High Accuracy Approximations of Burgers' Equation Using Adjoint Data Assimilation Methods

Alexander Recce
The Pingry School, 131 Martinsville Rd, Basking Ridge, NJ 07920, USA.

Email: recce.alexander@gmail.com
Manuscript submitted January 5, 2025; accepted March 10, 2025; published May 22, 2025.
doi: 10.17706/ijapm.2025.15.1.35-50

Abstract: Progress in modeling complex phenomena in many areas of physics, including fluid mechanics, electromagnetism, quantum mechanics, general relativity, and materials science requires solving non-linear Partial Differential Equations (PDEs). Burgers' equation is especially important, as it is one of the simplest non-linear equations in physics that captures essential features of more complex physical phenomena. In this work Burgers' equation is solved both exactly and numerically with the following as initial conditions: the sinusoidal function, Dirac delta function, and Heaviside step function. For the numerical solution, I use a predictor-correct scheme through time, and a finite difference scheme with 4th-order accuracy through space. Numerical results from both low viscosity and high viscosity conditions are compared with exact solutions. Before Data Assimilation (DA) is applied, even with decent numerical accuracy, the difference between the numerical result and exact solution is visible at the end of the analysis time window, which occurs due to the accumulation of numerical error while integrating. I then derive and apply the DA algorithm based on the adjoint method for Burgers' equation. The DA-based solution from the adjoint equation successfully corrects the initial condition and significantly improves the numerical results. The DA approach outlined in this research not only works for approximations of Burgers' equation, but it also has applications to other non-linear partial differential equations.

Keywords: Non-linear dynamics, Partial Differential Equations (PDEs), boundary conditions, numerical analysis

1. Introduction

Applying numerical methods to solve non-linear partial differential equations is fundamental to climate modeling [1–4], improvements to medical devices [5], designs of energy systems [6], and a variety of other scientific fields [7, 8]. Numerical methods suffer inherent precision and accuracy loss as the simulation length increases. With improvements in computer speed and storage over the last few decades, it has become increasingly possible to improve the real-time performance of numerical algorithms through methods that incorporate observational data in simulations. One such approach that has become popular is Data Assimilation (DA) [9, 10], a technique that allows a numerical algorithm to continuously correct its own results in order to provide the best possible prediction.

It is well known that numerical solutions of mathematical models require periodic corrections to maintain accuracy [11, 12]. In fact, typical weather forecasting models that rely on numerically solving the Navier-Stokes equation, a partial differential equation, lose prediction fidelity after 5–10 days without model state corrections. While using numerical models to predict the future is a relatively unambiguous process, implementing observational data to correct model states isn't as straightforward due to two main challenges. First, observations themselves are not error free. A naive blending of model prediction and observations does not always yield a correct result. Therefore, the observation error must be taken into account when applying observational data to correct the model state. Secondly, our information of the system being modelled is incomplete, either in the form of missing models for some of the physical processes or in the inability to acquire the necessary measurements due to engineering difficulties.

Data assimilation [13–15] is used to determine how likely certain predictions are based on a numerical model despite the fact that some parts of the system are not completely known a priori. In the variational method that I have adopted in this work, optimization is accomplished by minimizing a cost function that typically computes the L2 norm between the predicted state and observed state. I also assume the error statistics of the observed state are known, since the nonlinear equations have closed-form solutions for the cases studied here. Another approach, when applying DA, is statistical optimization, where the uncertainty of the model error is minimized through statistical estimation methods.

In this paper, I explain an adjoint variational algorithm [16, 17] and then apply that algorithm to solve the non-linear Burgers' equation. The variational algorithm is applied to three different initial conditions to demonstrate how it is able to improve the numerical predictions.

2. Burgers' Equation

Burgers' equation is a non-linear viscous equation that has been studied extensively due to its similarity to the Navier-Stokes equation. In addition, the study of its advection and diffusion terms also applies to certain fields of financial modeling, such as the Black-Scholes equation—a model that describes the stochastic volatility of individual investments.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0 \tag{1}$$

where u is the height of a wave and ν is the viscosity.

Under a suitable transformation [18, 19], Burgers' equation is reduced to the heat equation.

$$\frac{\partial T}{\partial t} = \nu\frac{\partial^2 T}{\partial x^2} \tag{2}$$

$$u = -2\nu\frac{\frac{\partial T}{\partial x}}{T} \tag{3}$$

Since the heat equation is linear and has analytical solutions, Burgers' equation can also be solved exactly for the general initial condition $f(x)$. The analytical solution has an integral term that may not have a closed-form solution.

$$u(x,t) = \frac{\int_{-\infty}^{\infty} f(\xi)\frac{x-\xi}{t}e^{-\frac{(x-\xi)^2}{4\nu t}}d\xi}{\int_{-\infty}^{\infty} f(\xi)e^{-\frac{(x-\xi)^2}{4\nu t}}d\xi} \tag{4}$$

In the case where discontinuity is present in the solution, the solution can be expressed as

$$u(x,t) = \frac{\int_{-\infty}^{\infty} f(\xi)\frac{x-\xi}{t}e^{-G(x,t,\xi)}d\xi}{\int_{-\infty}^{\infty} f(\xi)e^{-G(x,t,\xi)}d\xi} \tag{5}$$

I examine three characteristic initial conditions [20]. Any arbitrary boundary condition can be expressed using these characteristic initial conditions through either the Fourier transform or the Green's function.

2.1. Sinusoidal Initial Condition

The sinusoidal initial condition has a closed-form solution for Burgers' equation. Since a sinusoidal function is the basis function in a Fourier transform, I can apply a Fourier transform to obtain the solution to any continuous initial condition. Let the initial condition of the heat equation solution be

$$T(x,t) = b + ae^{-\nu k^2 t}\cos(kx) \tag{6}$$

where a, b, k are suitable constants. The exact solution for Burgers' equation from Eq. (4) is

$$u(x,t) = \frac{2\nu ak e^{-\mu k^2 t}\sin kx}{b + ae^{-\nu k^2 t}\cos kx} + \frac{2\nu ak\left(-\sqrt{1-a^2/b^2}\right)}{b - a^2/b} \tag{7}$$

This exact solution will be used to assimilate the numerical solution in the variational algorithm in the next section of this paper.

2.2. Dirac Delta Function Initial Condition

In the case where the initial condition is the Dirac delta function $u(x,0) = A\delta(x)$ where A is a chosen constant, its Green's function is

$$G(x,t,\xi) = \frac{1}{2\nu}\int_{-\infty}^{\xi} A\,\delta(\phi)d\phi + \frac{(x-\xi)^2}{4\nu t} = \begin{cases} \frac{(x-\xi)^2}{4\nu t} & \xi < 0 \\ \frac{(x-\xi)^2}{4\nu t} & \xi > 0 \end{cases} \tag{8}$$

After applying Eq. (8) to the solution in Eq. (5), I have the exact solution for Burgers' equation with a Dirac delta function as the initial condition.

$$u(x,t) = \sqrt{\frac{\nu}{t}}\left(\frac{\left(\frac{A}{2\nu}-1\right)e^{-x^2/4\nu t}}{\sqrt{\pi}+\left(\frac{A}{2\nu}-1\right)\sqrt{\frac{\pi}{2}}\mathrm{erfc}\left(\frac{x}{\sqrt{4\nu t}}\right)}\right) \tag{9}$$

where $\mathrm{erfc}(x)$ is the complement of the cumulative distribution function (CDF): $\mathrm{erfc}(x) = 1 - \mathrm{erf}(x) = 1 - \frac{2}{\sqrt{\pi}}\int_0^z e^{-r^2}\,dr = \frac{2}{\sqrt{\pi}}\int_z^{\infty} e^{-r^2}\,dr$. In practice, special care needs to be taken to evaluate the initial condition of the solution in Eq. (9), note that:

$$\lim_{t\to 0, x\to 0}\frac{x}{\sqrt{4\nu t}}=0 \tag{10}$$

$$\lim_{t\to 0, x\to 0}\mathrm{erfc}\left(\frac{x}{\sqrt{4\nu t}}\right)=1 \tag{11}$$

So blindly assigning $t = 0$ in the numerical calculation of the solution at $x = 0$ will cause floating point calculation errors.

