish (1+epsilon) from 1.
- void [calculate_integration_coefficients](#) ()
Modified from original DCFODE subroutine.
- void [interpolate_y](#) ()
Interpolates the zeroth-derivative of y Adapted from subroutine DINTDY, which was a general method for interpolating the K-th derivative of the dependent variable vector, y.
- void [jacobian_prep_init](#) ()
Modified from DPREPJ.
- bool [jacobian_prep_loop](#) ()
- bool [jacobian_prep_wrap_up](#) ()
- void [linear_chord_iteration](#) ()
Modified from DSOLSY.
- void [load_ew_values](#) ()
- double [magnitude_of_weighted_array](#) (const double *v)
returns the RMS value of {V dot W}, where V and W are N-vectors.
- double [magnitude_of_weighted_array](#) (unsigned int ix, double **v)
returns RMS value of v[][index]*
- int [gauss_elim_factor](#) ()
Factors a double array (arrays.lin_alg) by Gaussian elimination.
- void [linear_solver](#) ()
Solves the equation $Y' = A Y$, with $A = \text{arrays.lin_alg}$.
- unsigned int [index_of_max_magnitude](#) (unsigned int num_points, double **mx, int starting_ix)
Modified version of IDAMAX.
- void [load_derivatives](#) (double *derivs)
Load the externally generated derivative values (incoming as y_dot)i into the array derivs.

Protected Attributes

- [LsodeDataJacobianPrep data_prepj](#)
data used exclusively for the DPREPJ method.
- [LsodeDataArrays arrays](#)
data arrays, multiple purposes.
- [LsodeDataStode data_stode](#)
data used exclusively for the DSTODE method.
- unsigned int [num_steps_taken](#) {}
Was NST, in DLS001 common block.
- unsigned int [prior_num_steps](#) {}
Was NSLAST, in DLS001 common block.
- int [step_error](#) {}
Was KFLAG, in DLS001 common block 0: step was successful -1: requested accuracy could not be achieved.
- unsigned int [num_small_step_warnings](#) {}
Was NHNIL, in DLS001 common block.
- unsigned int [num_jacobian_evals](#) {}
Was NJE, in DLS001 common block Number of jacobian evaluations so far for the problem.
- double [iter_delta](#) {}
Was DEL, in DSTODE, local variable.
- double [prev_iter_delta](#) {}
Was DELP, in DSTODE, local variable.
- bool [first_pass](#) {true}
was ISTATE.
- [EntryPoint re_entry_point](#) {CycleStartFinish}
Indicates where in the integrator to return to following an exit to gether new derivatives.
- bool [initialized](#) {}
was INIT, in DLS001 common block.
- int [internal_state](#) {}
Was JSTART, in DLS001 common block.
- [CalculationTask calculation_task](#) {Normal}
Was ITASK.
- unsigned int [max_order_internal](#) {}
Was MAXORD.
- [LsodeControlDataInterface control_data](#)
- [LsodeControlDataInterface::IntegrationMethod prev_integration_method](#)
Was MEO, in DLS001 common block Integration method used in previous call (see integration_method).
- unsigned int [prev_method_order](#) {}
Was NQU, in DLS001 common block.
- double [method_coeff_first](#) {}
was EL0, in DLS001 common block.
- double [method_coeffs_current](#) [13] {}
was EL, in DLS001 common block.
- double [method_coeffs_complete](#) [13][12] {}
was ELCO, in DLS001 common block.
- double [test_coeffs_complete](#) [3][12] {}
was TESCO, in DLS001 common block.
- double [step_size](#) {}
was H, in DLS001 common block.
- double [prev_step_size](#) {}
was HOLD, in DLS001 common block.

- double `prev_good_step_size` {}
was HU, in DLS001 common block.
- double `max_step_size_inv` {}
was HMXI.
- bool `jacobian_current` {}
Was JCUR, in DLS001 common block Is the jacobian current.
- bool `update_jacobian` {true}
Was IPUP, in DLS001 common block.
- unsigned int `step_at_last_jacobian_update` {}
Was NSLP, in DLS001 common block.
- unsigned int `convergence_jacobian_flag` {}
Was ICF, in DLS001 common block.
- double `rel_change_since_jacobian` {}
was RC, in DLS001 common block.
- bool `iteration_matrix_singular` {}
Was IERPJ, in DLS001 common block.
- bool `modified_iteration_matrix_singular` {}
Was IERSL, in DLS001 common block.
- unsigned int `iteration_count` {}
- double `epsilon` {1.0E-12}
was UROUND, in DLS001 common block.
- double `sqrt_epsilon` {1.0E-6}
NEW.

Friends

- class `InputProcessor`
- void `init_attrjeod__LsodeFirstOrderODEIntegrator` ()

8.24.1 Detailed Description

Jeod-compatible version of the Livermore ODE solver, LSODE.

Definition at line 95 of file `lsode_first_order_ode_integrator.hh`.

8.24.2 Member Enumeration Documentation

8.24.2.1 CalculationTask

```
enum jeod::LsodeFirstOrderODEIntegrator::CalculationTask
```

Enumerator

Normal	Normal operation. Interpolate to target.
OneStep	Take only one step and return.
CompleteCycle	Stop at first mesh point at or beyond.
NormalWithSingularity	Normal computation, with safeguard on.
OneStepWithSingularity	Take one step without passing t_crit.

Definition at line 126 of file lsode_first_order_ode_integrator.hh.

8.24.2.2 EntryPoint

```
enum jeod::LsodeFirstOrderODEIntegrator::EntryPoint
```

Enumerator

CycleStartFinish	Default value. Assumption is that the current.
InitCalc	Reset during initialization. Valid only during.
JacobianPrep	Set at the end of the initialization of the.
ResetIterLoop	Set when the iteration loop (part of DSTODE) has.
IterationLoop	Set during the routine operation of the iteration.
DstodeResetStep	Set in dstode_640 when there have been too many.

Definition at line 103 of file lsode_first_order_ode_integrator.hh.

8.24.3 Constructor & Destructor Documentation

8.24.3.1 LsodeFirstOrderODEIntegrator()

```
jeod::LsodeFirstOrderODEIntegrator::LsodeFirstOrderODEIntegrator (
    const LsodeFirstOrderODEIntegrator & ) [delete]
```

8.24.4 Member Function Documentation

8.24.4.1 get_re_entry_point()

```
EntryPoint jeod::LsodeFirstOrderODEIntegrator::get_re_entry_point ( ) [inline]
```

Get re_entry_point member.

Definition at line 180 of file lsode_first_order_ode_integrator.hh.

8.24.4.2 operator=()

```
LsodeFirstOrderODEIntegrator& jeod::LsodeFirstOrderODEIntegrator::operator= (
    const LsodeFirstOrderODEIntegrator & ) [delete]
```

8.24.5 Friends And Related Function Documentation

8.24.5.1 init_attrjeod__LsodeFirstOrderODEIntegrator

```
void init_attrjeod__LsodeFirstOrderODEIntegrator ( ) [friend]
```

8.24.5.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 98 of file lsode_first_order_ode_integrator.hh.

8.24.6 Field Documentation

8.24.6.1 arrays

```
LsodeDataArrays jeod::LsodeFirstOrderODEIntegrator::arrays [protected]
```

data arrays, multiple purposes.

trick_units(-)

Definition at line 392 of file lsode_first_order_ode_integrator.hh.

Referenced by gauss_elim_factor(), integrate(), integrator_compute_new_order(), integrator_compute_new_order_prep(), integrator_core(), integrator_corrector_converged(), integrator_corrector_failed_part2(), integrator_corrector_iteration(), integrator_error_test_failed(), integrator_fail_reset_order_1_part1(), integrator_fail_reset_order_1_part2(), integrator_predict(), integrator_reset_iteration_loop_part1(), integrator_reset_iteration_loop_part2(), integrator_reset_yh(), integrator_wrapup(), interpolate_y(), jacobian_prep_init(), jacobian_prep_loop(), jacobian_prep_wrap_up(), linear_chord_iteration(), linear_solver(), load_ew_values(), magnitude_of_weighted_array(), manager_initialize_calculation_part1(), manager_initialize_calculation_part2(), manager_integration_loop_part1(), manager_integration_loop_part2(), reset_integrator(), and ~LsodeFirstOrderODEIntegrator().

8.24.6.2 calculation_task

```
CalculationTask jeod::LsodeFirstOrderODEIntegrator::calculation_task {Normal} [protected]
```

Was ITASK.

1: Normal 2:Take one step and return. 3:Stop at first mesh point at or beyond cycle_target_time and return 4:Normal computation, with safeguard on singularity time, t_crit 5:Take one step without passing t_crit. This implementation only allows for calculation_task = 1, so it is a protected variable until such time as it is extended to include additional options.

Only case 1 is supported.trick_units(-)

Definition at line 480 of file lsode_first_order_ode_integrator.hh.

Referenced by manager_check_stop_conditions(), and manager_integration_loop_part3().

8.24.6.3 control_data

```
LsodeControlDataInterface jeod::LsodeFirstOrderODEIntegrator::control_data [protected]
```

Definition at line 487 of file lsode_first_order_ode_integrator.hh.

Referenced by calculate_integration_coefficients(), gauss_elim_factor(), integrator_compute_new_order(), integrator_compute_new_order_prep(), integrator_core(), integrator_corrector_converged(), integrator_corrector_failed_part1(), integrator_corrector_failed_part2(), integrator_corrector_iteration(), integrator_error_test_failed(), integrator_fail_reset_order_1_part1(), integrator_fail_reset_order_1_part2(), integrator_predict(), integrator_reset_iteration_loop_part1(), integrator_reset_iteration_loop_part2(), integrator_reset_method_coeffs(), integrator_reset_yh(), integrator_set_new_order(), integrator_wrapup(), interpolate_y(), jacobian_prep_init(), jacobian_prep_loop(), jacobian_prep_wrap_up(), linear_chord_iteration(), linear_solver(), load_derivatives(), load_ew_values(), LsodeFirstOrderODEIntegrator(), magnitude_of_weighted_array(), manager_initialize_calculation_part1(), manager_initialize_calculation_part2(), manager_integration_loop_part1(), manager_integration_loop_part2(), manager_integration_loop_part3(), process_entry_point_cycle_start(), reset_integrator(), update_control_data(), and ~LsodeFirstOrderODEIntegrator().

8.24.6.4 convergence_factor

```
double jeod::LsodeFirstOrderODEIntegrator::convergence_factor {}
```

was CONIT, in DLS001 common block.

trick_units(-)

Definition at line 296 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), integrator_corrector_iteration(), and integrator_reset_method_coeffs().

8.24.6.5 convergence_jacobian_flag

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::convergence_jacobian_flag {} [protected]
```

Was ICF, in DLS001 common block.

0: Solution converged 1: Convergence failed; Jacobian is not current. 2: Convergence failed; Jacobian is current or not needed.trick_units(-)

Definition at line 568 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), integrator_corrector_failed_part1(), and integrator_corrector_failed_part2().

8.24.6.6 convergence_rate

```
double jeod::LsodeFirstOrderODEIntegrator::convergence_rate {}
```

was CRATE, in DLS001 common block.

trick_units(—)

Definition at line 300 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), integrator_corrector_iteration(), and jacobian_prep_wrap_up().

8.24.6.7 cycle_target_time

```
double jeod::LsodeFirstOrderODEIntegrator::cycle_target_time {}
```

Was TOUT.

The overall integration target time, reset on each externally-commanded cycle.trick_units(—)

Definition at line 288 of file lsode_first_order_ode_integrator.hh.

Referenced by integrate(), interpolate_y(), manager_check_stop_conditions(), manager_initialize_calculation_↵
part2(), manager_integration_loop_part3(), process_entry_point_cycle_start(), and reset_integrator().

8.24.6.8 data_prepj

```
LsodeDataJacobianPrep jeod::LsodeFirstOrderODEIntegrator::data_prepj [protected]
```

data used exclusively for the DPREPJ method.

trick_units(—)

Definition at line 388 of file lsode_first_order_ode_integrator.hh.

Referenced by jacobian_prep_init(), jacobian_prep_loop(), and jacobian_prep_wrap_up().

8.24.6.9 data_stode

```
LsodeDataStode jeod::LsodeFirstOrderODEIntegrator::data_stode [protected]
```

data used exclusively for the DSTODE method.

trick_units(—)

Definition at line 396 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_compute_new_order(), integrator_compute_new_order_check_step_error(), integrator_↵
_compute_new_order_prep(), integrator_core(), integrator_corrector_converged(), integrator_corrector_failed_↵
_part2(), integrator_error_test_failed(), integrator_fail_reset_order_1_part1(), integrator_fail_reset_order_1_↵
part2(), integrator_reset_method_coeffs(), integrator_reset_yh(), integrator_set_new_order(), and integrator_test_↵
_stepsize_change().

8.24.6.10 epsilon

```
double jeod::LsodeFirstOrderODEIntegrator::epsilon {1.0E-12} [protected]
```

was UROUND, in DLS001 common block.

Small number.trick_units(-)

Definition at line 603 of file lsode_first_order_ode_integrator.hh.

Referenced by calculate_epsilon(), interpolate_y(), jacobian_prep_init(), jacobian_prep_wrap_up(), manager_initialize_calculation_part2(), and manager_integration_loop_part2().

8.24.6.11 first_pass

```
bool jeod::LsodeFirstOrderODEIntegrator::first_pass {true} [protected]
```

was ISTATE.

true: was IASTATE = 1: first call for the problem, require initialization. false: was IASTATE = 2: subsequent call, no change to input parameters. not covered: IASTATE = 3: subsequent call, input parameters have changed.trick_units(-)

Definition at line 446 of file lsode_first_order_ode_integrator.hh.

Referenced by process_entry_point_cycle_start(), reset_integrator(), and ~LsodeFirstOrderODEIntegrator().

8.24.6.12 initialized

```
bool jeod::LsodeFirstOrderODEIntegrator::initialized {} [protected]
```

was INIT, in DLS001 common block.

Flag representing whether the problem has been initialized.trick_units(-)

Definition at line 456 of file lsode_first_order_ode_integrator.hh.

Referenced by manager_integration_loop_part3(), and process_entry_point_cycle_start().

8.24.6.13 internal_state

```
int jeod::LsodeFirstOrderODEIntegrator::internal_state {} [protected]
```

Was JSTART, in DLS001 common block.

0: First step for problem 1: Continue normal calculation -1: Next step has new values of step-size, order, or methods. -2: Undocumented.trick_units(-)

Definition at line 464 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), integrator_terminate(), and manager_initialize_calculation_part1().

8.24.6.14 iter_delta

```
double jeod::LsodeFirstOrderODEIntegrator::iter_delta {} [protected]
```

Was DEL, in DSTODE, local variable.

RMS value of {y dot error_weight_data}trick_units(-)

Definition at line 430 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_corrector_converged(), integrator_corrector_iteration(), and integrator_reset_iteration_↵ loop_part2().

8.24.6.15 iteration_count

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::iteration_count {} [protected]
```

Definition at line 597 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_corrector_converged(), integrator_corrector_iteration(), and integrator_reset_iteration_↵ loop_part1().

8.24.6.16 iteration_matrix_singular

```
bool jeod::LsodeFirstOrderODEIntegrator::iteration_matrix_singular {} [protected]
```

Was IERPJ, in DLS001 common block.

false: Iteration matrix was successfully LU-decomposed (iteration-method = 1,2,4,5) or inverted (iteration-method = 3). true: Matrix is singular.trick_units(-)

Definition at line 589 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), jacobian_prep_init(), and jacobian_prep_wrap_up().

8.24.6.17 jacobian_current

```
bool jeod::LsodeFirstOrderODEIntegrator::jacobian_current {} [protected]
```

Was JCUR, in DLS001 common block Is the jacobian current.

trick_units(-)

Definition at line 551 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), integrator_corrector_converged(), integrator_corrector_failed_part1(), and jacobian_prep_init().

8.24.6.18 max_history_size

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::max_history_size {12}
```

Was LMAX, in DLS001 common block Maximum allowable number of histories.

trick_units(-)

Definition at line 328 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_compute_new_order_prep(), integrator_core(), and integrator_corrector_converged().

8.24.6.19 max_order_internal

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::max_order_internal {} [protected]
```

Was MAXORD.

Populated from IWORK[5] Maximum order allowable.trick_units(-)

Definition at line 485 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), and update_control_data().

8.24.6.20 max_rel_change_without_jacobian

```
double jeod::LsodeFirstOrderODEIntegrator::max_rel_change_without_jacobian {0.3}
```

was CCMAX, in DLS001 common block.

Max relative change to (step_size * method_coeff_first) before Jacobian matrix is updated. see also rel_change↵_since_jacobian. // This value was set in DLSODE (line 1385) to 0.3 without any explanation. It can be changed, but takes 0.3 as default.trick_units(-)

Definition at line 364 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_predict().

8.24.6.21 max_step_increase_ratio

```
double jeod::LsodeFirstOrderODEIntegrator::max_step_increase_ratio {1.0}
```

was RMAX, in DLS001 common block.

Max ratio by which step size may be increased.trick_units(-)

Definition at line 354 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), integrator_corrector_failed_part2(), integrator_error_test_failed(), and integrator↵_reset_yh().

8.24.6.22 max_step_size_inv

```
double jeod::LsodeFirstOrderODEIntegrator::max_step_size_inv {} [protected]
```

was HMXI.

Inverse of maximum absolute step size allowable. Default to 0.0 (i.e. there is no upper bound), calculated from max_step_size if max_step_size is user-specified.trick_units(-)

Definition at line 544 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_reset_yh(), manager_initialize_calculation_part2(), and update_control_data().

8.24.6.23 method_coeff_first

```
double jeod::LsodeFirstOrderODEIntegrator::method_coeff_first {} [protected]
```

was EL0, in DLS001 common block.

method coefficient l_0 for current method and order.trick_units(-)

Definition at line 507 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), integrator_reset_method_coeffs(), jacobian_prep_init(), and linear_chord_↔ iteration().

8.24.6.24 method_coeffs_complete

```
double jeod::LsodeFirstOrderODEIntegrator::method_coeffs_complete[13][12] {} [protected]
```

was ELCO, in DLS001 common block.

The array of all of the method coefficients.trick_units(-)

Definition at line 516 of file lsode_first_order_ode_integrator.hh.

Referenced by calculate_integration_coefficients(), integrator_core(), and integrator_reset_method_coeffs().

8.24.6.25 method_coeffs_current

```
double jeod::LsodeFirstOrderODEIntegrator::method_coeffs_current[13] {} [protected]
```

was EL, in DLS001 common block.

trick_units(-)

Definition at line 511 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_compute_new_order(), integrator_core(), integrator_corrector_converged(), integrator_↔ corrector_iteration(), and integrator_reset_method_coeffs().

8.24.6.26 method_order_current

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::method_order_current {}
```

Was NQ, in DLS001 common block.

Method order being tried on this or next step.`trick_units(-)`

Definition at line 340 of file `lsode_first_order_ode_integrator.hh`.

Referenced by `integrator_compute_new_order()`, `integrator_compute_new_order_prep()`, `integrator_core()`, `integrator_corrector_converged()`, `integrator_corrector_failed_part2()`, `integrator_corrector_iteration()`, `integrator_error_test_failed()`, `integrator_fail_reset_order_1_part2()`, `integrator_predict()`, `integrator_reset_method_coeffs()`, `integrator_set_new_order()`, `interpolate_y()`, `manager_initialize_calculation_part2()`, and `reset_integrator()`.

8.24.6.27 modified_iteration_matrix_singular

```
bool jeod::LsodeFirstOrderODEIntegrator::modified_iteration_matrix_singular {} [protected]
```

Was IERSL, in DLS001 common block.

Like `iteration_matrix_singular`, only applied to the iteration matrix that has been modified to account for the new step for iteration-method 3.`trick_units(-)`

Definition at line 596 of file `lsode_first_order_ode_integrator.hh`.

Referenced by `integrator_core()`, `integrator_corrector_iteration()`, and `linear_chord_iteration()`.

8.24.6.28 num_equations

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::num_equations {1}
```

Was NYH, in DLS001 common block.

Number of ODEs to be solved in the current problem. In this implementation, `num_odes = num_equations`.`trick_units(-)`

Definition at line 316 of file `lsode_first_order_ode_integrator.hh`.

Referenced by `integrator_core()`, `integrator_corrector_failed_part2()`, `integrator_error_test_failed()`, `integrator_predict()`, `integrator_reset_method_coeffs()`, and `process_entry_point_cycle_start()`.

8.24.6.29 num_jacobian_evals

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::num_jacobian_evals {} [protected]
```

Was NJE, in DLS001 common block Number of jacobian evaluations so far for the problem.

trick_units(-)

Definition at line 425 of file lsode_first_order_ode_integrator.hh.

Referenced by jacobian_prep_init().

8.24.6.30 num_nordsiek_cols

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::num_nordsiek_cols {1}
```

Was L, in DLS001 common block Number of columns in Nordsiek array.

This appears to be a variable that s equal to the current order of the integrator + 1.trick_units(-)

Definition at line 323 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_compute_new_order(), integrator_compute_new_order_prep(), integrator_core(), integrator_corrector_converged(), integrator_fail_reset_order_1_part2(), integrator_reset_method_coeffs(), integrator_reset_yh(), integrator_set_new_order(), and interpolate_y().

8.24.6.31 num_predictor_elements

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::num_predictor_elements {1}
```

Was NQNYH, in DLS001 common block.

Number of elements of history array that are changed by predictor.trick_units(-)

Definition at line 333 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), and integrator_reset_method_coeffs().

8.24.6.32 num_small_step_warnings

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::num_small_step_warnings {} [protected]
```

Was NHNIL, in DLS001 common block.

Number of small-step encounters fo the problem so far.trick_units(-)

Definition at line 420 of file lsode_first_order_ode_integrator.hh.

Referenced by manager_integration_loop_part2().

8.24.6.33 num_steps_taken

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::num_steps_taken {} [protected]
```

Was NST, in DLS001 common block.

Number of steps taken for this problem.`trick_units(-)`

Definition at line 403 of file `lsode_first_order_ode_integrator.hh`.

Referenced by `integrator_corrector_converged()`, `integrator_predict()`, `jacobian_prep_wrap_up()`, `manager_check_↵_stop_conditions()`, and `manager_integration_loop_part1()`.

8.24.6.34 order_select_para

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::order_select_para {}
```

Was IALTH, in DLS001 common block.

0: Select optimal step size and method order 1: If `prev_success_order < maximum_order`, save vector so that an increase can be considered. `>1`: Perform neither.`trick_units(-)`

Definition at line 308 of file `lsode_first_order_ode_integrator.hh`.

Referenced by `integrator_compute_new_order()`, `integrator_compute_new_order_check_step_error()`, `integrator_↵_core()`, `integrator_corrector_converged()`, `integrator_fail_reset_order_1_part2()`, and `integrator_reset_yh()`.

8.24.6.35 prev_good_step_size

```
double jeod::LsodeFirstOrderODEIntegrator::prev_good_step_size {} [protected]
```

was HU, in DLS001 common block.

The size of the last successful step.`trick_units(-)`

Definition at line 537 of file `lsode_first_order_ode_integrator.hh`.

Referenced by `integrator_corrector_converged()`, `interpolate_y()`, and `reset_integrator()`.

8.24.6.36 prev_integration_method

```
LsodeControlDataInterface::IntegrationMethod jeod::LsodeFirstOrderODEIntegrator::prev_integration←
_method [protected]
```

Initial value:

```
{
    LsodeControlDataInterface::ImplicitAdamsNonStiff}
```

Was MEO, in DLS001 common block Integration method used in previous call (see integration_method).

trick_units(−)

Definition at line 494 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core().

8.24.6.37 prev_iter_delta

```
double jeod::LsodeFirstOrderODEIntegrator::prev_iter_delta {} [protected]
```

Was DELP, in DSTODE, local variable.

Previous value of iter_delta, used for comparison to identify rate at which iteration is converging / identifying divergence of iteration.trick_units(−)

Definition at line 437 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_corrector_iteration(), and integrator_reset_iteration_loop_part2().

8.24.6.38 prev_method_order

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::prev_method_order {} [protected]
```

Was NQU, in DLS001 common block.

Method order used in last successful step.trick_units(−)

Definition at line 500 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_corrector_converged(), and integrator_wrapup().

8.24.6.39 prev_step_size

```
double jeod::LsodeFirstOrderODEIntegrator::prev_step_size {} [protected]
```

was HOLD, in DLS001 common block.

trick_units(-)

Definition at line 532 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), integrator_terminate(), and integrator_test_stepsize_change().

8.24.6.40 prior_num_steps

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::prior_num_steps {} [protected]
```

Was NSLAST, in DLS001 common block.

Number of steps taken for the problem prior to this call to Lsode.trick_units(-)

Definition at line 408 of file lsode_first_order_ode_integrator.hh.

Referenced by manager_check_stop_conditions(), and manager_integration_loop_part1().

8.24.6.41 re_entry_point

```
EntryPoint jeod::LsodeFirstOrderODEIntegrator::re_entry_point {CycleStartFinish} [protected]
```

Indicates where in the integrator to return to following an exit to gether new derivatives.

trick_units(-)

Definition at line 451 of file lsode_first_order_ode_integrator.hh.

Referenced by integrate(), integrator_corrector_iteration(), integrator_fail_reset_order_1_part1(), integrator_reset↵
_iteration_loop_part1(), jacobian_prep_init(), manager_initialize_calculation_part1(), manager_integration_loop↵
part2(), and reset_integrator().

8.24.6.42 rel_change_since_jacobian

```
double jeod::LsodeFirstOrderODEIntegrator::rel_change_since_jacobian {} [protected]
```

was RC, in DLS001 common block.

Relative change to (step_size * method_coeff_first) since last update to Jacobian matrix.trick_units(-)

Definition at line 574 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), integrator_predict(), integrator_reset_method_coeffs(), integrator_reset_yh(), and jacobian_prep_wrap_up().

8.24.6.43 sqrt_epsilon

```
double jeod::LsodeFirstOrderODEIntegrator::sqrt_epsilon {1.0E-6} [protected]
```

NEW.

square root of epsilon.trick_units(-)

Definition at line 608 of file lsode_first_order_ode_integrator.hh.

Referenced by calculate_epsilon(), and manager_initialize_calculation_part1().

8.24.6.44 stage_target_time

```
double jeod::LsodeFirstOrderODEIntegrator::stage_target_time {}
```

was TN, in DLS001 common block.

Value of the independent variable, typically time, to which the integrator has successfully advanced, or to which it will advance in the next step/stage.trick_units(-)

Definition at line 349 of file lsode_first_order_ode_integrator.hh.

Referenced by integrate(), integrator_core(), integrator_corrector_failed_part2(), integrator_error_test_failed(), integrator_predict(), interpolate_y(), manager_check_stop_conditions(), manager_integration_loop_part1(), manager_integration_loop_part2(), manager_integration_loop_part3(), process_entry_point_cycle_start(), and reset_integrator().

8.24.6.45 step_at_last_jacobian_update

```
unsigned int jeod::LsodeFirstOrderODEIntegrator::step_at_last_jacobian_update {} [protected]
```

Was NSLP, in DLS001 common block.

Step number at last Jacobian update.trick_units(-)

Definition at line 561 of file lsode_first_order_ode_integrator.hh.

Referenced by integrator_core(), integrator_predict(), and jacobian_prep_wrap_up().

8.24.6.46 step_error

```
int jeod::LsodeFirstOrderODEIntegrator::step_error {} [protected]
```

Was KFLAG, in DLS001 common block 0: step was successful -1: requested accuracy could not be achieved.

<=-2: repeated convergence failures.trick_units(-)

Definition at line 415 of file lsode_first_order_ode_integrator.hh.

Referenced by `integrate()`, `integrator_compute_new_order_check_step_error()`, `integrator_core()`, `integrator_↵
corrector_converged()`, `integrator_corrector_failed_part2()`, `integrator_error_test_failed()`, `integrator_fail_reset_↵
order_1_part1()`, and `manager_integration_loop_part3()`.

8.24.6.47 step_size

```
double jeod::LsodeFirstOrderODEIntegrator::step_size {} [protected]
```

was H, in DLS001 common block.

Step size used on this step, or to be attempted on next.trick_units(-)

Definition at line 528 of file lsode_first_order_ode_integrator.hh.

Referenced by `integrator_core()`, `integrator_corrector_converged()`, `integrator_corrector_failed_part2()`, `integrator_↵
_corrector_iteration()`, `integrator_error_test_failed()`, `integrator_fail_reset_order_1_part1()`, `integrator_fail_reset_↵
_order_1_part2()`, `integrator_predict()`, `integrator_reset_method_coeffs()`, `integrator_reset_yh()`, `integrator_set_↵
_new_order()`, `integrator_terminate()`, `integrator_test_stepsize_change()`, `interpolate_y()`, `jacobian_prep_init()`,
`jacobian_prep_wrap_up()`, `linear_chord_iteration()`, `manager_check_stop_conditions()`, `manager_initialize_↵
calculation_part2()`, `manager_integration_loop_part2()`, and `manager_integration_loop_part3()`.

8.24.6.48 test_coeffs_complete

```
double jeod::LsodeFirstOrderODEIntegrator::test_coeffs_complete[3][12] {} [protected]
```

was TESCO, in DLS001 common block.

The array of all of the test coefficientstrick_units(-)

Definition at line 521 of file lsode_first_order_ode_integrator.hh.

Referenced by `calculate_integration_coefficients()`, `integrator_compute_new_order()`, `integrator_compute_new_↵
order_prep()`, `integrator_core()`, `integrator_corrector_converged()`, `integrator_corrector_iteration()`, and `integrator_↵
_wrapup()`.

8.24.6.49 update_jacobian

```
bool jeod::LsodeFirstOrderODEIntegrator::update_jacobian {true} [protected]
```

Was IPUP, in DLS001 common block.

Flag to indicate whether it is necessary to update the Jacobian.trick_units(-)

Definition at line 556 of file lsode_first_order_ode_integrator.hh.

Referenced by integrate(), integrator_core(), integrator_corrector_failed_part1(), integrator_corrector_failed_part2(), integrator_fail_reset_order_1_part2(), integrator_predict(), and jacobian_prep_wrap_up().

8.24.6.50 y

```
double* jeod::LsodeFirstOrderODEIntegrator::y {}
```

Was Y.

State vector (zeroth derivative).trick_units(-)

Definition at line 276 of file lsode_first_order_ode_integrator.hh.

Referenced by integrate(), integrator_corrector_iteration(), integrator_fail_reset_order_1_part1(), integrator_reset_iteration_loop_part1(), interpolate_y(), jacobian_prep_init(), jacobian_prep_loop(), linear_chord_iteration(), linear_solver(), and manager_initialize_calculation_part2().

8.24.6.51 y_dot

```
const double* jeod::LsodeFirstOrderODEIntegrator::y_dot {}
```

Was .

State vector (first derivative).trick_units(-)

Definition at line 281 of file lsode_first_order_ode_integrator.hh.

