A multi-level spectral deferred correction method

Ikrom Akramov

Chair Computational Mathematics
Institute of Mathematics (E-10)

TUHH
Hamburg
University of
Technology

16.05.2023

Research school on Iterative Methods for Partial Differential Equations

Spectral deferred correction (SDC) method[1]

☑ Convergence analysis

☑ Good stability properties

Multi-level spectral deferred correction (MLSDC) method[2]

💡 To use classical multigrid technique[3]

? Hoping for better convergence and lower computational cost

Agenda:

- Overview of SDC method
- An introduction to the MLSDC technique
- Numerical examples

[1]Dutt, A. *et al. Spectral Deferred Correction Methods for Ordinary Differential Equations.* in (2000).
[2]Speck, R. *et al.* A multi-level spectral deferred correction method. (2014).
[3]Briggs, W. *et al. A Multigrid Tutorial, 2nd Edition.* (2000).

Picard integral

TUHH

A generic **initial value problem (IVP)** given by

$$u'(t) = f(u(t), t)$$
$$u(0) = u_0$$

It is a corresponding to the **Picard integral form** on the interval $t \in [0, T]$

$$u(t) = u_0 + \int_0^t f(u(s), s) ds$$

- Focus on a single timestep $[T_n, T_{n+1}]$
- Introduce quadrature nodes $T_n = t_0 < t_1 < \cdots < t_M = T_{n+1}$
- Denote $u_m \approx u(t_m)$
- Define quadrature weights

$$q_{m,j} := \int_{T_n}^{t_m} l_j(s) ds, \ m = 0, \dots, M, \ j = 0, \dots, M$$

where $(l_j)_{j=0,\dots,M}$ are the **Lagrange polynomials**

What does SDC do?

SDC approximates fully implicit M-stage **Runge-Kutta** methods

$$\begin{array}{c|ccc|c} t_0 & q_{00} & \cdots & q_{0M} & \\ \vdots & \vdots & \ddots & \vdots & Q \\ t_M & q_{M0} & \cdots & q_{MM} & \\ \hline & q_0 & \cdots & q_M & \end{array}$$

Collocation problem

$$u_m = u_0 + \sum_{j=0}^{M} q_{m,j} f(u_j, t_j), \ m = 0, \ldots, M$$

Denote

- $U := [u_0, \ldots, u_M]^T$
- $U_0 := [u_0, \ldots, u_0]^T$
- $F(U) := [f(u_0, t_0), \ldots, f(u_M, t_M)]^T$

Matrix form of collocation formulation

$$U = U_0 + Q F(U)$$

Spectral deferred corrections (SDC)

SDC interation

$$u_{m+1}^{k+1} = u_m^{k+1} + \Delta t_m \underbrace{\left(f(u_m^{k+1}, t_m) - f(u_m^k, t_m)\right)}_{\to 0} + S_m^k$$

- $\Delta t_m := t_{m+1} - t_m$
- $S_m^k := \sum_{j=0}^{M} s_{m,j} f(u_m^k, t_m)$
- $s_{m,j} := q_{m,j} - q_{m-1,j}$

Convergence is monitored by computing the **SDC residual**

SDC residual

$$r^k = U_0 + QF(U^k) - U^k$$

Multi-level SDC (MLSDC)

MLSDC works as the **full approximation scheme (FAS)** for nonlinear multigrid methods.

- **SDC sweeps** are performed on levels to **solve the collocation problem**.
- **Define** levels $l = 1, \dots L$, where $l = 1$ is referred to generically as the fine level.
- The **operators** on level l are given by $A_l(U_l) \equiv U_l - Q_l F_l(U_l)$.
- Suitable **restriction** denote by R.
- **Interpolation** operators between levels are available.

