Numerical Simulation of Flow Past a Finite-Length Cylinder in Laminar Regime: Postprint

Authors: Guan Geng

Date: 2025-04-24T16:45:22+00:00

Abstract

Based on the computational fluid dynamics software Fluent, the incompressible Navier-Stokes equations were solved using the finite volume method to simulate the flow around a finite-length cylinder in a laminar flow regime with an aspect ratio of h/d=8 (where h and d denote the height and diameter of the cylinder, respectively). One end of the cylinder was fixed to the wall (i.e., the fixed end), while the other end was unconstrained (i.e., the free end). The Reynolds number range based on the cylinder diameter was 40≤Re$ $400. The simulation study obtained the wake flow field characteristics of the cylinder flow as well as its lift and drag coefficients, Strouhal number, etc. The results demonstrate that the peak lift coefficient, mean drag coefficient, and Strouhal number of the cylinder are all lower than the corresponding values for a two-dimensional cylinder, with the primary cause being the flow field structures at the cylinder's free end and fixed end. Additionally, the study discovered that the peak lift coefficient exhibits a trend of first increasing and then decreasing with increasing Re, and the paper provides a quantitative analysis of this issue.

Full Text

Preamble

The mathematical framework begins with the foundational expressions !N through !N567!;, establishing the parameter space for the KT and RKT transformations. The core relationship is defined by the equation sequence that serves as the primary constraint condition, where the RKT function operates on the input space. The technical implementation involves RKTF2 processing through multiple stages: TX5KT24 operates on the KT domain as [KT\10=.4>, while the parallel computation executes the transformation.

The derivative operation yields !N!#!, which completes the first computational block. The subsequent analysis involves recursive application of these transformations, with each iteration updating the state parameters according to the rules specified in MATH_{0005} .

The algorithm proceeds through three distinct phases: 1. Initial parameter estimation using MATH_{0006} . 2. Core transformation loop implementing through the transformation pipeline. 3. Final optimization applying MATH_{0010} , MATH_{0011} , and MATH_{0012} to generate the optimized solution.

Validation is performed using the standard test suite, with performance metrics recorded at each stage. The intermediate results demonstrate convergence properties, while the final output provides the optimized solution.

Extended analysis incorporates additional constraints through MATH_{0019} , MATH_{0020} , and MATH_{0021} , enabling multi-scale processing. The parallel implementations exhibit complementary performance characteristics, with error bounds specified by MATH_{0025} .

The refined methodology achieves superior results through the integration of MATH_{0026} . Comparative evaluation shows that configurations using MATH_{0027} and MATH_{0028} outperform baseline approaches by significant margins, with additional gains from MATH_{0029} and MATH_{0030} , exceeding predictions from MATH_{0031} .

Computational efficiency is enhanced via the HF* and HF# operators, with memory requirements scaling according to the system architecture. The complete framework integrates these components into a unified system, yielding final performance metrics documented through the evaluation pipeline.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv — Machine translation. Verify with original.