2.3. Heaviside Function Initial Condition

The Heaviside function is closely related to the Dirac delta function in that its derivative is the Dirac Delta function. The Heaviside function represents a shock initial condition to the nonlinear equation. Let

$$u(x,0) = aH(-x) + bH(x) = \begin{cases} a, & x < 0 \\ b, & x > 0 \end{cases} \tag{12}$$

Its Green's function is

$$G(x,t,\xi) = \frac{1}{2\nu}\int_0^{\xi} u\,(\phi, t=0)d\phi + \frac{(x-\xi)^2}{4\nu t} = \begin{cases} \frac{(x-\xi)^2}{4\nu t} - \frac{a}{2\nu}\xi, \xi < 0 \\ \frac{(x-\xi)^2}{4\nu t} - \frac{b}{2\nu}\xi, \xi < 0 \end{cases} \tag{13}$$

After applying the Green's function to Eq. (5), I have the exact solution for Burgers' equation with a Heaviside function as the initial condition.

$$u(x,t) = \frac{e^{\frac{b-a}{2\nu}}a\cdot\mathrm{erfc}\left(\frac{x-bt}{\sqrt{4\nu t}}\right)+b\cdot\mathrm{erfc}\left(\frac{x-at}{\sqrt{4\nu t}}\right)}{e^{\frac{b-a}{2\nu}}\mathrm{erfc}\left(\frac{x-bt}{\sqrt{4\nu t}}\right)+\mathrm{erfc}\left(\frac{x-at}{\sqrt{4\nu t}}\right)} \tag{14}$$

Once again, care needs to be taken when evaluating this numerically near $t = 0, x = 0$

$$\lim_{t\to 0, x\to 0} \frac{x-at}{\sqrt{4\nu t}}, \frac{x-bt}{\sqrt{4\nu t}} = 0 \tag{15}$$

$$\lim_{t\to 0, x\to 0} \operatorname{erfc}\left(\frac{x-at}{\sqrt{4\nu t}}\right), \operatorname{erfc}\left(\frac{x-bt}{\sqrt{4\nu t}}\right) = 1 \tag{16}$$

In the case $b = 0$, the above solution 14 simplifies to

$$u(x,t) = a - \frac{a}{1+\frac{\operatorname{erfc}\left(\frac{x}{\sqrt{4\nu t}}\right)}{\operatorname{erfc}\left(\frac{x-at}{\sqrt{4\nu t}}\right)} e^{-\frac{c}{\nu}(x-ct)}} \tag{17}$$

where c is the speed of the shock wave.

3. The Adjoint Data Assimilation Method

The fundamental premise of the data assimilation method is to improve the result of numerical integration by assimilating partial "truth" back into the initial condition over many DA time windows. In practice, the partial truth is often observational data from various sources relevant to the investigated system. Because of uncertainties in measurements, there cannot be absolute "truth". In order to reduce the bias in the assimilated result, the error variance of the observations must be treated appropriately when being assimilated with numerical predictions. The numerical prediction u^f and observation u^o are usually mixed through a weighted average to form the analysis $u^a = u^f + w\left(u^o - x^f\right)$ where the weights are based on their error covariance $w = \frac{\sigma_f^2}{\sigma_o^2 + \sigma_f^2}$. The error covariance of an observation describes how errors in one observation are related to errors in other observations. It's usually represented as a square matrix where the diagonal elements represent the variance of each observation, and the off-diagonal elements represent covariances. In this preliminary work, I have chosen to use exact solutions of Burgers' equation for the three different initial conditions as "truth" to carry out the assimilation calculation. Because the solution is exact and contains no error $\sigma_o = 0, w = 1$, I do not have to blend the numerical calculation with the "truth" by considering the error variances from both sources. In a sense, I have complete "truth" to guide the DA algorithm. In the case where the exact solution is not available, this method can still be applied, but it requires carefully considering the error covariance in both numerical prediction and observations.

The variational algorithm starts at $t = 0$ and computes the numerical solution of Burgers' equation up to a chosen time t_1, appropriate for the problem in the initial DA time window. Due to accumulation of numerical error, the numerical solution at t_1 can differ significantly from the exact solution at t_1. This solution at t_1 is used as the initial condition for numerical integration for the next DA time window until time approaches $2t_1$. The error in the solution at $2t_1$ has two sources: 1) the continual accumulation of numerical error; 2) the error in the initial condition at t_1. Since numerical error is inherent in any numerical scheme, our hope is to reduce the error in the initial condition at $t = t_1$ such that the numerical result at $2t_1$ does not drift too far away from the "truth"—in our case, the exact solution. This decrease of error in the initial condition at t_1 is accomplished through a reduction of the cost function between the solution at $2t_1$ and the exact solution. This cost function is defined as the usual L2 norm between the two solutions,

$$J(u, u^*) = \frac{1}{2}(u - u^*)^2 \tag{18}$$

where u^* is the exact solution or the "truth".

In order to reduce the cost function, I would like to find its gradient with respect to the initial condition at

t_1, $\vec{\nabla}_{u(x,t_1)} J$. Then, I can apply a gradient descent algorithm to reduce the cost function by altering the initial condition through iterations.

$$u(x,t_1)_{k+1} = u(x,t_1)_k - \alpha\ \vec{\nabla} J_{u(x,t_1)} \tag{19}$$

where k represents iteration and α is a learning rate with appropriate magnitude to reduce the cost function J without causing oscillation.

For a cost function that is not completely convex, some exploration of the function space is required. However, in the present problem, the cost function is, in fact, convex. These considerations are strikingly similar to what one typically faces in machine learning algorithms. While the engineering details are different, both approaches seek to minimize the cost function in the functional space.

For a more complex non-linear system that has multiple spatial dimensions, the boundary conditions can also be improved through the gradient descent algorithm. In the derivation, I will show that assimilating the boundary condition is not necessary for the 1D Burgers' equation.

Finding the gradient of the cost function with respect to the initial condition function at t_1 is equivalent to determining the optimal $u(x,t_1)$ such that the cost function at $2t_1$ can be minimized. This is clearly a variational problem—I must optimize a function $(u(x,t_1))$ in order to reduce the cost function at $2t_1$.

3.1. Variational Method

One of the simplest examples of a variational problem is determining the function that minimizes the L2 distance between two points on a plane. Intuitively, the solution is a line that connects the two points; however, these problems inspired Bernoulli, Euler, and Lagrange to generalize the approach to find an optimal function for variational problems. In ordinary calculus, optimization problems entail determining where a function is optimized for certain points, whereas in variational calculus, optimization problems seek to find an optimized function.

In the case of the L2 distance, its Euler-Lagrange equation is

$$I = \int_a^b F\,(x,y,y')dx = \int_a^b \sqrt{1+y'^2}\,dx \tag{20}$$

$$\frac{d}{dx}\frac{\partial F(x,y,y')}{\partial y'} - \frac{\partial F}{\partial y} = 0 \tag{21}$$

where the second equation is derived from the variational principle. I adapt and apply the principle to optimize a system governed by its state equation

$$\vec{\mathbf{L}}(\vec{\mathbf{u}},\ \vec{\boldsymbol{\gamma}}) = \vec{\mathbf{f}} \tag{22}$$

where $\vec{\boldsymbol{\gamma}}$ is a set of model parameters that I can assimilate to optimize the cost function, $\vec{\mathbf{L}}$ is the set of equations, and $\vec{\mathbf{f}}$ is the set of forcing terms.

