Efficient adaptive step size control for exponential integrators

Pranab Jyoti Deka

with
Lukas Einkemmer

Exponential Rosenbrock Schemes

Initial Value Problem (1D)

$$\frac{\partial u}{\partial t} = f(u) \qquad u(t=0) = u^0$$

$$\frac{\partial u}{\partial t} = \mathcal{J}(u)\, u + \mathcal{F}(u)$$

Linear term

Nonlinear remainder

Exponential Rosenbrock Schemes

Rosenbrock-Euler scheme (2nd order)

(Hochbruck 2006)

$$u^{n+1} = u^n + \varphi_1(\mathcal{J}(u^n)\Delta t)f(u^n)\,\Delta t$$

$$\varphi_{l+1}(z) = \frac{1}{z}\left(\varphi_l(z) - \frac{1}{l!}\right), \quad l \geq 1$$

$$\varphi_0(z) = e^z$$

Matrix Exponential

Exponential Rosenbrock Schemes

EXPRB43 (3rd order error estimate)

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2}\varphi_1(\frac{1}{2}\,\cdot) & & \\
1 & 0 & \varphi_1 & \\
\hline
 & \varphi_1 - 14\varphi_3 + 36\varphi_4 & 16\varphi_3 - 48\varphi_4 & -2\varphi_3 + 12\varphi_4 \\
 & \varphi_1 - 14\varphi_3 & 16\varphi_3 & -2\varphi_3
\end{array}
$$

Butcher Tableau

(Hochbruck & Ostermann 2010)

Internal Stages

$$
\begin{aligned}
a^n &= u^n + \varphi_1\left(\frac{1}{2}\mathcal{J}(u^n)\Delta t\right) f(u^n)\frac{1}{2}\Delta t \\
b^n &= u^n + \varphi_1(\mathcal{J}(u^n)\Delta t) f(u^n)\Delta t \\
&\quad + \varphi_1(\mathcal{J}(u^n)\Delta t)(\mathcal{F}(a^n) - \mathcal{F}(u^n))\Delta t
\end{aligned}
$$

$$
\begin{aligned}
u^{n+1} &= u^n + \varphi_1(\mathcal{J}(u^n)\Delta t) f(u^n)\Delta t \\
&\quad + \varphi_3(\mathcal{J}(u^n)\Delta t)(-14\mathcal{F}(u^n) + 16\mathcal{F}(a^n) - 2\mathcal{F}(b^n))\Delta t
\end{aligned}
$$

3rd order solution

$$
\begin{aligned}
u^{n+1} &= u^n + \varphi_1(\mathcal{J}(u^n)\Delta t) f(u^n)\Delta t \\
&\quad + \varphi_3(\mathcal{J}(u^n)\Delta t)(-14\mathcal{F}(u^n) + 16\mathcal{F}(a^n) - 2\mathcal{F}(b^n))\Delta t \\
&\quad + \varphi_4(\mathcal{J}(u^n)\Delta t)(36\mathcal{F}(u^n) - 48\mathcal{F}(a^n) + 12\mathcal{F}(b^n))\Delta t
\end{aligned}
$$

4th order solution

Leja Polynomial Interpolation

Leja points are defined recursively in a sequence

Given $K \subset C$ and $z \in K$

$$\xi_m \in \arg\max \prod_{i=0}^{m-1} |z - \xi_i|, \qquad m > 0$$

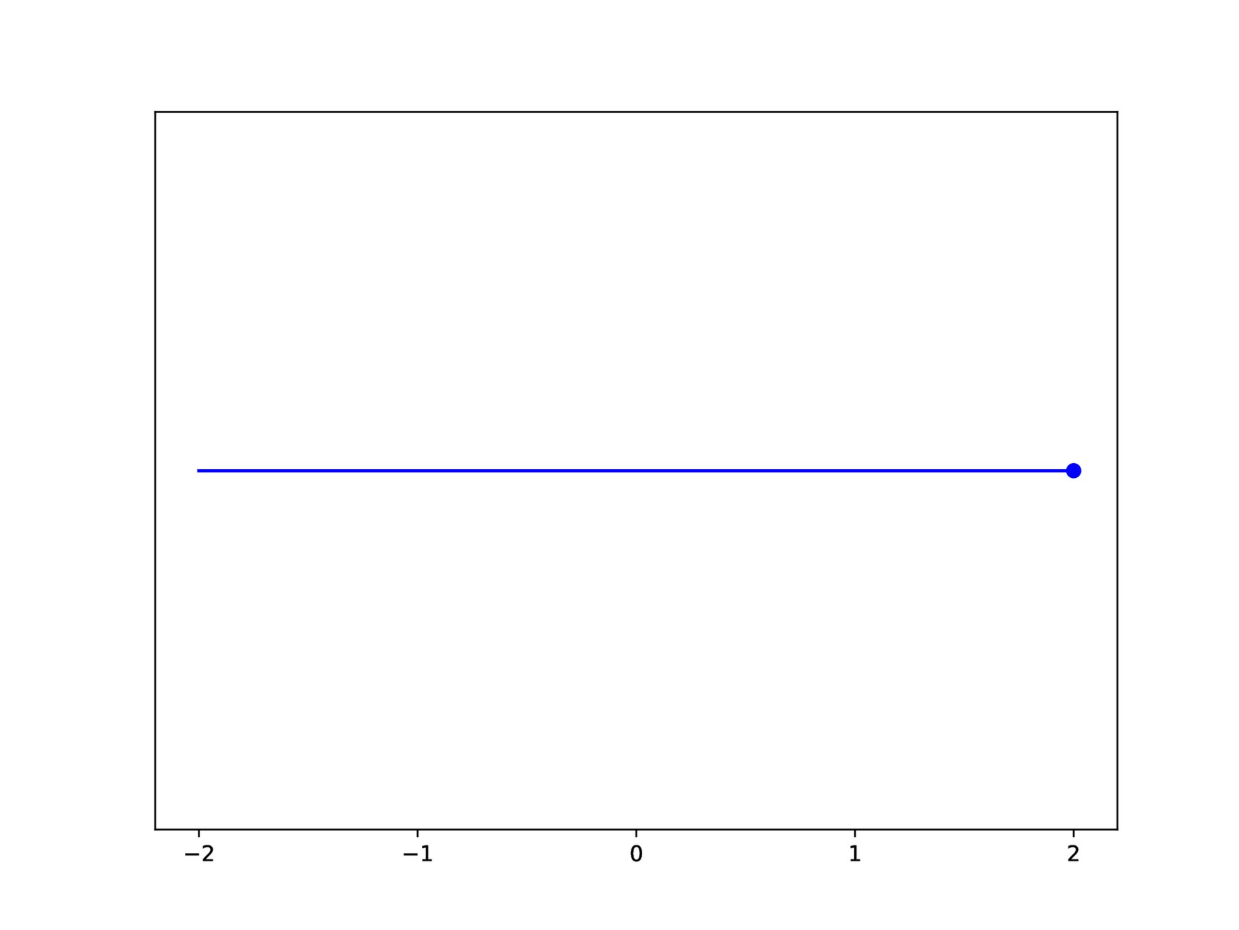
$$\xi_0 = \max|z|$$

Advantages:

