

SCD SEMISMOOTH* NEWTON METHOD FOR SOLVING STOKES PROBLEM WITH STICK-SLIP BOUNDARY CONDITION

Vladimir Arzt⁽¹⁾, Petr Beremlijski⁽²⁾

⁽¹⁾Ph.D. Student, VSB – Technical University of Ostrava, Czech Republic

⁽²⁾Associate Professor, VSB – Technical University of Ostrava, Czech Republic

Keywords: SCD semismooth* Newton method, Stokes problem, stick-slip boundary conditions, limiting coderivative, Schur complement, non-smooth optimization, cerebral aneurysm model, biomechanics.

Abstract

This contribution addresses the numerical solution of the Stokes flow problem characterized by nonlinear stick-slip boundary conditions. Traditional numerical approaches, such as regularization or penalization methods, often struggle with selecting smoothing parameters, which can lead to poor convergence or ill-conditioned linear systems.

The SCD semismooth* Newton method, initially developed by J. Outrata, H. Gfrerer, and J. Valdman, was designed for the numerical solution of generalized equations where the set-valued part is an SCD (subspace containing derivative) mapping. Originally applied to contact problems involving Coulomb friction, this solver is also well-suited for the Stokes problem, given the nonlinearity introduced by stick-slip boundary conditions. The method linearizes the set-valued term using the graph of the limiting coderivative, enabling a precise treatment of the stick-slip transition without artificial smoothing. To ensure computational efficiency, the linear system at each Newton step is solved using the Schur complement method.

The performance and reliability of the proposed algorithm are demonstrated through a series of numerical experiments of varying complexity, ranging from 2D benchmarks to a 3D cerebral aneurysm model. This complex, real-world geometry is used to verify the method's robustness and functionality in a practical biomechanical context. The method exhibits local superlinear convergence and outperforms other variants of semismooth Newton methods in terms of both the number of iterations and overall computational time.