The cost function is defined as the L2 norm between model predictions and the "truth". In the present case, $u^* = u^o = u_{\text{exact}}$.

$$J = \frac{1}{2}\int_0^T \int_0^L F(x,t,\vec{\mathbf{u}},\ \vec{\mathbf{u}}^o,\vec{\boldsymbol{\gamma}})\delta(x-x_i)dxdt \tag{23}$$

where x_i are the discrete points at which Burgers' equation is solved and $\delta(x-x_i)$ is the Dirac Delta function. $\vec{\boldsymbol{\gamma}}$ represents the initial condition of a DA window.

The variation of $\vec{\mathbf{u}}$ with respect to $\vec{\boldsymbol{\gamma}}$ in the direction of $\delta\vec{\boldsymbol{\gamma}}$ is

$$\delta\ \vec{\mathbf{u}} = \vec{\nabla}_{\vec{\gamma}}\ \vec{\mathbf{u}} \cdot \delta\ \vec{\boldsymbol{\gamma}} \tag{24}$$

where $\vec{\nabla}_{\vec{\gamma}}\ \vec{\mathbf{u}}$ is the gradient of $\vec{\mathbf{u}}$ with respect to the model parameter vector $\vec{\boldsymbol{\gamma}}$, leading to an Hessian matrix $\frac{\partial u_i}{\partial \gamma_j}$.

Similarly, the variation of the cost function with respect to the model parameter is

$$\delta J = \vec{\nabla}_{\vec{\gamma}} J \cdot \delta\ \vec{\boldsymbol{\gamma}} = \vec{\nabla}_{\vec{\mathbf{u}}} \cdot \delta\ \vec{\mathbf{u}} \tag{25}$$

Now, I proceed to optimize the variation of the state equation with respect to the model parameter.

$$\delta\vec{\mathbf{L}} = \vec{\boldsymbol{\nabla}}_{\vec{\gamma}}\vec{\mathbf{L}}\delta\vec{\gamma} + \vec{\boldsymbol{\nabla}}_{\vec{\mathbf{u}}}\vec{\mathbf{L}}\delta\vec{\mathbf{u}} = 0 \tag{26}$$

Multiplying this with the adjoint function $\vec{\mathbf{v}}$ (equivalent to the Lagrange multiplier) and integrating over space and time yields

$$\langle\vec{\boldsymbol{v}} \cdot \vec{\boldsymbol{\nabla}}_{\vec{\gamma}}\vec{\mathbf{L}}\delta\vec{\gamma}\rangle + \langle\vec{\boldsymbol{v}} \cdot \vec{\boldsymbol{\nabla}}_{\vec{\mathbf{u}}}\vec{\mathbf{L}}\delta\vec{\mathbf{u}}\rangle = 0 \tag{27}$$

The bracket symbol in the above equation means $\int_0^T \int_0^L u\,(x,t)dxdt$. After some manipulation, I can find the solution to the adjoint function (similar to the Lagrange multiplier).

$$\vec{\nabla}_{\vec{\mathbf{u}}}\vec{\mathbf{L}}(\vec{\mathbf{v}})\ \vec{\mathbf{v}} = -\ \vec{\nabla}_{\vec{\mathbf{u}}} J(\vec{\mathbf{v}}) \tag{28}$$

With the adjoint solution $\vec{\mathbf{v}}$, the directional derivative of the cost function can be evaluated.

$$\vec{\boldsymbol{\nabla}}_{\vec{\gamma}} J\delta\vec{\gamma} = \langle\vec{\boldsymbol{v}} \cdot \vec{\boldsymbol{\nabla}}_{\vec{\gamma}}\vec{\mathbf{L}}\delta\vec{\gamma}\rangle \tag{29}$$

With the gradient of the cost function, I can then apply a gradient descent algorithm to update the model parameter $\vec{\boldsymbol{\gamma}}$ through iterations.

$$\vec{\boldsymbol{\gamma}}_{k+1} = \vec{\boldsymbol{\gamma}}_k - \alpha\vec{\nabla}_{\vec{\gamma}} J\delta\vec{\boldsymbol{\gamma}} \tag{30}$$

I now apply this general approach to find the adjoint equation of Burgers' equation.

3.2. Adjoint Equation

I start with Burgers' equation and its initial and boundary conditions.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x\,\partial x} = 0, x \in [0, L] \tag{31}$$

$$u(0,t) = \phi_1(t), u(L,t) = \phi_2(t), u(x,t=0) = u_0(t) \tag{32}$$

where ϕ_i are the boundary conditions and u_0 is the initial condition at the beginning of each DA window.

First I introduce the Tangent Linear Model (TLM) equation that captures the effect of perturbing the initial condition and boundary condition similar to Eq. (26). Let $\tilde{u}$ be the solution to Burgers' equation after perturbing the initial condition u_0 and boundary condition ϕ_i by some small number α (learning rate). The TLM state function is defined as $\hat{u}$ as shown in Eq. (33) below.

$$\begin{gathered} u = u(u_0, \phi_1, \phi_2) \\ \tilde{u} = \tilde{u}(u_0 + \alpha\delta u_0, \phi_1 + \alpha\delta\phi_1, \phi_2 + \alpha\delta\phi_2) \\ \hat{u} = \lim_{\alpha\to 0}\frac{\tilde{u} - u}{\alpha} \end{gathered} \tag{33}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x\,\partial x} = 0 \tag{34}$$

$$\frac{\partial \tilde{u}}{\partial t} + \tilde{u}\frac{\partial \tilde{u}}{\partial x} - \nu\frac{\partial^2 \tilde{u}}{\partial x\,\partial x} = 0 \tag{35}$$

Using the definition of the TLM state function 33, the two sets of solutions from Burgers' equation are subtracted without perturbation 35 and with perturbations 35. Then, the limit is taken as $\alpha \to 0$, to yield

$$\frac{\partial \hat{u}}{\partial t} + \lim_{\alpha\to 0} \frac{\tilde{u}\frac{\partial \tilde{u}}{\partial x} - u\frac{\partial u}{\partial x}}{\alpha} - \nu \frac{\partial^2 \tilde{u}}{\partial x\, \partial x} = 0 \tag{36}$$

The second term further evaluates to

$$\lim_{\alpha\to 0} \frac{\tilde{u}\dfrac{\partial \tilde{u}}{\partial x} - u\dfrac{\partial u}{\partial x}}{\alpha} = \lim_{\alpha\to 0} \frac{1}{2} \frac{\dfrac{\partial \tilde{u}^2}{\partial x} - \dfrac{\partial u^2}{\partial x}}{\alpha} = \lim_{\alpha\to 0} \frac{1}{2} \frac{\dfrac{\partial \tilde{u}^2 - u^2}{\partial x}}{\alpha} = \lim_{\alpha\to 0} \frac{1}{2} \frac{\dfrac{\partial(\tilde{u}+u)(\tilde{u}-u)}{\partial x}}{\alpha} = \frac{1}{2}\frac{\partial(u+u)\hat{u}}{\partial x} = \frac{\partial u\hat{u}}{\partial x}$$

I used the definitions: $\lim_{\alpha\to 0} \tilde{u} = u, \hat{u} = \lim_{\alpha\to 0} \frac{\tilde{u}-u}{\alpha}$ in the last steps of the derivation. Then, I substitute this result back into Eq. (36). I can define the TLM equation with its associated boundary and initial conditions.