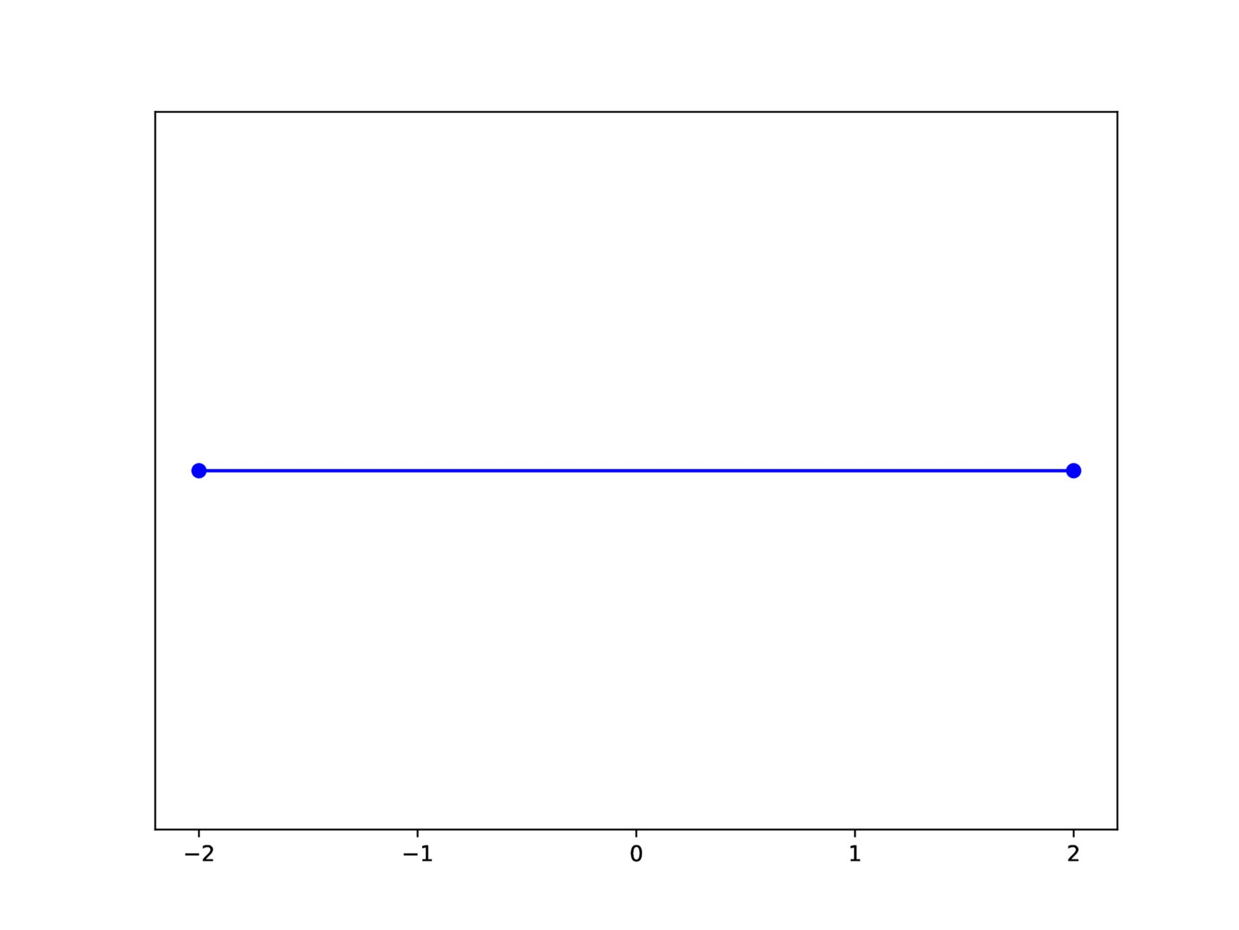
1. Computation of a polynomial at ‘m+1’ points does not require the re-computation at ‘m’ points (unlike Chebyshev points)