Referenced by integrate(), load_derivatives(), and manager_initialize_calculation_part2().

The documentation for this class was generated from the following files:

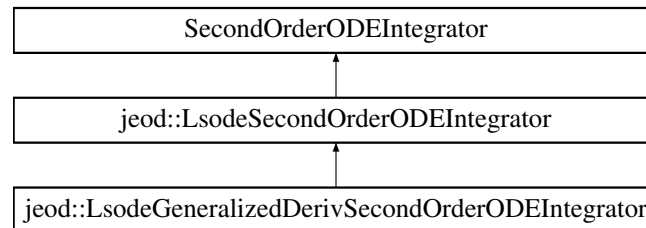
- [lsode_first_order_ode_integrator.hh](#)
- [lsode_first_order_ode_integrator__integrator.cc](#)
- [lsode_first_order_ode_integrator__manager.cc](#)
- [lsode_first_order_ode_integrator__support.cc](#)
- [lsode_first_order_ode_integrator__utility.cc](#)

8.25 jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator Class Reference

JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.

```
#include <lsode_generalized_second_order_ode_integrator.hh>
```

Inheritance diagram for jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator:



Public Member Functions

- [LsodeGeneralizedDerivSecondOrderODEIntegrator](#) ()=default
LsodeGeneralizedDerivSecondOrderODEIntegrator default constructor.
- [LsodeGeneralizedDerivSecondOrderODEIntegrator](#) (const [LsodeGeneralizedDerivSecondOrderODEIntegrator](#) &src)
LsodeGeneralizedDerivSecondOrderODEIntegrator copy constructor.
- [LsodeGeneralizedDerivSecondOrderODEIntegrator](#) (const [LsodeControlDataInterface](#) &data_in, [er7_utils::IntegrationControls](#) &controls, const [er7_utils::GeneralizedPositionDerivativeFunctions](#) &deriv_funs, unsigned int position_size, unsigned int velocity_size)
non-default constructor
- [~LsodeGeneralizedDerivSecondOrderODEIntegrator](#) () override
LsodeGeneralizedDerivSecondOrderODEIntegrator destructor.
- [LsodeGeneralizedDerivSecondOrderODEIntegrator](#) * [create_copy](#) () const override
Clone a LsodeGeneralizedDerivSecondOrderODEIntegrator.
- [er7_utils::IntegratorResult](#) [integrate](#) (double dyn_dt, unsigned int target_stage, const double *accel, double *velocity, double *position) override
Propagate state via Lsode's method.
- [LsodeGeneralizedDerivSecondOrderODEIntegrator](#) & [operator=](#) (const [LsodeGeneralizedDerivSecondOrderODEIntegrator](#) &)=delete

Data Fields

- double * [posdot](#) {}
Stash space for the result of the computation of the derivative of the zeroth-derivative.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__LsodeGeneralizedDerivSecondOrderODEIntegrator](#) ()

Additional Inherited Members

8.25.1 Detailed Description

JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.

Definition at line 90 of file `lsode_generalized_second_order_ode_integrator.hh`.

8.25.2 Constructor & Destructor Documentation

8.25.2.1 `LsodeGeneralizedDerivSecondOrderODEIntegrator()`

```
jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::LsodeGeneralizedDerivSecondOrderODE↔
Integrator ( ) [default]
```

[LsodeGeneralizedDerivSecondOrderODEIntegrator](#) default constructor.

8.25.3 Member Function Documentation

8.25.3.1 `operator=()`

```
LsodeGeneralizedDerivSecondOrderODEIntegrator& jeod::LsodeGeneralizedDerivSecondOrderODE↔
Integrator::operator= (
    const LsodeGeneralizedDerivSecondOrderODEIntegrator & ) [delete]
```

8.25.4 Friends And Related Function Documentation

8.25.4.1 `init_attrjeod__LsodeGeneralizedDerivSecondOrderODEIntegrator`

```
void init_attrjeod__LsodeGeneralizedDerivSecondOrderODEIntegrator ( ) [friend]
```

8.25.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 92 of file `lsode_generalized_second_order_ode_integrator.hh`.

8.25.5 Field Documentation

8.25.5.1 posdot

```
double* jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::posdot {}
```

Stash space for the result of the computation of the derivative of the zeroth-derivative.

Used with the Generalized derivative form, in which the derivative of the zeroth derivative is not equal to the first-derivative.trick_units(-)

Definition at line 149 of file lsode_generalized_second_order_ode_integrator.hh.

Referenced by `integrate()`, `LsodeGeneralizedDerivSecondOrderODEIntegrator()`, and `~LsodeGeneralizedDerivSecondOrderODEIntegrator()`.

The documentation for this class was generated from the following files:

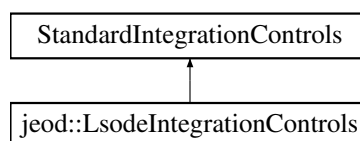
- [lsode_generalized_second_order_ode_integrator.hh](#)
- [lsode_generalized_second_order_ode_integrator.cc](#)

8.26 jeod::LsodeIntegrationControls Class Reference

Contains controls for an LSODE integrator.

```
#include <lsode_integration_controls.hh>
```

Inheritance diagram for jeod::LsodeIntegrationControls:



Public Member Functions

- [LsodeIntegrationControls](#) ()=default
- [LsodeIntegrationControls](#) (unsigned int num_stages)
- [~LsodeIntegrationControls](#) () override=default
- [LsodeIntegrationControls](#) & operator= (const [LsodeIntegrationControls](#) &)=delete
- [LsodeIntegrationControls](#) (const [LsodeIntegrationControls](#) &)=delete
- unsigned int [integrate](#) (double start_time, double sim_dt, er7_utils::TimeInterface &time_interface, er7_utils::IntegratorInterface &integ_interface, er7_utils::BaseIntegrationGroup &integ_group) override

Perform one step of the integration process.
- [LsodeIntegrationControls](#) * [create_copy](#) () const override

Create a copy of 'this' StandardIntegrationControls object.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__LsodeIntegrationControls](#) ()

8.26.1 Detailed Description

Contains controls for an LSODE integrator.

Definition at line 87 of file `lsode_integration_controls.hh`.

8.26.2 Constructor & Destructor Documentation

8.26.2.1 LsodeIntegrationControls() [1/2]

```
jeod::LsodeIntegrationControls::LsodeIntegrationControls ( ) [default]
```

8.26.2.2 ~LsodeIntegrationControls()

```
jeod::LsodeIntegrationControls::~~LsodeIntegrationControls ( ) [override], [default]
```

8.26.2.3 LsodeIntegrationControls() [2/2]

```
jeod::LsodeIntegrationControls::LsodeIntegrationControls (
    const LsodeIntegrationControls & ) [delete]
```

8.26.3 Member Function Documentation

8.26.3.1 operator=()

```
LsodeIntegrationControls& jeod::LsodeIntegrationControls::operator= (
    const LsodeIntegrationControls & ) [delete]
```

8.26.4 Friends And Related Function Documentation

8.26.4.1 init_attrjeod__LsodeIntegrationControls

```
void init_attrjeod__LsodeIntegrationControls ( ) [friend]
```

8.26.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 90 of file lsode_integration_controls.hh.

The documentation for this class was generated from the following files:

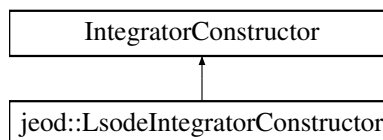
- [lsode_integration_controls.hh](#)
- [lsode_integration_controls.cc](#)

8.27 jeod::LsodeIntegratorConstructor Class Reference

Create state and time integrators that propagate using standard Lsode.

```
#include <lsode_integrator_constructor.hh>
```

Inheritance diagram for jeod::LsodeIntegratorConstructor:



Public Member Functions

- [LsodeIntegratorConstructor](#) ()=default
- [LsodeIntegratorConstructor](#) (const [LsodeIntegratorConstructor](#) &src)
- [LsodeIntegratorConstructor](#) & operator= (const [LsodeIntegratorConstructor](#) &)=delete
- const char * [get_class_name](#) () const override
Return the class name.
- bool [implements](#) (er7_utils::Integration::ODEProblemType problem_type) const override
Lsode currently does not implement a second order generalized step integrator.
- bool [provides](#) (er7_utils::Integration::ODEProblemType problem_type) const override
Lsode currently does not provide a second order generalized step integrator.
- er7_utils::IntegratorConstructor * [create_copy](#) () const override
Create a duplicate of the constructor.
- er7_utils::IntegrationControls * [create_integration_controls](#) () const override
Create an integration controls that guides the Lsode integration process.
- er7_utils::FirstOrderODEIntegrator * [create_first_order_ode_integrator](#) (unsigned int size, er7_utils::↵ IntegrationControls &controls) const override
Create an Lsode state integrator for a first order ODE.

- `er7_utils::SecondOrderODEIntegrator * create_second_order_ode_integrator` (unsigned int size, `er7_utils::IntegrationControls &controls`) const override
Create an Lsode state integrator for a simple second order ODE.
- `er7_utils::SecondOrderODEIntegrator * create_generalized_deriv_second_order_ode_integrator` (unsigned int position_size, unsigned int velocity_size, const `er7_utils::GeneralizedPositionDerivativeFunctions &deriv_funcs`, `er7_utils::IntegrationControls &controls`) const override
Create an Lsode state integrator for a generalized second order ODE where generalized position is advanced with the use of the position derivative function.
- unsigned int `get_transition_table_size` () const override
Lsode does not use a linear transition table.

Static Public Member Functions

- static `er7_utils::IntegratorConstructor * create_constructor` ()
Named constructor; create an [LsodeIntegratorConstructor](#) instance.

Data Fields

- [LsodeControlDataInterface data_interface](#)

Friends

- class [InputProcessor](#)
- void `init_attrjeod__LsodeIntegratorConstructor` ()

8.27.1 Detailed Description

Create state and time integrators that propagate using standard Lsode.

Definition at line 96 of file `lsode_integrator_constructor.hh`.

8.27.2 Constructor & Destructor Documentation

8.27.2.1 LsodeIntegratorConstructor()

```
jeod::LsodeIntegratorConstructor::LsodeIntegratorConstructor ( ) [default]
```

8.27.3 Member Function Documentation

8.27.3.1 get_class_name()

```
const char* jeod::LsodeIntegratorConstructor::get_class_name ( ) const [inline], [override]
```

Return the class name.

Definition at line 121 of file `lsode_integrator_constructor.hh`.

8.27.3.2 get_transition_table_size()

```
unsigned int jeod::LsodeIntegratorConstructor::get_transition_table_size ( ) const [inline],  
[override]
```

Lsode does not use a linear transition table.

Returns

Always returns 0.

Definition at line 201 of file `lsode_integrator_constructor.hh`.

8.27.3.3 implements()

```
bool jeod::LsodeIntegratorConstructor::implements (  
    er7_utils::Integration::ODEProblemType problem_type ) const [inline], [override]
```

Lsode currently does not implement a second order generalized step integrator.

Definition at line 130 of file `lsode_integrator_constructor.hh`.

8.27.3.4 operator=()

```
LsodeIntegratorConstructor& jeod::LsodeIntegratorConstructor::operator= (  
    const LsodeIntegratorConstructor & ) [delete]
```

8.27.3.5 provides()

```
bool jeod::LsodeIntegratorConstructor::provides (  
    er7_utils::Integration::ODEProblemType problem_type ) const [inline], [override]
```

Lsode currently does not provide a second order generalized step integrator.

Definition at line 140 of file `lsode_integrator_constructor.hh`.

8.27.4 Friends And Related Function Documentation

8.27.4.1 `init_attrjeod__LsodeIntegratorConstructor`

```
void init_attrjeod__LsodeIntegratorConstructor ( ) [friend]
```

8.27.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 98 of file `lsode_integrator_constructor.hh`.

8.27.5 Field Documentation

8.27.5.1 `data_interface`

```
LsodeControlDataInterface jeod::LsodeIntegratorConstructor::data_interface
```

Definition at line 206 of file `lsode_integrator_constructor.hh`.

Referenced by `create_first_order_ode_integrator()`, `create_generalized_deriv_second_order_ode_integrator()`, and `create_second_order_ode_integrator()`.

The documentation for this class was generated from the following files:

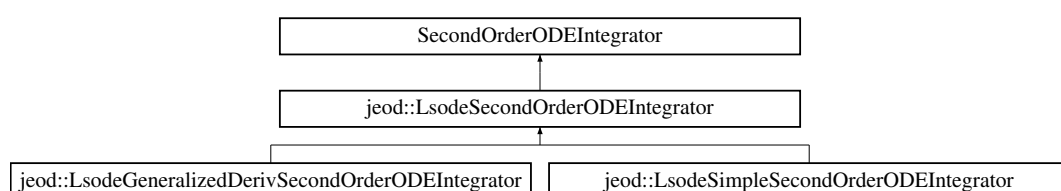
- [lsode_integrator_constructor.hh](#)
- [lsode_integrator_constructor.cc](#)

8.28 `jeod::LsodeSecondOrderODEIntegrator` Class Reference

JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.

```
#include <lsode_second_order_ode_integrator.hh>
```

Inheritance diagram for `jeod::LsodeSecondOrderODEIntegrator`:



Public Member Functions

- [~LsodeSecondOrderODEIntegrator](#) () override
LsodeSecondOrderODEIntegrator destructor.
- [LsodeSecondOrderODEIntegrator](#) & [operator=](#) (const [LsodeSecondOrderODEIntegrator](#) &)=delete
- [LsodeSecondOrderODEIntegrator](#) (const [LsodeSecondOrderODEIntegrator](#) &)=delete
- int [get_re_entry_point](#) ()
Get the integrator's reentry point.

Data Fields

- double * [y](#) {}
State vector (zeroth derivative).
- double * [y_dot](#) {}
State vector (first derivative).
- unsigned int [zeroth_derivative_size](#) {}
- unsigned int [first_derivative_size](#) {}
- [LsodeFirstOrderODEIntegrator](#) [first_order_integrator](#)
- bool [arrays_allocated](#) {}

Protected Member Functions

- [LsodeSecondOrderODEIntegrator](#) ()=default
- [LsodeSecondOrderODEIntegrator](#) (const [LsodeControlDataInterface](#) &data_in, er7_utils::IntegrationControls &controls, unsigned int size)
LsodeSecondOrderODEIntegrator non-default constructor.
- [LsodeSecondOrderODEIntegrator](#) (const [LsodeControlDataInterface](#) &data_in, er7_utils::IntegrationControls &controls, const er7_utils::GeneralizedPositionDerivativeFunctions &deriv_funs, unsigned int position_size, unsigned int velocity_size)
- void [reset_integrator](#) () override
Reset the integrator.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__LsodeSecondOrderODEIntegrator](#) ()

8.28.1 Detailed Description

JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.

Definition at line 90 of file `lsode_second_order_ode_integrator.hh`.

8.28.2 Constructor & Destructor Documentation

8.28.2.1 LsodeSecondOrderODEIntegrator() [1/2]

```
jeod::LsodeSecondOrderODEIntegrator::LsodeSecondOrderODEIntegrator (
    const LsodeSecondOrderODEIntegrator & ) [delete]
```

8.28.2.2 LsodeSecondOrderODEIntegrator() [2/2]

```
jeod::LsodeSecondOrderODEIntegrator::LsodeSecondOrderODEIntegrator ( ) [protected], [default]
```

8.28.3 Member Function Documentation**8.28.3.1 get_re_entry_point()**

```
int jeod::LsodeSecondOrderODEIntegrator::get_re_entry_point ( ) [inline]
```

Get the integrator's reentry point.

Definition at line 107 of file lsode_second_order_ode_integrator.hh.

8.28.3.2 operator=()

```
LsodeSecondOrderODEIntegrator& jeod::LsodeSecondOrderODEIntegrator::operator= (
    const LsodeSecondOrderODEIntegrator & ) [delete]
```

8.28.3.3 reset_integrator()

```
void jeod::LsodeSecondOrderODEIntegrator::reset_integrator ( ) [inline], [override], [protected]
```

Reset the integrator.

Definition at line 134 of file lsode_second_order_ode_integrator.hh.

8.28.4 Friends And Related Function Documentation

8.28.4.1 init_attrjeod__LsodeSecondOrderODEIntegrator

```
void init_attrjeod__LsodeSecondOrderODEIntegrator ( ) [friend]
```

8.28.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 92 of file lsode_second_order_ode_integrator.hh.

8.28.5 Field Documentation

8.28.5.1 arrays_allocated

```
bool jeod::LsodeSecondOrderODEIntegrator::arrays_allocated {}
```

Definition at line 158 of file lsode_second_order_ode_integrator.hh.

Referenced by LsodeSecondOrderODEIntegrator(), and ~LsodeSecondOrderODEIntegrator().

8.28.5.2 first_derivative_size

```
unsigned int jeod::LsodeSecondOrderODEIntegrator::first_derivative_size {}
```

Definition at line 154 of file lsode_second_order_ode_integrator.hh.

Referenced by jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::integrate().

8.28.5.3 first_order_integrator

```
LsodeFirstOrderODEIntegrator jeod::LsodeSecondOrderODEIntegrator::first_order_integrator
```

Definition at line 156 of file lsode_second_order_ode_integrator.hh.

Referenced by jeod::LsodeSimpleSecondOrderODEIntegrator::integrate(), and jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::integrate().

8.28.5.4 y

```
double* jeod::LsodeSecondOrderODEIntegrator::y {}
```

State vector (zeroth derivative).

trick_units(-)

Definition at line 146 of file lsode_second_order_ode_integrator.hh.

Referenced by `jeod::LsodeSimpleSecondOrderODEIntegrator::integrate()`, `jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::integrate()`, `LsodeSecondOrderODEIntegrator()`, and `~LsodeSecondOrderODEIntegrator()`.

8.28.5.5 y_dot

```
double* jeod::LsodeSecondOrderODEIntegrator::y_dot {}
```

State vector (first derivative).

trick_units(-)

Definition at line 151 of file lsode_second_order_ode_integrator.hh.

Referenced by `jeod::LsodeSimpleSecondOrderODEIntegrator::integrate()`, `jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::integrate()`, `LsodeSecondOrderODEIntegrator()`, and `~LsodeSecondOrderODEIntegrator()`.

8.28.5.6 zeroth_derivative_size

```
unsigned int jeod::LsodeSecondOrderODEIntegrator::zeroth_derivative_size {}
```

Definition at line 153 of file lsode_second_order_ode_integrator.hh.

Referenced by `jeod::LsodeSimpleSecondOrderODEIntegrator::integrate()`, and `jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator::integrate()`.

The documentation for this class was generated from the following files:

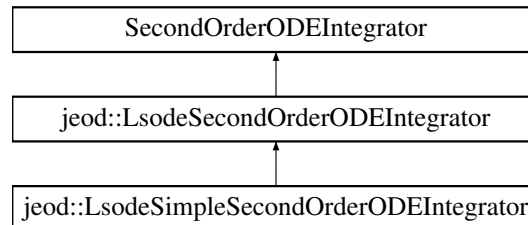
- [lsode_second_order_ode_integrator.hh](#)
- [lsode_second_order_ode_integrator.cc](#)

8.29 jeod::LsodeSimpleSecondOrderODEIntegrator Class Reference

JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.

```
#include <lsode_simple_second_order_ode_integrator.hh>
```

Inheritance diagram for jeod::LsodeSimpleSecondOrderODEIntegrator:



Public Member Functions

- [LsodeSimpleSecondOrderODEIntegrator](#) ()=default
- [~LsodeSimpleSecondOrderODEIntegrator](#) () override=default
- [LsodeSimpleSecondOrderODEIntegrator](#) & [operator=](#) (const [LsodeSimpleSecondOrderODEIntegrator](#) &)=delete
- [LsodeSimpleSecondOrderODEIntegrator](#) (const [LsodeSimpleSecondOrderODEIntegrator](#) &)=delete
- [LsodeSimpleSecondOrderODEIntegrator](#) (const [LsodeControlDataInterface](#) &data_in, [er7_utils::↔](#) IntegrationControls &controls, unsigned int size)
[LsodeSimpleSecondOrderODEIntegrator](#) non-default constructor.
- [LsodeSimpleSecondOrderODEIntegrator](#) * [create_copy](#) () const override
- [er7_utils::IntegratorResult](#) [integrate](#) (double dyn_dt, unsigned int target_stage, const double *accel, double *velocity, double *position) override
Propagate state via Lsode's method.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__LsodeSimpleSecondOrderODEIntegrator](#) ()

Additional Inherited Members

8.29.1 Detailed Description

JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.

Definition at line 93 of file `lsode_simple_second_order_ode_integrator.hh`.

8.29.2 Constructor & Destructor Documentation

8.29.2.1 LsodeSimpleSecondOrderODEIntegrator() [1/2]

```
jeod::LsodeSimpleSecondOrderODEIntegrator::LsodeSimpleSecondOrderODEIntegrator ( ) [default]
```

8.29.2.2 ~LsodeSimpleSecondOrderODEIntegrator()

```
jeod::LsodeSimpleSecondOrderODEIntegrator::~~LsodeSimpleSecondOrderODEIntegrator ( ) [override],  
[default]
```

8.29.2.3 LsodeSimpleSecondOrderODEIntegrator() [2/2]

```
jeod::LsodeSimpleSecondOrderODEIntegrator::LsodeSimpleSecondOrderODEIntegrator (   
    const LsodeSimpleSecondOrderODEIntegrator & ) [delete]
```

8.29.3 Member Function Documentation**8.29.3.1 operator=()**

```
LsodeSimpleSecondOrderODEIntegrator& jeod::LsodeSimpleSecondOrderODEIntegrator::operator= (   
    const LsodeSimpleSecondOrderODEIntegrator & ) [delete]
```

8.29.4 Friends And Related Function Documentation**8.29.4.1 init_attrjeod__LsodeSimpleSecondOrderODEIntegrator**

```
void init_attrjeod__LsodeSimpleSecondOrderODEIntegrator ( ) [friend]
```

8.29.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 95 of file `lsode_simple_second_order_ode_integrator.hh`.

The documentation for this class was generated from the following files:

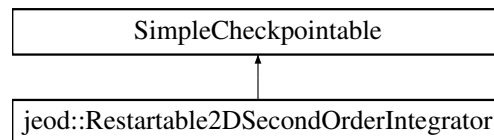
- [lsode_simple_second_order_ode_integrator.hh](#)
- [lsode_simple_second_order_ode_integrator.cc](#)

8.30 jeod::Restartable2DSecondOrderIntegrator Class Reference

Integrates a second order ODE in two dimensional space, $d^2x/dt^2 = a(x,t)$, where x is a two-vector.

```
#include <restartable_2d_second_order_integrator.hh>
```

Inheritance diagram for jeod::Restartable2DSecondOrderIntegrator:



Public Member Functions

- [Restartable2DSecondOrderIntegrator](#) ()
Default constructor.
- [~Restartable2DSecondOrderIntegrator](#) () override
Destructor.
- void [create_integrator](#) (const er7_utils::IntegratorConstructor &generator, er7_utils::IntegrationControls &controls)
Create the integrator to be managed.
- void [destroy_integrator](#) ()
Destroy the integrator.
- er7_utils::IntegratorResult [integrate](#) (double dyn_dt, unsigned int target_stage, const double *accel, double *velocity, double *position)
Propagate state to the specified stage of the integration process for an overall integration time interval of dyn_dt.
- void [reset_integrator](#) ()
Tell the integrator to reset itself.
- void [simple_restore](#) () override
Restore the integrator on restart.
- [Restartable2DSecondOrderIntegrator](#) (const [Restartable2DSecondOrderIntegrator](#) &)=delete
- [Restartable2DSecondOrderIntegrator](#) & operator= (const [Restartable2DSecondOrderIntegrator](#) &)=delete

Private Attributes

- er7_utils::SecondOrderODEIntegrator * [integrator](#) {}
The pointer to the object that performs integration.
- [RestartableSimpleSecondOrderODEIntegrator](#)< 2 > [integrator_manager](#)
The object that creates and manages the integrator object.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__Restartable2DSecondOrderIntegrator](#) ()

8.30.1 Detailed Description

Integrates a second order ODE in two dimensional space, $d^2x/dt^2 = a(x,t)$, where x is a two-vector.

Definition at line 82 of file `restartable_2d_second_order_integrator.hh`.

8.30.2 Constructor & Destructor Documentation

8.30.2.1 Restartable2DSecondOrderIntegrator() [1/2]

```
jeod::Restartable2DSecondOrderIntegrator::Restartable2DSecondOrderIntegrator ( ) [inline]
```

Default constructor.

Definition at line 88 of file `restartable_2d_second_order_integrator.hh`.

8.30.2.2 ~Restartable2DSecondOrderIntegrator()

```
jeod::Restartable2DSecondOrderIntegrator::~~Restartable2DSecondOrderIntegrator ( ) [inline],  
[override]
```

Destructor.

Definition at line 97 of file `restartable_2d_second_order_integrator.hh`.

8.30.2.3 Restartable2DSecondOrderIntegrator() [2/2]

```
jeod::Restartable2DSecondOrderIntegrator::Restartable2DSecondOrderIntegrator (   
    const Restartable2DSecondOrderIntegrator & ) [delete]
```

8.30.3 Member Function Documentation

8.30.3.1 create_integrator()

```
void jeod::Restartable2DSecondOrderIntegrator::create_integrator (   
    const er7_utils::IntegratorConstructor & generator,   
    er7_utils::IntegrationControls & controls ) [inline]
```

Create the integrator to be managed.

Parameters

in	<i>generator</i>	Generator used to create the integrator.
in, out	<i>controls</i>	Controls to be passed to the generator.

Definition at line 107 of file `restartable_2d_second_order_integrator.hh`.

8.30.3.2 `destroy_integrator()`

```
void jeod::Restartable2DSecondOrderIntegrator::destroy_integrator ( ) [inline]
```

Destroy the integrator.

Definition at line 116 of file `restartable_2d_second_order_integrator.hh`.

8.30.3.3 `integrate()`

```
er7_utils::IntegratorResult jeod::Restartable2DSecondOrderIntegrator::integrate (
    double dyn_dt,
    unsigned int target_stage,
    const double * accel,
    double * velocity,
    double * position ) [inline]
```

Propagate state to the specified stage of the integration process for an overall integration time interval of `dyn_dt`.

Note that this is a pass-through to the encapsulated integrator object. See `er7_utils::SecondOrderODEIntegrator::integrate` for details.

Parameters

in	<i>dyn_dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in	<i>accel</i>	Time derivative of the generalized velocity.
in, out	<i>velocity</i>	Generalized velocity vector.
in, out	<i>position</i>	Generalized position vector.

Returns

The status (time advance, pass/fail status) of the integration.

Definition at line 138 of file `restartable_2d_second_order_integrator.hh`.

8.30.3.4 operator=()

```
Restartable2DSecondOrderIntegrator& jeod::Restartable2DSecondOrderIntegrator::operator= (
    const Restartable2DSecondOrderIntegrator & ) [delete]
```

8.30.3.5 reset_integrator()

```
void jeod::Restartable2DSecondOrderIntegrator::reset_integrator ( ) [inline]
```

Tell the integrator to reset itself.

This should be called when the time step or time direction changes or upon a discrete change in state such docking/undocking. Such events invalidate saved state (if any). The integrator needs to restart from scratch when such events occur.

Definition at line 153 of file restartable_2d_second_order_integrator.hh.

8.30.3.6 simple_restore()

```
void jeod::Restartable2DSecondOrderIntegrator::simple_restore ( ) [inline], [override]
```

Restore the integrator on restart.

Definition at line 161 of file restartable_2d_second_order_integrator.hh.

8.30.4 Friends And Related Function Documentation

8.30.4.1 init_attrjeod__Restartable2DSecondOrderIntegrator

```
void init_attrjeod__Restartable2DSecondOrderIntegrator ( ) [friend]
```

8.30.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 84 of file restartable_2d_second_order_integrator.hh.

8.30.5 Field Documentation

8.30.5.1 integrator

```
er7_utils::SecondOrderODEIntegrator* jeod::Restartable2DSecondOrderIntegrator::integrator {}
[private]
```

The pointer to the object that performs integration.

This object is created managed by the integrator manager.trick_units(-)

Definition at line 176 of file restartable_2d_second_order_integrator.hh.

8.30.5.2 integrator_manager

```
RestartableSimpleSecondOrderODEIntegrator<2> jeod::Restartable2DSecondOrderIntegrator::integrator↔
_manager [private]
```

The object that creates and manages the integrator object.

```
trick_io(**)
```

Definition at line 181 of file restartable_2d_second_order_integrator.hh.

The documentation for this class was generated from the following file:

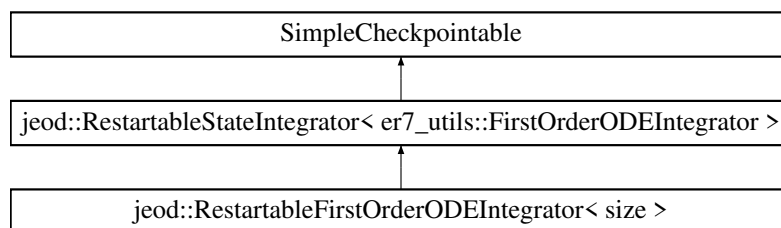
- [restartable_2d_second_order_integrator.hh](#)

8.31 jeod::RestartableFirstOrderODEIntegrator< size > Class Template Reference

A [RestartableFirstOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages an [er7_utils::FirstOrderODEIntegrator](#).

```
#include <restartable_state_integrator_templates.hh>
```

Inheritance diagram for jeod::RestartableFirstOrderODEIntegrator< size >:



Public Member Functions

- [RestartableFirstOrderODEIntegrator](#) ()=default
- [~RestartableFirstOrderODEIntegrator](#) () override=default
- [RestartableFirstOrderODEIntegrator](#) (const [RestartableFirstOrderODEIntegrator](#) &)=delete
- [RestartableFirstOrderODEIntegrator](#) & operator= (const [RestartableFirstOrderODEIntegrator](#) &)=delete
- [RestartableFirstOrderODEIntegrator](#) (er7_utils::FirstOrderODEIntegrator *&integ_ref)

Non-default constructor.

Private Member Functions

- `er7_utils::FirstOrderODEIntegrator * create_integrator_internal` (const `er7_utils::IntegratorConstructor` &generator, `er7_utils::IntegrationControls` &controls) override

Create the integrator to be managed.

Additional Inherited Members

8.31.1 Detailed Description

```
template<unsigned int size>
class jeod::RestartableFirstOrderODEIntegrator< size >
```

A [RestartableFirstOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages an `er7_utils::FirstOrderODEIntegrator`.

Template Parameters

<code>size</code>	Dimensionality of the state vector.
-------------------	-------------------------------------

Definition at line 306 of file `restartable_state_integrator_templates.hh`.