$$\frac{\partial \hat{u}}{\partial t} + \frac{\partial(u\hat{u})}{\partial x} - \nu \frac{\partial^2 \hat{u}}{\partial x\, \partial x} = 0 \tag{37}$$

$$\hat{u}(0,t) = \psi_1(t), \hat{u}(L,t) = \psi_2(t), \hat{u}(x,0) = \hat{u}_0(x) \tag{38}$$

Next, I multiply the TLM Eq. (37) with the adjoint function $\vec{\mathbf{v}}$ and integrate over the domain like in the general Eq. (27).

$$\int_0^L \int_0^T \left(\frac{\partial \hat{u}}{\partial t} + \frac{\partial(u\hat{u})}{\partial x} - \nu \frac{\partial^2 \hat{u}}{\partial x\, \partial x}\right) v dx dt = 0 \tag{39}$$

I dropped the vector arrow from $\vec{\mathbf{v}}$ in the above equation for clarity. It's a dot product between the TLM equation and the adjoint function in the space-time domain. I require the adjoint function to satisfy the following boundary and initial conditions

$$v(x=0) = v(x=L) = 0, v(t=T) = 0 \tag{40}$$

The derivation from Eq. (39) involves three separate evaluations using integration by parts. I will show how to evaluate the first term $\int_0^L \int_0^T \frac{\partial \hat{u}}{\partial t} v\, dx dt$; the other two terms can be evaluated similarly.

$$\begin{aligned}
\int_0^T \frac{\partial \hat{u}}{\partial t} v dt &= \hat{u}v|_0^T - \int \hat{u}\frac{\partial v}{\partial t} dt \\
&= \hat{u}(x,T)v(x,T) - \hat{u}(x,0)v(x,0) - \int \hat{u}\frac{\partial v}{\partial t} dt \\
&= -\hat{u}(x,0)v(x,0) - \int \hat{u}\frac{\partial v}{\partial t} dt \\
\int_0^L \int_0^T \frac{\partial \hat{u}}{\partial t} v dt dx &= -\int_0^L \hat{u}\,(x,0)v(x,0)dx - \int_0^L \int_0^T \hat{u}\frac{\partial v}{\partial t} dt dx
\end{aligned}$$

I used the initial condition for the adjoint function $v(t=T) = 0$ from Eq. (40). After some manipulation, Eq. (39) becomes

$$\int_0^L \int_0^T \hat{u}\left(-\frac{\partial v}{\partial t} - u\frac{\partial v}{\partial x} - \nu \frac{\partial^2 v}{\partial x\, \partial x}\right) dx dt = \int_0^L -\hat{u}(x,0)v(x,0)dx + \int_0^T \left(\nu\psi_2 \frac{\partial v}{\partial x} - \nu\psi_1 \frac{\partial v}{\partial x}\right) dt \tag{41}$$

If I introduce no perturbations for the boundary condition, then $\psi_1(0,t) = \psi_2(L,t) = 0$ and the above equation simplifies to

$$\int_0^L \int_0^T \hat{u}\left(-\frac{\partial v}{\partial t} - u\frac{\partial v}{\partial x} - \nu \frac{\partial^2 v}{\partial x\, \partial x}\right) dx dt = \int_0^L -\hat{u}(x,0)v(x,0)dx \tag{42}$$

The gradient $\vec{\nabla}_{\vec{\gamma}} J$ based on the L2 norm evaluates to

$$J(\vec{\gamma}) = \frac{1}{2}\int (u - u^o)^2 dx \tag{43}$$

$$\tilde{J}(\vec{\gamma}+\alpha\delta\vec{\gamma})-J(\vec{\gamma})=\frac{1}{2}\int(\tilde{u}-u^{o})^{2}-(u-u^{o})^{2}dx \tag{44}$$

I define $\hat{J}=\lim_{\alpha\to 0}\frac{\tilde{J}-J}{\alpha}$. Using the definitions $\lim_{\alpha\to 0}\tilde{u}=u, \hat{u}=\lim_{\alpha\to 0}\frac{\tilde{u}-u}{\alpha}$, I determine the directional derivative.

$$\hat{J}=\vec{\nabla}_{\vec{\gamma}}J=\vec{\nabla}J\cdot\ \vec{\gamma}=\lim_{\alpha\to 0}\frac{\tilde{J}-J}{\alpha}=\int\hat{u}(u-u^{o})dxdt \tag{45}$$

Note that at $t=0$,

$$\hat{u}(x,t=0)=\lim_{\alpha\to 0}\frac{\tilde{u}-u}{\alpha}=\lim_{\alpha t\to 0}\frac{u+\alpha\vec{\gamma}-u}{\alpha}=\vec{\gamma} \tag{46}$$

Comparing Eqs. (42), (45), and (46), I recognize that I can define the adjoint equation as

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial^{2}v}{\partial x\,\partial x}=u-u^{o} \tag{47}$$

With the boundary and initial conditions specified in Eq. (40), the gradient of the cost function can be evaluated using the right hand side of Eq. (42).

$$\begin{gathered}\int_{0}^{L}\nabla J\cdot\vec{\gamma}dx=\int_{0}^{L}-\hat{u}(x,0)v(x,0)dx=-\int_{0}^{L}\vec{\gamma}v(x,0)dx\\ \vec{\nabla}J=-v(x,0)\end{gathered} \tag{48}$$

Now I have the gradient of the cost function 48 which is the solution of the adjoint Eq. (47). I can now update the initial condition at the beginning of the DA window using the gradient descent algorithm 19.

$$u(x,0)_{k+1}=u(x,0)_{k}+\alpha v(x,0) \tag{49}$$

where $t=0$ is the beginning of each DA window corresponding to Eq. (30) in the general solution.

3.3. Numerical Evaluation

Next, I verify the adjoint Eq. (48) for Burgers' Eq. (1), applied during the data assimilation algorithm 49 using the three initial conditions discussed previously Sections 2.1, 2.2, 2.3. In this process, I need to solve Burgers' equation numerically. To do so, I have chosen to use the 2nd-order predictor-corrector algorithm for time $\frac{\partial u}{\partial t}$ and the 4th-order finite difference algorithm for $\frac{\partial u}{\partial x}$ and $\frac{\partial^2 u}{\partial x\,\partial x}$. At the boundaries, I use forward and backward difference approximations, while in the interior, I use central difference approximations. The algorithm used step sizes that were restricted by the Courant–Friedrichs–Lewy (CFL) condition to perform time integration. Other suitable methods for solving Burger's equation include finite volume method which is more tolerant of sharp gradient because it's integral based instead of difference based [21], spectral method since the analytical solution of the Burger's equation can be expressed in terms of Fourier series [22], and discontinuous Galerkin finite element method that is also integral based therefore well suited for problems with shock waves and sharp gradients [23]. A more complete survey of these methods can be found in [24–28].

Fig. 1 shows the result of the simulation compared with exact solutions for the sinusoidal initial condition at both low and high viscosities. The numerical result departs quite significantly at the end of the DA window. This is equivalent to a Numerical Weather Prediction (NWP) system that has lost fidelity in its predicted results. To show that this occurred over time, in Fig. 2, I demonstrate the numerical result and exact result at time steps 300 to 600. I can see that the deviation grows over increasing time integration steps.

