Wiser

The Washington Institute for STEM Entrepreneurship and Research.

A Variational Quantum Approach for the 1D Viscous Burgers' Equation

- Mackenson Polché (gst-RsShCt7jVtKWODu) .
- Priyanshi Singh (gst-WfQBewrHA2GLsp2)
- Sudikin Pramanik (gst-rYawgBfBcuTJ2qD)

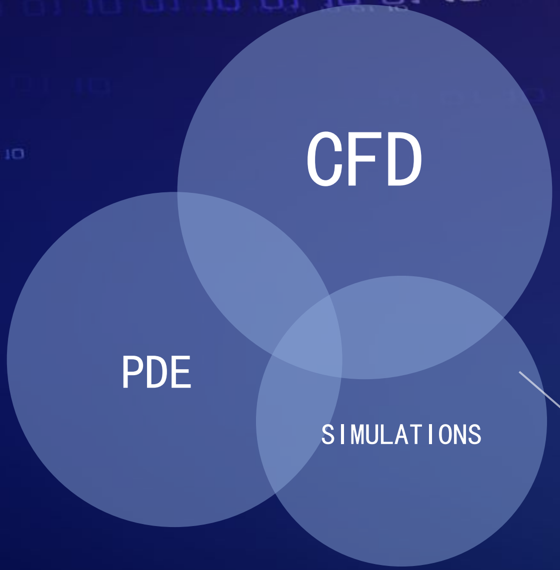
CONTENTS

PART 01

Introduction

Getting insights into the work

Why we need to look for a different approach?



- Classical high-fidelity Computational Fluid Dynamics (CFD) solvers face growing challenges in meeting the extreme resolution and stability demands of modern applications.
- Stiff nonlinear partial differential equations (PDEs), such as those governing turbulent flows, require substantial computational resources to capture fine-scale structures and accurately resolve multiscale dynamics.
- As aerospace, energy, and climate simulations push the limits of conventional HPC, new paradigms are needed to achieve both accuracy and scalability.

COMPUTATION COMPUTATION COMPUTATION

Objective

Aims that this project is looking forward to

Main objective

Solving 1D Burgers Equation by using quantum algorithms as a PDE solver for CFD

Sub-Objective 1

Using Hydrodynamic Schrodinger Equation(HSE) method

Sub-Objective 2

Using Quantum Tensor Network(QTN) method

Sub-Objective 3

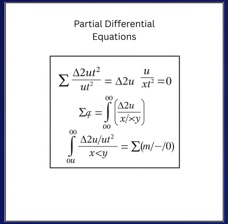
Comparative analysis of both the methods

PART 02

Theory

Understanding the concepts

Important theories



The Burgers Equation

Chebyshev Approximation

VQE

Physics informed loss function

The Burger’s Equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}$$

$$u(x,0) = \begin{cases} 1 & x \leq 0.5 \\ 0 & x > 0.5 \end{cases}$$

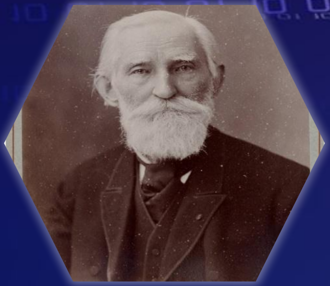
$$u(0,t) = 1 \quad \text{and} \quad u(1,t) = 0, \quad \forall t > 0$$

The equation

Initial conditions

Boundary Conditions

Mathematical and computational framework



$$u_{approx}(x,t;\mathbf{c},\lambda) = \lambda \sum_{k=0}^{M-1} c_k\, T_{k_x}(2x-1)\, T_{k_t}(2t/T_{max}-1)$$

