A first-order computational algorithm for reaction-diffusion-type equations via primal-dual hybrid gradient method

Shu Liu (UCLA), Siting Liu (UCLA), Xinzhe Zuo (UCLA),
Stanley Osher (UCLA), Wuchen Li (USC)

shuliu@math.ucla.edu

January 20, 2025

Table of Contents

Reaction-diffusion equations

- Time-dependent partial differential equation that models the density (or concentration) evolution of chemical systems with the reactions (substances get transformed to each other) and diffusions (substances get spread over).
- The equation has applications in broad scientific areas.
  - Phase-field models (depict the development of microstructures of multiple materials);
  - Evolution of species distribution in ecology system;
  - Reaction processes of multiple chemicals;
  - Modeling & prediction of crimes;
  - And many more.

Reaction-diffusion equations

General formulation of reaction-diffusion equation,

$$\frac{\partial u(x,t)}{\partial t} = -\mathcal{G}(a\mathcal{L}u(x,t) + bf(u(x,t))), \tag{1}$$

- defined on $\Omega \subset \mathbb{R}^2$ with initial condition u_0 and suitable boundary condition (Neumann, periodic, etc.).
- $\mathcal{G}, \mathcal{L}$ are usually non-negative definite, self-adjoint differential operators, $a, b \geq 0$, $f(\cdot)$ is a nonlinear reaction term.

Reaction-diffusion equations

Examples:

- Treat $\mathcal{G} = \mathrm{Id}, \mathcal{L} = -\Delta$. Take $f(u) = W'(u) = u^3 - u$ where $W(u) = \frac{1}{4}(1-u^2)^2$ is the double-well potential. This is the Allen-Cahn equation
$$\partial_t u = a\Delta u - b(u^3 - u).$$
- Treat $\mathcal{G} = -\Delta, \mathcal{L} = -\Delta$. Take $f(u) = W'(u) = u^3 - u$ as well. This is the Cahn-Hilliard equation
$$\partial_t u = -a\Delta\Delta u + b\Delta(u^3 - u).$$
- etc.

Many reaction-diffusion equation has gradient flow structure. Consider the free energy functional

$$\mathcal{E}(u) = \int_\Omega \|\nabla_x u(x)\|^2 + W(u(x))\ dx.$$

- Allen-Cahn equation is the L^2-gradient flow of $\mathcal{E}(u)$;
- Cahn-Hilliard equation is the $H^{-1}-$gradient flow of $\mathcal{E}(u)$;
- etc.

Reaction-diffusion equations

Take $f(u) = W'(u) = u^3 - u$ where $W(u) = \frac{1}{4}(1-u^2)^2$ is the double-well potential.

- Allen-Cahn (AC) equ: $\mathcal{G} = \mathrm{Id}, \mathcal{L} = -\Delta$, $\partial_t u = a\Delta u - b(u^3 - u)$.
- Cahn-Hilliard (CH) equ: $\mathcal{G} = -\Delta, \mathcal{L} = -\Delta$, $\partial_t u = -a\Delta\Delta u + b\Delta(u^3 - u)$.
- etc.

Many reaction-diffusion equation has gradient flow structure. Consider the free energy functional

$$\mathcal{E}(u) = \int_\Omega \|\nabla_x u(x)\|^2 + W(u(x))\, dx.$$

- Allen-Cahn equation is the L^2-gradient flow of $\mathcal{E}(u)$;
- Cahn-Hilliard equation is the $H^{-1}-$gradient flow of $\mathcal{E}(u)$;
- etc.

$$\mathcal{L}_h, \mathcal{G}_h$$

Numerical solution to reaction-diffusion equations

- Solve from Eulerian perspective:
  - (Finite difference) [Merriman et al. 1994], [Eyre et al. 1998], [Guillén-González et al. 2013], [Yang. 2016], [Shen et al. 2018], [Xu, et al. 2019], etc.
  - (Finite element) [Zhu et al. 2009], [Fu et al. 2023], etc.
  - (Spectral method) [Christlieb et al, 2014], etc.
- Solve from Lagrangian perspective (EnVarA): [Liu et al. 2020], [Liu et al. 2022], [Liu et al. 2022], etc.
- Comprehensive literature (books, benchmark problems, etc.) [Hundsdorfer et al. 2003], [Church et al. 2019], etc.