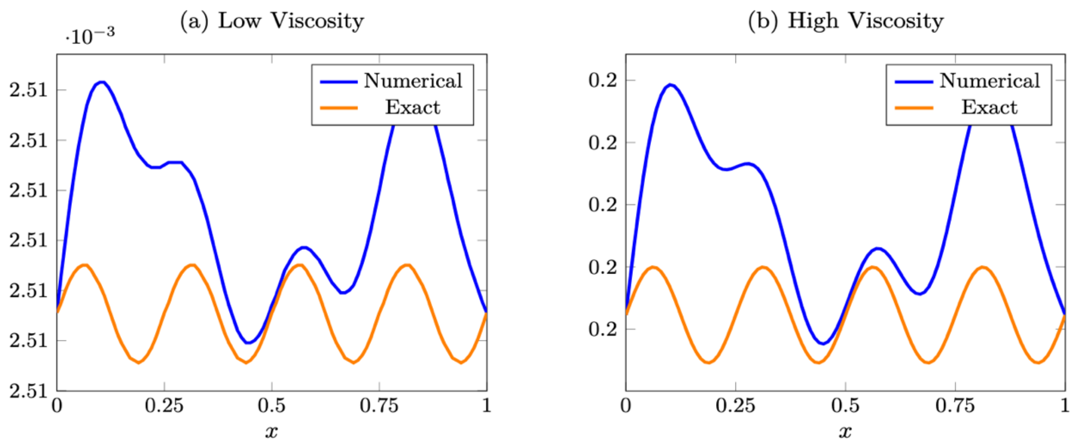

Fig. 1. Numerical and exact solutions of the sinusoidal initial condition Eq. (6) with low and high viscosities. (a) $\nu = 0.01, a = 1, b = 200, k = 8\pi$ starting from initial condition. Results shown after 600 steps of time integration. The numerical solution differs from the exact solution noticeably near the peak and trough of the sinusoidal wave. (b) similar to (a) but with higher viscosity $\nu = 0.8, a = 1, b = 200, k = 8\pi$ starting from initial condition 6. Note the difference in scale of the y-axis in the two panels. Created by student researcher.

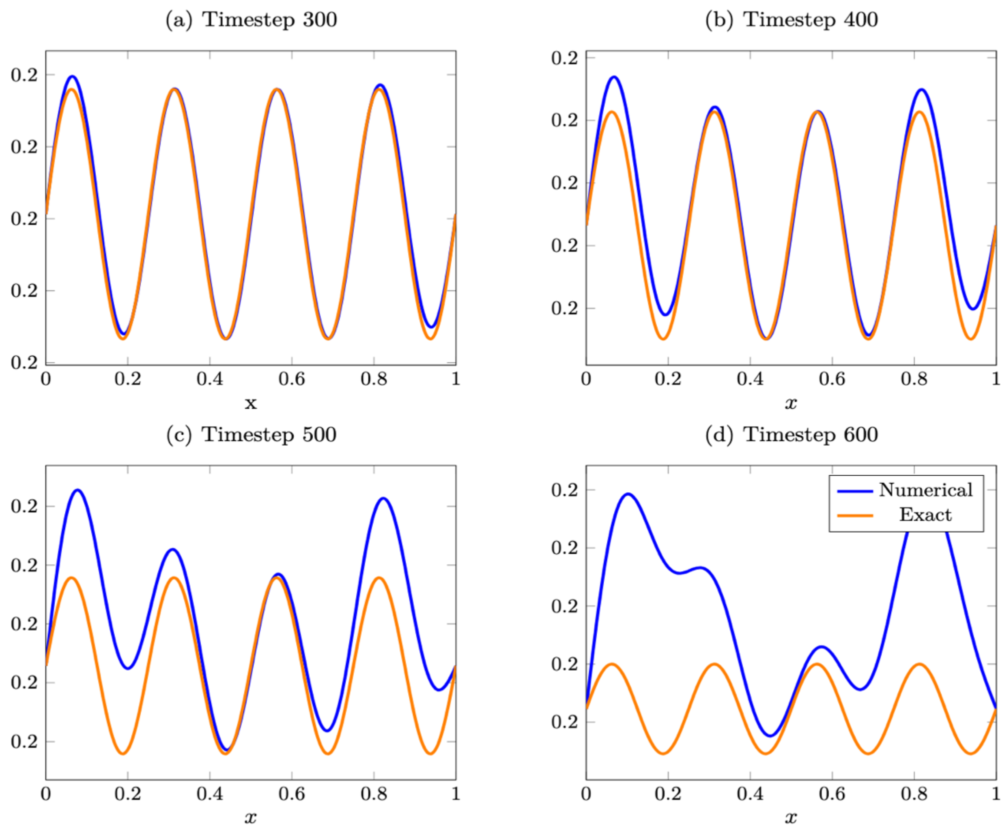

Fig. 2. Comparison between numerical calculation and exact values of u at various time steps [(a) 300; (b) 400; (c) 500; and (d) 600] for high viscosity $\nu = 0.8$. It can be seen that the numerical solution gradually deviates away from the exact solution to such an extent that the slope at the right boundary becomes reverted. However, as shown below in Fig. 6, the DA algorithm is able to correct the bias in the numerical solution after about 100 iterations. Created by student researcher.

Figs. 3 and 4 shows the comparison between numerical and exact solutions for the Dirac delta and Heaviside step initial conditions at both low and high viscosities. In both the delta and the step function solutions I can see the shock formation patterns more visibly compared to the sinusoidal solution. The non-linear term in Burgers' equation causes the solution to move to the right where $u > 0$ and left where $u < 0$. The magnitude of the movement depends on the magnitude of u. As a result, the more positive part of the solution shifts to the right more quickly than the less positive part, causing a shock boundary to form over time. This behavior is most obvious at low viscosity, causing difficulties in numerical solvers to capture the shock formation accurately. At high viscosity, the gradient in u is smeared out—the shock front barely forms. I can see that the agreement between the numerical and exact solutions at high viscosity is better for both the delta and the step function initial conditions.

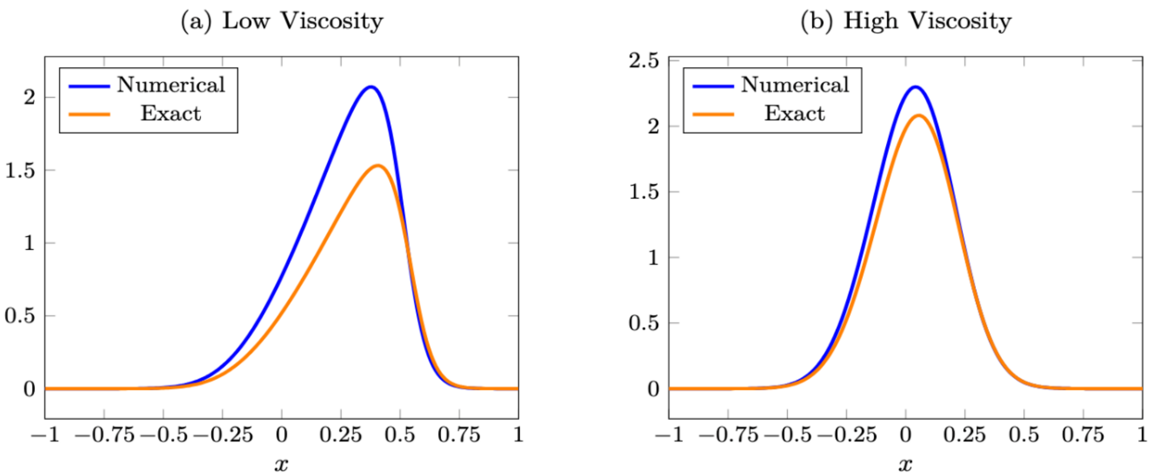

Fig. 3. Numerical and exact solutions of the Dirac delta function initial condition Eq. (9) with low and high viscosities. (a) Low viscosity: $\nu = 0.08, A = 1$ starting from initial condition. The solution shows correct shock formation behavior as the solution moves to the right as predicted by Burgers' equation. (b) High viscosity: $\nu = 0.8, A = 1$. Higher viscosity reduces the gradient in the function u. Created by student researcher.