2. Modest memory requirements

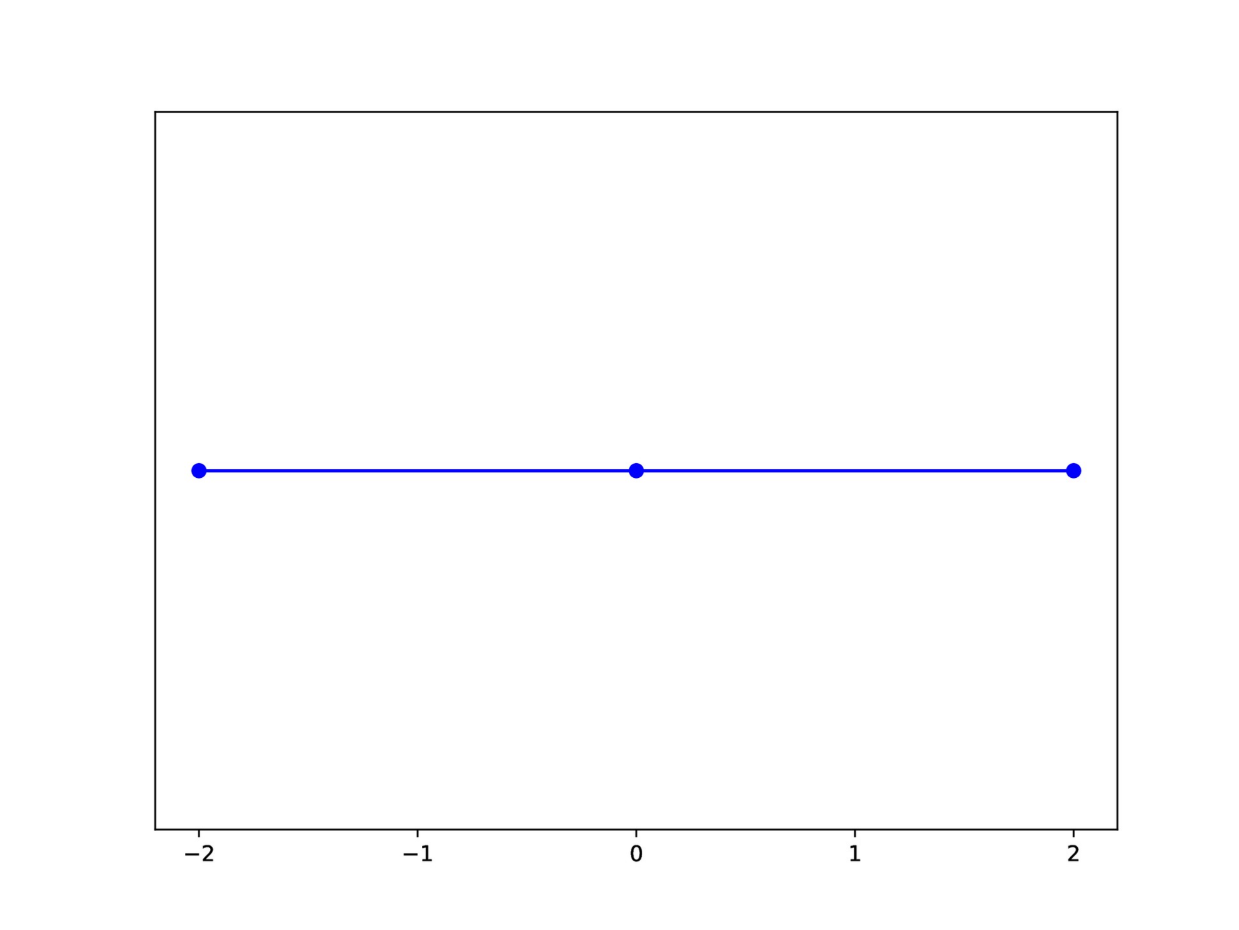
Leja Polynomial Interpolation



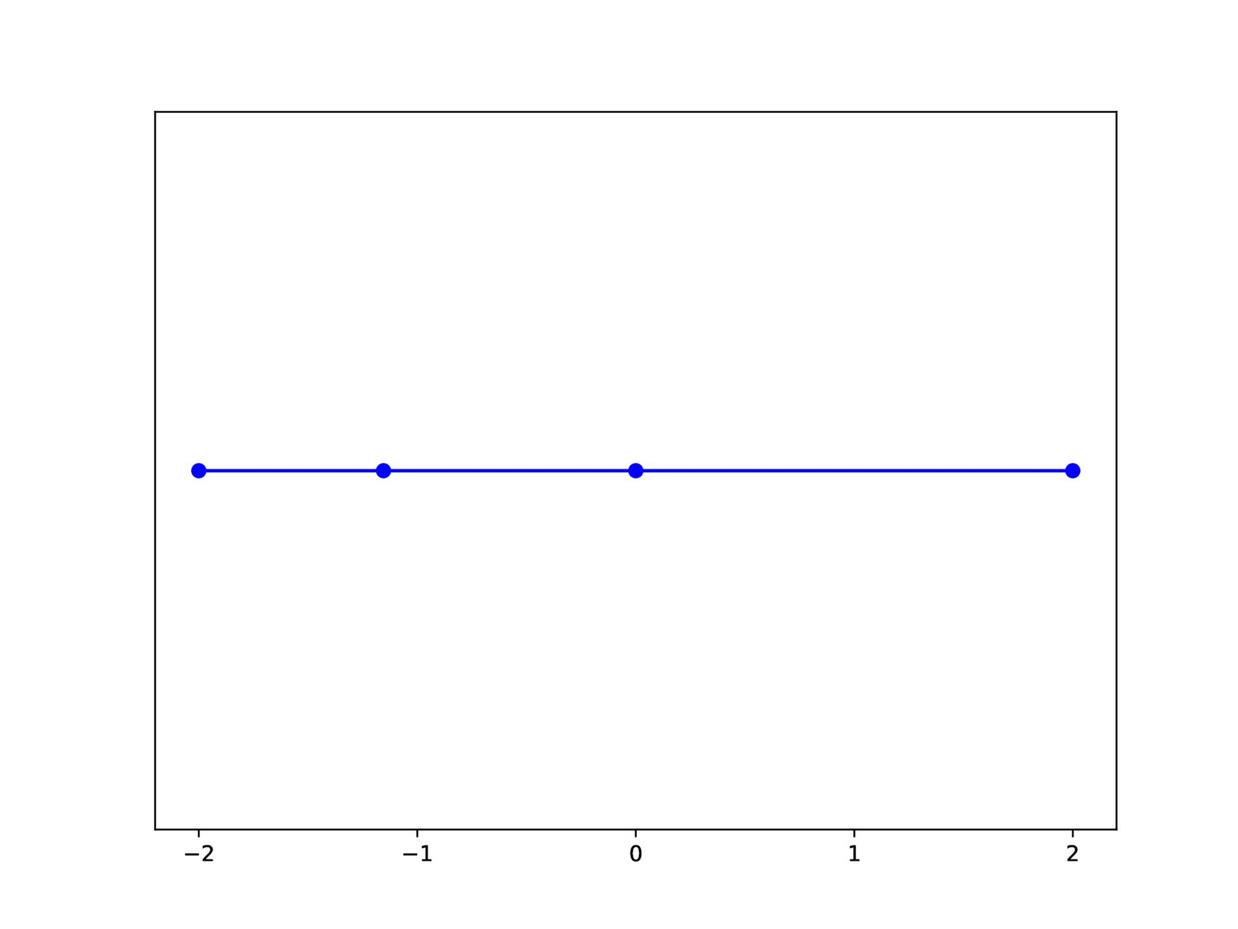
Leja Polynomial Interpolation



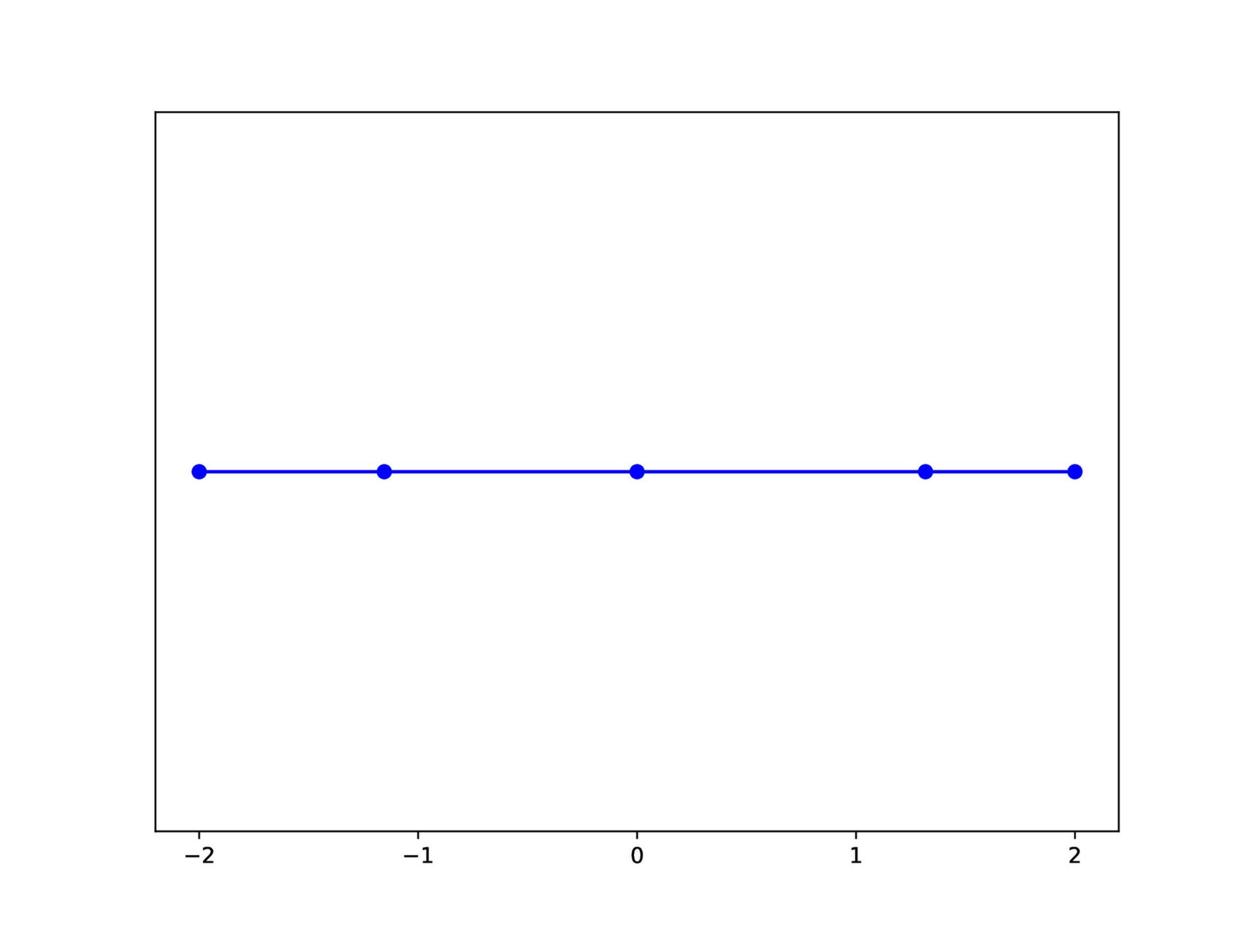
Leja Polynomial Interpolation



Leja Polynomial Interpolation



Leja Polynomial Interpolation



Step Size Controller

Why do we need them?

- Adapt step size depending on the needs
(if characteristic time scales vary drastically)
- Free users from selecting suitable step size
- Free users from having to determine the accuracy at every time step
- Able to detect onset of numerical instabilities and prevent them
- Increase computational efficiency

Step Size Controller

Traditional Step Size Controller

$$\Delta t^{n+1} = \Delta t^n \times \left(\frac{\text{tol}}{e^n}\right)^{1/(p+1)}$$

tol $\longrightarrow$ user-prescribed tolerance
e^n $\longrightarrow$ error incurred at time step ‘n’
p $\longrightarrow$ order of the integration method

Step size for the next time step is chosen based on the
error incurred and the step size
at the present time step.