Primal-dual hybrid gradients algorithm

The primal-dual hybrid gradient (PDHG) algorithm [Zhu et al. 2008], [Chambolle et al. 2011] is proposed to solve the saddle point problem,

$$\inf_y \sup_x f(x, y).$$

The algorithm uses proximal operators together with an extrapolation step to update (x_k, y_k), i.e.,

$$\begin{aligned}
x_{n+1} =& \operatorname*{argmin}_x \left\{ \frac{\|x - x_n\|^2}{2\tau_x} - f(x, y_n) \right\}, \\
\widetilde{x}_{n+1} =& x_{n+1} + \omega(x_{n+1} - x_n), \\
y_{n+1} =& \operatorname*{argmin}_y \left\{ \frac{\|y - y_n\|^2}{2\tau_y} + f(\widetilde{x}_{n+1}, y) \right\}.
\end{aligned}$$

Saddle point scheme for solving numerical PDEs

Solving numerical PDEs by coupling them with their dual variables and formulating a min-max saddle point scheme.

- Classical methods
  - Conservation Laws: [Liu et al. 2022].
  - Reaction-diffusion equations: Our research.
  - Hamilton-Jacobi equations: [Meng et al. 2023], etc.
- Machine (deep) learning methods
  - Introduce dual variable for the equation. Weak Adversarial Networks (WAN): [Zang et al. 2019] [Zang et al. 2020], etc.
  - Introduce dual variable directly for the residual: [McClenny et al. 2020], [Anagnostopoulos et al. 2023], etc.

Our goal

Numerical scheme: The implicit finite difference scheme.

Tools to resolve the scheme: PDHG algorithm.

Goal: Resolving the implicit scheme with the PDHG algorithm
+ proof convergence
+ verify the efficiency of the method.

Table of Contents

Background

Algorithm

Theoretical guarantees

Numerical Experiments

Summary & Future research

Motivation: why do we consider implicit scheme?

- Simplicity;
- Overcomes the Courant–Friedrichs–Lewy (CFL) condition on the time step size h_t in explicit or semi-explicit schemes, which leads to efficient computation of the steady state of the reaction-diffusion (RD) equation;
- Usually more stable than the explicit or semi-explicit schemes in phase field models with small diffusion coefficients and strong reaction coefficients;
- Preserves energy dissipation;

More discussions in [Xu et al. 2019].

Table of Contents

Background

Algorithm

Theoretical guarantees

Numerical Experiments

Summary & Future research

Motivation: why do we consider implicit scheme?

- Simplicity;
- Overcomes the Courant–Friedrichs–Lewy (CFL) condition, larger time stepsize which leads to efficient computation of the steady state of the reaction-diffusion (RD) equation;
- Usually more stable than the explicit or semi-explicit schemes in phase field models with small diffusion coefficients and strong reaction coefficients;
- Preserves energy dissipation;

More discussions in [Xu et al. 2019].

Motivation: why PDHG method for resolving the scheme?

- A first-order, easy-to-implement optimization algorithm with tunable hyperparameters;
- Does not require extra effort to compute the inverse of the Jacobian matrix;
- Convergence rate of the method is independent of grid resolution;
- Applicable to various types of numerical schemes with high flexibility.

Derivation of the algorithm

We solve (1) on $[0, T]$, we divide time interval into N_t subintervals, Suppose Ω is divided into $N_x \times N_x$ grids. The implicit scheme

$$\frac{U^{t+1} - U^t}{h_t} = -\mathcal{G}_h(a\mathcal{L}_h U^{t+1} + bf(U^{t+1})),$$

with $t = 1, \ldots, N_t$ with U_0 given.

This implicit scheme is equivalent to

$$F(U) = 0,$$

where $U = [U_1^\top, \ldots, U_{N_t}^\top]^\top \in \mathbb{R}^{N_t N_x^2}$.

$$F(U) = \mathscr{D}U + h_t \mathscr{G}_h(a\mathscr{L}_h U + bf(U)) - V.$$

$$F(U) = \mathscr{D}U + h_t\mathscr{G}_h(a\mathscr{L}_hU + bf(U)) - V.$$

Here $\mathscr{D} = D_{N_t} \otimes I_x$, where I_x denotes the identity matrix on $\mathbb{R}^{N_x^2}$,

$$D = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}.$$

On the other hand,$\mathscr{G}_h = I_t \otimes \mathcal{G}_h$; $\quad \mathscr{L}_h = I_t \otimes \mathcal{L}_h$, with I_{N_t} as the identity matrix on $\mathbb{R}^{N_t}$.
$f(U) = f((U^{1\top}, \dots, U^{N_t\top})^\top) = (\dots, f(U_{ij}^t), \dots)^\top$. Constant vector V depends on both the initial condition and the boundary condition.