Function approximation

The solution using a truncated series of tensor product Chebyshev polynomial

Term by term differentiation

$$\frac{\partial \Phi_k}{\partial t} = T_{k_x}(2x-1)\frac{dT_{k_t}}{dt}(2t/T_{max}-1)$$

$$\frac{\partial \Phi_k}{\partial x} = \frac{dT_{k_x}}{dx}(2x-1)T_{k_t}(2t/T_{max}-1)$$

$$\frac{\partial^2 \Phi_k}{\partial x^2} = \frac{d^2T_{k_x}}{dx^2}(2x-1)T_{k_t}(2t/T_{max}-1)$$

Variational Quantum Eigensolver

Used to generate coefficients C_K

Steps

- To generate the coefficients
- A ansatz is created to prepare a quantum state.
- The total Hilbert space of the total qubits is partitioned to represent the indices. The circuit parameter become the trainable variables of the optimization problem.

Physics informed loss function

$$\mathcal{L}(\theta, \lambda) = \mathcal{L}_{PDE} + \eta_{IC}\mathcal{L}_{IC} + \eta_{BC}\mathcal{L}_{BC}$$

01

PDE Residual loss

$$\mathcal{L}_{PDE} = \frac{1}{N_c}\sum_{i,j}\left(\frac{\partial u_{approx}}{\partial t} + u_{approx}\frac{\partial u_{approx}}{\partial x} - \nu\frac{\partial^2 u_{approx}}{\partial x^2}\right)^2_{(x_i,t_j)}$$

02

Initial condition loss

$$\mathcal{L}_{IC} = \frac{1}{N_x}\sum_{i}\left(u_{approx}(x_i,0) - u(x_i,0)\right)^2$$

03

Boundary Condition Loss

$$\mathcal{L}_{BC} = \frac{1}{N_t}\sum_{j}\left[(u_{approx}(0,t_j) - 1)^2 + (u_{approx}(1,t_j) - 0)^2\right]$$