Step Size Controller

Proposed Step Size Controller

Einkemmer (2018)

Principle: for iterative methods

Large step size ⟶ Higher cost
Small step size ⟶ Lower cost

Computational Cost

$$c^n = \frac{i^n}{\Delta t^n}$$

No. of matrix-vector products

$$T = \ln(\Delta t)$$
$$C(T) = \ln c(\Delta t)$$

$$T^{n+1} = T^n - \gamma \nabla C^n(T^n)$$

$$\nabla C^n(T^n) \approx \frac{C^n(T^n) - C^n(T^{n-1})}{T^n - T^{n-1}}$$

Advantages of using a smaller step size:

- Computationally cheap
- More accurate solution
- Global error:
  Proposed Cont. ≤ Traditional Cont.

$$\begin{aligned}
\nabla C^n(T^n) &\approx \frac{C^n(T^n) - C^n(T^{n-1})}{T^n - T^{n-1}} \\
&= \frac{C^n(T^n) - C^{n-1}(T^{n-1})}{T^n - T^{n-1}} + \frac{C^{n-1}(T^{n-1}) - C^n(T^{n-1})}{T^n - T^{n-1}} \\
&\approx \frac{C^n(T^n) - C^{n-1}(T^{n-1})}{T^n - T^{n-1}}
\end{aligned}$$

Step Size Controller

Proposed Step Size Controller

$$T^{n+1} = T^n - \gamma \frac{C^n(T^n) - C^{n-1}(T^{n-1})}{T^n - T^{n-1}}$$

$$\Delta t^{n+1} = \Delta t^n \exp(-\gamma \Delta)$$

$$\Delta = \frac{\ln c^n - \ln c^{n-1}}{\ln \Delta t^n - \ln \Delta t^{n-1}}$$

$$\Delta t^{n+1} = \Delta t^n \times \begin{cases} \lambda & \text{if } 1 \leq s < \lambda \\ \delta & \text{if } \delta \leq s < 1 \\ s & \text{otherwise} \end{cases}$$

$$s = \exp(-\alpha \tanh(\beta \Delta))$$

Non-penalized: parameters have been chosen to incur the minimum possible cost

Penalized: if trad. cont. performs better than non-penalized – penalty is imposed!

Non-penalized			
$\alpha = 0.65241444$	$\beta = 0.26862269$	$\lambda = 1.37412002$	$\delta = 0.64446017$
Penalized			
$\alpha = 1.19735982$	$\beta = 0.44611854$	$\lambda = 1.38440318$	$\delta = 0.73715227$

Factors δ and λ have been incorporated to ensure that step size changes by atleast δΔt or λΔt.

$$\Delta t = \min(\Delta t_{\text{traditional}}, \Delta t_{\text{proposed}})$$

Our Work

Application of the proposed step size controller on a few nonlinear problems

- Viscous Burgers’ Equation
- Inviscid Burgers’ Equation
- Porous Medium Equation

Periodic boundary conditions:
1D - [0, 1]
2D - [0, 1] x [0, 1]

Peclet Number (η): Ratio of advection to diffusion

Viscous Burgers' Equation

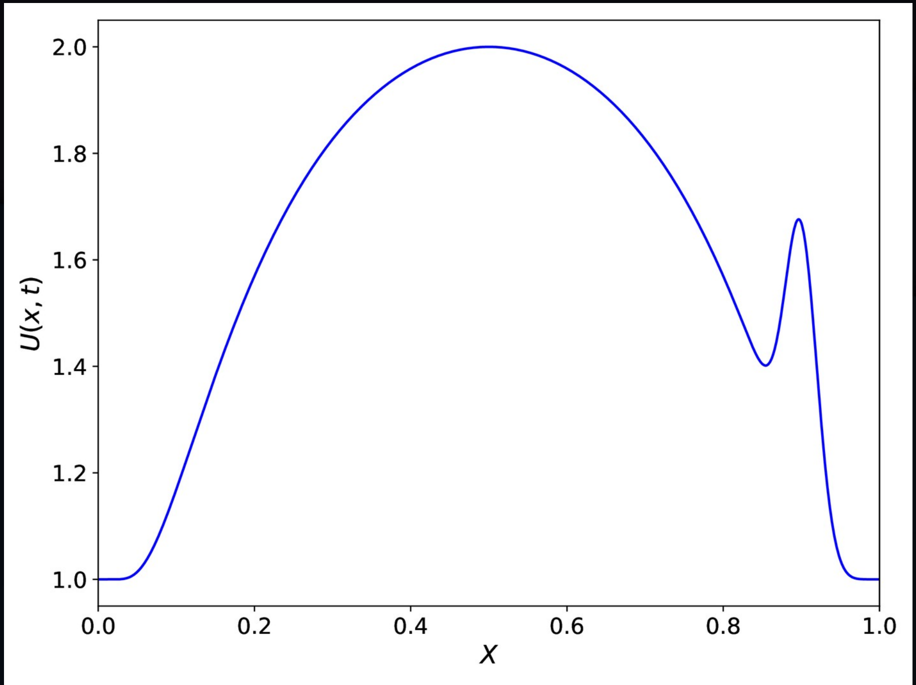
$t = 10^{-2}$

1D

$$\frac{\partial u}{\partial t} = \frac{1}{2}\eta \frac{\partial u^2}{\partial x} + \frac{\partial^2 u}{\partial x^2}$$

$$u(x, t=0) = 1 + \exp\left(1 - \frac{1}{1-(2x-1)^2}\right) + \frac{1}{2}\exp\left(-\frac{(x-x_0)^2}{2\sigma^2}\right)$$

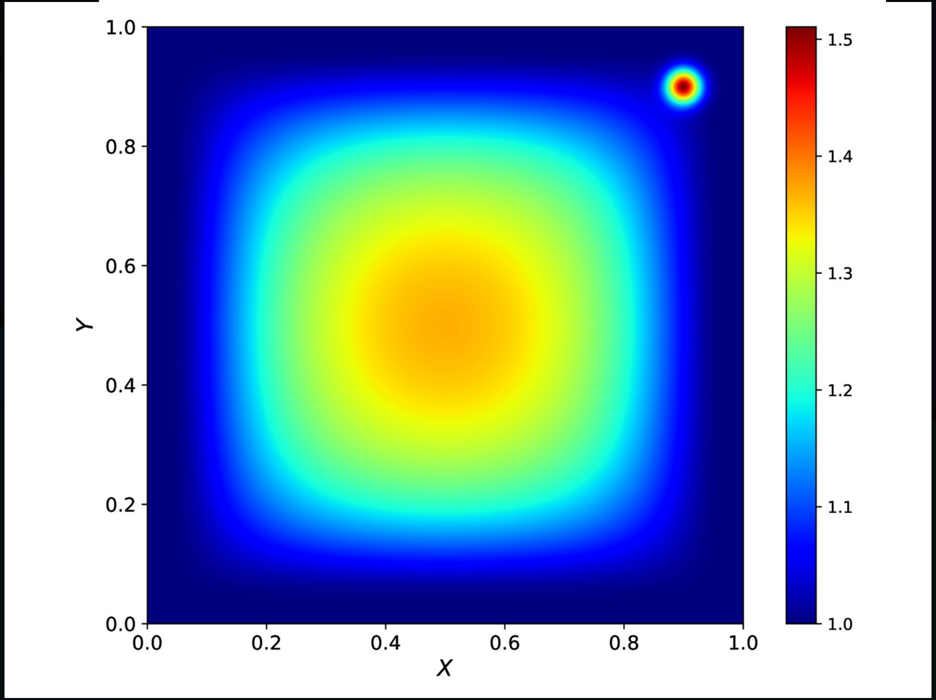
$x_0 = 0.9$ and $\sigma = 0.02$



2D

$$\frac{\partial u}{\partial t} = \frac{1}{2}\left(\eta_x \frac{\partial u^2}{\partial x} + \eta_y \frac{\partial u^2}{\partial y}\right) + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

$$u(x, y, t=0) = 1 + \exp\left(1 - \frac{1}{1-(2x-1)^2} - \frac{1}{1-(2y-1)^2}\right) + \frac{1}{2}\exp\left(\frac{-(x-x_0)^2 - (y-y_0)^2}{2\sigma^2}\right)$$