8.31.2 Constructor & Destructor Documentation

8.31.2.1 RestartableFirstOrderODEIntegrator() [1/3]

```
template<unsigned int size>
jeod::RestartableFirstOrderODEIntegrator< size >::RestartableFirstOrderODEIntegrator ( )
[default]
```

8.31.2.2 ~RestartableFirstOrderODEIntegrator()

```
template<unsigned int size>
jeod::RestartableFirstOrderODEIntegrator< size >::~~RestartableFirstOrderODEIntegrator ( )
[override], [default]
```

8.31.2.3 RestartableFirstOrderODEIntegrator() [2/3]

```
template<unsigned int size>
jeod::RestartableFirstOrderODEIntegrator< size >::RestartableFirstOrderODEIntegrator (
    const RestartableFirstOrderODEIntegrator< size > & ) [delete]
```

8.31.2.4 RestartableFirstOrderODEIntegrator() [3/3]

```
template<unsigned int size>
jeod::RestartableFirstOrderODEIntegrator< size >::RestartableFirstOrderODEIntegrator (
    er7_utils::FirstOrderODEIntegrator *& integ_ref ) [inline], [explicit]
```

Non-default constructor.

Parameters

in, out	<i>integ_ref</i>	Reference to the pointer to the integrator that is to be managed.
---------	------------------	---

Definition at line 319 of file restartable_state_integrator_templates.hh.

8.31.3 Member Function Documentation

8.31.3.1 create_integrator_internal()

```
template<unsigned int size>
er7_utils::FirstOrderODEIntegrator* jeod::RestartableFirstOrderODEIntegrator< size >::create↵
_integrator_internal (
    const er7_utils::IntegratorConstructor & generator,
    er7_utils::IntegrationControls & controls ) [inline], [override], [private],
[virtual]
```

Create the integrator to be managed.

Parameters

in	<i>generator</i>	Integrator constructor used to create the integrator.
in, out	<i>controls</i>	Integration controls to be passed to the generator.

Implements [jeod::RestartableStateIntegrator< er7_utils::FirstOrderODEIntegrator >](#).

Definition at line 332 of file restartable_state_integrator_templates.hh.

8.31.3.2 operator=()

```
template<unsigned int size>
RestartableFirstOrderODEIntegrator& jeod::RestartableFirstOrderODEIntegrator< size >::operator=
(
    const RestartableFirstOrderODEIntegrator< size > & ) [delete]
```

The documentation for this class was generated from the following file:

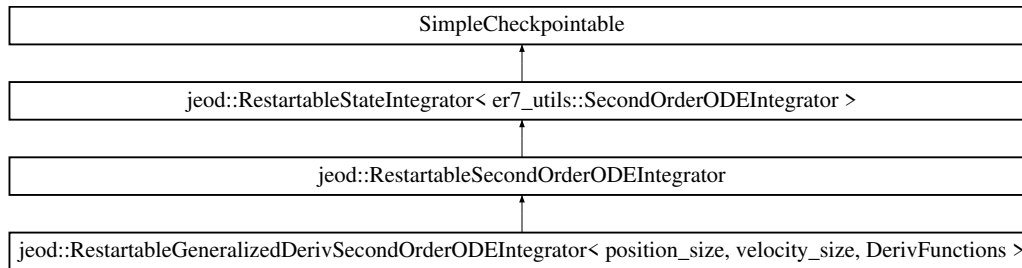
- [restartable_state_integrator_templates.hh](#)

8.32 jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions > Class Template Reference

A [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a generalized second order ODE problem, one in which the time derivative of the generalized position is some function of the generalized position and the generalized velocity.

```
#include <restartable_state_integrator_templates.hh>
```

Inheritance diagram for jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions >:



Public Member Functions

- [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) ()=default
- [~RestartableGeneralizedDerivSecondOrderODEIntegrator](#) () override=default
- [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) (const [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) &)=delete
- [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) & operator= (const [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) &)=delete
- [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) (er7_utils::SecondOrderODEIntegrator *&integ_ref)
 Non-default constructor.

Private Member Functions

- er7_utils::SecondOrderODEIntegrator * [create_integrator_internal](#) (const er7_utils::IntegratorConstructor &generator, er7_utils::IntegrationControls &controls) override
 Create the integrator to be managed.
- void [simple_restore_internal](#) (er7_utils::SecondOrderODEIntegrator *integrator_ptr) override
 Perform technique-specific restart actions.

Additional Inherited Members

8.32.1 Detailed Description

```
template<unsigned int position_size, unsigned int velocity_size, typename DerivFunctions>
class jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions >
```

A [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a generalized second order ODE problem, one in which the time derivative of the generalized position is some function of the generalized position and the generalized velocity.

Template Parameters

<i>position_size</i>	The dimensionality of the generalized position.
<i>velocity_size</i>	The dimensionality of the generalized velocity.
<i>DerivFunctions</i>	Class that derives from the class er7_utils::GeneralizedPositionDerivativeFunctions.

Definition at line 418 of file restartable_state_integrator_templates.hh.

8.32.2 Constructor & Destructor Documentation

8.32.2.1 RestartableGeneralizedDerivSecondOrderODEIntegrator() [1/3]

```
template<unsigned int position_size, unsigned int velocity_size, typename DerivFunctions>
jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions >::RestartableGeneralizedDerivSecondOrderODEIntegrator ( ) [default]
```

8.32.2.2 ~RestartableGeneralizedDerivSecondOrderODEIntegrator()

```
template<unsigned int position_size, unsigned int velocity_size, typename DerivFunctions>
jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions >::~~RestartableGeneralizedDerivSecondOrderODEIntegrator ( ) [override], [default]
```

8.32.2.3 RestartableGeneralizedDerivSecondOrderODEIntegrator() [2/3]

```
template<unsigned int position_size, unsigned int velocity_size, typename DerivFunctions>
jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions >::RestartableGeneralizedDerivSecondOrderODEIntegrator (
    const RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions > & ) [delete]
```

8.32.2.4 RestartableGeneralizedDerivSecondOrderODEIntegrator() [3/3]

```
template<unsigned int position_size, unsigned int velocity_size, typename DerivFunctions>
jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions >::RestartableGeneralizedDerivSecondOrderODEIntegrator (
    er7_utils::SecondOrderODEIntegrator *& integ_ref ) [inline], [explicit]
```

Non-default constructor.

Parameters

<i>in, out</i>	<i>integ_ref</i>	Reference to the pointer to the integrator that is to be managed.
----------------	------------------	---

Definition at line 433 of file `restartable_state_integrator_templates.hh`.

8.32.3 Member Function Documentation

8.32.3.1 `create_integrator_internal()`

```
template<unsigned int position_size, unsigned int velocity_size, typename DerivFunctions>
er7_utils::SecondOrderODEIntegrator* jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator<
position_size, velocity_size, DerivFunctions >::create_integrator_internal (
    const er7_utils::IntegratorConstructor & generator,
    er7_utils::IntegrationControls & controls ) [inline], [override], [private],
[virtual]
```

Create the integrator to be managed.

Parameters

<i>in</i>	<i>generator</i>	Integrator constructor used to create the integrator.
<i>in, out</i>	<i>controls</i>	Integration controls to be passed to the generator.

Implements [jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >](#).

Definition at line 446 of file `restartable_state_integrator_templates.hh`.

8.32.3.2 `operator=()`

```
template<unsigned int position_size, unsigned int velocity_size, typename DerivFunctions>
RestartableGeneralizedDerivSecondOrderODEIntegrator& jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator<
position_size, velocity_size, DerivFunctions >::operator= (
    const RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity↵
_size, DerivFunctions > & ) [delete]
```

8.32.3.3 `simple_restore_internal()`

```
template<unsigned int position_size, unsigned int velocity_size, typename DerivFunctions>
void jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size,
DerivFunctions >::simple_restore_internal (
```

```
er7_utils::SecondOrderODEIntegrator * integrator_ptr ) [inline], [override],  
[private], [virtual]
```

Perform technique-specific restart actions.

The generalized second order ODE integrators need the pointer to the derivative function to be restored.

Parameters

<code>in, out</code>	<code>integrator_ptr</code>	The base class's integrator data member
----------------------	-----------------------------	---

Reimplemented from [jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >](#).

Definition at line 461 of file `restartable_state_integrator_templates.hh`.

The documentation for this class was generated from the following file:

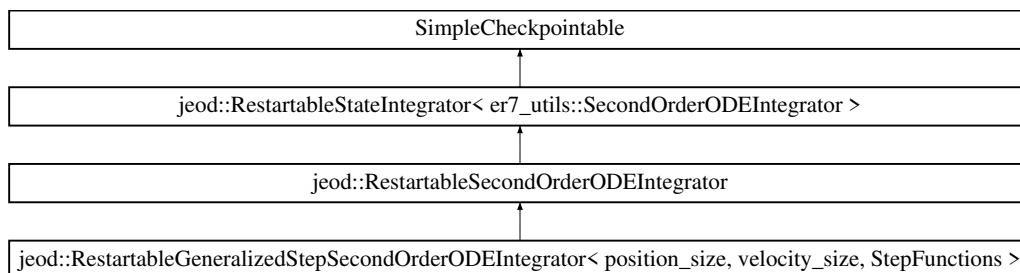
- [restartable_state_integrator_templates.hh](#)

8.33 jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, StepFunctions > Class Template Reference

A [RestartableGeneralizedStepSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a generalized second order ODE problem, one in which the time derivative of the generalized position is some function of the generalized position and the generalized velocity.

```
#include <restartable_state_integrator_templates.hh>
```

Inheritance diagram for `jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, StepFunctions >`:



Public Member Functions

- [RestartableGeneralizedStepSecondOrderODEIntegrator](#) ()=default
- [~RestartableGeneralizedStepSecondOrderODEIntegrator](#) () override=default
- [RestartableGeneralizedStepSecondOrderODEIntegrator](#) (const [RestartableGeneralizedStepSecondOrderODEIntegrator](#) &)=delete
- [RestartableGeneralizedStepSecondOrderODEIntegrator](#) & operator= (const [RestartableGeneralizedStepSecondOrderODEIntegrator](#) &)=delete
- [RestartableGeneralizedStepSecondOrderODEIntegrator](#) (er7_utils::SecondOrderODEIntegrator *&integ_ref)

Non-default constructor.

Private Member Functions

- er7_utils::SecondOrderODEIntegrator * [create_integrator_internal](#) (const er7_utils::IntegratorConstructor &generator, er7_utils::IntegrationControls &controls) override
 - void [simple_restore_internal](#) (er7_utils::SecondOrderODEIntegrator *integrator_ptr) override
- Create the integrator to be managed.*
- Perform technique-specific restart actions.*

Additional Inherited Members

8.33.1 Detailed Description

```
template<unsigned int position_size, unsigned int velocity_size, typename StepFunctions>
class jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, StepFunctions >
```

A [RestartableGeneralizedStepSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a generalized second order ODE problem, one in which the time derivative of the generalized position is some function of the generalized position and the generalized velocity.

Template Parameters

<i>position_size</i>	The dimensionality of the generalized position.
<i>velocity_size</i>	The dimensionality of the generalized velocity.
<i>StepFunctions</i>	Class that derives from <code>er7_utils::GeneralizedPositionStepFunctions</code> .

Definition at line 482 of file `restartable_state_integrator_templates.hh`.

8.33.2 Constructor & Destructor Documentation

8.33.2.1 RestartableGeneralizedStepSecondOrderODEIntegrator() [1/3]

```
template<unsigned int position_size, unsigned int velocity_size, typename StepFunctions>
jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, Step↵
Functions >::RestartableGeneralizedStepSecondOrderODEIntegrator ( ) [default]
```

8.33.2.2 ~RestartableGeneralizedStepSecondOrderODEIntegrator()

```
template<unsigned int position_size, unsigned int velocity_size, typename StepFunctions>
jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, Step↵
Functions >::~~RestartableGeneralizedStepSecondOrderODEIntegrator ( ) [override], [default]
```

8.33.2.3 RestartableGeneralizedStepSecondOrderODEIntegrator() [2/3]

```
template<unsigned int position_size, unsigned int velocity_size, typename StepFunctions>
jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, Step↵
Functions >::RestartableGeneralizedStepSecondOrderODEIntegrator (
    const RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity↵
_size, StepFunctions > & ) [delete]
```

8.33.2.4 RestartableGeneralizedStepSecondOrderODEIntegrator() [3/3]

```
template<unsigned int position_size, unsigned int velocity_size, typename StepFunctions>
jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, StepFunctions >::RestartableGeneralizedStepSecondOrderODEIntegrator (
    er7_utils::SecondOrderODEIntegrator *& integ_ref ) [inline], [explicit]
```

Non-default constructor.

Parameters

in, out	<i>integ_ref</i>	Reference to the pointer to the integrator that is to be managed.
---------	------------------	---

Definition at line 497 of file restartable_state_integrator_templates.hh.

8.33.3 Member Function Documentation

8.33.3.1 create_integrator_internal()

```
template<unsigned int position_size, unsigned int velocity_size, typename StepFunctions>
er7_utils::SecondOrderODEIntegrator* jeod::RestartableGeneralizedStepSecondOrderODEIntegrator<
position_size, velocity_size, StepFunctions >::create_integrator_internal (
    const er7_utils::IntegratorConstructor & generator,
    er7_utils::IntegrationControls & controls ) [inline], [override], [private],
[virtual]
```

Create the integrator to be managed.

Parameters

in	<i>generator</i>	Integrator constructor used to create the integrator.
in, out	<i>controls</i>	Integration controls to be passed to the generator.

Implements [jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >](#).

Definition at line 510 of file restartable_state_integrator_templates.hh.

8.33.3.2 operator=()

```
template<unsigned int position_size, unsigned int velocity_size, typename StepFunctions>
RestartableGeneralizedStepSecondOrderODEIntegrator& jeod::RestartableGeneralizedStepSecondOrderODEIntegrator<
position_size, velocity_size, StepFunctions >::operator= (
    const RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, StepFunctions > & ) [delete]
```

8.33.3.3 simple_restore_internal()

```
template<unsigned int position_size, unsigned int velocity_size, typename StepFunctions>
void jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size,
StepFunctions >::simple_restore_internal (
    er7_utils::SecondOrderODEIntegrator * integrator_ptr ) [inline], [override],
[private], [virtual]
```

Perform technique-specific restart actions.

The generalized second order ODE integrators need the pointer to the derivative function to be restored.

Parameters

<code>in, out</code>	<code>integrator_ptr</code>	The base class's integrator data member
----------------------	-----------------------------	---

Reimplemented from [jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >](#).

Definition at line 525 of file `restartable_state_integrator_templates.hh`.

The documentation for this class was generated from the following file:

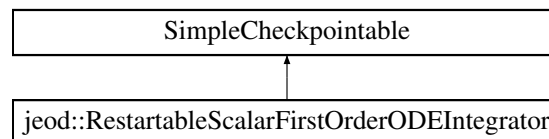
- [restartable_state_integrator_templates.hh](#)

8.34 jeod::RestartableScalarFirstOrderODEIntegrator Class Reference

A [RestartableScalarFirstOrderODEIntegrator](#) integrates a first order ODE, $dx/dt = v(x,t)$, where x is a scalar.

```
#include <restartable_state_integrator.hh>
```

Inheritance diagram for `jeod::RestartableScalarFirstOrderODEIntegrator`:



Public Member Functions

- [RestartableScalarFirstOrderODEIntegrator](#) ()
Default constructor.
- [~RestartableScalarFirstOrderODEIntegrator](#) () override=default
- [RestartableScalarFirstOrderODEIntegrator](#) (const [RestartableScalarFirstOrderODEIntegrator](#) &)=delete
- [RestartableScalarFirstOrderODEIntegrator](#) & operator= (const [RestartableScalarFirstOrderODEIntegrator](#) &)=delete
- void [create_integrator](#) (const er7_utils::IntegratorConstructor &generator, er7_utils::IntegrationControls &controls)
Create the integrator to be managed.

- void `destroy_integrator` ()
Destroy the integrator.
- `er7_utils::IntegratorResult integrate` (double dyn_dt, unsigned int target_stage, double *xdot, double *x)
Propagate state to the specified stage of the integration process for an overall integration time interval of dyn_dt.
- void `reset_integrator` ()
Tell the integrator to reset itself.
- void `simple_restore` () override
Restore the integrator on restart.

Private Attributes

- `er7_utils::FirstOrderODEIntegrator * integrator` {}
Pointer to the object that performs integration.
- `RestartableFirstOrderODEIntegrator< 1 > integrator_manager`
Object that creates and manages the integrator object.

Friends

- class `InputProcessor`
- void `init_attrjeod__RestartableScalarFirstOrderODEIntegrator` ()

8.34.1 Detailed Description

A `RestartableScalarFirstOrderODEIntegrator` integrates a first order ODE, $dx/dt = v(x,t)$, where x is a scalar.

Definition at line 89 of file `restartable_state_integrator.hh`.

8.34.2 Constructor & Destructor Documentation

8.34.2.1 `RestartableScalarFirstOrderODEIntegrator()` [1/2]

```
jeod::RestartableScalarFirstOrderODEIntegrator::RestartableScalarFirstOrderODEIntegrator ( )  
[inline]
```

Default constructor.

Definition at line 95 of file `restartable_state_integrator.hh`.

8.34.2.2 `~RestartableScalarFirstOrderODEIntegrator()`

```
jeod::RestartableScalarFirstOrderODEIntegrator::~~RestartableScalarFirstOrderODEIntegrator ( )  
[override], [default]
```


8.34.2.3 RestartableScalarFirstOrderODEIntegrator() [2/2]

```
jeod::RestartableScalarFirstOrderODEIntegrator::RestartableScalarFirstOrderODEIntegrator (
    const RestartableScalarFirstOrderODEIntegrator & ) [delete]
```

8.34.3 Member Function Documentation

8.34.3.1 create_integrator()

```
void jeod::RestartableScalarFirstOrderODEIntegrator::create_integrator (
    const er7_utils::IntegratorConstructor & generator,
    er7_utils::IntegrationControls & controls ) [inline]
```

Create the integrator to be managed.

Parameters

in	<i>generator</i>	Integrator constructor used to create the integrator.
in, out	<i>controls</i>	Integration controls to be passed to the generator.

Definition at line 110 of file restartable_state_integrator.hh.

8.34.3.2 destroy_integrator()

```
void jeod::RestartableScalarFirstOrderODEIntegrator::destroy_integrator ( ) [inline]
```

Destroy the integrator.

Definition at line 119 of file restartable_state_integrator.hh.

8.34.3.3 integrate()

```
er7_utils::IntegratorResult jeod::RestartableScalarFirstOrderODEIntegrator::integrate (
    double dyn_dt,
    unsigned int target_stage,
    double * xdot,
    double * x ) [inline]
```

Propagate state to the specified stage of the integration process for an overall integration time interval of dyn_dt.

Note that this is a pass-through to the encapsulated integrator object. See er7_utils::FirstOrderODEIntegrator↵::integrate for details.

Parameters

in	<i>dyn_dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in	<i>xdot</i>	Time derivative of x.
in, out	<i>x</i>	Item to be integrated.

Returns

The status (time advance, pass/fail status) of the integration.

Definition at line 139 of file restartable_state_integrator.hh.

8.34.3.4 operator=()

```
RestartableScalarFirstOrderODEIntegrator& jeod::RestartableScalarFirstOrderODEIntegrator↔
::operator= (
    const RestartableScalarFirstOrderODEIntegrator & ) [delete]
```

8.34.3.5 reset_integrator()

```
void jeod::RestartableScalarFirstOrderODEIntegrator::reset_integrator ( ) [inline]
```

Tell the integrator to reset itself.

This should be called when the time step or time direction changes or upon a discrete change in state such docking/undocking. Such events invalidate saved state (if any). The integrator needs to restart from scratch when such events occur.

Definition at line 153 of file restartable_state_integrator.hh.

8.34.3.6 simple_restore()

```
void jeod::RestartableScalarFirstOrderODEIntegrator::simple_restore ( ) [inline], [override]
```

Restore the integrator on restart.

Definition at line 161 of file restartable_state_integrator.hh.

8.34.4 Friends And Related Function Documentation

8.34.4.1 init_attrjeod__RestartableScalarFirstOrderODEIntegrator

```
void init_attrjeod__RestartableScalarFirstOrderODEIntegrator ( ) [friend]
```

8.34.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 91 of file restartable_state_integrator.hh.

8.34.5 Field Documentation

8.34.5.1 integrator

```
er7_utils::FirstOrderODEIntegrator* jeod::RestartableScalarFirstOrderODEIntegrator::integrator
{} [private]
```

Pointer to the object that performs integration.

The object is created managed by the integrator manager.trick_units(–)

Definition at line 172 of file restartable_state_integrator.hh.

8.34.5.2 integrator_manager

```
RestartableFirstOrderODEIntegrator<1> jeod::RestartableScalarFirstOrderODEIntegrator::integrator↔
_manager [private]
```

Object that creates and manages the integrator object.

```
trick_io(**)
```

Definition at line 177 of file restartable_state_integrator.hh.

The documentation for this class was generated from the following file:

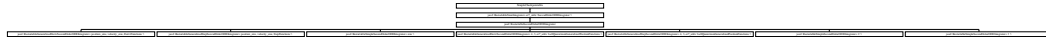
- [restartable_state_integrator.hh](#)

8.35 jeod::RestartableSecondOrderODEIntegrator Class Reference

A [RestartableSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a second order ODE problem.

```
#include <restartable_state_integrator_templates.hh>
```

Inheritance diagram for jeod::RestartableSecondOrderODEIntegrator:



Public Member Functions

- [~RestartableSecondOrderODEIntegrator](#) () override=default
- [RestartableSecondOrderODEIntegrator](#) (const [RestartableSecondOrderODEIntegrator](#) &)=delete
- [RestartableSecondOrderODEIntegrator](#) & operator= (const [RestartableSecondOrderODEIntegrator](#) &)=delete

Protected Member Functions

- [RestartableSecondOrderODEIntegrator](#) ()=default
- [RestartableSecondOrderODEIntegrator](#) (er7_utils::SecondOrderODEIntegrator *&integ_ref)

Non-default constructor.

8.35.1 Detailed Description

A [RestartableSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a second order ODE problem.

Definition at line 344 of file restartable_state_integrator_templates.hh.

8.35.2 Constructor & Destructor Documentation

8.35.2.1 ~RestartableSecondOrderODEIntegrator()

```
jeod::RestartableSecondOrderODEIntegrator::~~RestartableSecondOrderODEIntegrator ( ) [override],
[default]
```

8.35.2.2 RestartableSecondOrderODEIntegrator() [1/3]

```
jeod::RestartableSecondOrderODEIntegrator::RestartableSecondOrderODEIntegrator (
    const RestartableSecondOrderODEIntegrator & ) [delete]
```

8.35.2.3 RestartableSecondOrderODEIntegrator() [2/3]

```
jeod::RestartableSecondOrderODEIntegrator::RestartableSecondOrderODEIntegrator ( ) [protected],
[default]
```

8.35.2.4 RestartableSecondOrderODEIntegrator() [3/3]

```
jeod::RestartableSecondOrderODEIntegrator::RestartableSecondOrderODEIntegrator (
    er7_utils::SecondOrderODEIntegrator *& integ_ref ) [inline], [explicit], [protected]
```

Non-default constructor.

Parameters

<code>in, out</code>	<code>integ_ref</code>	Reference to the pointer to the integrator that is to be managed.
----------------------	------------------------	---

Definition at line 360 of file `restartable_state_integrator_templates.hh`.

8.35.3 Member Function Documentation

8.35.3.1 operator=()

```
RestartableSecondOrderODEIntegrator& jeod::RestartableSecondOrderODEIntegrator::operator= (
    const RestartableSecondOrderODEIntegrator & ) [delete]
```

The documentation for this class was generated from the following file:

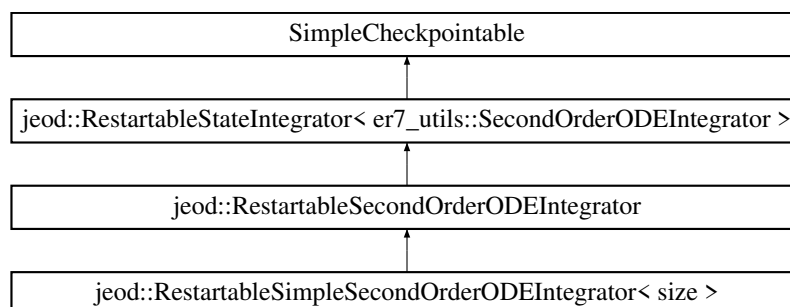
- [restartable_state_integrator_templates.hh](#)

8.36 jeod::RestartableSimpleSecondOrderODEIntegrator< size > Class Template Reference

A [RestartableSimpleSecondOrderODEIntegrator](#) is-a [RestartableSecondOrderODEIntegrator](#) that manages the integrator for a simple second order ODE problem, one in which the time derivative of the generalized position is the generalized velocity.

```
#include <restartable_state_integrator_templates.hh>
```

Inheritance diagram for `jeod::RestartableSimpleSecondOrderODEIntegrator< size >`:



Public Member Functions

- [RestartableSimpleSecondOrderODEIntegrator](#) ()=default
- [~RestartableSimpleSecondOrderODEIntegrator](#) () override=default
- [RestartableSimpleSecondOrderODEIntegrator](#) (const [RestartableSimpleSecondOrderODEIntegrator](#) &)=delete
- [RestartableSimpleSecondOrderODEIntegrator](#) & operator= (const [RestartableSimpleSecondOrderODEIntegrator](#) &)=delete
- [RestartableSimpleSecondOrderODEIntegrator](#) (er7_utils::SecondOrderODEIntegrator *&integ_ref)
Non-default constructor.

Private Member Functions

- er7_utils::SecondOrderODEIntegrator * [create_integrator_internal](#) (const er7_utils::IntegratorConstructor &generator, er7_utils::IntegrationControls &controls) override
Create the integrator to be managed.

Additional Inherited Members

8.36.1 Detailed Description

```
template<unsigned int size>
class jeod::RestartableSimpleSecondOrderODEIntegrator< size >
```

A [RestartableSimpleSecondOrderODEIntegrator](#) is-a [RestartableSecondOrderODEIntegrator](#) that manages the integrator for a simple second order ODE problem, one in which the time derivative of the generalized position is the generalized velocity.

Template Parameters

<i>size</i>	Size of the position vector (and also of velocity).
-------------	---

Definition at line 373 of file restartable_state_integrator_templates.hh.

8.36.2 Constructor & Destructor Documentation

8.36.2.1 RestartableSimpleSecondOrderODEIntegrator() [1/3]

```
template<unsigned int size>
jeod::RestartableSimpleSecondOrderODEIntegrator< size >::RestartableSimpleSecondOrderODEIntegrator
( ) [default]
```

8.36.2.2 ~RestartableSimpleSecondOrderODEIntegrator()

```
template<unsigned int size>
jeod::RestartableSimpleSecondOrderODEIntegrator< size >::~~RestartableSimpleSecondOrderODEIntegrator
( ) [override], [default]
```

8.36.2.3 RestartableSimpleSecondOrderODEIntegrator() [2/3]

```
template<unsigned int size>
jeod::RestartableSimpleSecondOrderODEIntegrator< size >::~RestartableSimpleSecondOrderODEIntegrator
(
    const RestartableSimpleSecondOrderODEIntegrator< size > & ) [delete]
```

8.36.2.4 RestartableSimpleSecondOrderODEIntegrator() [3/3]

```
template<unsigned int size>
jeod::RestartableSimpleSecondOrderODEIntegrator< size >::~RestartableSimpleSecondOrderODEIntegrator
(
    er7_utils::SecondOrderODEIntegrator *& integ_ref ) [inline], [explicit]
```

Non-default constructor.

Parameters

in, out	<i>integ_ref</i>	Reference to the pointer to the integrator that is to be managed.
---------	------------------	---

Definition at line 386 of file restartable_state_integrator_templates.hh.

8.36.3 Member Function Documentation

8.36.3.1 create_integrator_internal()

```
template<unsigned int size>
er7_utils::SecondOrderODEIntegrator* jeod::RestartableSimpleSecondOrderODEIntegrator< size >↔
::create_integrator_internal (
    const er7_utils::IntegratorConstructor & generator,
    er7_utils::IntegrationControls & controls ) [inline], [override], [private],
[virtual]
```

Create the integrator to be managed.

Parameters

in	<i>generator</i>	Integrator constructor used to create the integrator.
in, out	<i>controls</i>	Integration controls to be passed to the generator.

Implements [jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >](#).

Definition at line 399 of file restartable_state_integrator_templates.hh.

8.36.3.2 operator=()

```
template<unsigned int size>
RestartableSimpleSecondOrderODEIntegrator& jeod::RestartableSimpleSecondOrderODEIntegrator<
size >::operator= (
    const RestartableSimpleSecondOrderODEIntegrator< size > & ) [delete]
```

The documentation for this class was generated from the following file:

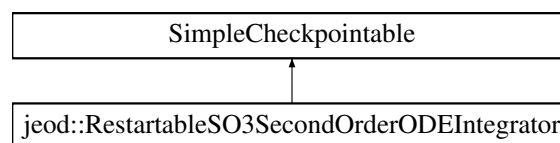
- [restartable_state_integrator_templates.hh](#)

8.37 jeod::RestartableSO3SecondOrderODEIntegrator Class Reference

A [RestartableSO3SecondOrderODEIntegrator](#) integrates a generalized second order ODE that describes rotation in three space.

```
#include <restartable_state_integrator.hh>
```

Inheritance diagram for jeod::RestartableSO3SecondOrderODEIntegrator:



Public Member Functions

- [RestartableSO3SecondOrderODEIntegrator \(\)](#)
Default constructor.
- [~RestartableSO3SecondOrderODEIntegrator \(\)](#) override
Destructor.
- [RestartableSO3SecondOrderODEIntegrator \(const RestartableSO3SecondOrderODEIntegrator &\)=delete](#)
- [RestartableSO3SecondOrderODEIntegrator & operator= \(const RestartableSO3SecondOrderODEIntegrator &\)=delete](#)
- void [create_integrator](#) ([GeneralizedSecondOrderODETechnique::TechniqueType](#) technique_in, const er7_utils::IntegratorConstructor &generator, er7_utils::IntegrationControls &controls)
Create the integrator to be managed.

- void [destroy_integrator](#) ()
Destroy the integrator.
- er7_utils::IntegratorResult [integrate](#) (double dyn_dt, unsigned int target_stage, const double *accel, double *velocity, double *position)
Propagate state to the specified stage of the integration process for an overall integration time interval of dyn_dt.
- void [reset_integrator](#) ()
Tell the integrator to reset itself.
- void [simple_restore](#) () override
Restore the integrator on restart.