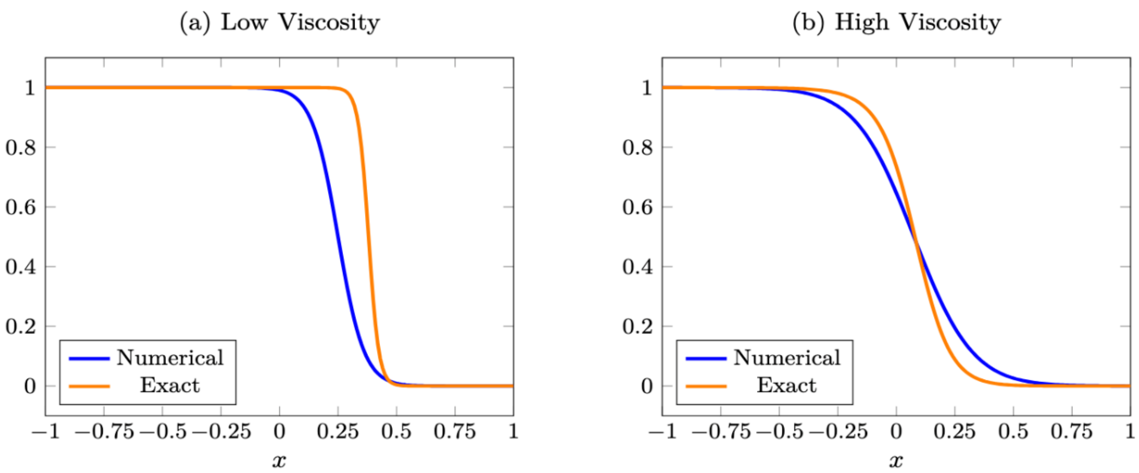

Fig. 4. Numerical and exact solutions of the Heaviside step function initial condition Eq. (17) with low and high viscosities. (a) Low viscosity: $\nu = 0.03, a = 1, c = 1$ starting from initial condition. The shock formation is distinct at low viscosity. (b) High viscosity: $\nu = 0.1, a = 1, c = 1$. Created by student researcher.

Fig. 5(a) shows the result of the simulation compared with exact solutions after DA correction is applied. The improvement to the numerical prediction is more striking when the graphs are animated—indeed, I can clearly observe the DA-corrected prediction approaches the exact solution during the iterations. Fig. 5(b) shows the decrease in cost function over the iteration steps. The solution of the adjoint equation in Fig. 5(c) indicates where the numerical prediction differs from the exact solution $u - u^o$. Over time, as the numerical prediction approaches the exact solution, the magnitude of the adjoint solution decreases.

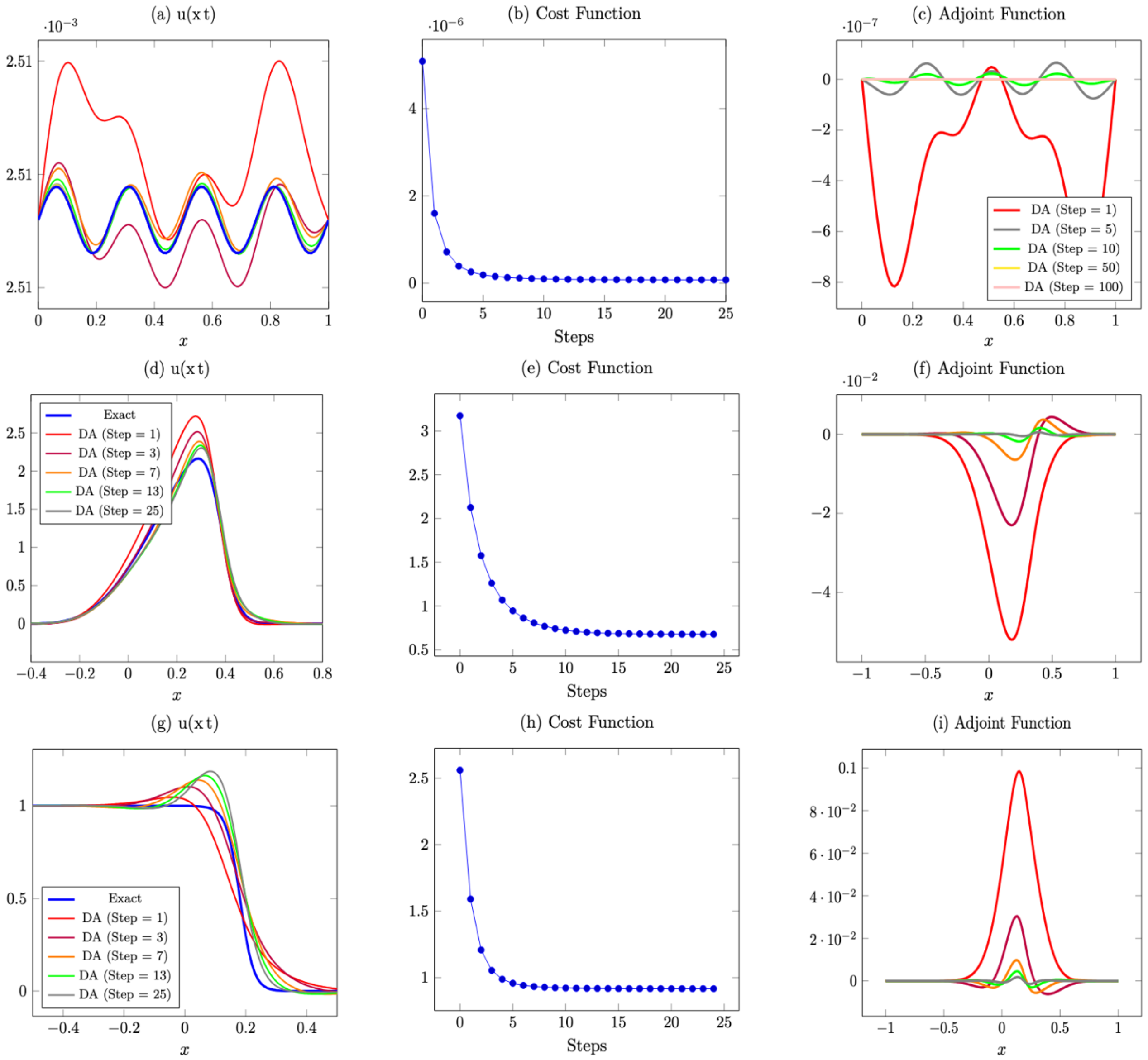

Fig. 5. (a) Sinusoidal Initial Condition, $\nu = 0.01$, Learning Rate = 2. The exact solution is shown in blue. Data assimilation approximation after 1 step is shown in red, after 2 steps is shown in purple, after 3 steps in orange, after 5 steps in green, after 10 steps in gray, and after 50 steps in yellow. (b) Sinusoidal Solution Cost Function (c) Sinusoidal Solution Adjoint Function. (d) Dirac Delta Function Initial Condition $\nu = 0.08$, Learning Rate = 5 (e) Dirac Delta Solution Cost Function (f) Dirac Delta Solution Adjoint Function (g) Heaviside Function Initial Condition $\nu = 0.03$, Learning Rate = 2 (h) Heaviside Solution Cost Function (i) Heaviside Solution Adjoint Function. Created by student researcher.