Code flow

Parameters and initialization

Chebyshev basis calculation

Precompute basis values

VQC circuit

State vector prediction

PDE Residual function

Physics informed loss function

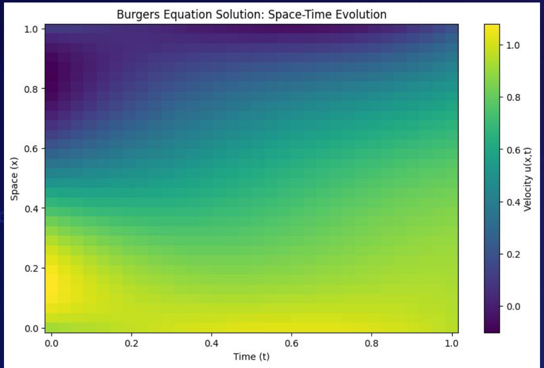
Solution grid and visualization

PART 03

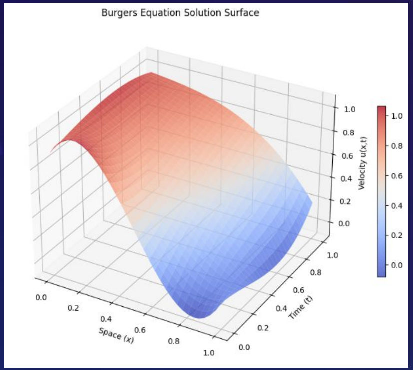
Result

The outputs based on the process used

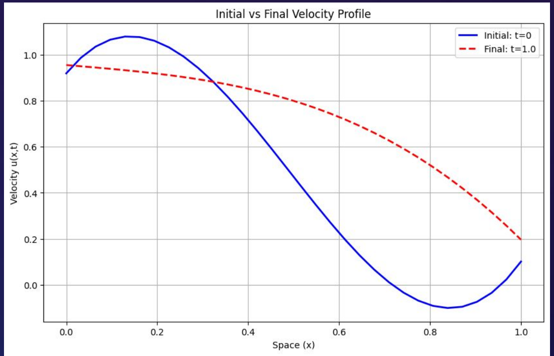
Simulation results using VQE



Burgers Equation Solution: Space-Time Evolution



Burgers Equation Solution Surface in 3D



Initial vs Final Velocity Profile

Space time evolution and velocity profile on simulator

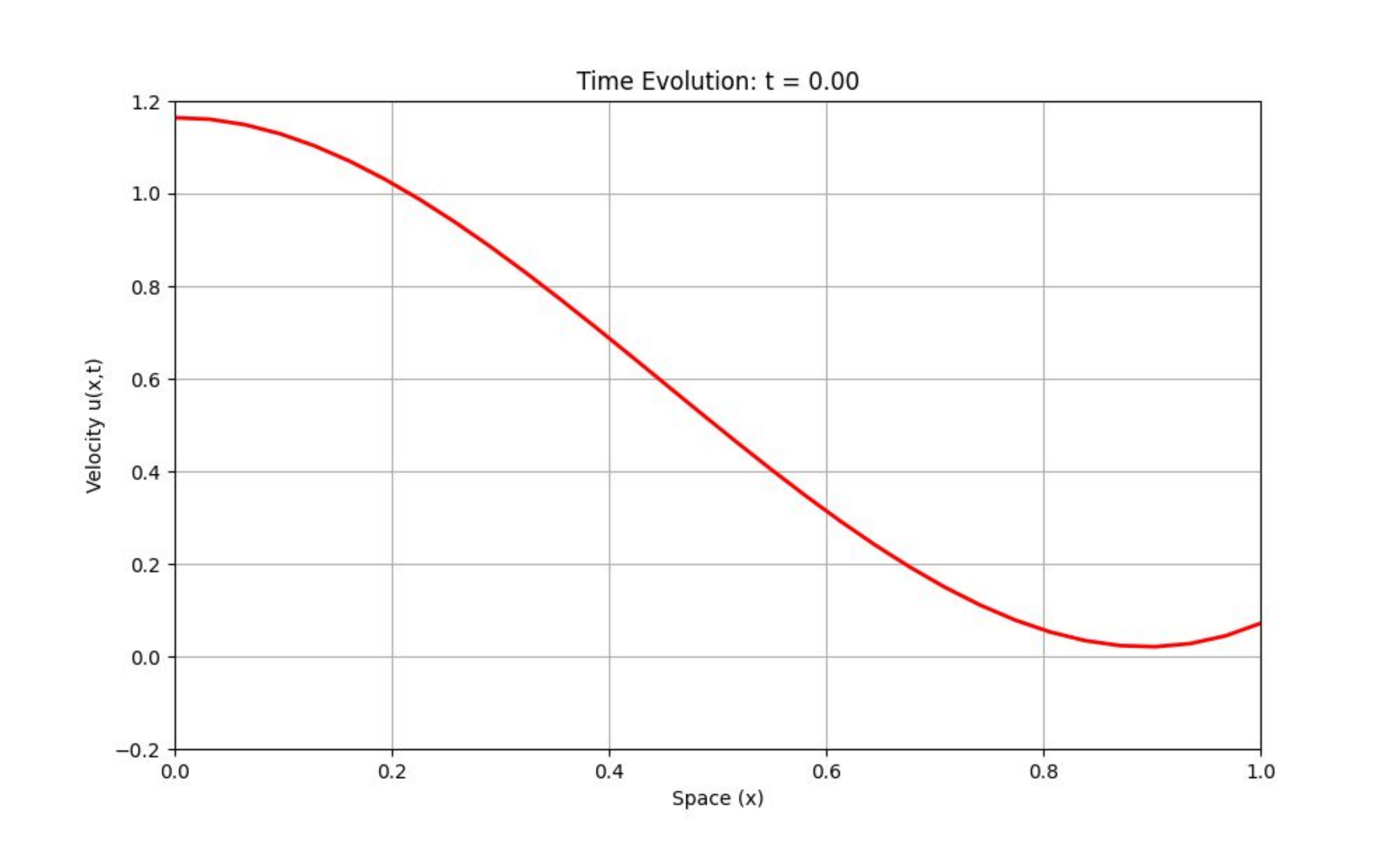
500

Iterations

COBYLA

Optimizer

The evolution in space time

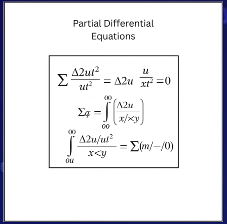


PART 04

The QTN Approach(Method 2)