FAS correction

TUHH

The FAS correction for coarse-grid sweeps

$$\tau_2 = A_2(RU_1) - RA_1(U_1) = RQ_1F_1(U_1)) - Q_2F_2(RU_1)$$

If the fine residual is zero (i.e., $U_1 \equiv \underline{U_{0,1} + \Delta t Q_1 F_1(U_1)}$), then

The FAS-corrected coarse equation

$$U_2 - Q_2F_2(U_2) = \underline{RU_{0,1} + RQ_1F_1(U_1)} - Q_2F_2(RU_1) = RU_1 - Q_2F_2(RU_1)$$

The MLSDC algorithm

TUHH

Fine level: Perform fine sweep

$$U_1^{k+1}, F_1^{k+1} \leftarrow \text{SDCSeep}(U_1^k, F_1^k)$$

Restriction:

$$U_2^k \leftarrow \text{Rest}(U_1^{k+1})$$
$$F_2^k \leftarrow \text{FEval}(U_2^{k+1})$$
$$\tilde{U}_2^k \leftarrow U_2^k$$

Coarse level: Compute FAS correction and sweep

$$\tau_2 \leftarrow \text{FAS}(F_1^{k+1}, F_2^{k+1})$$
$$U_2^{k+1}, F_2^{k+1} \leftarrow \text{SDCSeep}(U_2^k, F_2^k, \tau_2)$$

Interpolate coarse correction:

$$U_1^{k+1} \leftarrow U_1^{k+1} + \text{Inter}(U_2^{k+1} - \tilde{U}_2^k)$$
$$F_1^{k+1} \leftarrow \text{FEval}(U_1^{k+1})$$

Finest level:

$$U_1^{k+1}, F_1^{k+1} \leftarrow \text{SDCSeep}(U_1^k, F_1^k)$$

Dahlquist's equation

Stiff equation

$$u_t(t) = -11iu(t)$$
$$u(0) = 1.0$$

on $t \in [0, 0.3]$

- Gauss-Lobatto quadrature nodes
- 5 quadrature nodes on the **fine level**
- 3 quadrature nodes on the **coarse level**

