Abstract

This docum

1 Example

2 Function Overview

Name	bilinear_system_residual	
Call	rr = bilinear_system_residual(t, y, yp, Control_Volume, Parameters, element, nodes, BoundaryElements)	
Description	This function evaluates the residual of the discretised Poisson-Nernst-Planck equations.	
Inputs	t	Current time point.
	y	Vector of size (3*length(nodes)) consisting of the current solution values for each discretised.
	yp	Vector of size (3*length(nodes)) consisting of the current time derivatives for each discretised equation.
	Parameters	Structure defining all parameters required in simulation.
	element	Mesh connectivity matrix.
	nodes	Node numbers and location of nodes
	DL	Matrix defining the length of control volume faces in each element.
Outputs	rr	A vector of size (3*length(nodes)) that returns the value of the residual for each equation.
Notes		

Name	bilinear_flux_mex	
Call	Flux = bilinear_flux_mex (t, y, Parameters, element, nodes, DL)	
Description	BILINEAR_FLUX_MEX evaluates the integral of the flux term in the discretised Poisson-Nernst-Planck equations.	
Inputs	t	Current time point.
	y	A vector of size (3*length(nodes)) consisting of the current solution values at each node point.
	Parameters	Structure defining all parameters required in simulation.
	element	Mesh connectivity matrix.
	nodes	Node numbers and location of nodes
	DL	Matrix defining the length of control volume faces in each element.
Outputs	Flux	Vector of size (3*length(nodes)) that returns surface integral of the flux around each node in the mesh
Notes		

Name	Control_Volume_function
Call	Control_Volume = Control_Volume_function(element, nodes)
Description	CONTROL_VOLUME_FUNCTION calculates the volume of each control volume in the mesh defined by element and nodes.
Inputs	element Mesh connectivity matrix. nodes Node numbers and location of nodes
Outputs	
Notes	

Name	jacobian_calc(nodes, element)
Call	[dfdy, dfdyp] = jacobian_calc(nodes, element)
Description	JACOBIAN_CALC evaluates the connectivity between each element in the jacobian. This is used to efficiently calculate the Jacobian in the ode solver.
Inputs	nodes Node numbers and location of nodes element Mesh connectivity matrix.
Outputs	
Notes	

Name	length_element
Call	length = length_element(x, y)
Description	LENGTH_ELEMENT calculates the length of each face inside the quadrilateral with vertices defined by x and y.
Inputs	x Horizontal coordinate y Vertical coordinate
Outputs	
Notes	

Name	unsteady_bilinear	
Call	Unsteady = unsteady_bilinear(t, y, yp, Parameters, element, nodes)	
Description	UNSTEADY_BILINEAR evaluates the unsteady term in the finite volume discretisation of the Poisson-Nernst-Planck equations.	
Inputs	t	Current time point.
	y	A vector of size (3*length(nodes)) consisting of the current solution values at each node point.
	yp	Vector of size (3*length(nodes)) consisting of the current time derivatives for each discretised equation.
	Parameters	Structure defining all parameters required in simulation.
	element	Mesh connectivity matrix.
	nodes	Node numbers and location of nodes
Outputs		
Notes		

Name	source_bilinear
Call	Source = source_bilinear(t, y, yp, Parameters, element, nodes)
Description	SOURCE_BILINEAR evaluates the source term in the finite volume discretisation of the Poisson-Nernst-Planck equations
Inputs	t — Current time point. y — A vector of size (3*length(nodes)) consisting of the current solution values at each node point. yp — Vector of size (3*length(nodes)) consisting of the current time derivatives for each discretised equation. Parameters — Structure defining all parameters required in simulation. element — Mesh connectivity matrix. nodes — Node numbers and location of nodes
Outputs	
Notes	