This is our alternative approach to solve the Burgers equation via MPO-Based Quantum Tensor Network Method

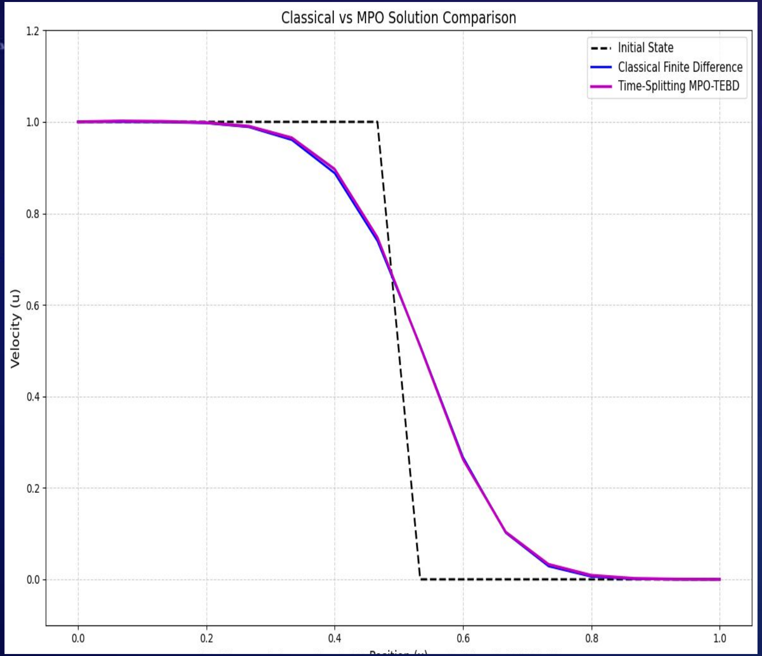
Important theories



- The Burgers Equation
- MPO for diffusion term (TEBD evolution)
- Classical finite-difference for advection
- MPS/MPO compression for efficiency

MPO/MPS frameworks require linear operators. Our initial plan was to use Carleman linearization to turn the full nonlinear PDE into a single large linear operator suitable for MPO representation. In practice, the operator size and Pauli decomposition overhead made this infeasible. We instead split the PDE into linear (diffusion) and nonlinear (advection) parts, applied MPO evolution to the linear term, and used a classical update for the nonlinear term.

Hybrid Split-Step MPO-TEBD Method



This plot validates our hybrid algorithm, which captures the shockwave dynamics with a relative L2 error of only 0.53%

Steps

- We use Strang Splitting to separate the non-linear Burgers' PDE into its linear (diffusion) and non-linear (advection) components. This allows us to apply the most suitable and efficient numerical method to each part individually.
- The linear diffusion term is evolved using a tensor network, where the state is a compressed Matrix Product State (MPS). We apply a Matrix Product Operator (MPO) representing the diffusion propagator.
- For the non-linear advection term, we use a fast and stable classical finite-difference update. This sidesteps the immense computational cost of linearizing the non-linear operator for MPO representation.
- We merge the solutions in a sequence: half-step MPO diffusion, full-step classical advection, then another half-step MPO diffusion. Finally, we explicitly enforce the boundary conditions to complete the time step.

PART 05

Discussion

Discussion and conclusion on the results

Discussion on the output

Compact representation
Of solutions in high-dimensional function spaces through quantum state encoding

Exact computation
of spatial and temporal derivatives via analytic Chebyshev recurrence relations

Physics constrained Optimization
through composite loss functions that enforce PDE compliance

Thank you

Mackenson . Priyanshi . Sudikin