Fig. 6(a)–(c) shows the result of the simulations at high viscosity for each setup. In general the result is similar to the low viscosity case as DA is able to correct the bias in the initial condition and improve the numerical result during the DA time window.

When analyzing the effect of DA corrections, I find the sinusoidal case to be the most interesting. As the time step integration continues, numerical results deviate significantly from the exact solution as shown in Fig. 2. When DA is applied, the solution from the adjoint equation is able to correct the deviation in the numerical solution as shown in Fig. 5(a) and Fig. 6(a) for the sinusoidal setup at both low and high viscosities.

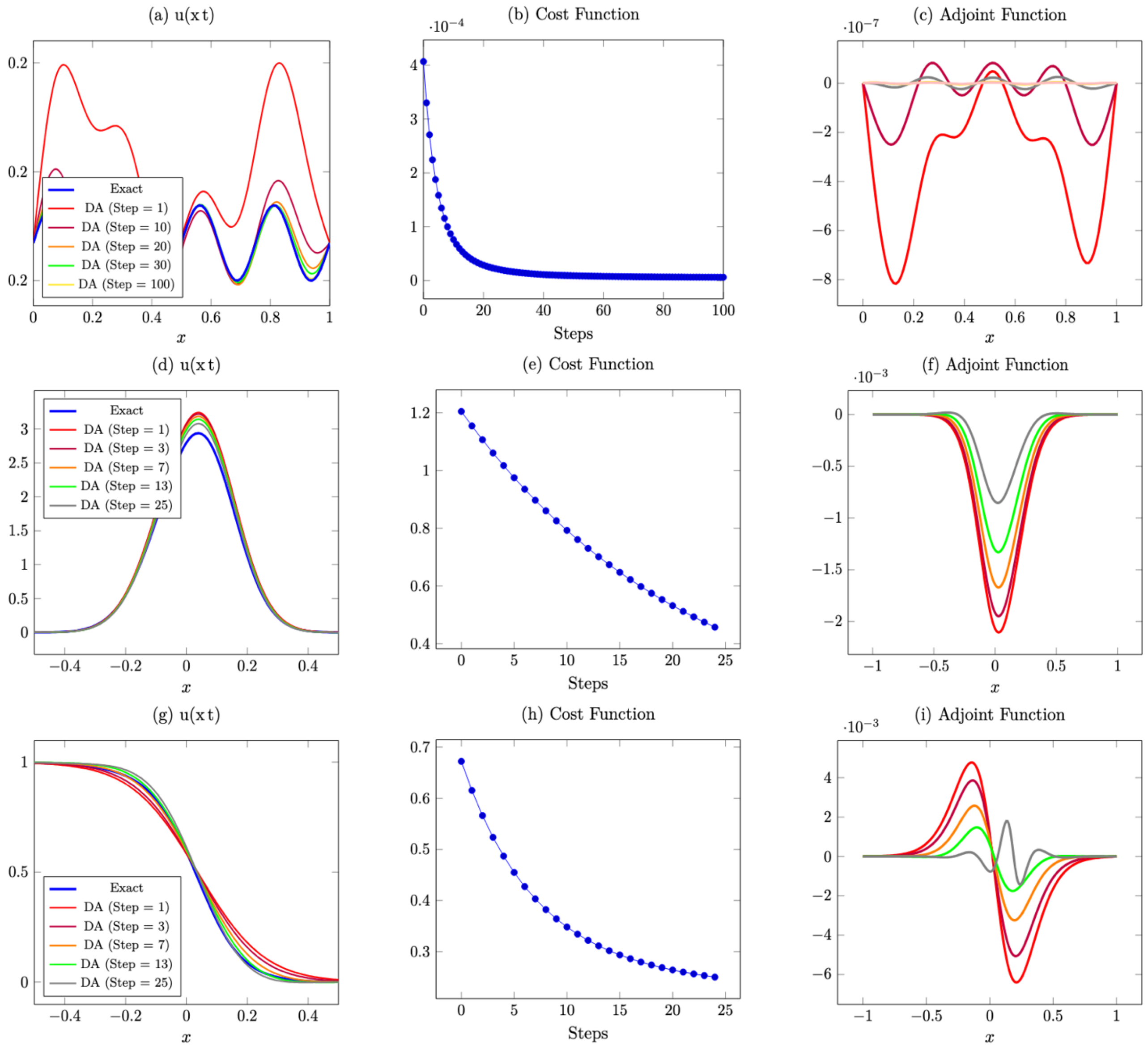

Fig. 6. (a) Sinusoidal Initial Condition, $\nu = 0.8$, Learning Rate = 30 (b) Sinusoidal Solution Cost Function (c) Sinusoidal Solution Adjoint Function. (d) Dirac Delta Function Initial Condition $\nu = 0.8$, Learning Rate = 5 (e) Dirac Delta Solution Cost Function (f) Dirac Delta Solution Adjoint Function (g) Heaviside Function Initial Condition $\nu = 0.1$, Learning Rate = 2 h Heaviside Solution Cost Function (i) Heaviside Solution Adjoint Function. Created by student researcher.

4. Conclusion

I examine Burgers' equation and its exact solutions for three different key initial conditions: the sinusoidal function, Dirac delta function, and Heaviside step function. These key initial conditions can be used as a basis to define any arbitrary initial conditions. Burgers' equation is unique in that it's a non-linear partial differential equation with known exact solutions. This allows for verification of results which makes quantifying the utility of new methods—from numerical to data assimilation techniques—relatively easy.

I presented the variational calculus method which I applied to both define the tangent linear equation and derive the adjoint equation for Burgers' equation. The gradient of the cost function with respect to the initial condition is the solution of the adjoint equation backward in time. Both Burgers' equation and the adjoint equation are solved with a predictor-corrector scheme in time and a 4th-order accurate finite difference scheme in space. The entire DA algorithm and all of its numerical complexities are implemented in a python computer program.

The integration of data assimilation in numerical simulation significantly improved the numerical approximation results as shown in Figures 5, 6, and 2. In the case of the positive definite sinusoidal function initial condition, the numerical solution deviated significantly from the exact solution at the end of the DA window. The adjoint solution is able to correct the initial condition at the beginning of the DA window and manages to successfully bring the numerical solution to close agreement with the exact solution again. In the cases of the shock-forming delta and step function initial conditions at low viscosity, data assimilation once again proves itself effective as it brings the numerical solution to close agreement with the exact solution.

It's worth noting that data assimilation can improve numerical models by providing better initial conditions, boundary conditions, or model parameters provided those statistics are available. It does not, however, improve the physics of the numerical model directly. Instead, the physics of the numerical model is captured by its governing equations. There are two approaches toward improving the overall quality of the numerical prediction: improving the physics of the numerical model, and better integrating the observations as part of the numerical simulation. DA effectively addresses the second approach towards solving the issue of poor numerical approximations—as shown in this work.

The data assimilation approach to Burger's equation is not limited to the 1D problem as shown in this work. It can easily be extended to 2D or 3D problems. It would be interesting to set up numerical validations of the Burger's equation in 2D and 3D based on the recent work by [22] where exact analytical solutions for 2D and 3D Burger's equation were found for a particular set of boundary and initial conditions. Continuing from here, the approach presented here can also be extended to solve the full 3D Navier-Stokes equations corrected by observational data.