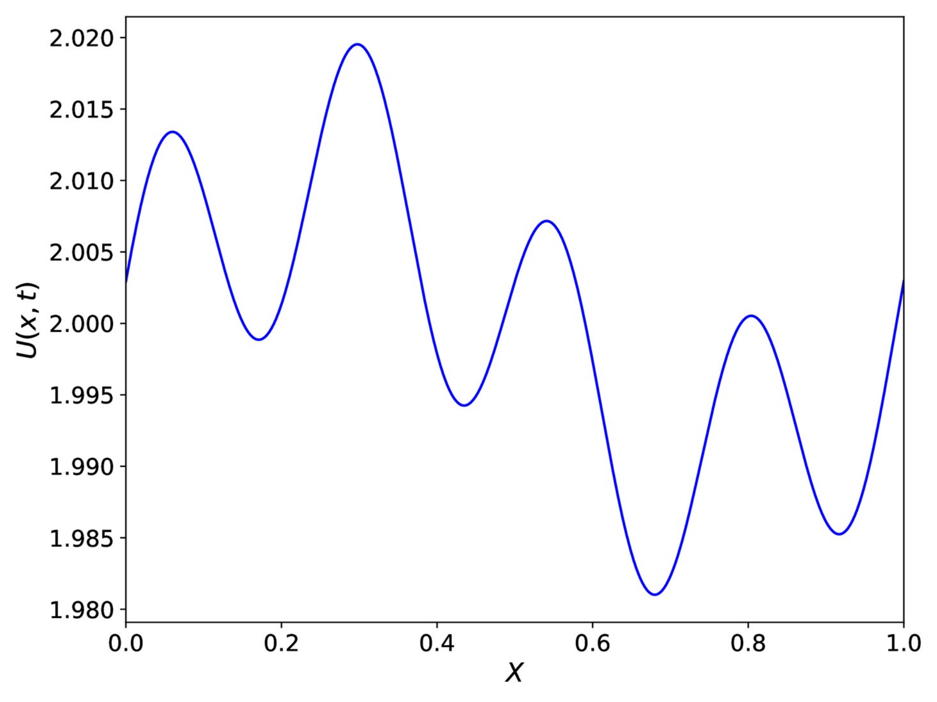
Inviscid Burgers' Equation

1D

$$\frac{\partial u}{\partial t} = \frac{1}{2}\frac{\partial u^2}{\partial x}$$

$$u(x, t=0) = 2 + \epsilon_1 \sin(\omega_1 x) + \epsilon_2 \sin(\omega_2 x + \phi)$$

$\omega_1 = 2\pi,\ \omega_2 = 8\pi,$ and $\phi = 0.3$ $\quad \epsilon_1 = \epsilon_2 = 10^{-2}$

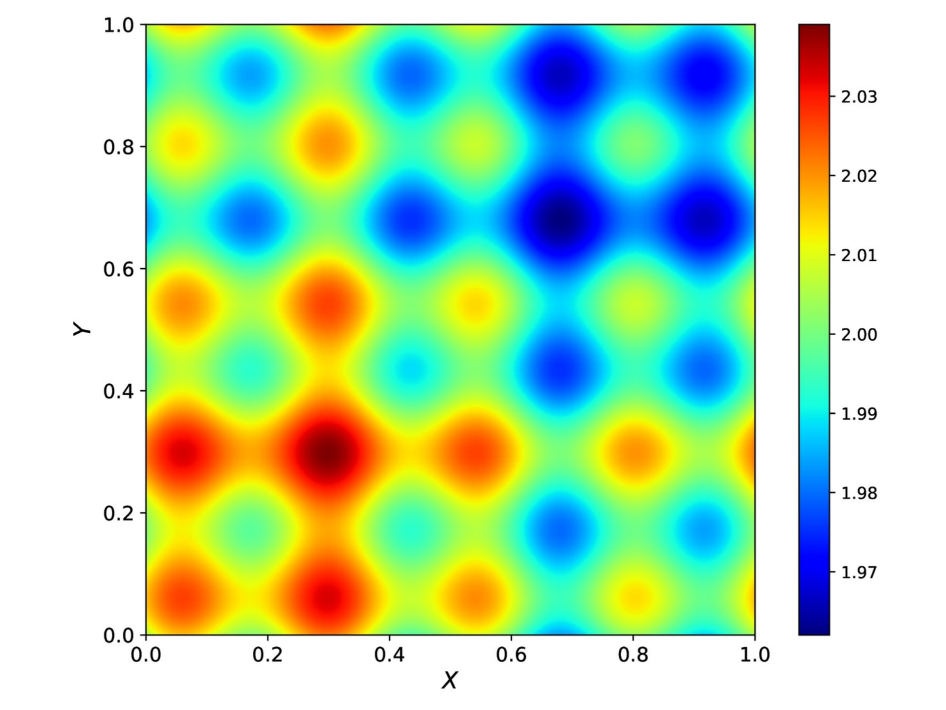


$$t = 3.25\eta \times 10^{-2}$$

2D

$$\frac{\partial u}{\partial t} = \frac{1}{2}\left(\frac{\partial u^2}{\partial x} + \frac{\partial u^2}{\partial y}\right)$$

$$u(x, y, t=0) = 2 + \epsilon_1 \sin(\omega_1 x) + \epsilon_2 \sin(\omega_2 x + \phi) + \epsilon_1 \sin(\omega_1 y) + \epsilon_2 \sin(\omega_2 y + \phi)$$