Private Attributes

- [GeneralizedSecondOrderODETechnique::TechniqueType](#) technique
Specifies the mechanism for integrating rotational state.
- er7_utils::SecondOrderODEIntegrator * [integrator](#) {}
Pointer to the object that performs integration.
- [RestartableGeneralizedDerivSecondOrderODEIntegrator](#)< 4, 3, er7_utils::LeftQuaternionGeneralizedPositionFunctions > [generalized_deriv_integrator_manager](#)
SO3 generalized derivative integrator.
- [RestartableGeneralizedStepSecondOrderODEIntegrator](#)< 4, 3, er7_utils::LeftQuaternionGeneralizedPositionFunctions > [generalized_step_integrator_manager](#)
SO3 Lie Group integrator.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__RestartableSO3SecondOrderODEIntegrator](#) ()

8.37.1 Detailed Description

A [RestartableSO3SecondOrderODEIntegrator](#) integrates a generalized second order ODE that describes rotation in three space.

Definition at line 281 of file restartable_state_integrator.hh.

8.37.2 Constructor & Destructor Documentation

8.37.2.1 RestartableSO3SecondOrderODEIntegrator() [1/2]

```
jeod::RestartableSO3SecondOrderODEIntegrator::RestartableSO3SecondOrderODEIntegrator ( ) [inline]
```

Default constructor.

Definition at line 287 of file restartable_state_integrator.hh.

8.37.2.2 ~RestartableSO3SecondOrderODEIntegrator()

```
jeod::RestartableSO3SecondOrderODEIntegrator::~~RestartableSO3SecondOrderODEIntegrator ( )
[inline], [override]
```

Destructor.

Definition at line 295 of file restartable_state_integrator.hh.

8.37.2.3 RestartableSO3SecondOrderODEIntegrator() [2/2]

```
jeod::RestartableSO3SecondOrderODEIntegrator::~RestartableSO3SecondOrderODEIntegrator (
    const RestartableSO3SecondOrderODEIntegrator & ) [delete]
```

8.37.3 Member Function Documentation

8.37.3.1 create_integrator()

```
void jeod::RestartableSO3SecondOrderODEIntegrator::create_integrator (
    GeneralizedSecondOrderODETechnique::TechniqueType technique_in,
    const er7_utils::IntegratorConstructor & generator,
    er7_utils::IntegrationControls & controls ) [inline]
```

Create the integrator to be managed.

Parameters

in	<i>technique↔ _in</i>	Integration technique; generalized step vs deriv.
in	<i>generator</i>	Integrator constructor used to create the integrator.
in, out	<i>controls</i>	Integration controls to be passed to the generator.

Definition at line 309 of file restartable_state_integrator.hh.

8.37.3.2 destroy_integrator()

```
void jeod::RestartableSO3SecondOrderODEIntegrator::destroy_integrator ( ) [inline]
```

Destroy the integrator.

Definition at line 339 of file restartable_state_integrator.hh.

8.37.3.3 integrate()

```
er7_utils::IntegratorResult jeod::RestartableSO3SecondOrderODEIntegrator::integrate (
    double dyn_dt,
    unsigned int target_stage,
    const double * accel,
    double * velocity,
    double * position ) [inline]
```

Propagate state to the specified stage of the integration process for an overall integration time interval of `dyn_dt`.

Note that this is a pass-through to the encapsulated integrator object. See `er7_utils::SecondOrderODEIntegrator::integrate` for details.

Parameters

in	<i>dyn_dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in	<i>accel</i>	Time derivative of the generalized velocity.
in, out	<i>velocity</i>	Generalized velocity vector.
in, out	<i>position</i>	Generalized position vector.

Returns

The status (time advance, pass/fail status) of the integration.

Definition at line 363 of file `restartable_state_integrator.hh`.

8.37.3.4 operator=()

```
RestartableSO3SecondOrderODEIntegrator& jeod::RestartableSO3SecondOrderODEIntegrator::operator=
(
    const RestartableSO3SecondOrderODEIntegrator & ) [delete]
```

8.37.3.5 reset_integrator()

```
void jeod::RestartableSO3SecondOrderODEIntegrator::reset_integrator ( ) [inline]
```

Tell the integrator to reset itself.

This should be called when the time step or time direction changes or upon a discrete change in state such docking/undocking. Such events invalidate saved state (if any). The integrator needs to restart from scratch when such events occur.

Definition at line 378 of file `restartable_state_integrator.hh`.

8.37.3.6 simple_restore()

```
void jeod::RestartableSO3SecondOrderODEIntegrator::simple_restore ( ) [inline], [override]
```

Restore the integrator on restart.

Definition at line 386 of file restartable_state_integrator.hh.

8.37.4 Friends And Related Function Documentation

8.37.4.1 init_attrjeod__RestartableSO3SecondOrderODEIntegrator

```
void init_attrjeod__RestartableSO3SecondOrderODEIntegrator ( ) [friend]
```

8.37.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 283 of file restartable_state_integrator.hh.

8.37.5 Field Documentation

8.37.5.1 generalized_deriv_integrator_manager

```
RestartableGeneralizedDerivSecondOrderODEIntegrator<4, 3, er7_utils::LeftQuaternionGeneralized↵  
PositionFunctions> jeod::RestartableSO3SecondOrderODEIntegrator::generalized_deriv_integrator↵  
_manager [private]
```

SO3 generalized derivative integrator.

```
trick_io(**)
```

Definition at line 423 of file restartable_state_integrator.hh.

8.37.5.2 generalized_step_integrator_manager

```
RestartableGeneralizedStepSecondOrderODEIntegrator<4, 3, er7_utils::LeftQuaternionGeneralized↵
PositionFunctions> jeod::RestartableSO3SecondOrderODEIntegrator::generalized_step_integrator↵
_manager [private]
```

SO3 Lie Group integrator.

trick_io(**)

Definition at line 429 of file restartable_state_integrator.hh.

8.37.5.3 integrator

```
er7_utils::SecondOrderODEIntegrator* jeod::RestartableSO3SecondOrderODEIntegrator::integrator
{} [private]
```

Pointer to the object that performs integration.

This object is created and managed by one of the integrator managers defined below. The technique dictates which of the two is used.
trick_units(-)

Definition at line 417 of file restartable_state_integrator.hh.

8.37.5.4 technique

```
GeneralizedSecondOrderODETechnique::TechniqueType jeod::RestartableSO3SecondOrderODEIntegrator↵
::technique [private]
```

Initial value:

```
{
    GeneralizedSecondOrderODETechnique::Unspecified}
```

Specifies the mechanism for integrating rotational state.

trick_units(-)

Definition at line 409 of file restartable_state_integrator.hh.

The documentation for this class was generated from the following file:

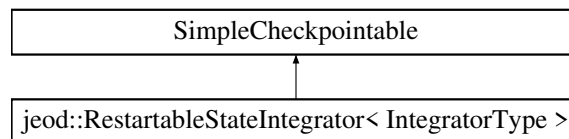
- [restartable_state_integrator.hh](#)

8.38 jeod::RestartableStateIntegrator< IntegratorType > Class Template Reference

A [RestartableStateIntegrator](#) establishes the basic capabilities needed to make a state integrator a managed resource.

```
#include <restartable_state_integrator_templates.hh>
```

Inheritance diagram for jeod::RestartableStateIntegrator< IntegratorType >:



Public Member Functions

- [~RestartableStateIntegrator](#) () override
Destructor.
- void [create_integrator](#) (const er7_utils::IntegratorConstructor &generator, er7_utils::IntegrationControls &controls)
Create the integrator to be managed.
- void [destroy_integrator](#) ()
Destroy the integrator.
- void [clear_integrator_reference](#) ()
Clear the pointer to the external integrator object.
- void [set_integrator_reference](#) (IntegratorType *&integ_ptr)
Set the pointer to the external integrator object.
- void [simple_restore](#) () override
Restore the integrator on restart.
- [RestartableStateIntegrator](#) (const [RestartableStateIntegrator](#) &)=delete
- [RestartableStateIntegrator](#) & operator= (const [RestartableStateIntegrator](#) &)=delete

Protected Member Functions

- [RestartableStateIntegrator](#) ()=default
- [RestartableStateIntegrator](#) (IntegratorType *&integ_ref)
Non-default constructor.

Private Member Functions

- virtual IntegratorType * [create_integrator_internal](#) (const er7_utils::IntegratorConstructor &generator, er7_utils::IntegrationControls &controls)=0
Create the integrator to be managed.
- virtual void [simple_restore_internal](#) (IntegratorType *integrator_ptr)
Perform technique-specific restart actions.

Private Attributes

- IntegratorType ** [integrator_handle](#) {}
Pointer to the containing object's integrator pointer.

8.38.1 Detailed Description

```
template<typename IntegratorType>
class jeod::RestartableStateIntegrator< IntegratorType >
```

A [RestartableStateIntegrator](#) establishes the basic capabilities needed to make a state integrator a managed resource.

This includes

- Creating the integrator object,
- Restoring the integrator's derivative function on restart, and
- Destroying the integrator when the [RestartableStateIntegrator](#) instance goes out of scope.

Template Parameters

<i>IntegratorType</i>	The type of integrator to be managed, either <code>er7_utils::FirstOrderODEIntegrator</code> or <code>er7_utils::SecondOrderODEIntegrator</code> .
-----------------------	--

Usage:

This base class template is not directly usable. One must instead use one of the three class templates that derive from this class template:

- [RestartableFirstOrderODEIntegrator](#) to manage an `er7_utils::FirstOrderODEIntegrator` pointer;
- [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) to manage an `er7_utils::SecondOrderODEIntegrator` pointer for a generalized second order ODE, one in which the time derivative of generalized position is a function of generalized position and generalized velocity; and
- [RestartableGeneralizedStepSecondOrderODEIntegrator](#) to manage an `er7_utils::SecondOrderODEIntegrator` pointer for a generalized second order ODE, one in which the time derivative of generalized position is a function of generalized position and generalized velocity; and
- [RestartableSimpleSecondOrderODEIntegrator](#) to manage an `er7_utils::SecondOrderODEIntegrator` pointer for a simple second order ODE, one in which the time derivative of generalized position is the generalized velocity.

Each state integrator to be used in some class needs a pair of data members declared in the definition of that class. The first of these pairs is a pointer to the appropriate state integrator type (either `er7_utils::FirstOrderODEIntegrator` or `er7_utils::SecondOrderODEIntegrator`). The second of the pairs of data members is an instance of the appropriate derived class of [RestartableStateIntegrator](#) that will manage the pointer. An example:

```
class MyClass {
...
    er7_utils::SecondOrderODEIntegrator * integrator;
    RestartableSimpleSecondOrderODEIntegrator<3> integ_manager;
...
};
```

The pointer itself must be exposed to Trick for checkpoint and restart. The `RestartableStateIntegrator`-derived object should be hidden from Trick.

The connection between the pointer and the manager for that pointer is made in the initializer lists of the constructors for the class. Use the `RestartableStateIntegrator` non-default constructor to tie the `RestartableStateIntegrator` object with the pointer it is to manage:

```
MyClass::MyClass ()
:
...
    integrator(nullptr),
    integ_manager(integrator),
...
{
    // Body of MyClass constructor
}
```

Note that there the `RestartableStateIntegrator` default constructors exist only for the sake of the simulation engine. The default constructors do not create viable instances. One must use the non-default constructor to initialize `RestartableStateIntegrator` instances.

The integrator itself is not created at construction time. Creating the integrator is the job of the `create_integrator` method. This method should be called at initialization time after having created the integration constructor and the integration controls objects.

The counterpart to the `create_integrator` method is `destroy_integrator`. This method must be called prior to calling `create_integrator` if the integrator has already been created via a previous call to `create_integrator`. (For example, switching to a different integration technique). The `destroy_integrator` can be called at shutdown or destruction time, but this call is not essential. The `RestartableStateIntegrator` object will call this method internally when it goes out of scope.

Class Design

This class template uses the non-virtual interface (NVI) design pattern, aka the template method design pattern (no relation to C++ class templates). The public `create_integrator` and `simple_restore` member functions use the private virtual `create_integrator_internal` and `simple_restore_internal` functions to create and restore the integrators.

Definition at line 171 of file `restartable_state_integrator_templates.hh`.

8.38.2 Constructor & Destructor Documentation

8.38.2.1 ~RestartableStateIntegrator()

```
template<typename IntegratorType>
jeod::RestartableStateIntegrator< IntegratorType >::~RestartableStateIntegrator ( ) [inline],
[override]
```

Destructor.

Definition at line 179 of file `restartable_state_integrator_templates.hh`.

8.38.2.2 RestartableStateIntegrator() [1/3]

```
template<typename IntegratorType>
jeod::RestartableStateIntegrator< IntegratorType >::RestartableStateIntegrator (
    const RestartableStateIntegrator< IntegratorType > & ) [delete]
```

8.38.2.3 RestartableStateIntegrator() [2/3]

```
template<typename IntegratorType>
jeod::RestartableStateIntegrator< IntegratorType >::RestartableStateIntegrator ( ) [protected],
[default]
```

8.38.2.4 RestartableStateIntegrator() [3/3]

```
template<typename IntegratorType>
jeod::RestartableStateIntegrator< IntegratorType >::RestartableStateIntegrator (
    IntegratorType *& integ_ref ) [inline], [explicit], [protected]
```

Non-default constructor.

Parameters

<i>in</i> , <i>out</i>	<i>integ_ref</i>	Reference to the pointer to the integrator that is to be managed.
------------------------	------------------	---

Definition at line 268 of file restartable_state_integrator_templates.hh.

8.38.3 Member Function Documentation**8.38.3.1 clear_integrator_reference()**

```
template<typename IntegratorType>
void jeod::RestartableStateIntegrator< IntegratorType >::clear_integrator_reference ( ) [inline]
```

Clear the pointer to the external integrator object.

This currently (pre-Trick 13.0) needs to be called on restart because both pointers point to invalid objects.

Definition at line 225 of file restartable_state_integrator_templates.hh.

8.38.3.2 create_integrator()

```
template<typename IntegratorType>
void jeod::RestartableStateIntegrator< IntegratorType >::create_integrator (
    const er7_utils::IntegratorConstructor & generator,
    er7_utils::IntegrationControls & controls ) [inline]
```

Create the integrator to be managed.

Parameters

in	<i>generator</i>	Integrator constructor used to create the integrator.
in, out	<i>controls</i>	Integration controls to be passed to the generator.

Definition at line 189 of file restartable_state_integrator_templates.hh.

8.38.3.3 create_integrator_internal()

```
template<typename IntegratorType>
virtual IntegratorType* jeod::RestartableStateIntegrator< IntegratorType >::create_integrator↵
_internal (
    const er7_utils::IntegratorConstructor & generator,
    er7_utils::IntegrationControls & controls ) [private], [pure virtual]
```

Create the integrator to be managed.

Parameters

in	<i>generator</i>	Integrator constructor used to create the integrator.
in, out	<i>controls</i>	Integration controls to be passed to the generator.

Returns

The constructed integrator.

Implemented in [jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, StepFunctions >](#), [jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< 4, 3, er7_utils::LeftQuaternionGeneralizedPositionFunctions >](#), [jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions >](#), [jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< 4, 3, er7_utils::LeftQuaternionGeneralizedPositionFunctions >](#), [jeod::RestartableSimpleSecondOrderODEIntegrator< size >](#), [jeod::RestartableSimpleSecondOrderODEIntegrator< 2 >](#), [jeod::RestartableSimpleSecondOrderODEIntegrator< 3 >](#), [jeod::RestartableFirstOrderODEIntegrator< size >](#), and [jeod::RestartableFirstOrderODEIntegrator< 1 >](#).

Referenced by [jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >::create_integrator\(\)](#).

8.38.3.4 destroy_integrator()

```
template<typename IntegratorType>
void jeod::RestartableStateIntegrator< IntegratorType >::destroy_integrator ( ) [inline]
```

Destroy the integrator.

Definition at line 212 of file restartable_state_integrator_templates.hh.

Referenced by [jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >::create_integrator\(\)](#), [jeod::RestartableT3SecondOrderODEIntegrator::destroy_integrator\(\)](#), and [jeod::RestartableStateIntegrator< er7↵_utils::SecondOrderODEIntegrator >::~~RestartableStateIntegrator\(\)](#).

8.38.3.5 operator=()

```
template<typename IntegratorType>
RestartableStateIntegrator& jeod::RestartableStateIntegrator< IntegratorType >::operator= (
    const RestartableStateIntegrator< IntegratorType > & ) [delete]
```

8.38.3.6 set_integrator_reference()

```
template<typename IntegratorType>
void jeod::RestartableStateIntegrator< IntegratorType >::set_integrator_reference (
    IntegratorType *& integ_ptr ) [inline]
```

Set the pointer to the external integrator object.

This currently (pre-Trick 13.0) needs to be called on restart because the `integrator_handle` is not properly restored by checkpoint.

Parameters

<code>in, out</code>	<code>integ_ptr</code>	Reference to the external integrator object.
----------------------	------------------------	--

Definition at line 236 of file `restartable_state_integrator_templates.hh`.

Referenced by `jeod::RestartableT3SecondOrderODEIntegrator::simple_restore()`.

8.38.3.7 simple_restore()

```
template<typename IntegratorType>
void jeod::RestartableStateIntegrator< IntegratorType >::simple_restore ( ) [inline], [override]
```

Restore the integrator on restart.

This currently (pre-Trick 13.0) needs to be called after calling `set_integrator_reference`.

Definition at line 246 of file `restartable_state_integrator_templates.hh`.

Referenced by `jeod::RestartableT3SecondOrderODEIntegrator::simple_restore()`.

8.38.3.8 simple_restore_internal()

```
template<typename IntegratorType>
virtual void jeod::RestartableStateIntegrator< IntegratorType >::simple_restore_internal (
    IntegratorType * integ_ptr ) [inline], [private], [virtual]
```

Perform technique-specific restart actions.

The default is to do nothing.

Parameters

<code>in, out</code>	<code>integrator_ptr</code>	The integrator object to be restored
----------------------	-----------------------------	--------------------------------------

Reimplemented in [jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< position_size, velocity_size, StepFunctions >](#), [jeod::RestartableGeneralizedStepSecondOrderODEIntegrator< 4, 3, er7_utils::LeftQuaternionGeneralizedPositionFunctions >](#), [jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< position_size, velocity_size, DerivFunctions >](#), and [jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator< 4, 3, er7_utils::LeftQuaternionGeneralizedPositionFunctions >](#).

Definition at line 290 of file `restartable_state_integrator_templates.hh`.

Referenced by `jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >::simple_restore()`.

8.38.4 Field Documentation

8.38.4.1 integrator_handle

```
template<typename IntegratorType>
IntegratorType** jeod::RestartableStateIntegrator< IntegratorType >::integrator_handle {}
[private]
```

Pointer to the containing object's integrator pointer.

`trick_io(**)`

Definition at line 297 of file `restartable_state_integrator_templates.hh`.

Referenced by `jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >::clear_integrator_↵reference()`, `jeod::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >::create_integrator()`, `jeod↵::RestartableStateIntegrator< er7_utils::SecondOrderODEIntegrator >::destroy_integrator()`, `jeod::Restartable↵StateIntegrator< er7_utils::SecondOrderODEIntegrator >::set_integrator_reference()`, and `jeod::Restartable↵StateIntegrator< er7_utils::SecondOrderODEIntegrator >::simple_restore()`.

The documentation for this class was generated from the following file:

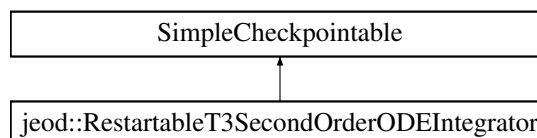
- [restartable_state_integrator_templates.hh](#)

8.39 jeod::RestartableT3SecondOrderODEIntegrator Class Reference

A [RestartableT3SecondOrderODEIntegrator](#) integrates a second order ODE in three space, $d^2x/dt^2 = a(x,t)$, where x is a three-vector.

```
#include <restartable_state_integrator.hh>
```

Inheritance diagram for `jeod::RestartableT3SecondOrderODEIntegrator`:



Public Member Functions

- [RestartableT3SecondOrderODEIntegrator](#) ()
Default constructor.
- [~RestartableT3SecondOrderODEIntegrator](#) () override=default
- [RestartableT3SecondOrderODEIntegrator](#) (const [RestartableT3SecondOrderODEIntegrator](#) &)=delete
- [RestartableT3SecondOrderODEIntegrator](#) & [operator=](#) (const [RestartableT3SecondOrderODEIntegrator](#) &)=delete
- void [create_integrator](#) (const er7_utils::IntegratorConstructor &generator, er7_utils::IntegrationControls &controls)
Create the integrator to be managed.
- void [destroy_integrator](#) ()
Destroy the integrator.
- er7_utils::IntegratorResult [integrate](#) (double dyn_dt, unsigned int target_stage, const double *accel, double *velocity, double *position)
Propagate state to the specified stage of the integration process for an overall integration time interval of dyn_dt.
- void [reset_integrator](#) ()
Tell the integrator to reset itself.
- void [simple_restore](#) () override
Restore the integrator on restart.

Private Attributes

- er7_utils::SecondOrderODEIntegrator * [integrator](#) {}
Pointer to the object that performs integration.
- [RestartableSimpleSecondOrderODEIntegrator](#)< 3 > [integrator_manager](#)
Object that creates and manages the integrator object.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__RestartableT3SecondOrderODEIntegrator](#) ()

8.39.1 Detailed Description

A [RestartableT3SecondOrderODEIntegrator](#) integrates a second order ODE in three space, $d^2x/dt^2 = a(x,t)$, where x is a three-vector.

Definition at line 184 of file restartable_state_integrator.hh.

8.39.2 Constructor & Destructor Documentation

8.39.2.1 RestartableT3SecondOrderODEIntegrator() [1/2]

```
jeod::RestartableT3SecondOrderODEIntegrator::RestartableT3SecondOrderODEIntegrator ( ) [inline]
```

Default constructor.

Definition at line 190 of file restartable_state_integrator.hh.

8.39.2.2 ~RestartableT3SecondOrderODEIntegrator()

```
jeod::RestartableT3SecondOrderODEIntegrator::~~RestartableT3SecondOrderODEIntegrator ( ) [override],
[default]
```

8.39.2.3 RestartableT3SecondOrderODEIntegrator() [2/2]

```
jeod::RestartableT3SecondOrderODEIntegrator::RestartableT3SecondOrderODEIntegrator (
    const RestartableT3SecondOrderODEIntegrator & ) [delete]
```

8.39.3 Member Function Documentation**8.39.3.1 create_integrator()**

```
void jeod::RestartableT3SecondOrderODEIntegrator::create_integrator (
    const er7_utils::IntegratorConstructor & generator,
    er7_utils::IntegrationControls & controls ) [inline]
```

Create the integrator to be managed.

Parameters

in	<i>generator</i>	Integrator constructor used to create the integrator.
in, out	<i>controls</i>	Integration controls to be passed to the generator.

Definition at line 205 of file restartable_state_integrator.hh.

8.39.3.2 destroy_integrator()

```
void jeod::RestartableT3SecondOrderODEIntegrator::destroy_integrator ( ) [inline]
```

Destroy the integrator.

Definition at line 214 of file `restartable_state_integrator.hh`.

References `jeod::RestartableStateIntegrator< IntegratorType >::destroy_integrator()`, and `integrator_manager`.

8.39.3.3 `integrate()`

```
er7_utils::IntegratorResult jeod::RestartableT3SecondOrderODEIntegrator::integrate (
    double dyn_dt,
    unsigned int target_stage,
    const double * accel,
    double * velocity,
    double * position ) [inline]
```

Propagate state to the specified stage of the integration process for an overall integration time interval of `dyn_dt`.

Note that this is a pass-through to the encapsulated integrator object. See `er7_utils::SecondOrderODEIntegrator::integrate` for details.

Parameters

in	<i>dyn_dt</i>	Dynamic time step, in dynamic time seconds.
in	<i>target_stage</i>	The stage of the integration process that the integrator should try to attain.
in	<i>accel</i>	Time derivative of the generalized velocity.
in, out	<i>velocity</i>	Generalized velocity vector.
in, out	<i>position</i>	Generalized position vector.

Returns

The status (time advance, pass/fail status) of the integration.

Definition at line 235 of file `restartable_state_integrator.hh`.

8.39.3.4 `operator=()`

```
RestartableT3SecondOrderODEIntegrator& jeod::RestartableT3SecondOrderODEIntegrator::operator=
(
    const RestartableT3SecondOrderODEIntegrator & ) [delete]
```


8.39.3.5 reset_integrator()

```
void jeod::RestartableT3SecondOrderODEIntegrator::reset_integrator ( ) [inline]
```

Tell the integrator to reset itself.

This should be called when the time step or time direction changes or upon a discrete change in state such docking/undocking. Such events invalidate saved state (if any). The integrator needs to restart from scratch when such events occur.

Definition at line 250 of file restartable_state_integrator.hh.

8.39.3.6 simple_restore()

```
void jeod::RestartableT3SecondOrderODEIntegrator::simple_restore ( ) [inline], [override]
```

Restore the integrator on restart.

Definition at line 258 of file restartable_state_integrator.hh.

References `integrator`, `integrator_manager`, `jeod::RestartableStateIntegrator< IntegratorType >::set_integrator_↵reference()`, and `jeod::RestartableStateIntegrator< IntegratorType >::simple_restore()`.

8.39.4 Friends And Related Function Documentation

8.39.4.1 init_attrjeod__RestartableT3SecondOrderODEIntegrator

```
void init_attrjeod__RestartableT3SecondOrderODEIntegrator ( ) [friend]
```

8.39.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 186 of file restartable_state_integrator.hh.

8.39.5 Field Documentation

8.39.5.1 integrator

```
er7_utils::SecondOrderODEIntegrator* jeod::RestartableT3SecondOrderODEIntegrator::integrator
{} [private]
```

Pointer to the object that performs integration.

The object is created managed by the integrator manager.`trick_units(-)`

Definition at line 269 of file `restartable_state_integrator.hh`.

Referenced by `simple_restore()`.

8.39.5.2 integrator_manager

```
RestartableSimpleSecondOrderODEIntegrator<3> jeod::RestartableT3SecondOrderODEIntegrator↔
::integrator_manager [private]
```

Object that creates and manages the integrator object.

`trick_io(**)`

Definition at line 274 of file `restartable_state_integrator.hh`.

Referenced by `destroy_integrator()`, and `simple_restore()`.

The documentation for this class was generated from the following file:

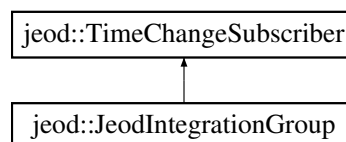
- [restartable_state_integrator.hh](#)

8.40 jeod::TimeChangeSubscriber Class Reference

A [TimeChangeSubscriber](#) is some object that wants to be notified of changes in the nature of time.

```
#include <time_change_subscriber.hh>
```

Inheritance diagram for `jeod::TimeChangeSubscriber`:



Public Member Functions

- [TimeChangeSubscriber](#) ()=default
- virtual [~TimeChangeSubscriber](#) ()=default
- [TimeChangeSubscriber](#) (const [TimeChangeSubscriber](#) &)=default
- [TimeChangeSubscriber](#) & operator= (const [TimeChangeSubscriber](#) &)=default
- virtual void [respond_to_time_change](#) ()=0

Somehow respond to a change in the nature of time.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__TimeChangeSubscriber](#) ()

8.40.1 Detailed Description

A [TimeChangeSubscriber](#) is some object that wants to be notified of changes in the nature of time.

Definition at line 74 of file `time_change_subscriber.hh`.

8.40.2 Constructor & Destructor Documentation

8.40.2.1 TimeChangeSubscriber() [1/2]

```
jeod::TimeChangeSubscriber::TimeChangeSubscriber ( ) [default]
```

8.40.2.2 ~TimeChangeSubscriber()

```
virtual jeod::TimeChangeSubscriber::~~TimeChangeSubscriber ( ) [virtual], [default]
```

8.40.2.3 TimeChangeSubscriber() [2/2]

```
jeod::TimeChangeSubscriber::TimeChangeSubscriber (
    const TimeChangeSubscriber & ) [default]
```

8.40.3 Member Function Documentation

8.40.3.1 operator=()

```
TimeChangeSubscriber& jeod::TimeChangeSubscriber::operator= (
    const TimeChangeSubscriber & ) [default]
```

8.40.3.2 respond_to_time_change()

```
virtual void jeod::TimeChangeSubscriber::respond_to_time_change ( ) [pure virtual]
```

Somehow respond to a change in the nature of time.

Implemented in [jeod::JeodIntegrationGroup](#).

8.40.4 Friends And Related Function Documentation

8.40.4.1 init_attrjeod__TimeChangeSubscriber

```
void init_attrjeod__TimeChangeSubscriber ( ) [friend]
```

8.40.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 76 of file time_change_subscriber.hh.

The documentation for this class was generated from the following file:

- [time_change_subscriber.hh](#)

8.41 er7_utils::TwoDArray< T > Class Template Reference

RAII template class that implements a rectangular two dimensional array.

```
#include <two_d_array.hh>
```

Public Member Functions

- [TwoDArray](#) ()=default
Default constructor.
- [TwoDArray](#) (const [TwoDArray](#)< T > &src)
Copy constructor.
- [~TwoDArray](#) ()
Destructor.
- [TwoDArray](#)< T > & [operator=](#) ([TwoDArray](#)< T > src)
Copy and swap assignment constructor.
- const T * [operator\[\]](#) (int N) const
Const overloaded index operator.
- T * [operator\[\]](#) (int N)
Non-const overloaded index operator.
- const T & [operator\(\)](#) (int N, int M) const
Const overloaded function operator.
- T & [operator\(\)](#) (int N, int M)
Non-const overloaded function operator.
- [operator](#) const T *const * () const
Const conversion operator to T const const*.*
- [operator](#) T*const * ()
*Non-const conversion operator to T*const*.*
- const T * [at](#) (int N) const
Range-checked equivalent of const T [operator\[\]](#)(int N) const.*
- T * [at](#) (int N)
Range-checked equivalent of T [operator\[\]](#)(int N).*
- const T & [at](#) (int N, int M) const
Range-checked equivalent of const T& [operator\(\)](#)(int N, int M) const.
- T & [at](#) (int N, int M)
Range-checked equivalent of T& [operator\(\)](#)(int N, int M).
- void [allocate](#) (std::size_t N, std::size_t M)
Allocate the array.
- void [rotate_down](#) (int limit)
Rotate elements 0 to limit downward, with array element 0 moved to array element limit.
- void [rotate_up](#) (int limit)
Rotate elements 0 to limit upward, with array element limit moved to array element 0.
- void [downsample](#) (int limit)
Downsample the array by swapping pointers.
- void [swap](#) ([TwoDArray](#)< T > &other)
*Swap the contents of *this with the other.*

Protected Attributes

- int [n](#) {}
The number of rows in the array.
- int [m](#) {}
The number of columns in the array.
- T * [data_array](#) {}
The array data, as an NxM array of T.
- T ** [row_array](#) {}
The rows in the array.

Private Member Functions

- void `allocate_internal` ()
Allocate memory for the array.
- void `deallocate_internal` ()
Deallocate memory for the array.