Visualization

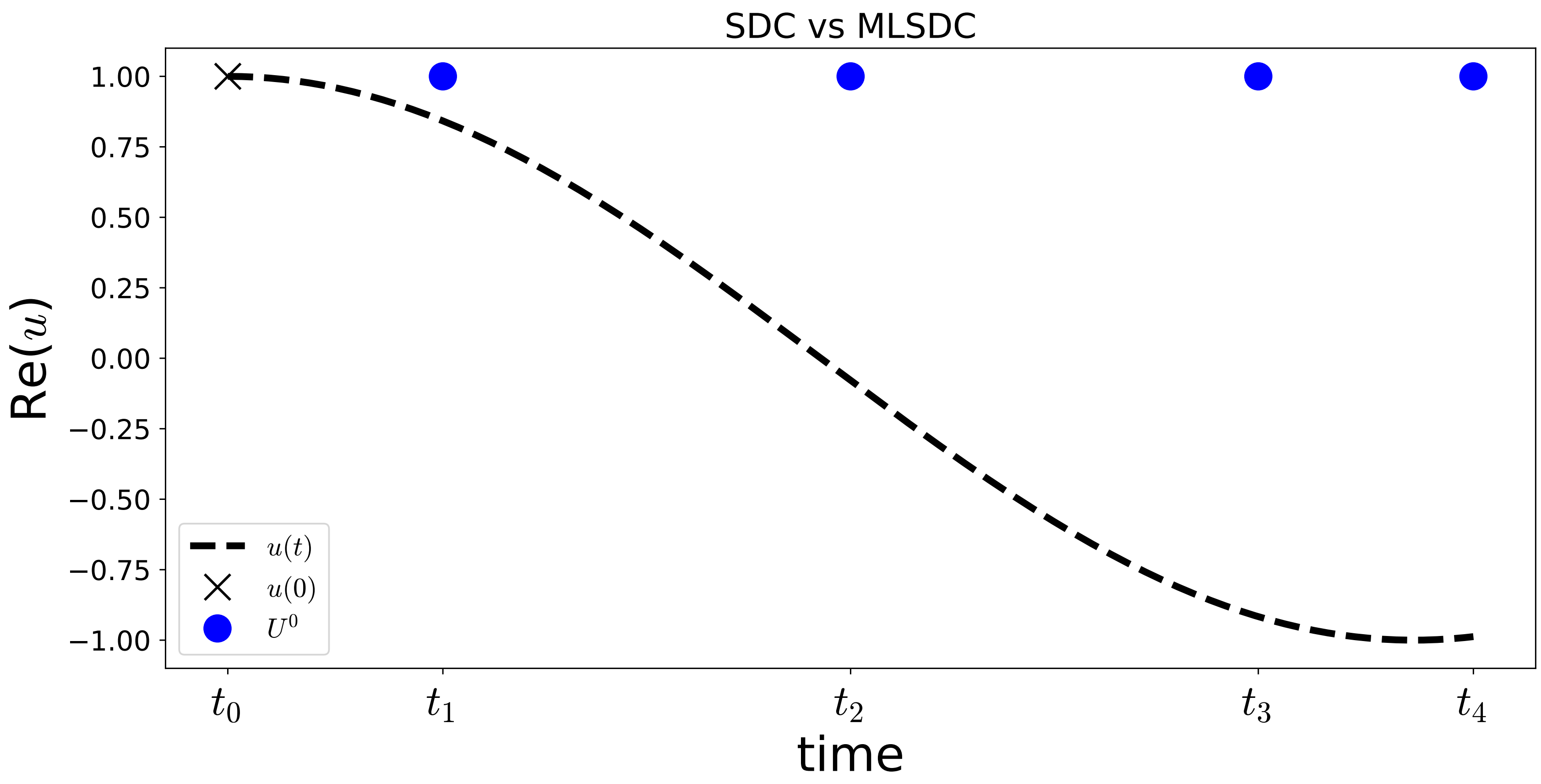

Figure 1: Initial guess for each nodes using the initial solution

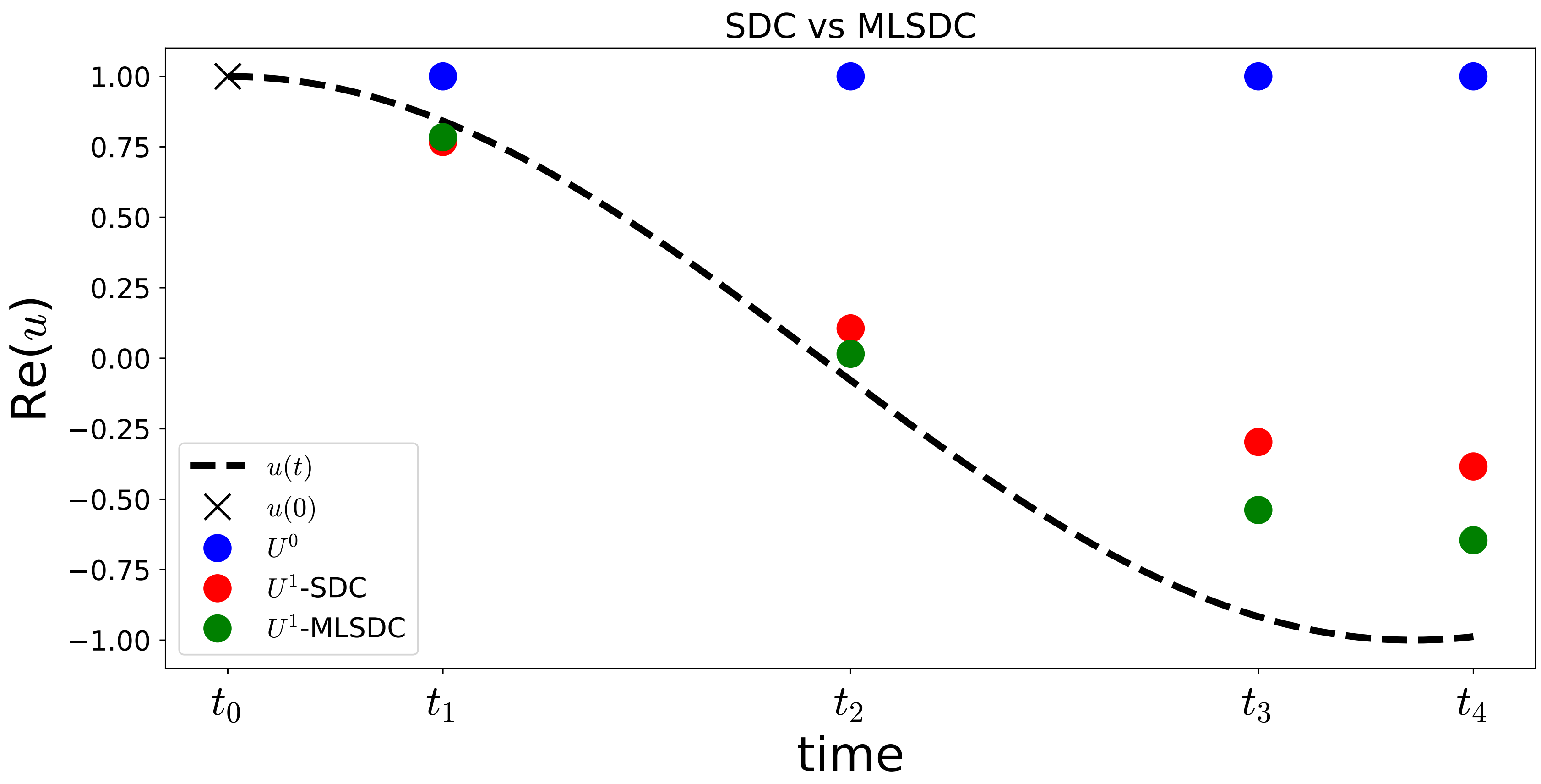

Figure 2: First sweep

Visualization

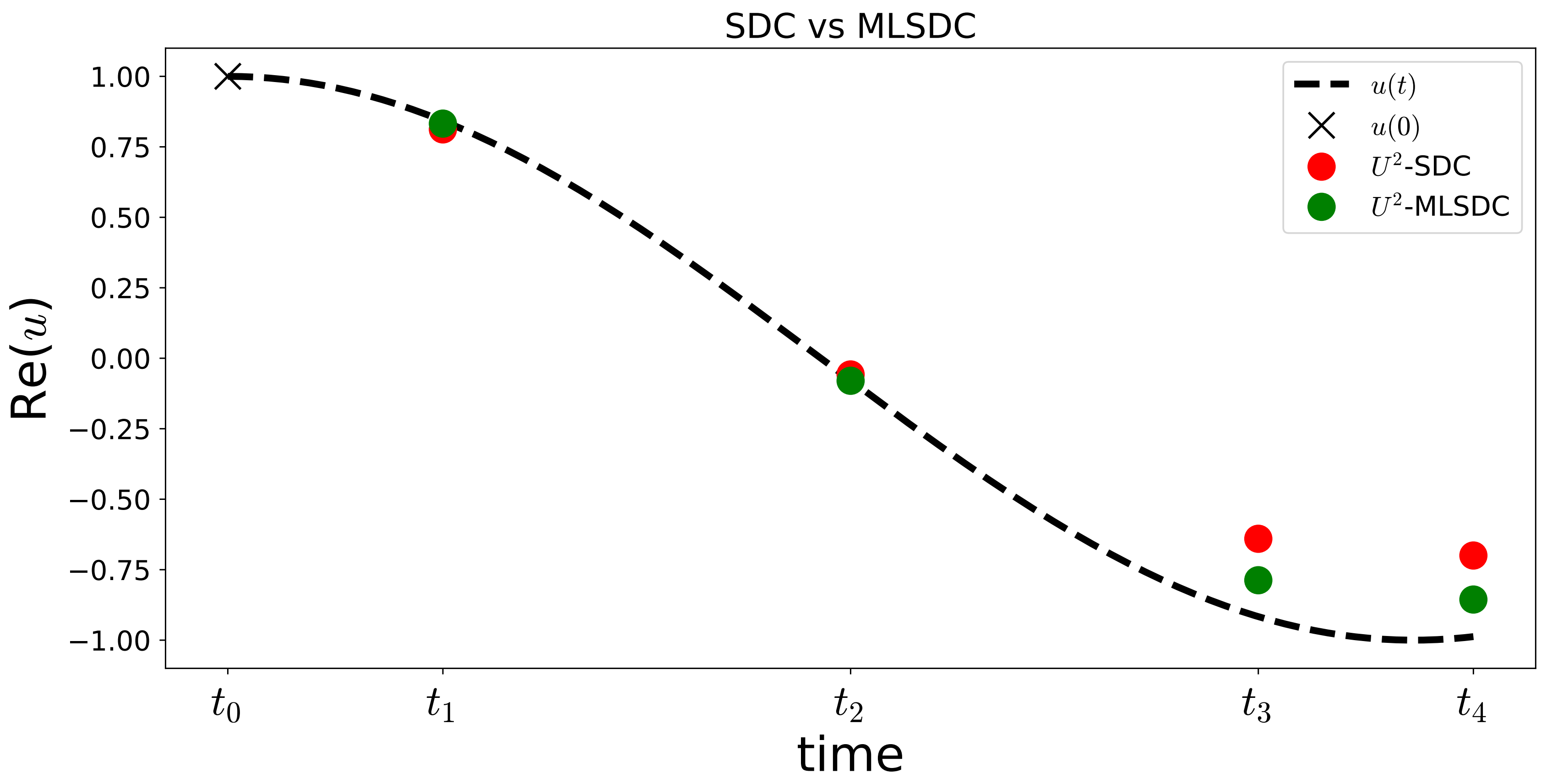

Figure 3: Second sweep

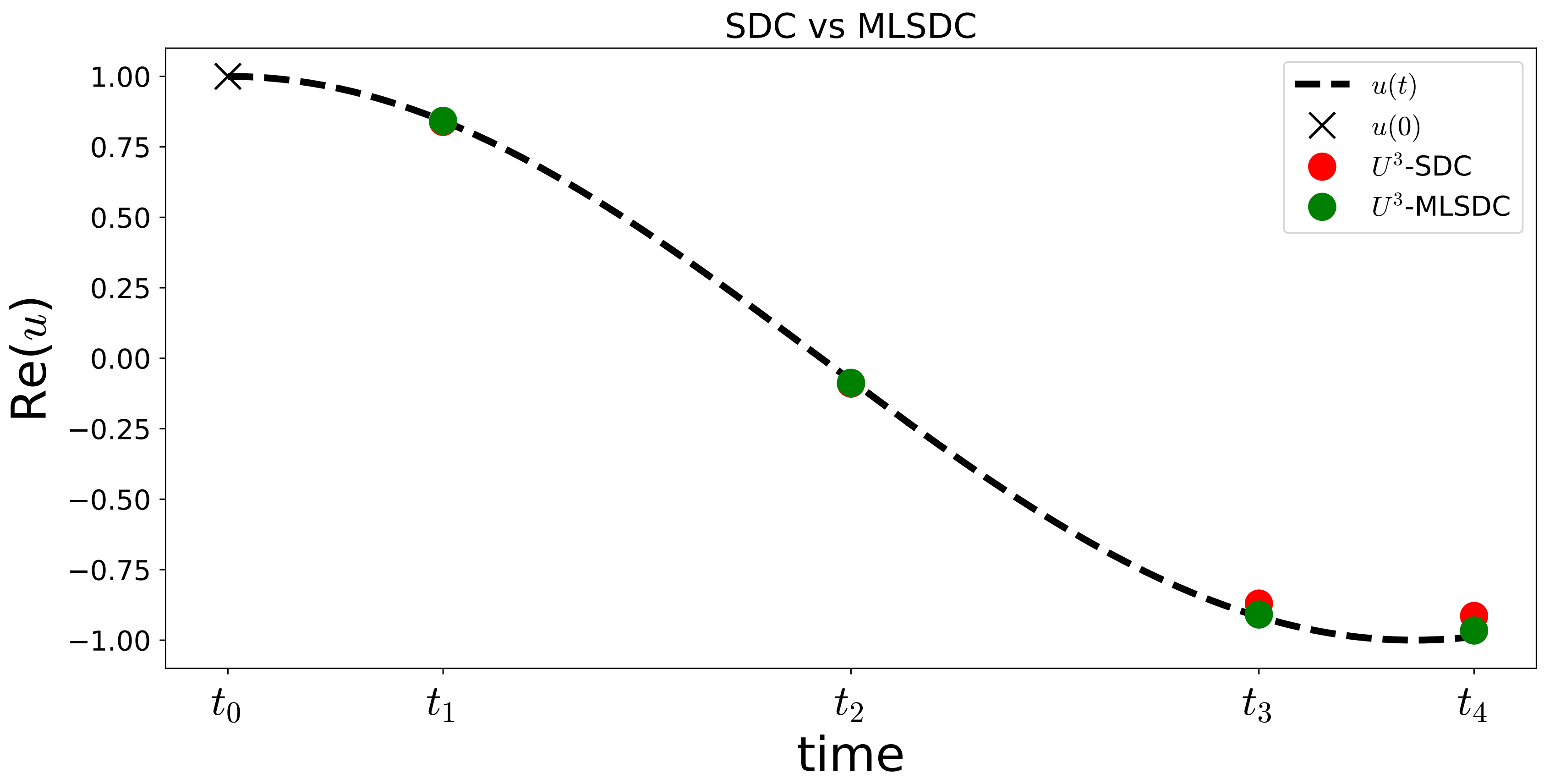

Figure 4: Third sweep

Residual

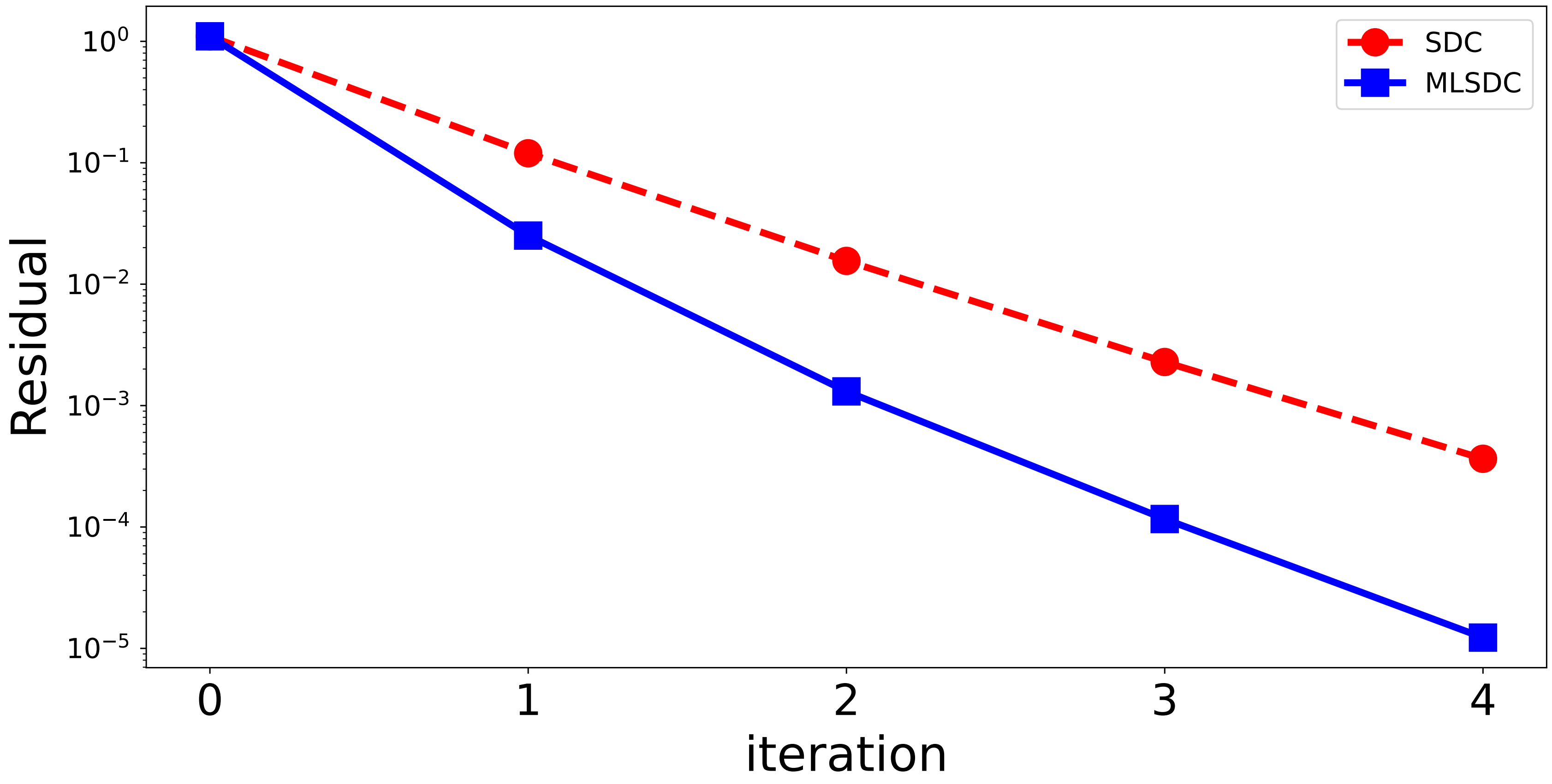

Figure 5: Residual computed with infinity norm

Consider the nonlinear viscous **Burgers' equation**