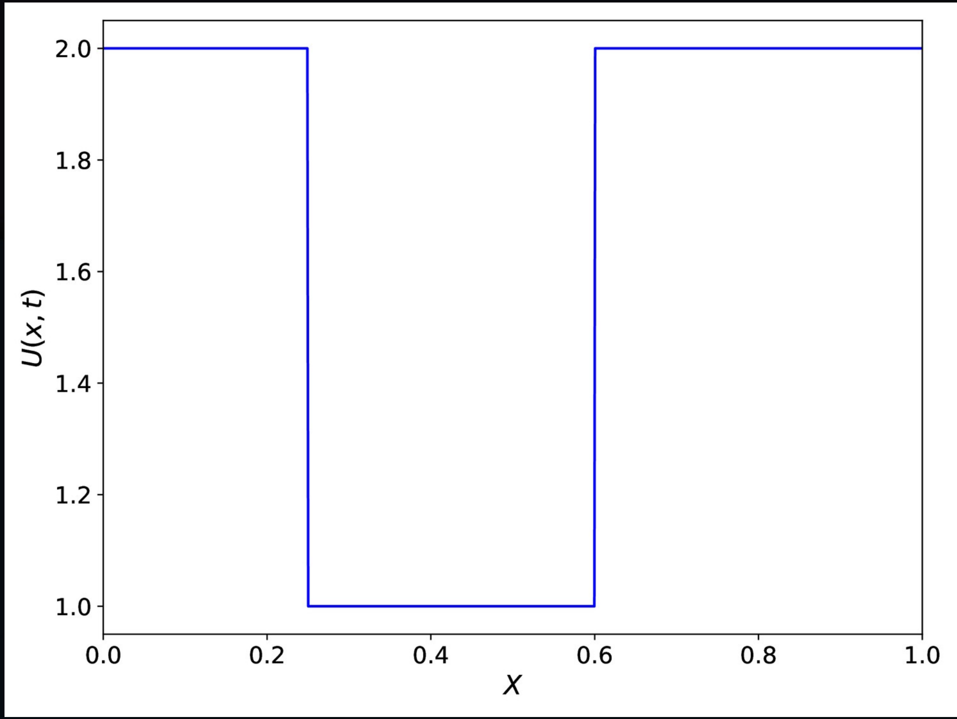
Porous Medium Equation

1D

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

$$u(x, t=0) = 1 + \Theta(x_1 - x) + \Theta(x - x_2)$$

$$x_1 = 0.25,\ x_2 = 0.6 \quad m = 2$$

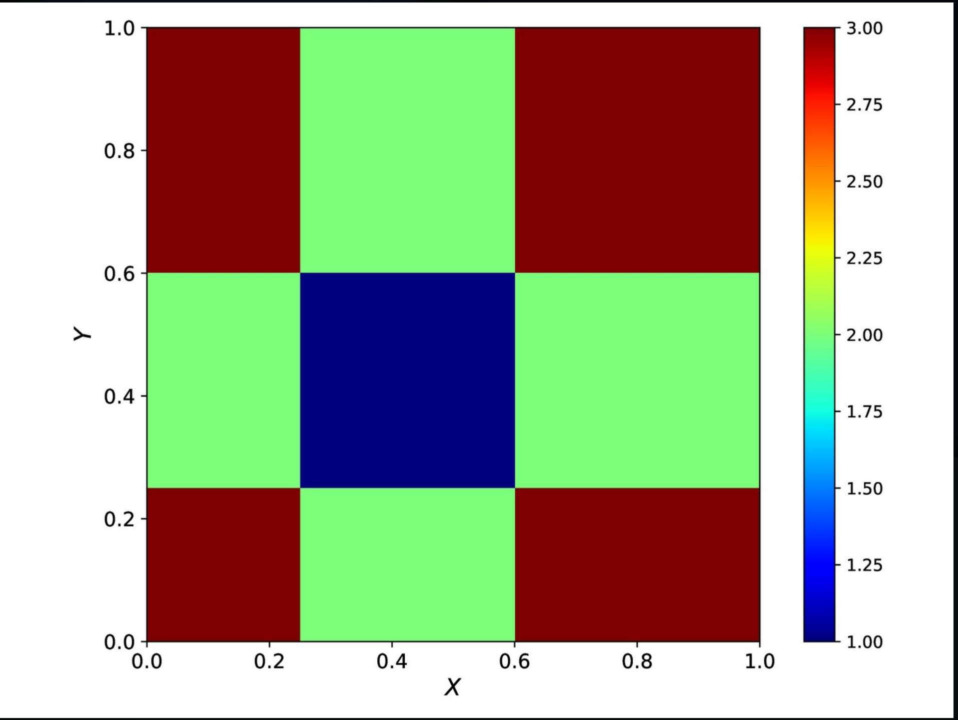


$$t = 10^{-2}$$

2D

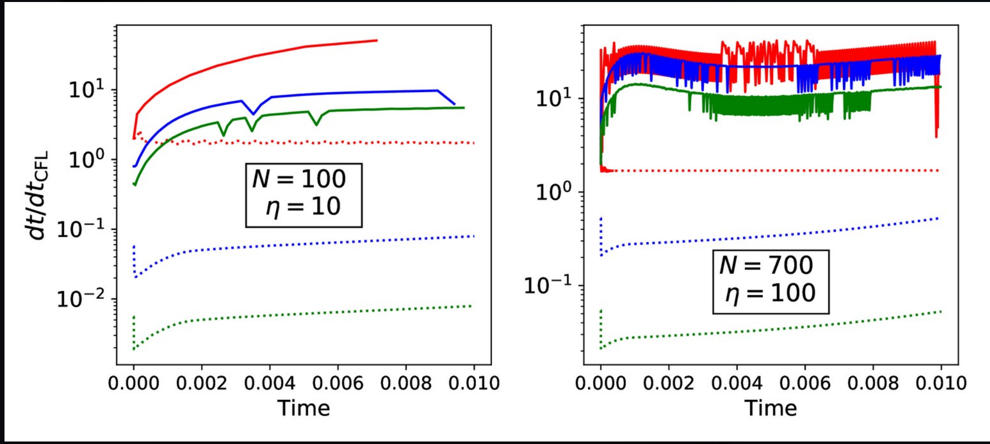
$$\frac{\partial u}{\partial t} = \eta_x \frac{\partial u}{\partial x} + \eta_y \frac{\partial u}{\partial y} + \frac{\partial^2 u^m}{\partial x^2} + \frac{\partial^2 u^m}{\partial y^2}$$

$$u(x, y, t=0) = 1 + \Theta(x_1 - x) + \Theta(x - x_2) + \Theta(y_1 - y) + \Theta(y - y_2)$$



Viscous Burgers' Equation

(EXPRB43 vs. RKF45)

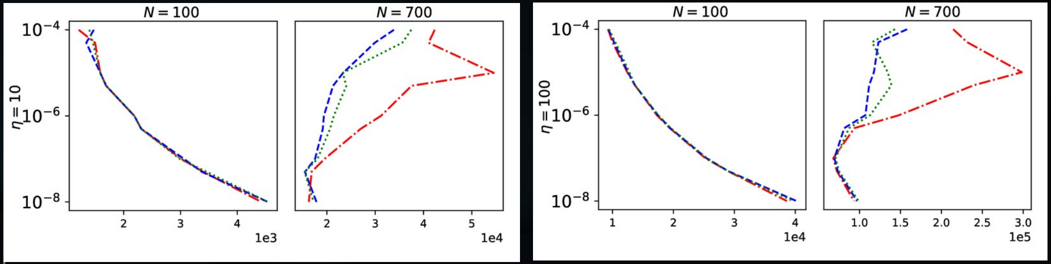


dashed lines – EXPRB43, dotted lines – RKF45

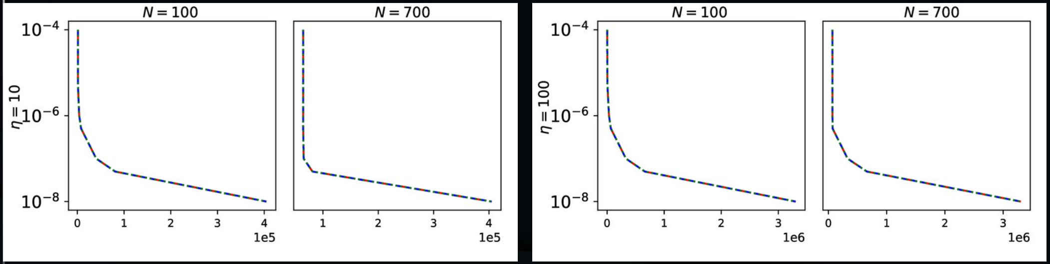
tol: 10^{-4}, 10^{-7}, 10^{-8}

Parameters	SDIRK23	EXPRB43	RKF45
$N = 100, \eta = 10$	$10^4 - 3 \cdot 10^4$	$10^3 - 4 \cdot 10^3$	$1.5 \cdot 10^3 - 4 \cdot 10^5$
$N = 100, \eta = 100$	$5 \cdot 10^4 - 2 \cdot 10^5$	$2 \cdot 10^4 - 3 \cdot 10^4$	$6 \cdot 10^4 - 4 \cdot 10^5$
$N = 700, \eta = 10$	$5 \cdot 10^4 - 1.5 \cdot 10^5$	$10^4 - 4 \cdot 10^4$	$2 \cdot 10^3 - 3.5 \cdot 10^6$
$N = 700, \eta = 100$	$5 \cdot 10^5 - 1.5 \cdot 10^6$	$10^5 - 2 \cdot 10^5$	$7 \cdot 10^4 - 3.5 \cdot 10^6$