To solve the implicit scheme, motivated by the treatment in [Zuo et al. 2023], it suffices to minimize

$$\min_{U\in\mathbb{R}^{N_t N_x^2}} \ \frac{1}{2\epsilon}\|F(U)\|^2.$$

Since $\frac{1}{2\epsilon}\|\cdot\|^2$ can be represented by using Legendre transform,

$$\frac{1}{2\epsilon}\|\cdot\|^2 = \sup_p \ (p,\cdot) - \frac{\epsilon}{2}\|p\|^2,$$

the original equation yields

$$\min_{U\in\mathbb{R}^{N_t N_x^2}} \max_{Q\in\mathbb{R}^{N_t N_x^2}} \ L(Q,U) \triangleq (Q,F(U)) - \frac{\epsilon}{2}\|Q\|^2.$$

One can then bring the aforementioned PDHG algorithm to deal with this min-max saddle point problem.

$$
\begin{aligned}
Q_{k+1} &= \frac{1}{1+\epsilon\tau_P}(Q_k + \tau_P F(U_k));\\
\widetilde{Q}_{k+1} &= Q_{k+1} + \omega(Q_{k+1} - Q_k);\\
U_{k+1} &= U_k - \tau_U (DF(U_k)^\top \widetilde{Q}_{k+1}).
\end{aligned}
$$

- We replace the original proximal step on U by explicit update.
  - Gain: Convenient to implement;
  - Loss: The convergence rate might be weaker (but still guaranteed to converge).
- Equilibrium of the PDHG method is $(U_*, 0)$, with $F(U_*) = 0$.

One can then bring the aforementioned PDHG algorithm to deal with this min-max saddle point problem.

$$
\begin{aligned}
Q_{k+1} &= \frac{1}{1+\epsilon\tau_P}(Q_k + \tau_P F(U_k));\\
\widetilde{Q}_{k+1} &= Q_{k+1} + \omega(Q_{k+1} - Q_k);\\
U_{k+1} &= U_k - \tau_U (DF(U_k))^\top \widetilde{Q}_{k+1}.
\end{aligned}
$$

- We replace the original proximal step on U by explicit update.
  - Gain: Convenient to implement;
  - Loss: The convergence rate might be weaker (but still guaranteed to converge).
- Equilibrium of the PDHG method is $(U_*, 0)$, with $F(U_*) = 0$.

$$F(U) \triangleq \mathscr{D}U + h_t\mathscr{G}_h(a\mathscr{L}_hU + bf(U)) - \mathrm{Const\ vector} = 0.$$

$$\mathscr{M} \triangleq \mathscr{D} + ah_t\mathscr{G}_h\mathscr{L}_h + bh_t\mathscr{G}_hJ_f$$

However, the significant conditional number of $F(\cdot)$ (even for linear case) may undermine the convergence of our method.

We need to **precondition** $F(\cdot)$.

Decompose $F(\cdot)$ as

$$\begin{aligned} F(U) &= \mathscr{D}U + h_t\mathscr{G}_h(a\mathscr{L}_hU + bf(U)) - V \\ &= (\mathscr{D} + ah_t\mathscr{G}_h\mathscr{L}_h)U + bh_t\mathscr{G}_h(f(\overline{U}) + J_f(U - \overline{U}) + R(U)) - V. \\ &= (\mathscr{D} + ah_t\mathscr{G}_h\mathscr{L}_h + bh_t\mathscr{G}_hJ_f)U + bh_t\mathscr{G}_hR(U) - (bh_t\mathscr{G}_h(f(\overline{U}) - J_f\overline{U}) - V) \\ &= \mathscr{M}U + bh_t\mathscr{G}_hR(U) - \widetilde{V}. \end{aligned}$$

- $\overline{U}$ is usually treated as a certain steady state of the reaction-diffusion equation.
- J_f is the Jacobian (or certain constant approximation of the Jacobian) of f at $\overline{U}$, $R(\cdot)$ is the remainder term. $Df(\overline{U}) \approx J_f = cI,\ (c \geq 0)$
- $\widetilde{V}$ is a constant vector.