Friends

- void `swap` (`TwoDArray`< T > &first, `TwoDArray`< T > &second)
Swap the contents of the two provided arrays.

8.41.1 Detailed Description

```
template<typename T>
class er7_utils::TwoDArray< T >
```

RAII template class that implements a rectangular two dimensional array.

The implementation provides two special-purpose features that are needed by some of the ER7 utilities and JEOD integrators. The rows of the array can be rotated and downsampled.

Template Parameters

<code>T</code>	Type of each element of the array.
----------------	------------------------------------

Definition at line 79 of file `two_d_array.hh`.

8.41.2 Constructor & Destructor Documentation

8.41.2.1 `TwoDArray()` [1/2]

```
template<typename T>
er7_utils::TwoDArray< T >::TwoDArray ( ) [default]
```

Default constructor.

8.41.2.2 `TwoDArray()` [2/2]

```
template<typename T>
er7_utils::TwoDArray< T >::TwoDArray (
    const TwoDArray< T > & src ) [inline]
```

Copy constructor.

Parameters

<i>src</i>	Item to be copied.
------------	--------------------

Definition at line 91 of file two_d_array.hh.

8.41.2.3 ~TwoDArray()

```
template<typename T>
er7_utils::TwoDArray< T >::~~TwoDArray ( ) [inline]
```

Destructor.

Definition at line 109 of file two_d_array.hh.

8.41.3 Member Function Documentation

8.41.3.1 allocate()

```
template<typename T>
void er7_utils::TwoDArray< T >::allocate (
    std::size_t N,
    std::size_t M ) [inline]
```

Allocate the array.

Parameters

<i>N</i>	Number of rows in the array.
<i>M</i>	Number of columns in the array.

Exceptions

<i>std::domain_error</i>	N and/or M won't fit in a signed int.
--------------------------	---------------------------------------

Definition at line 252 of file two_d_array.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::GaussJacksonIntegratorBase(), and er7_utils::TwoDArray< double >::TwoDArray().

8.41.3.2 `allocate_internal()`

```
template<typename T>
void er7_utils::TwoDArray< T >::allocate_internal ( ) [inline], [private]
```

Allocate memory for the array.

Definition at line 384 of file `two_d_array.hh`.

Referenced by `er7_utils::TwoDArray< double >::allocate()`.

8.41.3.3 `at()` [1/4]

```
template<typename T>
const T* er7_utils::TwoDArray< T >::at (
    int N ) const [inline]
```

Range-checked equivalent of `const T* operator[] (int N) const`.

Parameters

<i>N</i>	Row index.
----------	------------

Returns

Const pointer to the Nth row in the array.

Exceptions

<code>std::out_of_range</code>	If <i>N</i> is an invalid index.
--------------------------------	----------------------------------

Definition at line 190 of file `two_d_array.hh`.

8.41.3.4 `at()` [2/4]

```
template<typename T>
T* er7_utils::TwoDArray< T >::at (
    int N ) [inline]
```

Range-checked equivalent of `T* operator[] (int N)`.

Parameters

<i>N</i>	Row index.
----------	------------

Returns

Modifiable pointer to the Nth row in the array.

Exceptions

<i>std::out_of_range</i>	If <i>N</i> is an invalid index.
--------------------------	----------------------------------

Definition at line 205 of file two_d_array.hh.

8.41.3.5 at() [3/4]

```
template<typename T>
const T& er7_utils::TwoDArray< T >::at (
    int N,
    int M ) const [inline]
```

Range-checked equivalent of const T& [operator\(\)\(int N, int M\) const](#).

Parameters

<i>N</i>	Row index.
<i>M</i>	Column index.

Returns

Const reference to the N,M element of the array.

Exceptions

<i>std::out_of_range</i>	If <i>N</i> or <i>M</i> is an invalid index.
--------------------------	--

Definition at line 221 of file two_d_array.hh.

8.41.3.6 at() [4/4]

```
template<typename T>
T& er7_utils::TwoDArray< T >::at (
    int N,
    int M ) [inline]
```

Range-checked equivalent of T& [operator\(\)\(int N, int M\)](#).

Parameters

<i>N</i>	Row index.
<i>M</i>	Column index.

Returns

Reference to the N,M element of the array.

Exceptions

<code>std::out_of_range</code>	If <i>N</i> or <i>M</i> is an invalid index.
--------------------------------	--

Definition at line 237 of file `two_d_array.hh`.

8.41.3.7 deallocate_internal()

```
template<typename T>
void er7_utils::TwoDArray< T >::deallocate_internal ( ) [inline], [private]
```

Deallocate memory for the array.

Definition at line 393 of file `two_d_array.hh`.

Referenced by `er7_utils::TwoDArray< double >::allocate()`, and `er7_utils::TwoDArray< double >::~~TwoDArray()`.

8.41.3.8 downsample()

```
template<typename T>
void er7_utils::TwoDArray< T >::downsample (
    int limit ) [inline]
```

Downsample the array by swapping pointers.

Parameters

<i>limit</i>	Number of usable rows after downsample.
--------------	---

Exceptions

<code>std::out_of_range</code>	If <i>limit</i> represents an invalid index.
--------------------------------	--

Definition at line 321 of file `two_d_array.hh`.

Referenced by `jeod::GaussJacksonIntegratorBase< State, Primer >::downsample_hist()`.

8.41.3.9 operator const T *const *()

```
template<typename T>
er7_utils::TwoDArray< T >::operator const T *const * ( ) const [inline]
```

Const conversion operator to T const* const*.

Returns

Non-modifiable pointer to the array.

Definition at line 170 of file two_d_array.hh.

References er7_utils::TwoDArray< T >::row_array.

8.41.3.10 operator T*const *()

```
template<typename T>
er7_utils::TwoDArray< T >::operator T*const * ( ) [inline]
```

Non-const conversion operator to T*const*.

Returns

Modifiable pointer to the array.

Definition at line 179 of file two_d_array.hh.

References er7_utils::TwoDArray< T >::row_array.

8.41.3.11 operator()() [1/2]

```
template<typename T>
const T& er7_utils::TwoDArray< T >::operator() (
    int N,
    int M ) const [inline]
```

Const overloaded function operator.

Parameters

<i>N</i>	Row index.
<i>M</i>	Column index.

Returns

Const reference to the N,M element of the array.

Definition at line 150 of file two_d_array.hh.

8.41.3.12 operator() [2/2]

```
template<typename T>
T& er7_utils::TwoDArray< T >::operator() (
    int N,
    int M ) [inline]
```

Non-const overloaded function operator.

Parameters

<i>N</i>	Row index.
<i>M</i>	Column index.

Returns

Reference to the N,M element of the array.

Definition at line 161 of file two_d_array.hh.

8.41.3.13 operator=()

```
template<typename T>
TwoDArray<T>& er7_utils::TwoDArray< T >::operator= (
    TwoDArray< T > src ) [inline]
```

Copy and swap assignment constructor.

Parameters

<i>src</i>	Item to be copied.
------------	--------------------

Definition at line 118 of file two_d_array.hh.

8.41.3.14 operator[] [1/2]

```
template<typename T>
const T* er7_utils::TwoDArray< T >::operator[] (
    int N ) const [inline]
```

Const overloaded index operator.

Parameters

<i>N</i>	Row index.
----------	------------

Returns

Const pointer to the Nth row in the array.

Definition at line 129 of file two_d_array.hh.

8.41.3.15 operator[]() [2/2]

```
template<typename T>
T* er7_utils::TwoDArray< T >::operator[] (
    int N ) [inline]
```

Non-const overloaded index operator.

Parameters

<i>N</i>	Row index.
----------	------------

Returns

Modifiable pointer to the Nth row in the array.

Definition at line 139 of file two_d_array.hh.

8.41.3.16 rotate_down()

```
template<typename T>
void er7_utils::TwoDArray< T >::rotate_down (
    int limit ) [inline]
```

Rotate elements 0 to limit downward, with array element 0 moved to array element limit.

Parameters

<i>limit</i>	Index of last element participating in the rotation.
--------------	--

Exceptions

<i>std::out_of_range</i>	If <i>limit</i> is an invalid index.
--------------------------	--------------------------------------

Definition at line 280 of file two_d_array.hh.

Referenced by jeod::GaussJacksonIntegratorBase< State, Primer >::rotate_acc_hist().

8.41.3.17 rotate_up()

```
template<typename T>
void er7_utils::TwoDArray< T >::rotate_up (
    int limit ) [inline]
```

Rotate elements 0 to limit upward, with array element limit moved to array element 0.

Parameters

<i>limit</i>	Index of last element participating in the rotation.
--------------	--

Exceptions

<i>std::out_of_range</i>	If <i>limit</i> is an invalid index.
--------------------------	--------------------------------------

Definition at line 301 of file two_d_array.hh.

8.41.3.18 swap()

```
template<typename T>
void er7_utils::TwoDArray< T >::swap (
    TwoDArray< T > & other ) [inline]
```

Swap the contents of *this with the other.

Parameters

<i>other</i>	Other array.
--------------	--------------

Definition at line 338 of file two_d_array.hh.

Referenced by er7_utils::TwoDArray< double >::operator=(), and jeod::GaussJacksonIntegratorBase< State, Primer >::swap().

8.41.4 Friends And Related Function Documentation

8.41.4.1 swap

```
template<typename T>
void swap (
    TwoDArray< T > & first,
    TwoDArray< T > & second ) [friend]
```

Swap the contents of the two provided arrays.

Parameters

<i>first</i>	First array.
<i>second</i>	Second array.

Definition at line 351 of file two_d_array.hh.

8.41.5 Field Documentation

8.41.5.1 data_array

```
template<typename T>
T* er7_utils::TwoDArray< T >::data_array {} [protected]
```

The array data, as an NxM array of T.

trick_units(-)

Definition at line 373 of file two_d_array.hh.

Referenced by er7_utils::TwoDArray< double >::allocate(), er7_utils::TwoDArray< double >::allocate_internal(), er7_utils::TwoDArray< double >::deallocate_internal(), and er7_utils::TwoDArray< double >::swap().

8.41.5.2 m

```
template<typename T>
int er7_utils::TwoDArray< T >::m {} [protected]
```

The number of columns in the array.

trick_units(-)

Definition at line 368 of file two_d_array.hh.

Referenced by er7_utils::TwoDArray< double >::allocate(), er7_utils::TwoDArray< double >::allocate_internal(), er7_utils::TwoDArray< double >::at(), er7_utils::TwoDArray< double >::swap(), and er7_utils::TwoDArray< double >::TwoDArray().

8.41.5.3 `n`

```
template<typename T>
int er7_utils::TwoDArray< T >::n {} [protected]
```

The number of rows in the array.

trick_units(—)

Definition at line 363 of file `two_d_array.hh`.

Referenced by `er7_utils::TwoDArray< double >::allocate()`, `er7_utils::TwoDArray< double >::allocate_internal()`, `er7_utils::TwoDArray< double >::at()`, `er7_utils::TwoDArray< double >::downsample()`, `er7_utils::TwoDArray< double >::rotate_down()`, `er7_utils::TwoDArray< double >::rotate_up()`, `er7_utils::TwoDArray< double >::swap()`, and `er7_utils::TwoDArray< double >::TwoDArray()`.

8.41.5.4 `row_array`

```
template<typename T>
T** er7_utils::TwoDArray< T >::row_array {} [protected]
```

The rows in the array.

trick_units(—)

Definition at line 378 of file `two_d_array.hh`.

Referenced by `er7_utils::TwoDArray< double >::allocate()`, `er7_utils::TwoDArray< double >::allocate_internal()`, `er7_utils::TwoDArray< double >::at()`, `er7_utils::TwoDArray< double >::deallocate_internal()`, `er7_utils::TwoDArray< double >::downsample()`, `er7_utils::TwoDArray< T >::operator const T *const *()`, `er7_utils::TwoDArray< T >::operator T*const *()`, `er7_utils::TwoDArray< double >::operator()()`, `er7_utils::TwoDArray< double >::operator[]()`, `er7_utils::TwoDArray< double >::rotate_down()`, `er7_utils::TwoDArray< double >::rotate_up()`, `er7_utils::TwoDArray< double >::swap()`, and `er7_utils::TwoDArray< double >::TwoDArray()`.

The documentation for this class was generated from the following file:

- [two_d_array.hh](#)

Chapter 9

File Documentation

9.1 gauss_jackson_coefficients_pair.cc File Reference

Defines member functions for the class GaussJacksonCoefficientsPair.

```
#include "../include/gauss_jackson_coefficients_pair.hh"
#include "../include/gauss_jackson_two_state.hh"
#include "er7_utils/interface/include/alloc.hh"
#include <algorithm>
#include <iostream>
```

Namespaces

- [jeod](#)

Namespace jeod.

9.1.1 Detailed Description

Defines member functions for the class GaussJacksonCoefficientsPair.

9.2 gauss_jackson_coefficients_pair.hh File Reference

Defines the class GaussJacksonCoefficientsPair, which contains summed Adams and Gauss-Jackson coefficient pair.

```
#include "gauss_jackson_one_state.hh"
#include "gauss_jackson_two_state.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/interface/include/config.hh"
#include <cstdint>
#include <iosfwd>
```

Data Structures

- class [jeod::GaussJacksonCoefficientsPair](#)
Contains a summed Adams and Gauss-Jackson coefficient pair.

Namespaces

- [jeod](#)
Namespace jeod.

9.2.1 Detailed Description

Defines the class GaussJacksonCoefficientsPair, which contains summed Adams and Gauss-Jackson coefficient pair.

9.3 gauss_jackson_coeffs.cc File Reference

Defines member functions for the class GaussJacksonCoeffs.

```
#include "../include/jeod/jeod.h"
#include "../include/jeod/jeod_coeffs.h"
#include "../include/jeod/jeod_coeffs_pair.h"
#include "er7_utils/interface/include/alloc.h"
#include "er7_utils/math/include/n_choose_m.h"
#include <algorithm>
#include <cassert>
#include <iostream>
```

Namespaces

- [jeod](#)
Namespace jeod.

Functions

- `std::ostream & jeod::operator<< (std::ostream &stream, const GaussJacksonCoeffs &coeff)`

9.3.1 Detailed Description

Defines member functions for the class GaussJacksonCoeffs.

9.4 gauss_jackson_coefs.hh File Reference

Defines the class GaussJacksonCoefs, which contains the Gauss-Jackson predictor and corrector coefficients.

```
#include "gauss_jackson_coefficients_pair.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include <iosfwd>
```

Data Structures

- class [jeod::GaussJacksonCoefs](#)

Contains the Gauss-Jackson predictor and corrector coefficients.

Namespaces

- [jeod](#)

Namespace jeod.

9.4.1 Detailed Description

Defines the class GaussJacksonCoefs, which contains the Gauss-Jackson predictor and corrector coefficients.

9.5 gauss_jackson_config.cc File Reference

Defines member functions for the class GaussJacksonIntegratorConstructor.

```
#include "utils/math/include/numerical.hh"
#include "../include/gauss_jackson_config.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include <algorithm>
#include <string>
```

Namespaces

- [jeod](#)

Namespace jeod.

Functions

- static GaussJacksonConfig [jeod::set_default_config_values](#) (const GaussJacksonConfig &config)
Swap the negative ones in the supplied config with the default values, some of which are computed.
- static unsigned int [jeod::validate_config](#) (const GaussJacksonConfig &config)
Check for invalid values in the supplied config.

9.5.1 Detailed Description

Defines member functions for the class GaussJacksonIntegratorConstructor.

9.6 gauss_jackson_config.hh File Reference

Defines the class GaussJacksonConfig, which specifies Gauss-Jackson configuration data.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/integration_technique.hh"
```

Data Structures

- class [jeod::GaussJacksonConfig](#)
Contains Gauss-Jackson configuration data.

Namespaces

- [jeod](#)
Namespace jeod.

9.6.1 Detailed Description

Defines the class GaussJacksonConfig, which specifies Gauss-Jackson configuration data.

9.7 gauss_jackson_first_order_ode_integrator.hh File Reference

Defines the class GaussJacksonFirstOrderODEIntegrator, which integrates a first order ODE using the summed Adams technique.

```
#include "gauss_jackson_integrator_base_first.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/first_order_ode_integrator.hh"
```

Data Structures

- class [jeod::GaussJacksonFirstOrderODEIntegrator](#)
Integrates a first order ODE using the summed Adams technique that is embedded within the Gauss-Jackson technique.

Namespaces

- [jeod](#)
Namespace jeod.

9.7.1 Detailed Description

Defines the class GaussJacksonFirstOrderODEIntegrator, which integrates a first order ODE using the summed Adams technique.

- Note: This is a header-only implementation. There is no source file that corresponds to this header.

9.8 gauss_jackson_generalized_second_order_ode_integrator.cc File Reference

Defines member functions for the class GaussJacksonGeneralizedDerivSecondOrderODEIntegrator.

```
#include "../include/gauss_jackson_generalized_second_order_ode_integrator.↵
hh"
#include "er7_utils/interface/include/alloc.hh"
#include <algorithm>
```

Namespaces

- [jeod](#)
Namespace jeod.

9.8.1 Detailed Description

Defines member functions for the class GaussJacksonGeneralizedDerivSecondOrderODEIntegrator.

9.9 gauss_jackson_generalized_second_order_ode_integrator.hh File Reference

Defines the class GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, which integrates a generalized derivative second order ODE using Gauss-Jackson.

```
#include "gauss_jackson_first_order_ode_integrator.hh"
#include "gauss_jackson_simple_second_order_ode_integrator.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/second_order_ode_integrator.↵
hh"
```

Data Structures

- class [jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator](#)
Integrates a generalized derivative second order ODE using Gauss-Jackson.

Namespaces

- [jeod](#)

Namespace jeod.

9.9.1 Detailed Description

Defines the class GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, which integrates a generalized derivative second order ODE using Gauss-Jackson.

9.10 gauss_jackson_integration_controls.cc File Reference

Defines member functions for the class GaussJacksonIntegrationControls.

```
#include "utils/math/include/numerical.hh"
#include "../include/gauss_jackson_integration_controls.hh"
#include "er7_utils/integration/core/include/base_integration_group.hh"
#include "er7_utils/integration/core/include/integrator_constructor.hh"
#include "er7_utils/integration/core/include/integrator_result.hh"
#include "er7_utils/integration/core/include/time_interface.hh"
#include "er7_utils/interface/include/alloc.hh"
#include <cassert>
```

Namespaces

- [jeod](#)

Namespace jeod.

9.10.1 Detailed Description

Defines member functions for the class GaussJacksonIntegrationControls.

9.11 gauss_jackson_integration_controls.hh File Reference

Defines the class GaussJacksonIntegrationControls, which controls Gauss-Jackson integration process.

```
#include "gauss_jackson_coeffs.hh"
#include "gauss_jackson_config.hh"
#include "gauss_jackson_state_machine.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/integration_controls.hh"
#include "er7_utils/integration/core/include/integrator_constructor.hh"
```

Data Structures

- class [jeod::GaussJacksonIntegrationControls](#)
IntegrationControls specialized for Gauss-Jackson integration.

Namespaces

- [jeod](#)
Namespace jeod.

9.11.1 Detailed Description

Defines the class GaussJacksonIntegrationControls, which controls Gauss-Jackson integration process.

9.12 gauss_jackson_integrator_base.hh File Reference

Defines the template class GaussJacksonIntegratorBase, which provides the basis for Gauss-Jackson integration.

```
#include "gauss_jackson_coeffs.hh"
#include "gauss_jackson_config.hh"
#include "gauss_jackson_integration_controls.hh"
#include "gauss_jackson_state_machine.hh"
#include "two_d_array.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/integ_utils.hh"
#include "er7_utils/integration/core/include/integrator_result.hh"
#include "er7_utils/interface/include/alloc.hh"
#include <algorithm>
#include <cassert>
```

Data Structures

- class [jeod::GaussJacksonIntegratorBase< State, Primer >](#)
Base template class for integrating state via the Gauss-Jackson technique.

Namespaces

- [jeod](#)
Namespace jeod.

9.12.1 Detailed Description

Defines the template class GaussJacksonIntegratorBase, which provides the basis for Gauss-Jackson integration.

9.13 gauss_jackson_integrator_base_first.hh File Reference

Defines the template specializations of GaussJacksonIntegratorBase for a first order ODE.

```
#include "gauss_jackson_integrator_base.hh"
#include "gauss_jackson_one_state.hh"
#include "er7_utils/integration/core/include/first_order_ode_integrator.hh"
#include "er7_utils/integration/core/include/integrator_constructor.hh"
#include "er7_utils/interface/include/alloc.hh"
#include <algorithm>
#include <cmath>
```

Namespaces

- [jeod](#)
Namespace jeod.

Typedefs

- using [jeod::GaussJacksonIntegratorBaseFirst](#) = GaussJacksonIntegratorBase< GaussJacksonOneState, er7_utils::FirstOrderODEIntegrator >
Alias for a first order Gauss Jackson integrator.

9.13.1 Detailed Description

Defines the template specializations of GaussJacksonIntegratorBase for a first order ODE.

9.14 gauss_jackson_integrator_base_second.hh File Reference

Defines the template specializations of GaussJacksonIntegratorBase for a first order ODE.

```
#include "gauss_jackson_integrator_base.hh"
#include "gauss_jackson_two_state.hh"
#include "er7_utils/integration/core/include/integrator_constructor.hh"
#include "er7_utils/integration/core/include/second_order_ode_integrator.↵
hh"
#include "er7_utils/interface/include/alloc.hh"
#include <algorithm>
#include <cmath>
```

Namespaces

- [jeod](#)
Namespace jeod.

Typedefs

- using [jeod::GaussJacksonIntegratorBaseSecond](#) = GaussJacksonIntegratorBase< GaussJacksonTwoState, er7_utils::SecondOrderODEIntegrator >

Alias for a second order Gauss Jackson integrator.

9.14.1 Detailed Description

Defines the template specializations of GaussJacksonIntegratorBase for a first order ODE.

9.15 gauss_jackson_integrator_constructor.cc File Reference

Defines member functions for the class GaussJacksonIntegratorConstructor.

```
#include "../include/gauss_jackson_integrator_constructor.hh"
#include "../include/gauss_jackson_first_order_ode_integrator.hh"
#include "../include/gauss_jackson_generalized_second_order_ode_integrator.↵
hh"
#include "../include/gauss_jackson_simple_second_order_ode_integrator.hh"
#include "er7_utils/integration/core/include/integration_controls.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/integration/core/include/integrator_constructor_factory.↵
hh"
#include "er7_utils/integration/core/include/integrator_constructor_utils.↵
hh"
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include <algorithm>
```

Namespaces

- [jeod](#)

Namespace jeod.

Functions

- static GaussJacksonIntegrationControls * [jeod::cast_to_gj_controls](#) (er7_utils::IntegrationControls &controls)

Cast the provided integration controls to a [GaussJacksonIntegrationControls](#).

9.15.1 Detailed Description

Defines member functions for the class GaussJacksonIntegratorConstructor.

9.16 gauss_jackson_integrator_constructor.hh File Reference

Defines the class GaussJacksonIntegratorConstructor, which constructs integrators that use Gauss-Jackson integration.

```
#include "gauss_jackson_config.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/integrator_constructor.hh"
#include <string>
```

Data Structures

- class [jeod::GaussJacksonIntegratorConstructor](#)
Create state and time integrators that propagate using Gauss-Jackson.

Namespaces

- [jeod](#)
Namespace jeod.

9.16.1 Detailed Description

Defines the class GaussJacksonIntegratorConstructor, which constructs integrators that use Gauss-Jackson integration.

9.17 gauss_jackson_one_state.hh File Reference

Defines the class GaussJacksonOneState, which contains a double* pointer.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include <cstddef>
```

Data Structures

- class [jeod::GaussJacksonOneState](#)
Essentially just a double.*

Namespaces

- [jeod](#)
Namespace jeod.

9.17.1 Detailed Description

Defines the class GaussJacksonOneState, which contains a double* pointer.

9.18 gauss_jackson_rational_coefs.cc File Reference

Defines member functions for the class GaussJacksonRationalCoefficients.

```
#include "../include/jeod/jeod_class.hh"
#include <cassert>
```

Namespaces

- [jeod](#)

Namespace jeod.

9.18.1 Detailed Description

Defines member functions for the class GaussJacksonRationalCoefficients.

9.19 gauss_jackson_rational_coefs.hh File Reference

Defines the class GaussJacksonRationalCoefficients, which contains a set of Adams or Stormer-Cowell coefficients.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/math/include/n_choose_m.hh"
#include "er7_utils/math/include/ratio128.hh"
#include <vector>
```

Data Structures

- class [jeod::GaussJacksonRationalCoefficients](#)

Contains a set of Adams or Stormer-Cowell coefficients.

Namespaces

- [jeod](#)

Namespace jeod.

9.19.1 Detailed Description

Defines the class GaussJacksonRationalCoefficients, which contains a set of Adams or Stormer-Cowell coefficients.

9.20 gauss_jackson_simple_second_order_ode_integrator.hh File Reference

Defines the class GaussJacksonSimpleSecondOrderODEIntegrator, which integrates a simple second order ODE using Gauss-Jackson.

```
#include "gauss_jackson_integrator_base_second.hh"
#include "gauss_jackson_two_state.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/second_order_ode_integrator.↵
hh"
```

Data Structures

- class [jeod::GaussJacksonSimpleSecondOrderODEIntegrator](#)
Integrates a simple second order ODE using the Gauss-Jackson technique.

Namespaces

- [jeod](#)
Namespace jeod.

9.20.1 Detailed Description

Defines the class GaussJacksonSimpleSecondOrderODEIntegrator, which integrates a simple second order ODE using Gauss-Jackson.

9.21 gauss_jackson_state_machine.cc File Reference

Defines member functions for the class GaussJacksonStateMachine.

```
#include "../include/gauss_jackson_state_machine.hh"
#include "../include/gauss_jackson_config.hh"
#include <algorithm>
#include <cassert>
#include <string>
```

Namespaces

- [jeod](#)
Namespace jeod.

9.21.1 Detailed Description

Defines member functions for the class GaussJacksonStateMachine.

9.22 gauss_jackson_state_machine.hh File Reference

Defines the class GaussJacksonStateMachine, which guides the Gauss-Jackson integration process.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include <string>
```

Data Structures

- class [jeod::GaussJacksonStateMachine](#)
Guides the behavior of the Gauss-Jackson integration process via a finite state machine.

Namespaces

- [jeod](#)
Namespace jeod.

9.22.1 Detailed Description

Defines the class GaussJacksonStateMachine, which guides the Gauss-Jackson integration process.

9.23 gauss_jackson_two_state.hh File Reference

Defines the class GaussJacksonTwoState, which contains a pair of double* pointers.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include <cstddef>
```

Data Structures

- class [jeod::GaussJacksonTwoState](#)
Essentially just std::pair<double>.*

Namespaces

- [jeod](#)
Namespace jeod.

9.23.1 Detailed Description

Defines the class GaussJacksonTwoState, which contains a pair of double* pointers.

9.24 generalized_second_order_ode_technique.cc File Reference

Define class GeneralizedSecondOrderODETechnique methods.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "../include/generalized_second_order_ode_technique.hh"
#include "../include/integration_messages.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.24.1 Detailed Description

Define class GeneralizedSecondOrderODETechnique methods.

9.25 generalized_second_order_ode_technique.hh File Reference

Define the static class GeneralizedSecondOrderODETechnique.

```
#include "er7_utils/integration/core/include/integration_technique.hh"
#include "er7_utils/integration/core/include/integrator_constructor.hh"
```

Data Structures

- class [jeod::GeneralizedSecondOrderODETechnique](#)

Enumerates the integration schemes, generalized Cartesian or Lie group integration, and provides simple utilities that work with this.

Namespaces

- [jeod](#)

Namespace jeod.

9.25.1 Detailed Description

Define the static class GeneralizedSecondOrderODETechnique.

9.26 integration_messages.cc File Reference

Implement the class IntegrationMessages.

```
#include "utils/message/include/make_message_code.hh"
#include "../include/integration_messages.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

Macros

- #define [MAKE_INTEGRATION_MESSAGE_CODE](#)(id) JEOD_MAKE_MESSAGE_CODE(IntegrationMessages, "utils/integration/", id)

9.26.1 Detailed Description

Implement the class IntegrationMessages.

9.26.2 Macro Definition Documentation

9.26.2.1 MAKE_INTEGRATION_MESSAGE_CODE

```
#define MAKE_INTEGRATION_MESSAGE_CODE(  
    id ) JEOD_MAKE_MESSAGE_CODE(IntegrationMessages, "utils/integration/", id)
```

Definition at line 37 of file integration_messages.cc.

9.27 integration_messages.hh File Reference

Define the class IntegrationMessages, the class that specifies the message IDs used in the integration model.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::IntegrationMessages](#)
Declares messages associated with the integration test model.

Namespaces

- [jeod](#)

Namespace jeod.

9.27.1 Detailed Description

Define the class IntegrationMessages, the class that specifies the message IDs used in the integration model.

9.28 jeod_integration_group.cc File Reference

Define JeodIntegrationGroup methods.

```
#include "../include/jeod_integration_group.hh"
#include "../include/integration_messages.hh"
#include "../include/jeod_integration_time.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/integrable_object.hh"
#include "er7_utils/integration/core/include/integrator_constructor.hh"
#include <algorithm>
#include <cstdint>
```

Namespaces

- [jeod](#)

Namespace jeod.

9.28.1 Detailed Description

Define JeodIntegrationGroup methods.

9.29 jeod_integration_group.hh File Reference

Define the extensible class IntegrationGroup, an instance of which is responsible for integrating the states of a set of DynBody objects.

```
#include "time_change_subscriber.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/container/include/pointer_vector.hh"
#include "utils/sim_interface/include/jeod_integrator_interface.hh"
#include "er7_utils/integration/core/include/base_integration_group.hh"
#include "er7_utils/integration/core/include/integrable_object.hh"
#include "er7_utils/integration/core/include/integrator_constructor.hh"
#include "er7_utils/integration/core/include/integrator_result_merger_↵
container.hh"
#include <cstdint>
```


Data Structures

- class [jeod::JeodIntegrationGroupOwner](#)
The abstract class *IntegrationGroupOwner* contains an *IntegrationGroup*.
- class [jeod::JeodIntegrationGroup](#)
A *JeodIntegrationGroup* integrates the state of a set of objects over time.

Namespaces

- [jeod](#)
Namespace *jeod*.

9.29.1 Detailed Description

Define the extensible class *IntegrationGroup*, an instance of which is responsible for integrating the states of a set of *DynBody* objects.

9.30 jeod_integration_time.cc File Reference

Define *JeodIntegrationTime* methods.

```
#include <algorithm>
#include <cstdlib>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/integration_messages.hh"
#include "../include/jeod_integration_time.hh"
#include "../include/time_change_subscriber.hh"
```

Namespaces

- [jeod](#)
Namespace *jeod*.

9.30.1 Detailed Description

Define *JeodIntegrationTime* methods.