SDIRK23
(Einkemmer 2018)



EXPRB43

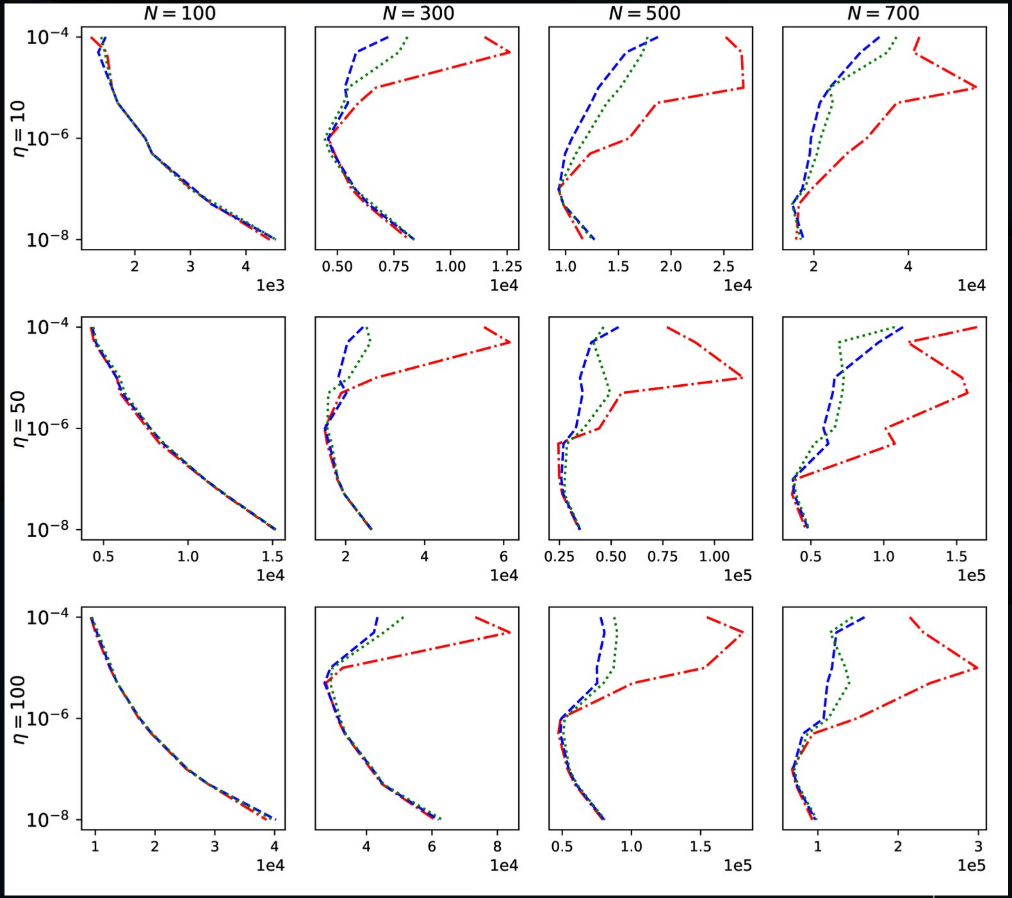


RKF45

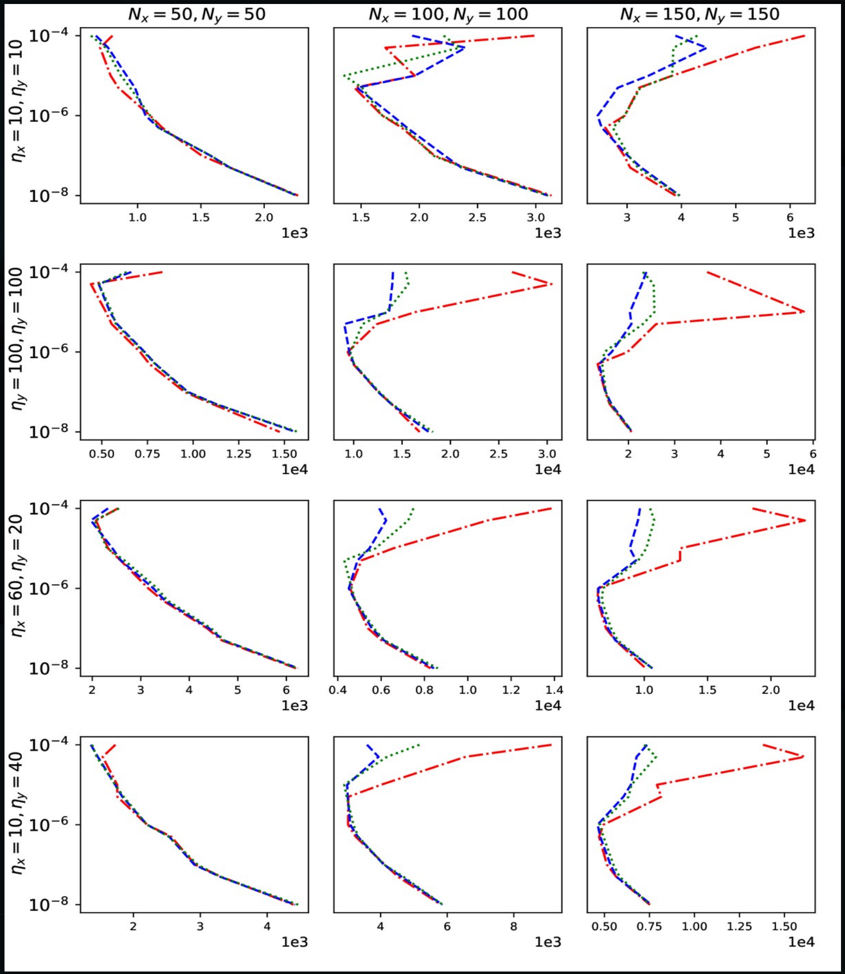
Traditional; Non-penalized; Penalized

Viscous Burgers' Equation

1D



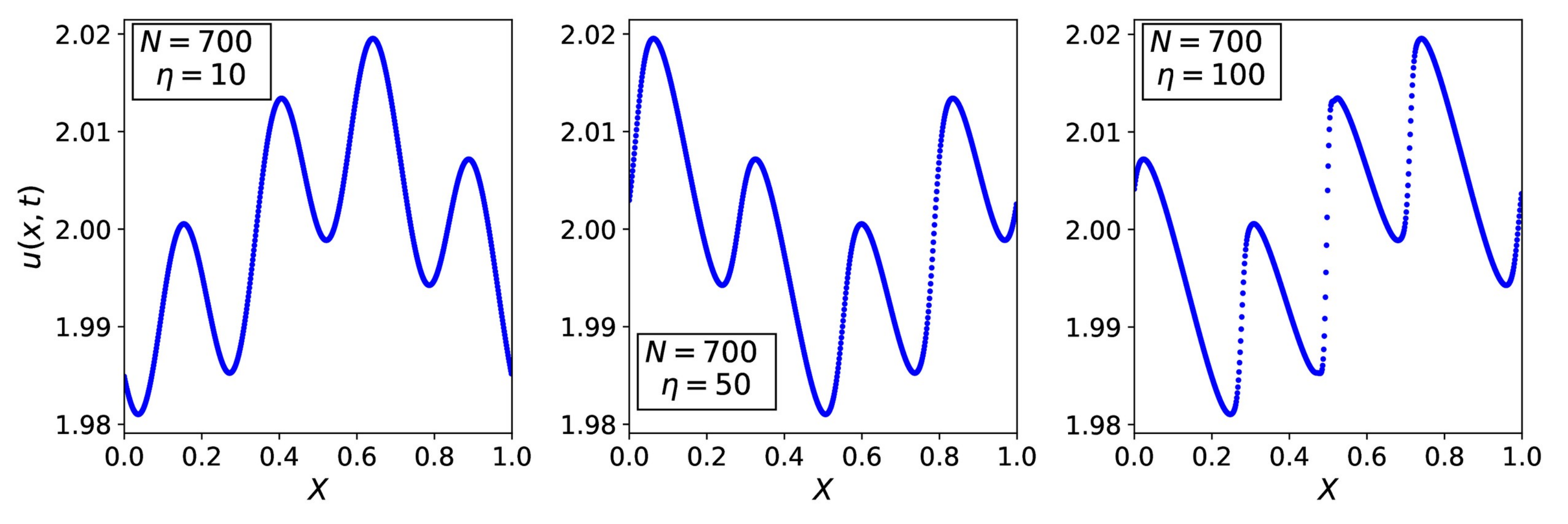
2D



Traditional; Non-penalized; Penalized

Inviscid Burgers' Equation

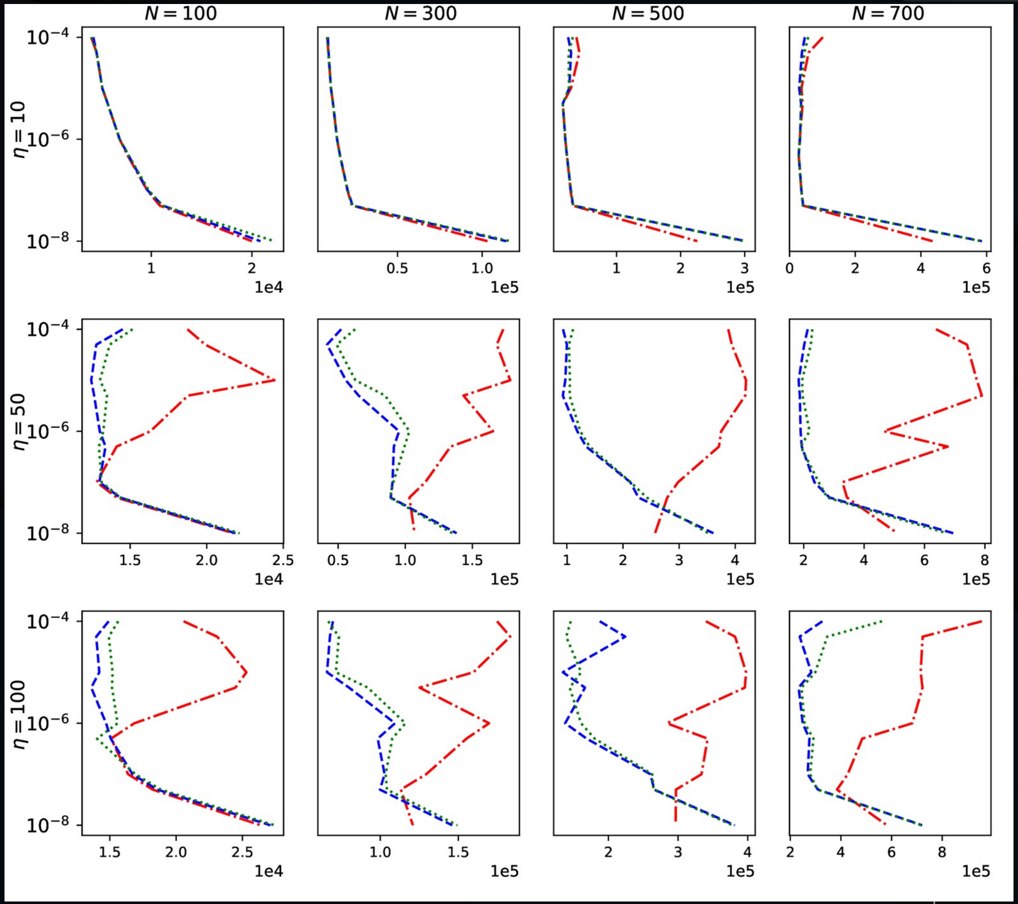
1D



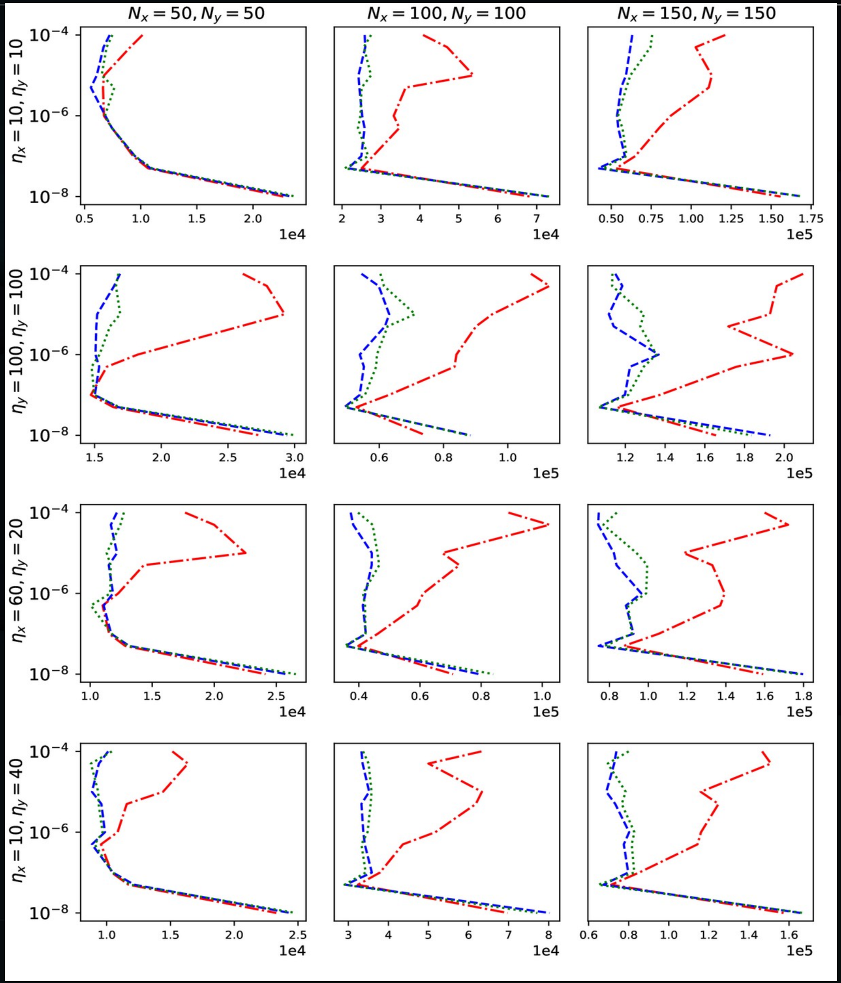
Results not yet available!

Porous Medium Equation

1D



2D



Traditional; Non-penalized; Penalized

Conclusions

- Proposed step size controller has better performance than traditional controller in majority of the cases (lenient to intermediate tolerance)

$\Downarrow$

- Multiple small step sizes incur less computational effort than 1 large step size

- ‘Similar shapes’ in 1D and 2D $\longrightarrow$ Proposed controller is reliable and effective

- EXPRB43 has superior performance than (explicit) RKF45 and (implicit) SDIRK23