We then precondition $F(\cdot)$ with $\mathscr{M}^{-1}$, i.e., consider

$$\widehat{F}(\cdot) = \mathscr{M}^{-1} F(\cdot),$$

and apply the PDHG algorithm to $\widehat{F}(\cdot)$.

E.g.

- For Allen-Cahn equation, $\mathcal{G}_h = I$, $\mathcal{L}_h = \Delta_h$, and $\overline{U} = \pm 1$, $J_f = Df(\overline{U}) = W''(\overline{U}) = 2I$. $\mathscr{M} = \mathscr{D} + ah_t I_t \otimes \Delta_h + 2bh_t I$.
- For Cahn-Hilliard equation, $\mathcal{G}_h = \Delta_h$, $\mathcal{L}_h = \Delta_h$, $J_f = 2I$. $\mathscr{M} = \mathscr{D} + ah_t(I_t \otimes \Delta_h)^2 + 2bh_t I_t \otimes \Delta_h$.
- Some more general cases in numerical examples.

Both linear systems w.r.t. $\mathscr{M}$ can be efficiently solved by backward substitution + FFT (periodic bc) / DCT (Neumann bc).

The PDHG algorithm for $\widehat{F}(\cdot)$:

$$
\begin{aligned}
Q_{k+1} &= \frac{1}{1+\epsilon\tau_P}(Q_k + \tau_P(\mathscr{M}^{-1}F(U_k)));\\
\widetilde{Q}_{k+1} &= Q_{k+1} + \omega(Q_{k+1} - Q_k);\\
U_{k+1} &= U_k - \tau_U((\mathscr{M}^{-1}DF(U_k))^\top \widetilde{Q}_{k+1}).
\end{aligned}
$$

This is equivalent to the following version of the G-prox PDHG method [Jacobs et al. 2018] with $P_k = \mathscr{M}^{-\top}Q_k$.

$$
\begin{aligned}
P_{k+1} &= \frac{1}{1+\epsilon\tau_P}(P_k + \tau_P((\mathscr{M}\mathscr{M}^\top)^{-1}F(V_k)));\\
\widetilde{P}_{k+1} &= P_{k+1} + \omega(P_{k+1} - P_k);\\
V_{k+1} &= V_k - \tau_U(DF(V_k)^\top \widetilde{P}_{k+1}).
\end{aligned}
$$

- Termination condition: whenever the norm of the residual term is less than tolerance *tol*, i.e., $\|\mathrm{Res}(U_k)\|_\infty < tol$.

We choose to terminate our PDHG iteration whenever the ℓ^∞ norm of the residual term

$$\mathrm{Res}(U_k) = \left[\dots, \left(\frac{U_k^{t+1} - U_k^t}{h_t} + \mathcal{G}_h(a\mathcal{L}_h U_k^{t+1} + bf(U_k^{t+1}))\right)^\top, \dots\right]^\top_{0\le t\le N_t-1}.$$

We choose $tol = 10^{-6}$ for most of our numerical examples.

- Complexity of each iteration of the proposed PDHG method is $\mathcal{O}(N_t \cdot N_x^2 \log(N_x))$.

N_t from back-substitution, $N_x^2 \log(N_x)$ from FFT.

Table of Contents

Background

Algorithm

Theoretical guarantees

Numerical Experiments

Summary & Future research

We mainly address the following three problems:

- Unique existence of the solution to the root-finding problem $F(U) = 0$;
- Convergence of $\|\widehat{F}(U_t)\|_2$ of the PDHG flow (time continuous version of the PDHG algorithm);
- Convergence of $\|U_k - U_*\|_2$ of the PDHG algorithm.