9.31 jeod_integration_time.hh File Reference

Define the class *JeodIntegrationTime*.

```
#include "er7_utils/integration/core/include/time_interface.hh"
#include "utils/container/include/pointer_vector.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::JeodIntegrationTime](#)

The class [JeodIntegrationTime](#) adds the concepts of timestamping and notification of changes in the nature of time to the ER7 numerical utilities *TimeInterface* class.

Namespaces

- [jeod](#)

Namespace *jeod*.

9.31.1 Detailed Description

Define the class *JeodIntegrationTime*.

9.32 *lsode_control_data_interface.cc* File Reference

Define member functions for the class *LsodeControlDataInterface*.

```
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "../include/lsode_control_data_interface.hh"
```

9.32.1 Detailed Description

Define member functions for the class *LsodeControlDataInterface*.

9.33 *lsode_control_data_interface.hh* File Reference

Define the class *LsodeControlDataInterface*.

```
#include <vector>
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::LsodeControlDataInterface](#)

Specifies controls for an LSODE integrator.

Namespaces

- [jeod](#)

Namespace jeod.

9.33.1 Detailed Description

Define the class LsodeControlDataInterface.

9.34 Isode_data_classes.cc File Reference

Define member functions for the data-grouping classes specified in Isode_data_classes.

```
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "er7_utils/interface/include/alloc.hh"
#include "../include/lsode_data_classes.hh"
```

9.34.1 Detailed Description

Define member functions for the data-grouping classes specified in Isode_data_classes.

9.35 Isode_data_classes.hh File Reference

Define LSODE classes that contain just data members.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "lsode_control_data_interface.hh"
```

Data Structures

- class [jeod::LsodeDataJacobianPrep](#)
Data associated with the method DPREPJ.
- class [jeod::LsodeDataArrays](#)
The data arrays.
- class [jeod::LsodeDataStode](#)
The data associated with method Dstode.

Namespaces

- [jeod](#)

Namespace jeod.

9.35.1 Detailed Description

Define LSODE classes that contain just data members.

9.36 lside_first_order_ode_integrator.hh File Reference

Define the class LsodeFirstOrderODEIntegrator, the Jeod-compatible version of the Livermore ODE solver, LSODE.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/first_order_ode_integrator.hh"
#include "er7_utils/integration/core/include/integration_controls.hh"
#include "er7_utils/integration/core/include/integrator_result.hh"
#include "lsode_control_data_interface.hh"
#include "lsode_data_classes.hh"
#include "lsode_integration_controls.hh"
```

Data Structures

- class [jeod::LsodeFirstOrderODEIntegrator](#)
Jeod-compatible version of the Livermore ODE solver, LSODE.

Namespaces

- [jeod](#)
Namespace jeod.

9.36.1 Detailed Description

Define the class LsodeFirstOrderODEIntegrator, the Jeod-compatible version of the Livermore ODE solver, LSODE.

9.37 lside_first_order_ode_integrator__integrator.cc File Reference

Define member functions for the class LsodeFirstOrderODEIntegrator.

```
#include <cmath>
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/integration/core/include/integ_utils.hh"
#include "er7_utils/integration/core/include/integration_controls.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "utils/math/include/numerical.hh"
#include "../include/lsode_first_order_ode_integrator.hh"
```

9.37.1 Detailed Description

Define member functions for the class LsodeFirstOrderODEIntegrator.

9.38 Isode_first_order_ode_integrator__manager.cc File Reference

Define member functions for the class LsodeFirstOrderODEIntegrator.

```
#include <cmath>
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/integration/core/include/integ_utils.hh"
#include "er7_utils/integration/core/include/integration_controls.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "utils/math/include/numerical.hh"
#include "../include/lsode_first_order_ode_integrator.hh"
```

9.38.1 Detailed Description

Define member functions for the class LsodeFirstOrderODEIntegrator.

9.39 Isode_first_order_ode_integrator__support.cc File Reference

Define member functions for the class LsodeFirstOrderODEIntegrator.

```
#include <cmath>
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/integration/core/include/integ_utils.hh"
#include "er7_utils/integration/core/include/integration_controls.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "utils/math/include/numerical.hh"
#include "../include/lsode_first_order_ode_integrator.hh"
```

9.39.1 Detailed Description

Define member functions for the class LsodeFirstOrderODEIntegrator.

9.40 Isode_first_order_ode_integrator__utility.cc File Reference

Define member functions for the class LsodeFirstOrderODEIntegrator.

```
#include <cmath>
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/integration/core/include/integ_utils.hh"
#include "er7_utils/integration/core/include/integration_controls.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "../include/lsode_first_order_ode_integrator.hh"
```

9.40.1 Detailed Description

Define member functions for the class `LsodeFirstOrderODEIntegrator`.

9.41 `lsode_generalized_second_order_ode_integrator.cc` File Reference

Define member functions for the class `LsodeGeneralizedDerivSecondOrderODEIntegrator`.

```
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/integration/core/include/integ_utils.hh"
#include "er7_utils/integration/core/include/integration_controls.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "../include/lsode_control_data_interface.hh"
#include "../include/lsode_generalized_second_order_ode_integrator.hh"
```

9.41.1 Detailed Description

Define member functions for the class `LsodeGeneralizedDerivSecondOrderODEIntegrator`.

9.42 `lsode_generalized_second_order_ode_integrator.hh` File Reference

Define the class `LsodeGeneralizedDerivSecondOrderODEIntegrator`.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/second_order_ode_integrator.↵
hh"
#include "lsode_second_order_ode_integrator.hh"
```

Data Structures

- class [jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator](#)
JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.

Namespaces

- [jeod](#)
Namespace jeod.

9.42.1 Detailed Description

Define the class `LsodeGeneralizedDerivSecondOrderODEIntegrator`.

9.43 lsode_integration_controls.cc File Reference

Define the methods for the class LsodeIntegrationControls.

```
#include "er7_utils/integration/core/include/base_integration_group.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/integration/core/include/integrator_interface.hh"
#include "er7_utils/integration/core/include/time_interface.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "utils/math/include/numerical.hh"
#include "../include/lsode_integration_controls.hh"
```

9.43.1 Detailed Description

Define the methods for the class LsodeIntegrationControls.

9.44 lsode_integration_controls.hh File Reference

Define the class LsodeIntegrationControls.

```
#include "er7_utils/interface/include/alloc.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/generalized_position_derivative.↵
hh"
#include "er7_utils/integration/core/include/standard_integration_controls.↵
hh"
```

Data Structures

- class [jeod::LsodeIntegrationControls](#)
Contains controls for an LSODE integrator.

Namespaces

- [jeod](#)
Namespace jeod.

9.44.1 Detailed Description

Define the class LsodeIntegrationControls.

9.45 lsode_integrator_constructor.cc File Reference

Define the methods in the class LsodeIntegratorConstructor.

```
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/integration/core/include/integrator_constructor_utils.↵
hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "../include/lsode_first_order_ode_integrator.hh"
#include "../include/lsode_generalized_second_order_ode_integrator.hh"
#include "../include/lsode_integration_controls.hh"
#include "../include/lsode_integrator_constructor.hh"
#include "../include/lsode_second_order_ode_integrator.hh"
#include "../include/lsode_simple_second_order_ode_integrator.hh"
```

9.45.1 Detailed Description

Define the methods in the class LsodeIntegratorConstructor.

9.46 lsode_integrator_constructor.hh File Reference

Define the class LsodeIntegratorConstructor, the class that constructs the integration controls and the integrators for the LSODE method.

```
#include <vector>
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/integrator_constructor.hh"
#include "lsode_control_data_interface.hh"
#include "lsode_first_order_ode_integrator.hh"
#include "lsode_generalized_second_order_ode_integrator.hh"
#include "lsode_second_order_ode_integrator.hh"
#include "lsode_simple_second_order_ode_integrator.hh"
```

Data Structures

- class [jeod::LsodeIntegratorConstructor](#)
Create state and time integrators that propagate using standard Lsode.

Namespaces

- [jeod](#)
Namespace jeod.

9.46.1 Detailed Description

Define the class LsodeIntegratorConstructor, the class that constructs the integration controls and the integrators for the LSODE method.

9.47 `lsode_second_order_ode_integrator.cc` File Reference

Define member functions for the class `LsodeSecondOrderODEIntegrator`.

```
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/integration/core/include/integ_utils.hh"
#include "er7_utils/integration/core/include/integration_controls.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "../include/lsode_control_data_interface.hh"
#include "../include/lsode_second_order_ode_integrator.hh"
```

9.47.1 Detailed Description

Define member functions for the class `LsodeSecondOrderODEIntegrator`.

9.48 `lsode_second_order_ode_integrator.hh` File Reference

Define the class `LsodeSecondOrderODEIntegrator`.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/second_order_ode_integrator.↵
hh"
#include "lsode_first_order_ode_integrator.hh"
#include "lsode_integration_controls.hh"
```

Data Structures

- class [jeod::LsodeSecondOrderODEIntegrator](#)
JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.

Namespaces

- [jeod](#)
Namespace jeod.

9.48.1 Detailed Description

Define the class `LsodeSecondOrderODEIntegrator`.

9.49 lsode_simple_second_order_ode_integrator.cc File Reference

Define member functions for the class LsodeSimpleSecondOrderODEIntegrator.

```
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/integration/core/include/integration_messages.hh"
#include "er7_utils/interface/include/message_handler.hh"
#include "../include/lsode_control_data_interface.hh"
#include "../include/lsode_simple_second_order_ode_integrator.hh"
```

9.49.1 Detailed Description

Define member functions for the class LsodeSimpleSecondOrderODEIntegrator.

9.50 lsode_simple_second_order_ode_integrator.hh File Reference

Define the class LsodeSimpleSecondOrderODEIntegrator.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/second_order_ode_integrator.↵
hh"
#include "lsode_second_order_ode_integrator.hh"
```

Data Structures

- class [jeod::LsodeSimpleSecondOrderODEIntegrator](#)
JEOD-compatible version of the Livermore ODE solver, LSODE, capable of integrating second-order ODEs.

Namespaces

- [jeod](#)
Namespace jeod.

9.50.1 Detailed Description

Define the class LsodeSimpleSecondOrderODEIntegrator.

9.51 restartable_2d_second_order_integrator.hh File Reference

Defines the class Restartable2DSecondOrderODEIntegrator.

```
#include "generalized_second_order_ode_technique.hh"
#include "integration_messages.hh"
#include "restartable_state_integrator_templates.hh"
#include "utils/container/include/simple_checkpointable.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/first_order_ode_integrator.hh"
#include "er7_utils/integration/core/include/left_quaternion_functions.hh"
#include "er7_utils/integration/core/include/second_order_ode_integrator.↵
hh"
#include <cstdlib>
```

Data Structures

- class [jeod::Restartable2DSecondOrderIntegrator](#)

Integrates a second order ODE in two dimensional space, $d^2x/dt^2 = a(x,t)$, where x is a two-vector.

Namespaces

- [jeod](#)

Namespace jeod.

9.51.1 Detailed Description

Defines the class Restartable2DSecondOrderODEIntegrator.

9.52 restartable_state_integrator.hh File Reference

Define classes that encapsulate the construction, destruction, checkpointing, and restarting of state integrators.

```
#include "generalized_second_order_ode_technique.hh"
#include "integration_messages.hh"
#include "restartable_state_integrator_templates.hh"
#include "utils/container/include/simple_checkpointable.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "er7_utils/integration/core/include/first_order_ode_integrator.hh"
#include "er7_utils/integration/core/include/left_quaternion_functions.hh"
#include "er7_utils/integration/core/include/second_order_ode_integrator.↵
hh"
#include <cstdlib>
```

Data Structures

- class [jeod::RestartableScalarFirstOrderODEIntegrator](#)
A [RestartableScalarFirstOrderODEIntegrator](#) integrates a first order ODE, $dx/dt = v(x,t)$, where x is a scalar.
- class [jeod::RestartableT3SecondOrderODEIntegrator](#)
A [RestartableT3SecondOrderODEIntegrator](#) integrates a second order ODE in three space, $d^2x/dt^2 = a(x,t)$, where x is a three-vector.
- class [jeod::RestartableSO3SecondOrderODEIntegrator](#)
A [RestartableSO3SecondOrderODEIntegrator](#) integrates a generalized second order ODE that describes rotation in three space.

Namespaces

- [jeod](#)
Namespace *jeod*.

9.52.1 Detailed Description

Define classes that encapsulate the construction, destruction, checkpointing, and restarting of state integrators.

9.53 restartable_state_integrator_templates.hh File Reference

Define template classes that encapsulate the construction, destruction, checkpointing, and restarting of state integrators.

```
#include <cstdlib>
#include "er7_utils/integration/core/include/first_order_ode_integrator.hh"
#include "er7_utils/integration/core/include/integrator_constructor.hh"
#include "er7_utils/integration/core/include/second_order_ode_integrator.↵
hh"
#include "utils/container/include/simple_checkpointable.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "integration_messages.hh"
```

Data Structures

- class [jeod::RestartableStateIntegrator< IntegratorType >](#)
A [RestartableStateIntegrator](#) establishes the basic capabilities needed to make a state integrator a managed resource.
- class [jeod::RestartableFirstOrderODEIntegrator< size >](#)
A [RestartableFirstOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages an [er7_utils::FirstOrderODE↵Integrator](#).
- class [jeod::RestartableSecondOrderODEIntegrator](#)
A [RestartableSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a second order ODE problem.
- class [jeod::RestartableSimpleSecondOrderODEIntegrator< size >](#)

A [RestartableSimpleSecondOrderODEIntegrator](#) is-a [RestartableSecondOrderODEIntegrator](#) that manages the integrator for a simple second order ODE problem, one in which the time derivative of the generalized position is the generalized velocity.

- class [jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator](#)< position_size, velocity_size, DerivFunctions >

A [RestartableGeneralizedDerivSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a generalized second order ODE problem, one in which the time derivative of the generalized position is some function of the generalized position and the generalized velocity.

- class [jeod::RestartableGeneralizedStepSecondOrderODEIntegrator](#)< position_size, velocity_size, StepFunctions >

A [RestartableGeneralizedStepSecondOrderODEIntegrator](#) is-a [RestartableStateIntegrator](#) that manages the integrator for a generalized second order ODE problem, one in which the time derivative of the generalized position is some function of the generalized position and the generalized velocity.

Namespaces

- [jeod](#)

Namespace [jeod](#).

9.53.1 Detailed Description

Define template classes that encapsulate the construction, destruction, checkpointing, and restarting of state integrators.

9.54 time_change_subscriber.hh File Reference

Define the class TimeChangeSubscriber.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class [jeod::TimeChangeSubscriber](#)

A [TimeChangeSubscriber](#) is some object that wants to be notified of changes in the nature of time.

Namespaces

- [jeod](#)

Namespace [jeod](#).

9.54.1 Detailed Description

Define the class TimeChangeSubscriber.

9.55 two_d_array.hh File Reference

Defines the template class [er7_utils::TwoDArray](#), which implements an RAI rectangular 2D array.