$$\bar{u}_t + \bar{u} \cdot \bar{u}_x = \nu \bar{u}_{xx}, \;\; x \in [-1, 1], \;\; t \in [0, t_{end}]$$
$$\bar{u}(x, 0) = u^0(x)$$
$$\bar{u}(-1, t) = \bar{u}(1, t)$$

with $\nu > 0$ and initial condition

$$u^0(x) = \exp\left(-\frac{x^2}{\sigma^2}\right), \;\; \sigma = 0.1$$

- $t_{end} = \Delta t = 0.01$
- **Gauss-Lobatto** collocation nodes
- a spatial mesh of $N = 256$ points on the **fine level**
- a spatial mesh of $N = 128$ points on the **coarse level**

Required iterations

Number of fine level sweeps required to reach a residual of $\|r^k\|_\infty \leq 10^{-5}$.		
Method	# Fine sweeps, $\nu = 0.1$	# Fine sweeps, $\nu = 1.0$
SDC	4	12
MLSDC	3	7

Three-dimensional viscous Burgers' equation

TUHH

Consider the nonlinear viscous **Burgers' equation** in **three dimensions**

$$\bar{u}_t(\mathbf{x}, t) + \bar{u}(\mathbf{x}, t) \cdot \nabla \bar{u}(\mathbf{x}, t) = \nu \nabla^2 \bar{u}(\mathbf{x}, t), \ \mathbf{x} \in [0, 1]^3, \ t \in [0, 1]$$

with $\mathbf{x} = (x, y, z)$ and initial value

$$\bar{u}(x, t) = \exp\left(-\frac{(x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2}{\sigma^2}\right)$$

homogeneous **Dirichlet boundary condition**

- The fine level uses a 255^3 point mesh
- The coarse level 127^3

Required iterations and runtimes

TUHH

Number of fine level sweeps required to reach a residual of $\|r^k\|_\infty \leq 10^{-5}$.

Method $(M = 3)$	# Fine sweeps, $\nu = 0.1$	# Fine sweeps, $\nu = 1.0$
SDC	$9\ (39.4s)$	$16\ (74.1s)$
MLSDC	$4\ (26.2s)$	$8\ (49.1s)$

Method $(M = 5)$	# Fine sweeps, $\nu = 0.1$	# Fine sweeps, $\nu = 1.0$
SDC	$7\ (59.5s)$	$18\ (162.7s)$
MLSDC	$3\ (40.8s)$	$9\ (105.6s)$

Summary

- MLSDC significantly reduce the number of fine sweeps required for convergence in comparison to single-level SDC
- Runtime savings small

Summary

- MLSDC significantly reduce the number of fine sweeps required for convergence in comparison to single-level SDC
- Runtime savings small

Thank You for Attention!