Unique existence of the solution to $F(U) = 0$

Recall the implicit scheme of the reaction-diffusion equation as

$$\frac{U^{t+1} - U^{t}}{h_t} = -\mathcal{G}_h(a\mathcal{L}_h U^{t+1} + bf(U^{t+1})),$$

- Suppose that the reaction term $f \in C^1(\mathbb{R}^1)$ can be decomposed as the sum

  $$f = V' + \phi,$$

  where $V \in C^1(\mathbb{R})$ is convex, and $\phi \in C(\mathbb{R})$ is Lipschitz. Furthermore, suppose V satisfies

  $$(V'(x) - V'(y), x - y) \geq K|x - y|^2,$$

  with $K \geq 0$.

- Suppose the spectral decomposition of $\mathcal{G}_h$ as

  $$\mathcal{G}_h = \left[\begin{array}{c|c} Q_1 & Q_2 \end{array}\right] \begin{bmatrix} \Lambda & \\ & O \end{bmatrix} \begin{bmatrix} Q_1^\top \\ Q_2^\top \end{bmatrix}.$$

Unique existence of the solution to $F(U) = 0$

Theorem 1 (Existence and uniqueness of $F(U) = 0$)

We assume that $\mathcal{G}_h$, $\mathcal{L}_h$ used in our finite difference scheme are non-negative definite and self-adjoint. If the time stepsize h_t of the scheme satisfies

$$\lambda_{\min}\left(\frac{\Lambda^{-1}}{h_t} + a\, Q_1^\top \mathcal{L}_h Q_1\right) + bK > b\,\mathrm{Lip}(\phi), \tag{2}$$

then the root-finding problem $F(U) = 0$ admits a unique solution.

E.g. (Similar results on AC, CH equations are also in [Xu et al. 2019].)

- (Allen-Cahn with periodic b.c.) Recall $f(x) = W'(x) = V'(x) + \phi(x)$,

$$V(x) = \begin{cases} \frac{1}{4}(x^2-1)^2, & |x| > 1; \\ 0 & |x| \leq 1. \end{cases} \qquad \phi(x) = \begin{cases} 0, & |x| > 1; \\ x^3 - x, & |x| \leq 1. \end{cases}$$

  In this case, (2) yields $h_t < \frac{1}{2b}$.

- (Cahn-Hilliard with periodic b.c.) (2) yields $h_t < \frac{a^2}{b^2}$.

Convergence of PDHG flow

As we send $\tau_U, \tau_P \to 0$ and $(1+\omega)\tau_P \to \gamma > 0$, the original PDHG algorithm reduce to a time-continuous dynamic on (P, U)-space.

$$\begin{aligned} \dot{Q}_t &= -\epsilon Q_t + \widehat{F}(U_t), \\ \dot{U}_t &= -D\widehat{F}(U_t)^\top (Q_t + \gamma \dot{Q}_t). \end{aligned}$$

By analyzing the Lyapunov functional

$$\mathcal{I}(U_t, P_t) = \frac{1}{2}\|\widehat{F}(U_t)\|^2 + \frac{\mu}{2}\|Q_t\|^2,$$

we are able to prove the convergence of $\|\widehat{F}(U_t)\|$ as long as

$$\underline{\sigma} = \inf_{U \in \mathbb{R}^{N_x^2}} \{\sigma_{\min}(D\widehat{F}(U))\} > 0; \quad \overline{\sigma} = \sup_{U \in \mathbb{R}^{N_x^2}} \{\sigma_{\max}(D\widehat{F}(U))\} < \infty.$$

Consider the following assumptions:

- (A) a, b are non-negative;
- (B) The reaction term $f(\cdot)$ is Lipschitz;
- (C) $\mathcal{L}_h \succeq O, \mathcal{G}_h \succeq O$ are self-adjoint, and commute, i.e., $\mathcal{G}_h\mathcal{L}_h = \mathcal{L}_h\mathcal{G}_h$;
- (D) J_f is a constant diagonal matrix cI with $c \geq 0$.

