

main()	using namespace
	input/output files
	initialize MPI
	construct elastic3D
	elastic3D.run_mesh()
	elastic3D.run()
PROGRAM END	

run_mesh()	read_domain()
	output_mesh()
run()	setup_system()
	assemble_system()
	solve()
	output_results()

setup_system()	distribute dofs	
	setup_quadrature_point_history()	
	sparsity_pattern(args)	
	local dof reinit	system_matrix
		system_rhs
	reinit inc. disp.	
	constraints.clear()	
	nodal_bcs()	
assemble_system()	system_rhs=0	
	system_matrix=0	
	inst.	fe_values
		cell_matrix
		cell_rhs
		local_dof_indiced
		body_force
		body_force_values
		lambda_values
		mu_values
		lambda
		mu
		rhs_values
		pltd_rhs
		ptld_idx
		load1
		matchver
		z_component
		boundary_values
	Loop	cells in dof_handler
	compress, vector add	system_matrix
		system_rhs

nodal_bcs()	create		vector of BC pts		
			vector of tuples: (DOF,Mag)		
			vector for BC removal		
	read file into vectors				
	set affected radius				
	loop	cells	unpack tuple (DOF,Mag)		
			constraints.add_line		
		vertices	if non-zero	constraints.set_inhomogeneity	
			add BC for removal		
		BC points	remove BC		bcpoints
					bcmags
		BC mags	clear removal list		
			constraints.close()		

cell loop	check locally owned			
	cell_matrix=0			
	cell_rhs=0			
	fe_values.reinit(cell)			
	loop	i	get_strain()	
		j	cell_matrix(i,j)	
		q		
	local_quadrature_points_data			
	body_force			
	loop	i	old_stress	cell_rhs
		q		
	get_dof_indices			
	constraints.dist._local_to_global			
compress, vector add			system_matrix	
			system_rhs	

solve()	dist._incr._disp
	solver_control
	Solver cg
	preconditioner(system_matrix)
	cg.solve
	incremental_displacement

output_results()	
------------------	--