```
#include "er7_utils/interface/include/alloc.hh"
#include "er7_utils/interface/include/er7_class.hh"
#include "utils/sim_interface/include/config.hh"
#include <algorithm>
#include <cstdlib>
#include <cstring>
#include <limits>
#include <stdexcept>
```

Data Structures

- class [er7_utils::TwoDArray< T >](#)
RAII template class that implements a rectangular two dimensional array.
- class [er7_utils::DoubleTwoDArray](#)
2D array, specialized for doubles.

Namespaces

- [er7_utils](#)

9.55.1 Detailed Description

Defines the template class [er7_utils::TwoDArray](#), which implements an RAI rectangular 2D array.

Index

- ~GaussJacksonCoefficientsPair
 - jeod::GaussJacksonCoefficientsPair, [65](#)
- ~GaussJacksonCoeffs
 - jeod::GaussJacksonCoeffs, [71](#)
- ~GaussJacksonFirstOrderODEIntegrator
 - jeod::GaussJacksonFirstOrderODEIntegrator, [81](#)
- ~GaussJacksonGeneralizedDerivSecondOrderODE↔Integrator
 - jeod::GaussJacksonGeneralizedDerivSecond↔OrderODEIntegrator, [85](#)
- ~GaussJacksonIntegrationControls
 - jeod::GaussJacksonIntegrationControls, [92](#)
- ~GaussJacksonIntegratorBase
 - jeod::GaussJacksonIntegratorBase, [107](#)
- ~GaussJacksonIntegratorConstructor
 - jeod::GaussJacksonIntegratorConstructor, [136](#)
- ~GaussJacksonSimpleSecondOrderODEIntegrator
 - jeod::GaussJacksonSimpleSecondOrderODE↔Integrator, [150](#)
- ~GeneralizedSecondOrderODETechnique
 - jeod::GeneralizedSecondOrderODETechnique, [172](#)
- ~JeodIntegrationGroup
 - jeod::JeodIntegrationGroup, [180](#)
- ~JeodIntegrationGroupOwner
 - jeod::JeodIntegrationGroupOwner, [187](#)
- ~JeodIntegrationTime
 - jeod::JeodIntegrationTime, [189](#)
- ~LsodeControlDataInterface
 - jeod::LsodeControlDataInterface, [194](#)
- ~LsodeDataArrays
 - jeod::LsodeDataArrays, [202](#)
- ~LsodeDataJacobianPrep
 - jeod::LsodeDataJacobianPrep, [207](#)
- ~LsodeDataStode
 - jeod::LsodeDataStode, [210](#)
- ~LsodeFirstOrderODEIntegrator
 - Lsode, [54](#)
- ~LsodeGeneralizedDerivSecondOrderODEIntegrator
 - Lsode, [54](#)
- ~LsodeIntegrationControls
 - jeod::LsodeIntegrationControls, [238](#)
- ~LsodeSecondOrderODEIntegrator
 - Lsode, [55](#)
- ~LsodeSimpleSecondOrderODEIntegrator
 - jeod::LsodeSimpleSecondOrderODEIntegrator, [248](#)
- ~Restartable2DSecondOrderIntegrator
 - jeod::Restartable2DSecondOrderIntegrator, [250](#)
- ~RestartableFirstOrderODEIntegrator
 - jeod::RestartableFirstOrderODEIntegrator, [254](#)
- ~RestartableGeneralizedDerivSecondOrderODE↔Integrator
 - jeod::RestartableGeneralizedDerivSecondOrder↔ODEIntegrator, [257](#)
- ~RestartableGeneralizedStepSecondOrderODE↔Integrator
 - jeod::RestartableGeneralizedStepSecondOrder↔ODEIntegrator, [261](#)
- ~RestartableSO3SecondOrderODEIntegrator
 - jeod::RestartableSO3SecondOrderODEIntegrator, [273](#)
- ~RestartableScalarFirstOrderODEIntegrator
 - jeod::RestartableScalarFirstOrderODEIntegrator, [264](#)
- ~RestartableSecondOrderODEIntegrator
 - jeod::RestartableSecondOrderODEIntegrator, [268](#)
- ~RestartableSimpleSecondOrderODEIntegrator
 - jeod::RestartableSimpleSecondOrderODE↔Integrator, [270](#)
- ~RestartableStateIntegrator
 - jeod::RestartableStateIntegrator, [280](#)
- ~RestartableT3SecondOrderODEIntegrator
 - jeod::RestartableT3SecondOrderODEIntegrator, [287](#)
- ~TimeChangeSubscriber
 - jeod::TimeChangeSubscriber, [291](#)
- ~TwoDArray
 - er7_utils::TwoDArray, [295](#)
- abs_tolerance_error_control
 - jeod::LsodeControlDataInterface, [195](#)
- abs_tolerance_error_control_vec
 - jeod::LsodeControlDataInterface, [196](#)
- absolute_tolerance
 - jeod::GaussJacksonConfig, [77](#)
 - jeod::GaussJacksonIntegratorBase, [128](#)
- acc_hist
 - jeod::GaussJacksonIntegratorBase, [128](#)
- accum_correction
 - jeod::LsodeDataArrays, [203](#)
- add_integrable_object
 - jeod::JeodIntegrationGroup, [181](#)
- add_time_change_subscriber
 - jeod::JeodIntegrationTime, [189](#)
- advance_edit_integration_constants
 - jeod::GaussJacksonIntegratorBase, [107](#)
- advance_predictor_integration_constants
 - jeod::GaussJacksonIntegratorBase, [108](#)

- allocate
 - er7_utils::TwoDArray, 295
- allocate_arrays
 - jeod::GaussJacksonCoefficientsPair, 65
 - Lsode, 24
- allocate_internal
 - er7_utils::TwoDArray, 295
- allocate_state_contents
 - jeod::GaussJacksonIntegratorBase, 108, 109
- allocated
 - jeod::LsodeDataArrays, 203
- apply
 - jeod::GaussJacksonCoefficientsPair, 66
- arrays
 - jeod::LsodeFirstOrderODEIntegrator, 220
- arrays_allocated
 - jeod::LsodeSecondOrderODEIntegrator, 245
- at
 - er7_utils::TwoDArray, 296, 297
- at_downsample
 - jeod::GaussJacksonStateMachine, 163
- at_end_of_tour
 - jeod::GaussJacksonIntegrationControls, 97
 - jeod::GaussJacksonStateMachine, 163
- at_order_change
 - jeod::GaussJacksonStateMachine, 163
- at_reinitialize
 - jeod::GaussJacksonStateMachine, 164
- base_integrate
 - jeod::GaussJacksonIntegratorBase, 109
- base_reset
 - jeod::GaussJacksonIntegratorBase, 110
- bootstrap_edit_redo_needed
 - jeod::GaussJacksonStateMachine, 164
- calculate_epsilon
 - Lsode, 25
- calculate_integration_coefficients
 - Lsode, 25
- calculation_task
 - jeod::LsodeFirstOrderODEIntegrator, 220
- CalculationTask
 - jeod::LsodeFirstOrderODEIntegrator, 218
- cast_to_gj_controls
 - jeod, 60
- check_interface_data
 - Lsode, 26
- clear_integrator_reference
 - jeod::RestartableStateIntegrator, 281
- coeff
 - jeod::GaussJacksonIntegrationControls, 97
 - jeod::GaussJacksonIntegratorBase, 128
- coefficients
 - jeod::GaussJacksonRationalCoefficients, 148
- compute_coeffs
 - jeod::GaussJacksonCoeffs, 71
- config
 - jeod::GaussJacksonIntegrationControls, 97
 - jeod::GaussJacksonIntegratorConstructor, 141
- configure
 - jeod::GaussJacksonCoefficientsPair, 67
 - jeod::GaussJacksonCoeffs, 72
 - jeod::GaussJacksonIntegratorConstructor, 136
 - jeod::GaussJacksonStateMachine, 158
- configure_adams_corrector
 - jeod::GaussJacksonRationalCoefficients, 145
- construct_predictor
 - jeod::GaussJacksonRationalCoefficients, 145
- construct_stormer_cowell_corrector
 - jeod::GaussJacksonRationalCoefficients, 146
- control_data
 - jeod::LsodeFirstOrderODEIntegrator, 220
- convergence_factor
 - jeod::LsodeFirstOrderODEIntegrator, 221
- convergence_jacobian_flag
 - jeod::LsodeFirstOrderODEIntegrator, 221
- convergence_rate
 - jeod::LsodeFirstOrderODEIntegrator, 221
- convert_to_ordinate_form
 - jeod::GaussJacksonRationalCoefficients, 146
- correct
 - jeod::GaussJacksonIntegratorBase, 110, 111
- correction_iterations
 - jeod::GaussJacksonStateMachine, 164
- corrector
 - jeod::GaussJacksonCoeffs, 74
- corrector_method
 - jeod::LsodeControlDataInterface, 196
- corrector_sum
 - jeod::GaussJacksonIntegratorBase, 129
- CorrectorMethod
 - jeod::LsodeControlDataInterface, 193
- create_constructor
 - jeod::GaussJacksonIntegratorConstructor, 137
 - Lsode, 26
- create_copy
 - jeod::GaussJacksonFirstOrderODEIntegrator, 81
 - jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, 86
 - jeod::GaussJacksonIntegrationControls, 92
 - jeod::GaussJacksonIntegratorConstructor, 137
 - jeod::GaussJacksonSimpleSecondOrderODEIntegrator, 151
 - Lsode, 26, 27
- create_first_order_ode_integrator
 - jeod::GaussJacksonIntegratorConstructor, 137
 - Lsode, 28
- create_generalized_deriv_second_order_ode_integrator
 - jeod::GaussJacksonIntegratorConstructor, 138
 - Lsode, 28
- create_integration_controls
 - jeod::GaussJacksonIntegratorConstructor, 138
 - Lsode, 29
- create_integrator
 - jeod::Restartable2DSecondOrderIntegrator, 250

- jeod::RestartableSO3SecondOrderODEIntegrator, 274
- jeod::RestartableScalarFirstOrderODEIntegrator, 265
- jeod::RestartableStateIntegrator, 281
- jeod::RestartableT3SecondOrderODEIntegrator, 287
- create_integrator_internal
 - jeod::RestartableFirstOrderODEIntegrator, 255
 - jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator, 258
 - jeod::RestartableGeneralizedStepSecondOrderODEIntegrator, 262
 - jeod::RestartableSimpleSecondOrderODEIntegrator, 271
 - jeod::RestartableStateIntegrator, 283
- create_primer
 - jeod::GaussJacksonIntegratorBase, 111, 112
- create_second_order_ode_integrator
 - jeod::GaussJacksonIntegratorConstructor, 139
 - Lsode, 29
- current_order
 - jeod::GaussJacksonStateMachine, 164
- cycle_dyndt
 - jeod::GaussJacksonIntegrationControls, 98
- cycle_scale
 - jeod::GaussJacksonStateMachine, 165
- cycle_simdt
 - jeod::GaussJacksonIntegrationControls, 98
- cycle_start_time
 - jeod::GaussJacksonStateMachine, 165
- cycle_starttime
 - jeod::GaussJacksonIntegrationControls, 98
- cycle_target_time
 - jeod::LsodeFirstOrderODEIntegrator, 222
- data_array
 - er7_utils::TwoDArray, 303
- data_interface
 - jeod::LsodeIntegratorConstructor, 242
- data_prepj
 - jeod::LsodeFirstOrderODEIntegrator, 222
- data_stode
 - jeod::LsodeFirstOrderODEIntegrator, 222
- deallocate_arrays
 - jeod::GaussJacksonCoefficientsPair, 67
- deallocate_internal
 - er7_utils::TwoDArray, 298
- deallocate_state_contents
 - jeod::GaussJacksonIntegratorBase, 112, 113
- default_configuration
 - jeod::GaussJacksonConfig, 76
- delinv
 - jeod::GaussJacksonIntegratorBase, 129
- destroy_allocated_arrays
 - Lsode, 30
- destroy_integrator
 - jeod::Restartable2DSecondOrderIntegrator, 251
- jeod::RestartableSO3SecondOrderODEIntegrator, 274
- jeod::RestartableScalarFirstOrderODEIntegrator, 265
- jeod::RestartableStateIntegrator, 283
- jeod::RestartableT3SecondOrderODEIntegrator, 287
- discard_extra_terms
 - jeod::GaussJacksonRationalCoefficients, 147
- displace_back
 - jeod::GaussJacksonRationalCoefficients, 147
- downsample
 - er7_utils::TwoDArray, 298
- downsample_hist
 - jeod::GaussJacksonIntegratorBase, 113
- dsm
 - jeod::LsodeDataStode, 211
- edit_count
 - jeod::GaussJacksonIntegrationControls, 98
- edit_point
 - jeod::GaussJacksonIntegratorBase, 114
- EntryPoint
 - jeod::LsodeFirstOrderODEIntegrator, 219
- epsilon
 - jeod::LsodeFirstOrderODEIntegrator, 222
- er7_utils, 57
- er7_utils::DoubleTwoDArray, 63
 - init_atr7_utils__DoubleTwoDArray, 63
 - InputProcessor, 64
- er7_utils::TwoDArray
 - ~TwoDArray, 295
 - allocate, 295
 - allocate_internal, 295
 - at, 296, 297
 - data_array, 303
 - deallocate_internal, 298
 - downsample, 298
 - m, 303
 - n, 303
 - operator const T *const *, 298
 - operator T*const *, 299
 - operator(), 299, 300
 - operator=, 300
 - operator[], 300, 301
 - rotate_down, 301
 - rotate_up, 302
 - row_array, 304
 - swap, 302
 - TwoDArray, 294
- er7_utils::TwoDArray< T >, 292
- error_control_indicator
 - jeod::LsodeControlDataInterface, 196
- error_control_vector_copied_over
 - jeod::LsodeControlDataInterface, 196
- error_weight
 - jeod::LsodeDataArrays, 203
- ErrorControlIndicator
 - jeod::LsodeControlDataInterface, 194

- exit_bootstrap_edit
 - jeod::GaussJacksonStateMachine, 158
- exit_bootstrap_step
 - jeod::GaussJacksonStateMachine, 158
- exit_priming
 - jeod::GaussJacksonStateMachine, 158
- fac
 - jeod::LsodeDataJacobianPrep, 208
- final_order
 - jeod::GaussJacksonConfig, 78
 - jeod::GaussJacksonStateMachine, 165
- first
 - jeod::GaussJacksonOneState, 143
 - jeod::GaussJacksonTwoState, 170
- first_derivative_size
 - jeod::LsodeSecondOrderODEIntegrator, 245
- first_order_integrator
 - jeod::LsodeSecondOrderODEIntegrator, 245
- first_pass
 - jeod::LsodeFirstOrderODEIntegrator, 223
- fsm_state
 - jeod::GaussJacksonIntegrationControls, 99
 - jeod::GaussJacksonIntegratorBase, 129
 - jeod::GaussJacksonStateMachine, 165
- FsmState
 - jeod::GaussJacksonStateMachine, 157
- gauss_elim_factor
 - Lsode, 30
- gauss_jackson_coefficients_pair.cc, 305
- gauss_jackson_coefficients_pair.hh, 305
- gauss_jackson_coeffs.cc, 306
- gauss_jackson_coeffs.hh, 307
- gauss_jackson_config.cc, 307
- gauss_jackson_config.hh, 308
- gauss_jackson_first_order_ode_integrator.hh, 308
- gauss_jackson_generalized_second_order_ode_↔
 - integrator.cc, 309
- gauss_jackson_generalized_second_order_ode_↔
 - integrator.hh, 309
- gauss_jackson_integration_controls.cc, 310
- gauss_jackson_integration_controls.hh, 310
- gauss_jackson_integrator_base.hh, 311
- gauss_jackson_integrator_base_first.hh, 312
- gauss_jackson_integrator_base_second.hh, 312
- gauss_jackson_integrator_constructor.cc, 313
- gauss_jackson_integrator_constructor.hh, 314
- gauss_jackson_one_state.hh, 314
- gauss_jackson_rational_coeffs.cc, 315
- gauss_jackson_rational_coeffs.hh, 315
- gauss_jackson_simple_second_order_ode_integrator.↔
 - hh, 316
- gauss_jackson_state_machine.cc, 316
- gauss_jackson_state_machine.hh, 317
- gauss_jackson_two_state.hh, 317
- GaussJackson, 18
- GaussJacksonCoefficientsPair
 - jeod::GaussJacksonCoefficientsPair, 65
- GaussJacksonCoeffs
 - jeod::GaussJacksonCoeffs, 70, 71
- GaussJacksonConfig
 - jeod::GaussJacksonConfig, 76
- GaussJacksonFirstOrderODEIntegrator
 - jeod::GaussJacksonFirstOrderODEIntegrator, 81
- GaussJacksonGeneralizedDerivSecondOrderODE↔
 - Integrator
 - jeod::GaussJacksonGeneralizedDerivSecond↔
 - OrderODEIntegrator, 85
- GaussJacksonIntegrationControls
 - jeod::GaussJacksonIntegrationControls, 91, 92
- GaussJacksonIntegratorBase
 - jeod::GaussJacksonIntegratorBase, 105, 106
- GaussJacksonIntegratorBaseFirst
 - jeod, 59
- GaussJacksonIntegratorBaseSecond
 - jeod, 60
- GaussJacksonIntegratorConstructor
 - jeod::GaussJacksonIntegratorConstructor, 136
- GaussJacksonOneState
 - jeod::GaussJacksonOneState, 143
- GaussJacksonRationalCoefficients
 - jeod::GaussJacksonRationalCoefficients, 145
- GaussJacksonSimpleSecondOrderODEIntegrator
 - jeod::GaussJacksonSimpleSecondOrderODE↔
 - Integrator, 149, 150
- GaussJacksonStateMachine
 - jeod::GaussJacksonStateMachine, 157
- GaussJacksonTwoState
 - jeod::GaussJacksonTwoState, 169
- generalized_deriv_integrator_manager
 - jeod::RestartableSO3SecondOrderODEIntegrator, 276
- generalized_second_order_ode_technique.cc, 318
- generalized_second_order_ode_technique.hh, 318
- generalized_step_integrator_manager
 - jeod::RestartableSO3SecondOrderODEIntegrator, 276
- GeneralizedSecondOrderODETechnique
 - jeod::GeneralizedSecondOrderODETechnique, 172
- get_at_downsample
 - jeod::GaussJacksonStateMachine, 159
- get_at_end_of_tour
 - jeod::GaussJacksonStateMachine, 159
- get_at_order_change
 - jeod::GaussJacksonStateMachine, 159
- get_at_reinitialize
 - jeod::GaussJacksonStateMachine, 159
- get_buffer_size
 - jeod::GaussJacksonIntegratorConstructor, 139
- get_class_name
 - jeod::GaussJacksonIntegratorConstructor, 139
 - jeod::LsodeIntegratorConstructor, 240
- get_coeff
 - jeod::GaussJacksonIntegrationControls, 92
- get_config

- jeod::GaussJacksonIntegrationControls, 93
- get_current_order
 - jeod::GaussJacksonStateMachine, 160
- get_cycle_scale
 - jeod::GaussJacksonStateMachine, 160
- get_cycle_start_time
 - jeod::GaussJacksonStateMachine, 160
- get_fsm_state
 - jeod::GaussJacksonStateMachine, 160
- get_history_length
 - jeod::GaussJacksonStateMachine, 161
- get_max_history_size
 - jeod::GaussJacksonStateMachine, 161
- get_priming_controls
 - jeod::GaussJacksonIntegrationControls, 93
- get_re_entry_point
 - jeod::LsodeFirstOrderODEIntegrator, 219
 - jeod::LsodeSecondOrderODEIntegrator, 244
- get_state_machine
 - jeod::GaussJacksonIntegrationControls, 93
- get_timestamp_time
 - jeod::JeodIntegrationTime, 190
- get_transition_table_size
 - jeod::GaussJacksonIntegratorConstructor, 140
 - jeod::LsodeIntegratorConstructor, 241
- gj_coefs
 - jeod::GaussJacksonCoefficientsPair, 69
- group_owner
 - jeod::JeodIntegrationGroup, 185
- history
 - jeod::LsodeDataArrays, 204
- history_length
 - jeod::GaussJacksonIntegratorBase, 130
 - jeod::GaussJacksonStateMachine, 166
- history_size
 - jeod::GaussJacksonStateMachine, 166
- hl0
 - jeod::LsodeDataJacobianPrep, 208
- implements
 - jeod::GaussJacksonIntegratorConstructor, 140
 - jeod::LsodeIntegratorConstructor, 241
- index
 - jeod::LsodeDataJacobianPrep, 208
- index_max
 - jeod::LsodeDataJacobianPrep, 209
- index_of_max_magnitude
 - Lsode, 30
- information
 - jeod::IntegrationMessages, 176
- init_attrer7_utils__DoubleTwoDArray
 - er7_utils::DoubleTwoDArray, 63
- init_attrjeod__GaussJacksonCoefficientsPair
 - jeod::GaussJacksonCoefficientsPair, 68
- init_attrjeod__GaussJacksonCoeffs
 - jeod::GaussJacksonCoeffs, 73
- init_attrjeod__GaussJacksonConfig
 - jeod::GaussJacksonConfig, 77
- init_attrjeod__GaussJacksonGeneralizedDeriv↔
 - SecondOrderODEIntegrator
 - jeod::GaussJacksonGeneralizedDerivSecond↔
 - OrderODEIntegrator, 87
- init_attrjeod__GaussJacksonIntegrationControls
 - jeod::GaussJacksonIntegrationControls, 96
- init_attrjeod__GaussJacksonIntegratorConstructor
 - jeod::GaussJacksonIntegratorConstructor, 141
- init_attrjeod__GaussJacksonOneState
 - jeod::GaussJacksonOneState, 143
- init_attrjeod__GaussJacksonRationalCoefficients
 - jeod::GaussJacksonRationalCoefficients, 147
- init_attrjeod__GaussJacksonSimpleSecondOrderOD↔
 - EIntegrator
 - jeod::GaussJacksonSimpleSecondOrderODE↔
 - Integrator, 154
- init_attrjeod__GaussJacksonStateMachine
 - jeod::GaussJacksonStateMachine, 163
- init_attrjeod__GaussJacksonTwoState
 - jeod::GaussJacksonTwoState, 170
- init_attrjeod__IntegrationMessages
 - jeod::IntegrationMessages, 176
- init_attrjeod__JeodIntegrationGroup
 - jeod::JeodIntegrationGroup, 185
- init_attrjeod__JeodIntegrationTime
 - jeod::JeodIntegrationTime, 191
- init_attrjeod__LsodeControlDataInterface
 - jeod::LsodeControlDataInterface, 195
- init_attrjeod__LsodeDataArrays
 - jeod::LsodeDataArrays, 203
- init_attrjeod__LsodeDataJacobianPrep
 - jeod::LsodeDataJacobianPrep, 208
- init_attrjeod__LsodeDataStode
 - jeod::LsodeDataStode, 211
- init_attrjeod__LsodeFirstOrderODEIntegrator
 - jeod::LsodeFirstOrderODEIntegrator, 220
- init_attrjeod__LsodeGeneralizedDerivSecondOrderO↔
 - DEIntegrator
 - jeod::LsodeGeneralizedDerivSecondOrderODE↔
 - Integrator, 236
- init_attrjeod__LsodeIntegrationControls
 - jeod::LsodeIntegrationControls, 238
- init_attrjeod__LsodeIntegratorConstructor
 - jeod::LsodeIntegratorConstructor, 242
- init_attrjeod__LsodeSecondOrderODEIntegrator
 - jeod::LsodeSecondOrderODEIntegrator, 244
- init_attrjeod__LsodeSimpleSecondOrderODEIntegrator
 - jeod::LsodeSimpleSecondOrderODEIntegrator, 248
- init_attrjeod__Restartable2DSecondOrderIntegrator
 - jeod::Restartable2DSecondOrderIntegrator, 252
- init_attrjeod__RestartableSO3SecondOrderODE↔
 - Integrator
 - jeod::RestartableSO3SecondOrderODEIntegrator, 276
- init_attrjeod__RestartableScalarFirstOrderODE↔
 - Integrator
 - jeod::RestartableScalarFirstOrderODEIntegrator,

- 266
- init_attrjeod__RestartableT3SecondOrderODE↵
 - Integrator
 - jeod::RestartableT3SecondOrderODEIntegrator, 289
- init_attrjeod__TimeChangeSubscriber
 - jeod::TimeChangeSubscriber, 292
- init_state
 - jeod::GaussJacksonIntegratorBase, 130
- initial_order
 - jeod::GaussJacksonConfig, 78
 - jeod::GaussJacksonIntegrationControls, 99
 - jeod::GaussJacksonIntegratorBase, 130
 - jeod::GaussJacksonStateMachine, 166
- initial_step_size
 - jeod::LsodeControlDataInterface, 197
- initialize_edit_integration_constants
 - jeod::GaussJacksonIntegratorBase, 114, 115
- initialize_group
 - jeod::JeodIntegrationGroup, 181
- initialize_predictor_integration_constants
 - jeod::GaussJacksonIntegratorBase, 115
- initialized
 - jeod::LsodeFirstOrderODEIntegrator, 223
- InputProcessor
 - er7_utils::DoubleTwoDArray, 64
 - jeod::GaussJacksonCoefficientsPair, 68
 - jeod::GaussJacksonCoeffs, 73
 - jeod::GaussJacksonConfig, 77
 - jeod::GaussJacksonGeneralizedDerivSecond↵
 - OrderODEIntegrator, 87
 - jeod::GaussJacksonIntegrationControls, 97
 - jeod::GaussJacksonIntegratorConstructor, 141
 - jeod::GaussJacksonOneState, 143
 - jeod::GaussJacksonRationalCoefficients, 148
 - jeod::GaussJacksonSimpleSecondOrderODE↵
 - Integrator, 154
 - jeod::GaussJacksonStateMachine, 163
 - jeod::GaussJacksonTwoState, 170
 - jeod::IntegrationMessages, 176
 - jeod::JeodIntegrationGroup, 185
 - jeod::JeodIntegrationTime, 191
 - jeod::LsodeControlDataInterface, 195
 - jeod::LsodeDataArrays, 203
 - jeod::LsodeDataJacobianPrep, 208
 - jeod::LsodeDataStode, 211
 - jeod::LsodeFirstOrderODEIntegrator, 220
 - jeod::LsodeGeneralizedDerivSecondOrderODE↵
 - Integrator, 236
 - jeod::LsodeIntegrationControls, 239
 - jeod::LsodeIntegratorConstructor, 242
 - jeod::LsodeSecondOrderODEIntegrator, 245
 - jeod::LsodeSimpleSecondOrderODEIntegrator, 248
 - jeod::Restartable2DSecondOrderIntegrator, 252
 - jeod::RestartableSO3SecondOrderODEIntegrator, 276
- jeod::RestartableScalarFirstOrderODEIntegrator, 267
- jeod::RestartableT3SecondOrderODEIntegrator, 289
- jeod::TimeChangeSubscriber, 292
- integ_merger
 - jeod::JeodIntegrationGroup, 185
- integrable_objects
 - jeod::JeodIntegrationGroup, 186
- integrate
 - jeod::GaussJacksonFirstOrderODEIntegrator, 82
 - jeod::GaussJacksonGeneralizedDerivSecond↵
 - OrderODEIntegrator, 86
 - jeod::GaussJacksonIntegrationControls, 94
 - jeod::GaussJacksonSimpleSecondOrderODE↵
 - Integrator, 151
 - jeod::Restartable2DSecondOrderIntegrator, 251
 - jeod::RestartableSO3SecondOrderODEIntegrator, 274
 - jeod::RestartableScalarFirstOrderODEIntegrator, 265
 - jeod::RestartableT3SecondOrderODEIntegrator, 288
 - Lsode, 31, 32
- integrate_bodies
 - jeod::JeodIntegrationGroup, 181
- integrate_container
 - jeod::JeodIntegrationGroup, 182
- integrate_edit
 - jeod::GaussJacksonIntegrationControls, 94
- integrate_gj
 - jeod::GaussJacksonIntegrationControls, 95
 - jeod::GaussJacksonIntegratorBase, 116
- integrate_primer
 - jeod::GaussJacksonIntegratorBase, 117
- Integration, 17
- integration_messages.cc, 319
 - MAKE_INTEGRATION_MESSAGE_CODE, 319
- integration_messages.hh, 319
- integration_method
 - jeod::LsodeControlDataInterface, 197
- IntegrationMessages
 - jeod::IntegrationMessages, 175
- IntegrationMethod
 - jeod::LsodeControlDataInterface, 194
- integrator
 - jeod::Restartable2DSecondOrderIntegrator, 252
 - jeod::RestartableSO3SecondOrderODEIntegrator, 277
 - jeod::RestartableScalarFirstOrderODEIntegrator, 267
 - jeod::RestartableT3SecondOrderODEIntegrator, 289
- integrator_compute_new_order
 - Lsode, 33
- integrator_compute_new_order_check_step_error
 - Lsode, 34
- integrator_compute_new_order_prep

- Lsode, [34](#)
 - integrator_core
 - Lsode, [34](#)
 - integrator_corrector_converged
 - Lsode, [35](#)
 - integrator_corrector_failed_part1
 - Lsode, [36](#)
 - integrator_corrector_failed_part2
 - Lsode, [36](#)
 - integrator_corrector_iteration
 - Lsode, [37](#)
 - integrator_error_test_failed
 - Lsode, [37](#)
 - integrator_fail_reset_order_1_part1
 - Lsode, [38](#)
 - integrator_fail_reset_order_1_part2
 - Lsode, [38](#)
 - integrator_handle
 - jeod::RestartableStateIntegrator, [285](#)
 - integrator_manager
 - jeod::Restartable2DSecondOrderIntegrator, [253](#)
 - jeod::RestartableScalarFirstOrderODEIntegrator, [267](#)
 - jeod::RestartableT3SecondOrderODEIntegrator, [290](#)
 - integrator_predict
 - Lsode, [39](#)
 - integrator_reset_iteration_loop_part1
 - Lsode, [39](#)
 - integrator_reset_iteration_loop_part2
 - Lsode, [40](#)
 - integrator_reset_method_coeffs
 - Lsode, [40](#)
 - integrator_reset_yh
 - Lsode, [40](#)
 - integrator_set_new_order
 - Lsode, [41](#)
 - integrator_terminate
 - Lsode, [41](#)
 - integrator_test_stepsize_change
 - Lsode, [41](#)
 - integrator_wrapup
 - Lsode, [42](#)
 - internal_error
 - jeod::IntegrationMessages, [176](#)
 - internal_state
 - jeod::LsodeFirstOrderODEIntegrator, [223](#)
 - interpolate_y
 - Lsode, [42](#)
 - invalid_item
 - jeod::IntegrationMessages, [177](#)
 - invalid_request
 - jeod::IntegrationMessages, [177](#)
 - iredo
 - jeod::LsodeDataStode, [211](#)
 - iret
 - jeod::LsodeDataStode, [212](#)
 - is_corrector_method_functional_iteration
 - jeod::LsodeControlDataInterface, [195](#)
 - is_provided_by
 - jeod::GeneralizedSecondOrderODETechnique, [173](#)
 - iter_delta
 - jeod::LsodeFirstOrderODEIntegrator, [223](#)
 - iteration_count
 - jeod::LsodeFirstOrderODEIntegrator, [224](#)
 - iteration_matrix_singular
 - jeod::LsodeFirstOrderODEIntegrator, [224](#)
 - jacobian_current
 - jeod::LsodeFirstOrderODEIntegrator, [224](#)
 - jacobian_prep_init
 - Lsode, [43](#)
 - jacobian_prep_loop
 - Lsode, [44](#)
 - jacobian_prep_wrap_up
 - Lsode, [44](#)
 - jeod, [57](#)
 - cast_to_gj_controls, [60](#)
 - GaussJacksonIntegratorBaseFirst, [59](#)
 - GaussJacksonIntegratorBaseSecond, [60](#)
 - operator<<, [60](#)
 - set_default_config_values, [62](#)
 - validate_config, [62](#)
 - jeod::GaussJacksonCoefficientsPair, [64](#)
 - ~GaussJacksonCoefficientsPair, [65](#)
 - allocate_arrays, [65](#)
 - apply, [66](#)
 - configure, [67](#)
 - deallocate_arrays, [67](#)
 - GaussJacksonCoefficientsPair, [65](#)
 - gj_coefs, [69](#)
 - init_attrjeod__GaussJacksonCoefficientsPair, [68](#)
 - InputProcessor, [68](#)
 - operator=, [67](#)
 - printOut, [67](#)
 - sa_coefs, [69](#)
 - swap, [68](#)
- jeod::GaussJacksonCoeffs, [69](#)
 - ~GaussJacksonCoeffs, [71](#)
- compute_coefs, [71](#)
- configure, [72](#)
- corrector, [74](#)
- GaussJacksonCoeffs, [70](#), [71](#)
- init_attrjeod__GaussJacksonCoeffs, [73](#)
- InputProcessor, [73](#)
- max_order, [74](#)
- operator<<, [73](#)
- operator=, [72](#)
- order, [74](#)
- predictor, [74](#)
- swap, [73](#)
- jeod::GaussJacksonConfig, [75](#)
 - absolute_tolerance, [77](#)
 - default_configuration, [76](#)
 - final_order, [78](#)
 - GaussJacksonConfig, [76](#)

- init_attrjeod__GaussJacksonConfig, 77
- initial_order, 78
- InputProcessor, 77
- max_correction_iterations, 78
- ndoubling_steps, 78
- priming_technique, 79
- relative_tolerance, 79
- standard_configuration, 76
- validate_configuration, 77
- jeod::GaussJacksonFirstOrderODEIntegrator, 80
 - ~GaussJacksonFirstOrderODEIntegrator, 81
 - create_copy, 81
 - GaussJacksonFirstOrderODEIntegrator, 81
 - integrate, 82
 - operator=, 82
 - reset_integrator, 82
 - swap, 82, 83
- jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, 83
 - ~GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, 85
 - create_copy, 86
 - GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, 85
 - init_attrjeod__GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, 87
 - InputProcessor, 87
 - integrate, 86
 - operator=, 86
 - pos_integrator, 87
 - posdot, 88
 - posdotdot, 88
 - reset_integrator, 86
 - swap, 87
 - vel_integrator, 88
- jeod::GaussJacksonIntegrationControls, 89
 - ~GaussJacksonIntegrationControls, 92
 - at_end_of_tour, 97
 - coeff, 97
 - config, 97
 - create_copy, 92
 - cycle_dyndt, 98
 - cycle_simdt, 98
 - cycle_starttime, 98
 - edit_count, 98
 - fsm_state, 99
 - GaussJacksonIntegrationControls, 91, 92
 - get_coeff, 92
 - get_config, 93
 - get_priming_controls, 93
 - get_state_machine, 93
 - init_attrjeod__GaussJacksonIntegrationControls, 96
 - initial_order, 99
 - InputProcessor, 97
 - integrate, 94
 - integrate_edit, 94
 - integrate_gj, 95
 - max_correction_iterations, 99
 - operator=, 95
 - order, 99
 - priming_controls, 100
 - reset_integrator, 95
 - reset_time, 100
 - start_cycle, 96
 - state_machine, 100
 - swap, 96
- jeod::GaussJacksonIntegratorBase
 - ~GaussJacksonIntegratorBase, 107
 - absolute_tolerance, 128
 - acc_hist, 128
 - advance_edit_integration_constants, 107
 - advance_predictor_integration_constants, 108
 - allocate_state_contents, 108, 109
 - base_integrate, 109
 - base_reset, 110
 - coeff, 128
 - correct, 110, 111
 - corrector_sum, 129
 - create_primer, 111, 112
 - deallocate_state_contents, 112, 113
 - delinv, 129
 - downsample_hist, 113
 - edit_point, 114
 - fsm_state, 129
 - GaussJacksonIntegratorBase, 105, 106
 - history_length, 130
 - init_state, 130
 - initial_order, 130
 - initialize_edit_integration_constants, 114, 115
 - initialize_predictor_integration_constants, 115
 - integrate_gj, 116
 - integrate_primer, 117
 - max_history_size, 131
 - mid_correct, 118
 - operator=, 119
 - order, 131
 - pos_hist, 131
 - position_corrector, 132
 - predict, 119
 - primer, 132
 - relative_tolerance, 132
 - replicate_primer, 120
 - replicate_state, 121
 - rotate_acc_hist, 122
 - save_comparison_data, 122
 - save_epoch_data, 123
 - size, 133
 - start_cycle, 124
 - state_machine, 133
 - swap, 124
 - swap_state, 125
 - test_for_convergence, 127
 - velocity_corrector, 133
- jeod::GaussJacksonIntegratorBase< State, Primer >, 101

- jeod::GaussJacksonIntegratorConstructor, 134
 - ~GaussJacksonIntegratorConstructor, 136
 - config, 141
 - configure, 136
 - create_constructor, 137
 - create_copy, 137
 - create_first_order_ode_integrator, 137
 - create_generalized_deriv_second_order_ode_integrator, 138
 - create_integration_controls, 138
 - create_second_order_ode_integrator, 139
 - GaussJacksonIntegratorConstructor, 136
 - get_buffer_size, 139
 - get_class_name, 139
 - get_transition_table_size, 140
 - implements, 140
 - init_attrjeod__GaussJacksonIntegratorConstructor, 141
 - InputProcessor, 141
 - operator=, 140
 - priming_constructor, 141
 - provides, 140
 - swap, 140
- jeod::GaussJacksonOneState, 142
 - first, 143
 - GaussJacksonOneState, 143
 - init_attrjeod__GaussJacksonOneState, 143
 - InputProcessor, 143
- jeod::GaussJacksonRationalCoefficients, 144
 - coefficients, 148
 - configure_adams_corrector, 145
 - construct_predictor, 145
 - construct_stormer_cowell_corrector, 146
 - convert_to_ordinate_form, 146
 - discard_extra_terms, 147
 - displace_back, 147
 - GaussJacksonRationalCoefficients, 145
 - init_attrjeod__GaussJacksonRationalCoefficients, 147
 - InputProcessor, 148
- jeod::GaussJacksonSimpleSecondOrderODEIntegrator, 148
 - ~GaussJacksonSimpleSecondOrderODEIntegrator, 150
 - create_copy, 151
 - GaussJacksonSimpleSecondOrderODEIntegrator, 149, 150
 - init_attrjeod__GaussJacksonSimpleSecondOrderODEIntegrator, 154
 - InputProcessor, 154
 - integrate, 151
 - operator=, 151
 - reset_integrator, 153
 - swap, 153
- jeod::GaussJacksonStateMachine, 154
 - at_downsample, 163
 - at_end_of_tour, 163
 - at_order_change, 163
 - at_reinitialize, 164
 - bootstrap_edit_redo_needed, 164
 - configure, 158
 - correction_iterations, 164
 - current_order, 164
 - cycle_scale, 165
 - cycle_start_time, 165
 - exit_bootstrap_edit, 158
 - exit_bootstrap_step, 158
 - exit_priming, 158
 - final_order, 165
 - fsm_state, 165
 - FsmState, 157
 - GaussJacksonStateMachine, 157
 - get_at_downsample, 159
 - get_at_end_of_tour, 159