Theorem 2 (Convergence of PDHG flow (general RD equation))

Suppose the conditions (A), (B), (C), (D) hold. Pick h_t and $T = N_t h_t$ with

$$bT\mathrm{Lip}(R)\zeta_{a,b,c}(h_t) < 1.$$

Then there exists a unique root of $\widehat{F}(U) = 0$. Furthermore, denote $\theta = bT\mathrm{Lip}(R)\zeta_{a,b,c}(h_t) < 1$; and set $\epsilon = \kappa - \frac{1}{2}$ and $\gamma = \frac{1}{\kappa} - \frac{1}{2\kappa^2}$; then

$$\|\widehat{F}(U_t)\| \leq \exp\left(-\frac{5}{32}\cdot\frac{(1-\theta)^3}{1+\theta}\,t\right)\sqrt{\|\widehat{F}(U_0)\|^2 + (1+\theta)\|Q_0\|^2}.$$

$$\text{Here,}\quad \kappa = \frac{\overline{\sigma}}{\underline{\sigma}}, \quad \zeta_{a,b,c}(h_t) = \max_{1\leq k\leq N_x^2}\left\{\frac{\lambda_k(\mathcal{G}_h)}{1 + h_t(a\lambda_k(\mathcal{G}_h)\lambda_k(\mathcal{L}_h) + bc\lambda_k(\mathcal{G}_h))}\right\}.$$

Theorem 3 (Convergence of PDHG flow (AC/CH-type equation))

Suppose (A), (B), (C), (D) hold, We pick $T = N_t h_t$ ($N_t \in \mathbb{N}_+$) such that

- (Allen-Cahn type, $\mathcal{G}_h = I$, $\mathcal{L}_h$ is self-adjoint, non-negative definite)

$$T < \frac{1}{\mathrm{bLip}(R)}, \quad \text{or } h_t < \frac{1}{\mathrm{bLip}(R)} \text{ and } N_t \le \left\lfloor \frac{1}{\mathrm{bLip}(R) h_t} \right\rfloor.$$

  Denote $\widetilde{\theta} = \mathrm{bLip}(R)T < 1$;

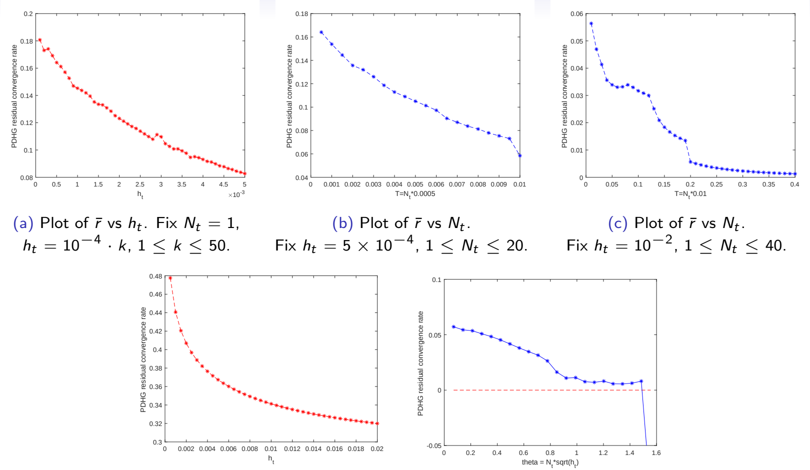
- (Cahn-Hilliard type, $\mathcal{G}_h = \mathcal{L}_h$ are self-adjoint, and non-negative definite)

$$T < \frac{2\sqrt{ah_t} + bch_t}{\mathrm{bLip}(R)}, \quad \text{or } h_t < \frac{4a}{b^2(\mathrm{Lip}(R) - c)_+^2} \text{ and } N_t \le \left\lfloor \frac{2\sqrt{a/h_t} + bc}{\mathrm{bLip}(R)} \right\rfloor.$$

  Denote $\widetilde{\theta} = \frac{\mathrm{bLip}(R)T}{2\sqrt{ah_t}+bch_t} = \frac{\mathrm{bLip}(R)N_t\sqrt{h_t}}{2\sqrt{a}+bc\sqrt{h_t}} < 1$;

Suppose further that $\epsilon = \kappa - \frac{1}{2}$ and $\gamma = \frac{1}{\kappa} - \frac{1}{2\kappa^2}$, then

$$\|\widehat{F}(U_t)\| \le \exp\left(-\frac{5}{32} \cdot \frac{(1-\widetilde{\theta})^3}{1+\widetilde{\theta}}\ t\right)\sqrt{\|\widehat{F}(U_0)\|^2 + (1+\widetilde{\theta})\|Q_0\|^2}.$$



(a) Plot of $\bar{r}$ vs h_t. Fix $N_t = 1$, $h_t = 10^{-4} \cdot k$, $1 \leq k \leq 50$.