Appendix of Data Assimilation

The conception of data assimilation originated in the numerical weather forecasting community through its access to both numerical weather forecasting systems and observational networks of atmospheric and ocean states. DA has shown its continual value to improve the quality and accuracy of Numerical Weather Predictions (NWPs) by mixing numerical results with various sources of measurements. In recent years, apart from its applications in traditional NWP systems, more applications of DA have been explored in 1) Global Positioning System (GPS)/Meteorology (MET) Data Assimilation; 2) Hurricane Initialization by Data Assimilation; 3) Estimation of Ocean Mixed Layer by Data Assimilation [7]; 4) Physics-based hydrological modeling [29]. While the DA method can significantly improve the quality of the numerical prediction, it requires iterations performed backward in time and storing the observational data within the DA window. This makes the DA approach more computationally expensive than conventional methods.

Machine Learning and Deep Learning (ML/DL) [30] is also being integrated with DA to further the prediction skills of NWPs. While DA has the advantage of incorporating domain-specific physics knowledge of the system, it is incapable of improving the physics of the model while also being computationally expensive. However, ML/DL can provide better insight into parameters needed in physical models while reducing computational cost once trained. ML/DL is capable of extracting non-linear information about a system that scientists still have inadequate knowledge about. Therefore, combining ML/DL with DA may provide a better overall solution to improving physical aspects—known and unknown—of a NWP system.

In addition, scientists at the Large Helical Device (LHD) are able to incorporate DA algorithms [31] to provide advanced control in future fusion reactors. To address the high computational demand when performing DA—due to the requirement of many iterations to minimize cost function—researchers are looking into the quantum annealing process [32].

The data assimilation approach also goes by different names in various other fields such as state estimation [33]; optimization [34]; history matching [35]; retrieval production [36]; and inverse modelling [37]. Data assimilation attempts to solve a very fundamental problem in any predictive system—how to best mix model predictions and real-world observations to deal with uncertainties. Therefore, I expect data assimilation techniques to be explored in many more fields in the future.

Conflict of Interest

The author declares no conflict of interest.

Acknowledgment

I am very grateful for the support I received from my mentor, Dr. Fei Liu, who provided inspiration and guidance in this research work. I would also like to thank my parents for their continued encouragement and support.

References

[1] Lorenc, A. C. (1981). A global three-dimensional multivariate statistical interpolation scheme. *Monthly Weather Review*, *109(4)*, 701–721.

[2] Lorenc, A. C. (1984). Analysis methods for the quality control of observations. *Proceedings of the Workshop on Use and Quality Control of Meteorological Observations*.

[3] Lorenc, A. C. (1986). Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, *112(474)*, 1177–1194.

[4] Dimet, F.-X., & Talagrand, O. Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects. *Tellus A*, *38A(2)*, 1986.

[5] Tang, W.-H., *et al*. (2018). Data assimilation and multisource decision-making in systems biology based on unobtrusive internet-of-things devices. *Biomed Eng Online*.

[6] Heaney, C. E., *et al.* (2024). Data assimilation with machine learning for dynamical systems: Modelling indoor ventilation. *Physica A: Statistical Mechanics and its Applications*.

[7] Wang, B., Zou, X., & Zhu, J. (2000). Data assimilation and its applications. *Proceedings of the National Academy of Sciences*, *97(21)*, 11143–11144.

[8] Jin, X., *et al*. A review of data assimilation of remote sensing and crop models. *European Journal of Agronomy*, 2018.

[9] Bouttier, F., & Courtier, P. (1999). *Data Assimilation Concepts and Methods*.

[10] Bannister, R. N. (2017). A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*.

[11] Lorenz, E. N. (1996). Predictability: A problem partly solved. *Proceedings of the Seminar on predictability*.

[12] Baumhefner, D. P., & Tribbia, J. J. (2004). Scale interactions and atmospheric predictability: An updated perspective. *Mon. Weather Rev*.

[13] Daley, R. (1991). *Atmospheric Data Analysis*. Cambridge University Press: Cambridge, UK.

[14] Kalnay, E. (2002). *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press: Cambridge, UK.

[15] Rabier, F. (2005). Overview of global data assimilation developments in numerical weather prediction centres. *Q. J. R. Meteorol. Soc*.

[16] Courtier, P., & Talagrand, O. (1987). Variational assimilation of meteorological observations with the adjoint vorticity equation. ii: Numerical results. *Quarterly Journal of the Royal Meteorological Society*, *113(478)*, 1329–1347.

[17] Talagrand, O. & Courtier, P. (1987). Variational assimilation of meteorological observations with the adjoint vorticity equation. i: Theory. *Quarterly Journal of the Royal Meteorological Society*, *113(478)*, 1311–1328.

[18] Hopf, E. (1950). The partial differential eauation $u_t + uu_x = u_{xx}$. *Communications on Pure and Applied Mathematics*.

[19] Cole, J. D. (1951). On a quasi-linear parabolic equation occuring in aerodynamics. *Quarterly of Applied Mathematics*.

[20] Salih, A. (2016). *Burger's Equation*.

[21] Sheng, Y., & Zhang, T. (2018). The finite volume method for two-dimensional Burgers' equation. *Personal and Ubiquitous Computing*, *22*, 1133–1139.

[22] Qiang, G., & Zou, M. Y. (2017). An analytical solution for two and three dimensional nonlinear Burgers' equation. *Applied Mathematical Modelling*, *45*, 255–270.

[23] Shao, L., Feng, X., & He, Y. (2011). The local discontinuous Galerkin finite element method for Burger's equation. *Mathematical and Computer Modelling*, *54(11–12)*, 2943–2954.

[24] Liao, W. Y. (2008). An implicit fourth-order compact finite difference scheme for one-dimensional Burger's equation, *Appl. Math. Comput., 206(2008)*, 755–764.

[25] Gustafson, B., Kress, H. O., & Oliger, J. (1995). Time dependent problems and difference methods. New York: *John Wiley and Sons*.

[26] Hassanien, I. A., Salama, A. A., & Hosham, H. A. (2005). Fourth-order finite difference method for soving Burger's equation. *Appl. Math. Comput., 170(2005)*, 781–800.

[27] Kutluay, S., Bahadir, A. R., & Ozdes, A. (1999). Numerical solution of one-dimensional Burger's equation: Explicit and exact-explicit finite difference methods. *J. Comput. Appl. Math., 103(1999)*, 251–261.

[28] Caldwell, J., Wanless, P., & Cook, A. E. (1981). A finite element approach to Burger's equation. *Appl. Math. Model*, *5(1981)*, 189–193.

[29] Camporese, M., & Girotto, M. (2022). Recent advances and opportunities in data assimilation for physics-based hydrological modeling. *Frontiers in Water*.

[30] Bonavita, M., Geer, A., Laloyaux, P., Massart, S., & Chrust, M. (2021). Data assimilation or machine learning? *ECMWF Newsletter*, *167*.

[31] Morishita, Y., Murakami, S., Kenmochi, N., *et al*. (2024). First application of data assimilation-based control to fusion plasma. *Nature*.

[32] Kotsuki, S., Kawasaki, F., & Ohashi, M. (2024). Quantum data assimilation: A new approach to solving data assimilation on quantum annealers. *Nonlin. Processes Geophys*.

[33] Wunsch, C. (2012). *Discrete Inverse and State Estimation Problems: With Geophysical Fluid Applications.* Cambridge, UK: Cambridge University Press.

[34] Biegler, L. T. (1997). *Large-Scale Optimization with Applications: Part I: Optimization in Inverse Pro*

blems and Design. New York, NY: Springer-Verlag.

[35] Emerick, A. (2012). *History Matching and Uncertainty Characterization.* Saarbrucken, Germany: Lambert Academic Publishing.

[36] Rodgers, C. D. (2000). *Inverse Methods for Atmospheric Sounding: Theory and Practice.* Singapore: World Scientific Publishing.

[37] Tarantola, A. (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation.* Philadelphia, PA: SIAM.