 - get_at_order_change, 159
 - get_at_reinitialize, 159
 - get_current_order, 160
 - get_cycle_scale, 160
 - get_cycle_start_time, 160
 - get_fsm_state, 160
 - get_history_length, 161
 - get_max_history_size, 161
 - history_length, 166
 - history_size, 166
 - init_attrjeod__GaussJacksonStateMachine, 163
 - initial_order, 166
 - InputProcessor, 163
 - max_correction_iterations, 166
 - max_history_size, 167
 - ndoubling_steps, 167
 - perform_step, 161
 - reset, 161
 - scale_factor, 167
 - set_bootstrap_edit_redo_needed, 162
 - state_name, 162
 - step_increment, 167
 - steps_since_reset, 168
 - tour_count, 168
 - transition_state, 162
- jeod::GaussJacksonTwoState, 168
 - first, 170
 - GaussJacksonTwoState, 169
 - init_attrjeod__GaussJacksonTwoState, 170
 - InputProcessor, 170
 - second, 170
- jeod::GeneralizedSecondOrderODETechnique, 171
 - ~GeneralizedSecondOrderODETechnique, 172
 - GeneralizedSecondOrderODETechnique, 172
 - is_provided_by, 173
 - operator=, 173
 - TechniqueType, 172
 - validate_technique, 173
- jeod::IntegrationMessages, 174
 - information, 176
 - init_attrjeod__IntegrationMessages, 176
 - InputProcessor, 176

- IntegrationMessages, 175
- internal_error, 176
- invalid_item, 177
- invalid_request, 177
- operator=, 176
- unsupported_option, 177
- jeod::JeodIntegrationGroup, 178
 - ~JeodIntegrationGroup, 180
 - add_integrable_object, 181
 - group_owner, 185
 - init_attrjeod__JeodIntegrationGroup, 185
 - initialize_group, 181
 - InputProcessor, 185
 - integ_merger, 185
 - integrable_objects, 186
 - integrate_bodies, 181
 - integrate_container, 182
 - jeod_integ_interface, 186
 - jeod_time_manager, 186
 - JeodIntegrationGroup, 180, 181
 - merge_integrator_result, 182
 - need_first_step_derivatives, 183
 - operator=, 183
 - register_classes, 183
 - remove_integrable_object, 183
 - reset_body_integrators, 184
 - reset_container, 184
 - respond_to_time_change, 184
 - update_from_owner, 185
- jeod::JeodIntegrationGroupOwner, 187
 - ~JeodIntegrationGroupOwner, 187
 - update_integration_group, 187
- jeod::JeodIntegrationTime, 188
 - ~JeodIntegrationTime, 189
 - add_time_change_subscriber, 189
 - get_timestamp_time, 190
 - init_attrjeod__JeodIntegrationTime, 191
 - InputProcessor, 191
 - JeodIntegrationTime, 189
 - notify_time_change_subscribers, 190
 - operator=, 190
 - remove_time_change_subscriber, 190
 - time_change_subscribers, 191
- jeod::LsodeControlDataInterface, 192
 - ~LsodeControlDataInterface, 194
 - abs_tolerance_error_control, 195
 - abs_tolerance_error_control_vec, 196
 - corrector_method, 196
 - CorrectorMethod, 193
 - error_control_indicator, 196
 - error_control_vector_copied_over, 196
 - ErrorControlIndicator, 194
 - init_attrjeod__LsodeControlDataInterface, 195
 - initial_step_size, 197
 - InputProcessor, 195
 - integration_method, 197
 - IntegrationMethod, 194
 - is_corrector_method_functional_iteration, 195
 - LsodeControlDataInterface, 194
 - max_correction_iters, 197
 - max_num_conv_failure, 197
 - max_num_small_step_warnings, 198
 - max_num_steps, 198
 - max_num_steps_jacobian, 198
 - max_order, 198
 - max_step_size, 199
 - min_step_size, 199
 - num_odes, 199
 - num_odes_at_alloc, 200
 - operator=, 195
 - rel_tolerance_error_control, 200
 - rel_tolerance_error_control_vec, 200
- jeod::LsodeDataArrays, 201
 - ~LsodeDataArrays, 202
 - accum_correction, 203
 - allocated, 203
 - error_weight, 203
 - history, 204
 - init_attrjeod__LsodeDataArrays, 203
 - InputProcessor, 203
 - lin_alg, 204
 - lin_alg_1, 204
 - lin_alg_2, 205
 - lin_alg_index1, 205
 - LsodeDataArrays, 202
 - num_odes, 205
 - operator=, 202
 - pivots, 205
 - save, 206
- jeod::LsodeDataJacobianPrep, 206
 - ~LsodeDataJacobianPrep, 207
 - fac, 208
 - hl0, 208
 - index, 208
 - index_max, 209
 - init_attrjeod__LsodeDataJacobianPrep, 208
 - InputProcessor, 208
 - LsodeDataJacobianPrep, 207
 - operator=, 207
 - r0, 209
 - yj, 209
- jeod::LsodeDataStode, 209
 - ~LsodeDataStode, 210
 - dsm, 211
 - init_attrjeod__LsodeDataStode, 211
 - InputProcessor, 211
 - iredo, 211
 - iret, 212
 - LsodeDataStode, 210
 - ncf, 212
 - new_method_order, 212
 - operator=, 211
 - step_ratio, 212
 - step_ratio_order_inc, 213
 - told, 213
- jeod::LsodeFirstOrderODEIntegrator, 213

- arrays, [220](#)
- calculation_task, [220](#)
- CalculationTask, [218](#)
- control_data, [220](#)
- convergence_factor, [221](#)
- convergence_jacobian_flag, [221](#)
- convergence_rate, [221](#)
- cycle_target_time, [222](#)
- data_prepj, [222](#)
- data_stode, [222](#)
- EntryPoint, [219](#)
- epsilon, [222](#)
- first_pass, [223](#)
- get_re_entry_point, [219](#)
- init_attrjeod__LsodeFirstOrderODEIntegrator, [220](#)
- initialized, [223](#)
- InputProcessor, [220](#)
- internal_state, [223](#)
- iter_delta, [223](#)
- iteration_count, [224](#)
- iteration_matrix_singular, [224](#)
- jacobian_current, [224](#)
- LsodeFirstOrderODEIntegrator, [219](#)
- max_history_size, [224](#)
- max_order_internal, [225](#)
- max_rel_change_without_jacobian, [225](#)
- max_step_increase_ratio, [225](#)
- max_step_size_inv, [225](#)
- method_coeff_first, [226](#)
- method_coeffs_complete, [226](#)
- method_coeffs_current, [226](#)
- method_order_current, [226](#)
- modified_iteration_matrix_singular, [227](#)
- num_equations, [227](#)
- num_jacobian_evals, [227](#)
- num_nordsiek_cols, [228](#)
- num_predictor_elements, [228](#)
- num_small_step_warnings, [228](#)
- num_steps_taken, [228](#)
- operator=, [219](#)
- order_select_para, [229](#)
- prev_good_step_size, [229](#)
- prev_integration_method, [229](#)
- prev_iter_delta, [230](#)
- prev_method_order, [230](#)
- prev_step_size, [230](#)
- prior_num_steps, [231](#)
- re_entry_point, [231](#)
- rel_change_since_jacobian, [231](#)
- sqrt_epsilon, [231](#)
- stage_target_time, [232](#)
- step_at_last_jacobian_update, [232](#)
- step_error, [232](#)
- step_size, [233](#)
- test_coeffs_complete, [233](#)
- update_jacobian, [233](#)
- y, [234](#)
- y_dot, [234](#)
- jeod::LsodeGeneralizedDerivSecondOrderODE↔
 - Integrator, [235](#)
 - init_attrjeod__LsodeGeneralizedDerivSecond↔
 - OrderODEIntegrator, [236](#)
 - InputProcessor, [236](#)
 - LsodeGeneralizedDerivSecondOrderODE↔
 - Integrator, [236](#)
 - operator=, [236](#)
 - posdot, [237](#)
- jeod::LsodeIntegrationControls, [237](#)
 - ~LsodeIntegrationControls, [238](#)
 - init_attrjeod__LsodeIntegrationControls, [238](#)
 - InputProcessor, [239](#)
 - LsodeIntegrationControls, [238](#)
 - operator=, [238](#)
- jeod::LsodeIntegratorConstructor, [239](#)
 - data_interface, [242](#)
 - get_class_name, [240](#)
 - get_transition_table_size, [241](#)
 - implements, [241](#)
 - init_attrjeod__LsodeIntegratorConstructor, [242](#)
 - InputProcessor, [242](#)
 - LsodeIntegratorConstructor, [240](#)
 - operator=, [241](#)
 - provides, [241](#)
- jeod::LsodeSecondOrderODEIntegrator, [242](#)
 - arrays_allocated, [245](#)
 - first_derivative_size, [245](#)
 - first_order_integrator, [245](#)
 - get_re_entry_point, [244](#)
 - init_attrjeod__LsodeSecondOrderODEIntegrator, [244](#)
 - InputProcessor, [245](#)
 - LsodeSecondOrderODEIntegrator, [243](#), [244](#)
 - operator=, [244](#)
 - reset_integrator, [244](#)
 - y, [245](#)
 - y_dot, [246](#)
 - zeroth_derivative_size, [246](#)
- jeod::LsodeSimpleSecondOrderODEIntegrator, [247](#)
 - ~LsodeSimpleSecondOrderODEIntegrator, [248](#)
 - init_attrjeod__LsodeSimpleSecondOrderODE↔
 - Integrator, [248](#)
 - InputProcessor, [248](#)
 - LsodeSimpleSecondOrderODEIntegrator, [247](#), [248](#)
 - operator=, [248](#)
- jeod::Restartable2DSecondOrderIntegrator, [249](#)
 - ~Restartable2DSecondOrderIntegrator, [250](#)
 - create_integrator, [250](#)
 - destroy_integrator, [251](#)
 - init_attrjeod__Restartable2DSecondOrder↔
 - Integrator, [252](#)
 - InputProcessor, [252](#)
 - integrate, [251](#)
 - integrator, [252](#)
 - integrator_manager, [253](#)
 - operator=, [251](#)
 - reset_integrator, [252](#)

- Restartable2DSecondOrderIntegrator, 250
- simple_restore, 252
- jeod::RestartableFirstOrderODEIntegrator
 - ~RestartableFirstOrderODEIntegrator, 254
 - create_integrator_internal, 255
 - operator=, 255
 - RestartableFirstOrderODEIntegrator, 254
- jeod::RestartableFirstOrderODEIntegrator< size >, 253
- jeod::RestartableGeneralizedDerivSecondOrderODE↔Integrator
 - ~RestartableGeneralizedDerivSecondOrderODE↔Integrator, 257
 - create_integrator_internal, 258
 - operator=, 258
 - RestartableGeneralizedDerivSecondOrderODE↔Integrator, 257
 - simple_restore_internal, 258
- jeod::RestartableGeneralizedDerivSecondOrderODE↔Integrator< position_size, velocity_size, DerivFunctions >, 256
- jeod::RestartableGeneralizedStepSecondOrderODE↔Integrator
 - ~RestartableGeneralizedStepSecondOrderODE↔Integrator, 261
 - create_integrator_internal, 262
 - operator=, 262
 - RestartableGeneralizedStepSecondOrderODE↔Integrator, 261
 - simple_restore_internal, 262
- jeod::RestartableGeneralizedStepSecondOrderODE↔Integrator< position_size, velocity_size, StepFunctions >, 260
- jeod::RestartableSO3SecondOrderODEIntegrator, 272
 - ~RestartableSO3SecondOrderODEIntegrator, 273
 - create_integrator, 274
 - destroy_integrator, 274
 - generalized_deriv_integrator_manager, 276
 - generalized_step_integrator_manager, 276
 - init_attrjeod__RestartableSO3SecondOrderODE↔Integrator, 276
 - InputProcessor, 276
 - integrate, 274
 - integrator, 277
 - operator=, 275
 - reset_integrator, 275
 - RestartableSO3SecondOrderODEIntegrator, 273, 274
 - simple_restore, 275
 - technique, 277
- jeod::RestartableScalarFirstOrderODEIntegrator, 263
 - ~RestartableScalarFirstOrderODEIntegrator, 264
 - create_integrator, 265
 - destroy_integrator, 265
 - init_attrjeod__RestartableScalarFirstOrderODE↔Integrator, 266
 - InputProcessor, 267
 - integrate, 265
 - integrator, 267
 - integrator_manager, 267
 - operator=, 266
 - reset_integrator, 266
 - RestartableScalarFirstOrderODEIntegrator, 264
 - simple_restore, 266
- jeod::RestartableSecondOrderODEIntegrator, 268
 - ~RestartableSecondOrderODEIntegrator, 268
 - operator=, 269
 - RestartableSecondOrderODEIntegrator, 268, 269
- jeod::RestartableSimpleSecondOrderODEIntegrator
 - ~RestartableSimpleSecondOrderODEIntegrator, 270
 - create_integrator_internal, 271
 - operator=, 272
 - RestartableSimpleSecondOrderODEIntegrator, 270, 271
- jeod::RestartableSimpleSecondOrderODEIntegrator< size >, 269
- jeod::RestartableStateIntegrator
 - ~RestartableStateIntegrator, 280
 - clear_integrator_reference, 281
 - create_integrator, 281
 - create_integrator_internal, 283
 - destroy_integrator, 283
 - integrator_handle, 285
 - operator=, 283
 - RestartableStateIntegrator, 280, 281
 - set_integrator_reference, 284
 - simple_restore, 284
 - simple_restore_internal, 284
- jeod::RestartableStateIntegrator< IntegratorType >, 278
- jeod::RestartableT3SecondOrderODEIntegrator, 285
 - ~RestartableT3SecondOrderODEIntegrator, 287
 - create_integrator, 287
 - destroy_integrator, 287
 - init_attrjeod__RestartableT3SecondOrderODE↔Integrator, 289
 - InputProcessor, 289
 - integrate, 288
 - integrator, 289
 - integrator_manager, 290
 - operator=, 288
 - reset_integrator, 288
 - RestartableT3SecondOrderODEIntegrator, 286, 287
 - simple_restore, 289
- jeod::TimeChangeSubscriber, 290
 - ~TimeChangeSubscriber, 291
 - init_attrjeod__TimeChangeSubscriber, 292
 - InputProcessor, 292
 - operator=, 291
 - respond_to_time_change, 291
 - TimeChangeSubscriber, 291
- jeod_integ_interface
 - jeod::JeodIntegrationGroup, 186
- jeod_integration_group.cc, 320

- jeod_integration_group.hh, 320
- jeod_integration_time.cc, 321
- jeod_integration_time.hh, 321
- jeod_time_manager
 - jeod::JeodIntegrationGroup, 186
- JeodIntegrationGroup
 - jeod::JeodIntegrationGroup, 180, 181
- JeodIntegrationTime
 - jeod::JeodIntegrationTime, 189
- lin_alg
 - jeod::LsodeDataArrays, 204
- lin_alg_1
 - jeod::LsodeDataArrays, 204
- lin_alg_2
 - jeod::LsodeDataArrays, 205
- lin_alg_index1
 - jeod::LsodeDataArrays, 205
- linear_chord_iteration
 - Lsode, 45
- linear_solver
 - Lsode, 45
- load_derivatives
 - Lsode, 46
- load_ew_values
 - Lsode, 46
- Lsode, 20
 - ~LsodeFirstOrderODEIntegrator, 54
 - ~LsodeGeneralizedDerivSecondOrderODE↔Integrator, 54
 - ~LsodeSecondOrderODEIntegrator, 55
 - allocate_arrays, 24
 - calculate_epsilon, 25
 - calculate_integration_coefficients, 25
 - check_interface_data, 26
 - create_constructor, 26
 - create_copy, 26, 27
 - create_first_order_ode_integrator, 28
 - create_generalized_deriv_second_order_ode↔integrator, 28
 - create_integration_controls, 29
 - create_second_order_ode_integrator, 29
 - destroy_allocated_arrays, 30
 - gauss_elim_factor, 30
 - index_of_max_magnitude, 30
 - integrate, 31, 32
 - integrator_compute_new_order, 33
 - integrator_compute_new_order_check_step_error, 34
 - integrator_compute_new_order_prep, 34
 - integrator_core, 34
 - integrator_corrector_converged, 35
 - integrator_corrector_failed_part1, 36
 - integrator_corrector_failed_part2, 36
 - integrator_corrector_iteration, 37
 - integrator_error_test_failed, 37
 - integrator_fail_reset_order_1_part1, 38
 - integrator_fail_reset_order_1_part2, 38
 - integrator_predict, 39
 - integrator_reset_iteration_loop_part1, 39
 - integrator_reset_iteration_loop_part2, 40
 - integrator_reset_method_coeffs, 40
 - integrator_reset_yh, 40
 - integrator_set_new_order, 41
 - integrator_terminate, 41
 - integrator_test_stepsize_change, 41
 - integrator_wrapup, 42
 - interpolate_y, 42
 - jacobian_prep_init, 43
 - jacobian_prep_loop, 44
 - jacobian_prep_wrap_up, 44
 - linear_chord_iteration, 45
 - linear_solver, 45
 - load_derivatives, 46
 - load_ew_values, 46
 - LsodeControlDataInterface, 46
 - LsodeFirstOrderODEIntegrator, 47
 - LsodeGeneralizedDerivSecondOrderODE↔Integrator, 47, 48
 - LsodeIntegrationControls, 48
 - LsodeIntegratorConstructor, 48
 - LsodeSecondOrderODEIntegrator, 48, 49
 - LsodeSimpleSecondOrderODEIntegrator, 49
 - magnitude_of_weighted_array, 49, 50
 - manager_check_stop_conditions, 50
 - manager_initialize_calculation_part1, 51
 - manager_initialize_calculation_part2, 51
 - manager_integration_loop_part1, 51
 - manager_integration_loop_part2, 52
 - manager_integration_loop_part3, 52
 - manager_set_calculation_phase_eq_2_reload, 52
 - process_entry_point_cycle_start, 53
 - reset_integrator, 53
 - set_abs_tol, 53
 - set_rel_tol, 53
 - update_control_data, 54
- lsode_control_data_interface.cc, 322
- lsode_control_data_interface.hh, 322
- lsode_data_classes.cc, 323
- lsode_data_classes.hh, 323
- lsode_first_order_ode_integrator.hh, 324
- lsode_first_order_ode_integrator__integrator.cc, 324
- lsode_first_order_ode_integrator__manager.cc, 325
- lsode_first_order_ode_integrator__support.cc, 325
- lsode_first_order_ode_integrator__utility.cc, 325
- lsode_generalized_second_order_ode_integrator.cc, 326
- lsode_generalized_second_order_ode_integrator.hh, 326
- lsode_integration_controls.cc, 327
- lsode_integration_controls.hh, 327
- lsode_integrator_constructor.cc, 328
- lsode_integrator_constructor.hh, 328
- lsode_second_order_ode_integrator.cc, 329
- lsode_second_order_ode_integrator.hh, 329
- lsode_simple_second_order_ode_integrator.cc, 330
- lsode_simple_second_order_ode_integrator.hh, 330

- LsodeControlDataInterface
 - jeod::LsodeControlDataInterface, [194](#)
 - Lsode, [46](#)
- LsodeDataArrays
 - jeod::LsodeDataArrays, [202](#)
- LsodeDataJacobianPrep
 - jeod::LsodeDataJacobianPrep, [207](#)
- LsodeDataStode
 - jeod::LsodeDataStode, [210](#)
- LsodeFirstOrderODEIntegrator
 - jeod::LsodeFirstOrderODEIntegrator, [219](#)
 - Lsode, [47](#)
- LsodeGeneralizedDerivSecondOrderODEIntegrator
 - jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator, [236](#)
 - Lsode, [47](#), [48](#)
- LsodeIntegrationControls
 - jeod::LsodeIntegrationControls, [238](#)
 - Lsode, [48](#)
- LsodeIntegratorConstructor
 - jeod::LsodeIntegratorConstructor, [240](#)
 - Lsode, [48](#)
- LsodeSecondOrderODEIntegrator
 - jeod::LsodeSecondOrderODEIntegrator, [243](#), [244](#)
 - Lsode, [48](#), [49](#)
- LsodeSimpleSecondOrderODEIntegrator
 - jeod::LsodeSimpleSecondOrderODEIntegrator, [247](#), [248](#)
 - Lsode, [49](#)
- m
 - er7_utils::TwoDArray, [303](#)
- MAKE_INTEGRATION_MESSAGE_CODE
 - integration_messages.cc, [319](#)
- magnitude_of_weighted_array
 - Lsode, [49](#), [50](#)
- manager_check_stop_conditions
 - Lsode, [50](#)
- manager_initialize_calculation_part1
 - Lsode, [51](#)
- manager_initialize_calculation_part2
 - Lsode, [51](#)
- manager_integration_loop_part1
 - Lsode, [51](#)
- manager_integration_loop_part2
 - Lsode, [52](#)
- manager_integration_loop_part3
 - Lsode, [52](#)
- manager_set_calculation_phase_eq_2_reload
 - Lsode, [52](#)
- max_correction_iterations
 - jeod::GaussJacksonConfig, [78](#)
 - jeod::GaussJacksonIntegrationControls, [99](#)
 - jeod::GaussJacksonStateMachine, [166](#)
- max_correction_iters
 - jeod::LsodeControlDataInterface, [197](#)
- max_history_size
 - jeod::GaussJacksonIntegratorBase, [131](#)
 - jeod::GaussJacksonStateMachine, [167](#)
 - jeod::LsodeFirstOrderODEIntegrator, [224](#)
- max_num_conv_failure
 - jeod::LsodeControlDataInterface, [197](#)
- max_num_small_step_warnings
 - jeod::LsodeControlDataInterface, [198](#)
- max_num_steps
 - jeod::LsodeControlDataInterface, [198](#)
- max_num_steps_jacobian
 - jeod::LsodeControlDataInterface, [198](#)
- max_order
 - jeod::GaussJacksonCoeffs, [74](#)
 - jeod::LsodeControlDataInterface, [198](#)
- max_order_internal
 - jeod::LsodeFirstOrderODEIntegrator, [225](#)
- max_rel_change_without_jacobian
 - jeod::LsodeFirstOrderODEIntegrator, [225](#)
- max_step_increase_ratio
 - jeod::LsodeFirstOrderODEIntegrator, [225](#)
- max_step_size
 - jeod::LsodeControlDataInterface, [199](#)
- max_step_size_inv
 - jeod::LsodeFirstOrderODEIntegrator, [225](#)
- merge_integrator_result
 - jeod::JeodIntegrationGroup, [182](#)
- method_coeff_first
 - jeod::LsodeFirstOrderODEIntegrator, [226](#)
- method_coeffs_complete
 - jeod::LsodeFirstOrderODEIntegrator, [226](#)
- method_coeffs_current
 - jeod::LsodeFirstOrderODEIntegrator, [226](#)
- method_order_current
 - jeod::LsodeFirstOrderODEIntegrator, [226](#)
- mid_correct
 - jeod::GaussJacksonIntegratorBase, [118](#)
- min_step_size
 - jeod::LsodeControlDataInterface, [199](#)
- Models, [15](#)
- modified_iteration_matrix_singular
 - jeod::LsodeFirstOrderODEIntegrator, [227](#)
- n
 - er7_utils::TwoDArray, [303](#)
- ncf
 - jeod::LsodeDataStode, [212](#)
- ndoubling_steps
 - jeod::GaussJacksonConfig, [78](#)
 - jeod::GaussJacksonStateMachine, [167](#)
- need_first_step_derivatives
 - jeod::JeodIntegrationGroup, [183](#)
- new_method_order
 - jeod::LsodeDataStode, [212](#)
- notify_time_change_subscribers
 - jeod::JeodIntegrationTime, [190](#)
- num_equations
 - jeod::LsodeFirstOrderODEIntegrator, [227](#)
- num_jacobian_evals
 - jeod::LsodeFirstOrderODEIntegrator, [227](#)
- num_nordsiek_cols
 - jeod::LsodeFirstOrderODEIntegrator, [228](#)

- num_odes
 - jeod::LsodeControlDataInterface, [199](#)
 - jeod::LsodeDataArrays, [205](#)
- num_odes_at_alloc
 - jeod::LsodeControlDataInterface, [200](#)
- num_predictor_elements
 - jeod::LsodeFirstOrderODEIntegrator, [228](#)
- num_small_step_warnings
 - jeod::LsodeFirstOrderODEIntegrator, [228](#)
- num_steps_taken
 - jeod::LsodeFirstOrderODEIntegrator, [228](#)
- operator const T *const *
 - er7_utils::TwoDArray, [298](#)
- operator T*const *
 - er7_utils::TwoDArray, [299](#)
- operator<<
 - jeod, [60](#)
 - jeod::GaussJacksonCoeffs, [73](#)
- operator()
 - er7_utils::TwoDArray, [299](#), [300](#)
- operator=
 - er7_utils::TwoDArray, [300](#)
 - jeod::GaussJacksonCoefficientsPair, [67](#)
 - jeod::GaussJacksonCoeffs, [72](#)
 - jeod::GaussJacksonFirstOrderODEIntegrator, [82](#)
 - jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, [86](#)
 - jeod::GaussJacksonIntegrationControls, [95](#)
 - jeod::GaussJacksonIntegratorBase, [119](#)
 - jeod::GaussJacksonIntegratorConstructor, [140](#)
 - jeod::GaussJacksonSimpleSecondOrderODEIntegrator, [151](#)
 - jeod::GeneralizedSecondOrderODETechnique, [173](#)
 - jeod::IntegrationMessages, [176](#)
 - jeod::JeodIntegrationGroup, [183](#)
 - jeod::JeodIntegrationTime, [190](#)
 - jeod::LsodeControlDataInterface, [195](#)
 - jeod::LsodeDataArrays, [202](#)
 - jeod::LsodeDataJacobianPrep, [207](#)
 - jeod::LsodeDataStode, [211](#)
 - jeod::LsodeFirstOrderODEIntegrator, [219](#)
 - jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator, [236](#)
 - jeod::LsodeIntegrationControls, [238](#)
 - jeod::LsodeIntegratorConstructor, [241](#)
 - jeod::LsodeSecondOrderODEIntegrator, [244](#)
 - jeod::LsodeSimpleSecondOrderODEIntegrator, [248](#)
 - jeod::Restartable2DSecondOrderIntegrator, [251](#)
 - jeod::RestartableFirstOrderODEIntegrator, [255](#)
 - jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator, [258](#)
 - jeod::RestartableGeneralizedStepSecondOrderODEIntegrator, [262](#)
 - jeod::RestartableSO3SecondOrderODEIntegrator, [275](#)
 - jeod::RestartableScalarFirstOrderODEIntegrator, [266](#)
 - jeod::RestartableSecondOrderODEIntegrator, [269](#)
 - jeod::RestartableSimpleSecondOrderODEIntegrator, [272](#)
 - jeod::RestartableStateIntegrator, [283](#)
 - jeod::RestartableT3SecondOrderODEIntegrator, [288](#)
 - jeod::TimeChangeSubscriber, [291](#)
- operator[]
 - er7_utils::TwoDArray, [300](#), [301](#)
- order
 - jeod::GaussJacksonCoeffs, [74](#)
 - jeod::GaussJacksonIntegrationControls, [99](#)
 - jeod::GaussJacksonIntegratorBase, [131](#)
- order_select_para
 - jeod::LsodeFirstOrderODEIntegrator, [229](#)
- perform_step
 - jeod::GaussJacksonStateMachine, [161](#)
- pivots
 - jeod::LsodeDataArrays, [205](#)
- pos_hist
 - jeod::GaussJacksonIntegratorBase, [131](#)
- pos_integrator
 - jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, [87](#)
- posdot
 - jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, [88](#)
 - jeod::LsodeGeneralizedDerivSecondOrderODEIntegrator, [237](#)
- posdotdot
 - jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, [88](#)
- position_corrector
 - jeod::GaussJacksonIntegratorBase, [132](#)
- predict
 - jeod::GaussJacksonIntegratorBase, [119](#)
- predictor
 - jeod::GaussJacksonCoeffs, [74](#)
- prev_good_step_size
 - jeod::LsodeFirstOrderODEIntegrator, [229](#)
- prev_integration_method
 - jeod::LsodeFirstOrderODEIntegrator, [229](#)
- prev_iter_delta
 - jeod::LsodeFirstOrderODEIntegrator, [230](#)
- prev_method_order
 - jeod::LsodeFirstOrderODEIntegrator, [230](#)
- prev_step_size
 - jeod::LsodeFirstOrderODEIntegrator, [230](#)
- primer
 - jeod::GaussJacksonIntegratorBase, [132](#)
- priming_constructor
 - jeod::GaussJacksonIntegratorConstructor, [141](#)
- priming_controls
 - jeod::GaussJacksonIntegrationControls, [100](#)
- priming_technique
 - jeod::GaussJacksonConfig, [79](#)

- printOut
 - jeod::GaussJacksonCoefficientsPair, [67](#)
- prior_num_steps
 - jeod::LsodeFirstOrderODEIntegrator, [231](#)
- process_entry_point_cycle_start
 - Lsode, [53](#)
- provides
 - jeod::GaussJacksonIntegratorConstructor, [140](#)
 - jeod::LsodeIntegratorConstructor, [241](#)
- r0
 - jeod::LsodeDataJacobianPrep, [209](#)
- re_entry_point
 - jeod::LsodeFirstOrderODEIntegrator, [231](#)
- register_classes
 - jeod::JeodIntegrationGroup, [183](#)
- rel_change_since_jacobian
 - jeod::LsodeFirstOrderODEIntegrator, [231](#)
- rel_tolerance_error_control
 - jeod::LsodeControlDataInterface, [200](#)
- rel_tolerance_error_control_vec
 - jeod::LsodeControlDataInterface, [200](#)
- relative_tolerance
 - jeod::GaussJacksonConfig, [79](#)
 - jeod::GaussJacksonIntegratorBase, [132](#)
- remove_integrable_object
 - jeod::JeodIntegrationGroup, [183](#)
- remove_time_change_subscriber
 - jeod::JeodIntegrationTime, [190](#)
- replicate_primer
 - jeod::GaussJacksonIntegratorBase, [120](#)
- replicate_state
 - jeod::GaussJacksonIntegratorBase, [121](#)
- reset
 - jeod::GaussJacksonStateMachine, [161](#)
- reset_body_integrators
 - jeod::JeodIntegrationGroup, [184](#)
- reset_container
 - jeod::JeodIntegrationGroup, [184](#)
- reset_integrator
 - jeod::GaussJacksonFirstOrderODEIntegrator, [82](#)
 - jeod::GaussJacksonGeneralizedDerivSecondOrderODEIntegrator, [86](#)
 - jeod::GaussJacksonIntegrationControls, [95](#)
 - jeod::GaussJacksonSimpleSecondOrderODEIntegrator, [153](#)
 - jeod::LsodeSecondOrderODEIntegrator, [244](#)
 - jeod::Restartable2DSecondOrderIntegrator, [252](#)
 - jeod::RestartableSO3SecondOrderODEIntegrator, [275](#)
 - jeod::RestartableScalarFirstOrderODEIntegrator, [266](#)
 - jeod::RestartableT3SecondOrderODEIntegrator, [288](#)
 - Lsode, [53](#)
- reset_time
 - jeod::GaussJacksonIntegrationControls, [100](#)
- respond_to_time_change
 - jeod::JeodIntegrationGroup, [184](#)
- jeod::TimeChangeSubscriber, [291](#)
- Restartable2DSecondOrderIntegrator
 - jeod::Restartable2DSecondOrderIntegrator, [250](#)
- restartable_2d_second_order_integrator.hh, [331](#)
- restartable_state_integrator.hh, [331](#)
- restartable_state_integrator_templates.hh, [332](#)
- RestartableFirstOrderODEIntegrator
 - jeod::RestartableFirstOrderODEIntegrator, [254](#)
- RestartableGeneralizedDerivSecondOrderODEIntegrator
 - jeod::RestartableGeneralizedDerivSecondOrderODEIntegrator, [257](#)
- RestartableGeneralizedStepSecondOrderODEIntegrator
 - jeod::RestartableGeneralizedStepSecondOrderODEIntegrator, [261](#)
- RestartableSO3SecondOrderODEIntegrator
 - jeod::RestartableSO3SecondOrderODEIntegrator, [273](#), [274](#)
- RestartableScalarFirstOrderODEIntegrator
 - jeod::RestartableScalarFirstOrderODEIntegrator, [264](#)
- RestartableSecondOrderODEIntegrator
 - jeod::RestartableSecondOrderODEIntegrator, [268](#), [269](#)
- RestartableSimpleSecondOrderODEIntegrator
 - jeod::RestartableSimpleSecondOrderODEIntegrator, [270](#), [271](#)
- RestartableStateIntegrator
 - jeod::RestartableStateIntegrator, [280](#), [281](#)
- RestartableT3SecondOrderODEIntegrator
 - jeod::RestartableT3SecondOrderODEIntegrator, [286](#), [287](#)
- rotate_acc_hist
 - jeod::GaussJacksonIntegratorBase, [122](#)
- rotate_down
 - er7_utils::TwoDArray, [301](#)
- rotate_up
 - er7_utils::TwoDArray, [302](#)
- row_array
 - er7_utils::TwoDArray, [304](#)
- sa_coefs
 - jeod::GaussJacksonCoefficientsPair, [69](#)
- save
 - jeod::LsodeDataArrays, [206](#)
- save_comparison_data
 - jeod::GaussJacksonIntegratorBase, [122](#)
- save_epoch_data
 - jeod::GaussJacksonIntegratorBase, [123](#)
- scale_factor
 - jeod::GaussJacksonStateMachine, [167](#)
- second
 - jeod::GaussJacksonTwoState, [170](#)
- set_abs_tol
 - Lsode, [53](#)
- set_bootstrap_edit_redo_needed
 - jeod::GaussJacksonStateMachine, [162](#)
- set_default_config_values

- jeod, 62
- set_integrator_reference
 - jeod::RestartableStateIntegrator, 284
- set_rel_tol
 - Lsode, 53
- simple_restore
 - jeod::Restartable2DSecondOrderIntegrator, 252
 - jeod::RestartableSO3SecondOrderODEIntegrator, 275
 - jeod::RestartableScalarFirstOrderODEIntegrator, 266
 - jeod::RestartableStateIntegrator, 284
 - jeod::RestartableT3SecondOrderODEIntegrator, 289
- simple_restore_internal
 - jeod::RestartableGeneralizedDerivSecondOrder↔ODEIntegrator, 258
 - jeod::RestartableGeneralizedStepSecondOrder↔ODEIntegrator, 262
 - jeod::RestartableStateIntegrator, 284
- size
 - jeod::GaussJacksonIntegratorBase, 133
- sqrt_epsilon
 - jeod::LsodeFirstOrderODEIntegrator, 231
- stage_target_time
 - jeod::LsodeFirstOrderODEIntegrator, 232
- standard_configuration
 - jeod::GaussJacksonConfig, 76
- start_cycle
 - jeod::GaussJacksonIntegrationControls, 96
 - jeod::GaussJacksonIntegratorBase, 124
- state_machine
 - jeod::GaussJacksonIntegrationControls, 100
 - jeod::GaussJacksonIntegratorBase, 133
- state_name
 - jeod::GaussJacksonStateMachine, 162
- step_at_last_jacobian_update
 - jeod::LsodeFirstOrderODEIntegrator, 232
- step_error
 - jeod::LsodeFirstOrderODEIntegrator, 232
- step_increment
 - jeod::GaussJacksonStateMachine, 167
- step_ratio
 - jeod::LsodeDataStode, 212
- step_ratio_order_inc
 - jeod::LsodeDataStode, 213
- step_size
 - jeod::LsodeFirstOrderODEIntegrator, 233
- steps_since_reset
 - jeod::GaussJacksonStateMachine, 168
- swap
 - er7_utils::TwoDArray, 302
 - jeod::GaussJacksonCoefficientsPair, 68
 - jeod::GaussJacksonCoeffs, 73
 - jeod::GaussJacksonFirstOrderODEIntegrator, 82, 83
 - jeod::GaussJacksonGeneralizedDerivSecond↔OrderODEIntegrator, 87
 - jeod::GaussJacksonIntegrationControls, 96
 - jeod::GaussJacksonIntegratorBase, 124
 - jeod::GaussJacksonIntegratorConstructor, 140
 - jeod::GaussJacksonSimpleSecondOrderODE↔Integrator, 153
- swap_state
 - jeod::GaussJacksonIntegratorBase, 125
- technique
 - jeod::RestartableSO3SecondOrderODEIntegrator, 277
- TechniqueType
 - jeod::GeneralizedSecondOrderODETechnique, 172
- test_coeffs_complete
 - jeod::LsodeFirstOrderODEIntegrator, 233
- test_for_convergence
 - jeod::GaussJacksonIntegratorBase, 127
- time_change_subscriber.hh, 333
- time_change_subscribers
 - jeod::JeodIntegrationTime, 191
- TimeChangeSubscriber
 - jeod::TimeChangeSubscriber, 291
- told
 - jeod::LsodeDataStode, 213
- tour_count
 - jeod::GaussJacksonStateMachine, 168
- transition_state
 - jeod::GaussJacksonStateMachine, 162
- two_d_array.hh, 334
- TwoDArray
 - er7_utils::TwoDArray, 294
- unsupported_option
 - jeod::IntegrationMessages, 177
- update_control_data
 - Lsode, 54
- update_from_owner
 - jeod::JeodIntegrationGroup, 185
- update_integration_group
 - jeod::JeodIntegrationGroupOwner, 187
- update_jacobian
 - jeod::LsodeFirstOrderODEIntegrator, 233
- Utils, 16
- validate_config
 - jeod, 62
- validate_configuration
 - jeod::GaussJacksonConfig, 77
- validate_technique
 - jeod::GeneralizedSecondOrderODETechnique, 173
- vel_integrator
 - jeod::GaussJacksonGeneralizedDerivSecond↔OrderODEIntegrator, 88
- velocity_corrector
 - jeod::GaussJacksonIntegratorBase, 133
- y

- [jeod::LsodeFirstOrderODEIntegrator, 234](#)
 - [jeod::LsodeSecondOrderODEIntegrator, 245](#)
- [y_dot](#)
 - [jeod::LsodeFirstOrderODEIntegrator, 234](#)
 - [jeod::LsodeSecondOrderODEIntegrator, 246](#)
- [yj](#)
 - [jeod::LsodeDataJacobianPrep, 209](#)
- [zeroth_derivative_size](#)
 - [jeod::LsodeSecondOrderODEIntegrator, 246](#)