(b) Plot of $\bar{r}$ vs N_t. Fix $h_t = 5 \times 10^{-4}$, $1 \leq N_t \leq 20$.

(c) Plot of $\bar{r}$ vs N_t. Fix $h_t = 10^{-2}$, $1 \leq N_t \leq 40$.

(d) Plot of $\bar{r}$ vs h_t. Fix $N_t = 1$, $h_t = 5 \times 10^{-4} \cdot k$, $1 \leq k \leq 40$.

(e) Plot of $\bar{r}$ vs N_t. Fix $h_t = 0.005$, $1 \leq N_t \leq 22$.

Figure: Plot of the convergence rate of the residual term $\|\widehat{F}(U_k)\|$ w.r.t. h_t, N_t. (Up) Allen-Cahn (AC), (Down) Cahn-Hilliard (CH1).

Convergence of PDHG algorithm

Now we switch to the PDHG algorithm, we prove convergence for the error $\|U_k - U_*\|_2$.

Theorem 4 (Convergence of PDHG algorithm (AC/CH-type equ))

Suppose (A), (B), (C), (D) hold. Assume h_t, N_t satisfy

- (Allen-Cahn type) $N_t \cdot h_t < \frac{\sqrt{2}-1}{b\mathrm{Lip}(R)}$. Denote $\theta = b\mathrm{Lip}(R)N_t h_t < \sqrt{2}-1$;
- (Cahn-Hilliard type) $h_t < \frac{4(\sqrt{2}-1)^2 a}{b^2(\mathrm{Lip}(R)-(\sqrt{2}-1)c)_+^2}$, $N_t \le \left\lfloor (\sqrt{2}-1)\frac{2\sqrt{a/h_t}+bc}{b\mathrm{Lip}(R)} \right\rfloor$. Denote $\theta = \frac{b\mathrm{Lip}(R)N_t\sqrt{h_t}}{2\sqrt{a}+bc\sqrt{h_t}} < \sqrt{2}-1$.

Then, there is unique U_* with $\widehat{F}(U_*) = 0$. Choose $u \in (\frac{\theta^2}{1-2\theta}, 1)$ and set

$$\tau_P = \frac{u(1-2\theta)-\theta^2}{8\sqrt{u(1-u)}(1+\theta)^2 \max\{u(1+\theta)^2, 1-u\}}, \tau_U = \frac{\tau_P}{1-u}, \omega = \frac{\sqrt{u(1-u)}}{\tau_U}, \epsilon = \sqrt{\frac{u}{1-u}}.$$

Then U_k converges linearly to U_* with the rate $1 - \Phi/2 + \mathcal{O}(\Phi^2)$, i.e.,

$$\|U_k - U_*\|_2^2 \le C_0 \left(\frac{2}{\Phi + \sqrt{\Phi^2+4}}\right)^{k+1}, \quad \Phi = \frac{(1-2\theta)^2}{8(1+\theta)^2} \cdot \frac{\left(1 - \frac{\theta^2}{1-2\theta} \cdot \frac{1}{u}\right)^2}{\max\{(1+\theta)^2, (1-u)/u\}}.$$

Convergence of PDHG algorithm (simplified version)

Now we switch to the PDHG algorithm, we prove convergence for the error $\|U_k - U_*\|_2$.

Theorem

Suppose $f(\cdot)$ is Lipschitz, $R(U) = f(U) - (f(\overline{U}) - J_f(U - \overline{U}))$; $\mathcal{L}_h\mathcal{G}_h = \mathcal{G}_h\mathcal{L}_h$.
Assume h_t, N_t satisfy

- (AC type $\mathcal{G}_h = I$) $N_t \cdot h_t < \frac{\sqrt{2}-1}{b\mathrm{Lip}(R)}$;
- (CH type $\mathcal{L}_h = \mathcal{G}_h$) $h_t < \frac{4(\sqrt{2}-1)^2 a}{b^2(\mathrm{Lip}(R)-(\sqrt{2}-1)c)_+^2}$, $N_t \leq \left\lfloor (\sqrt{2}-1)\frac{2\sqrt{a/h_t}+bc}{b\mathrm{Lip}(R)} \right\rfloor$.

Then, there exists a unique solution U_* to $F(U) = 0$.
We can pick suitable hyperparameters $\tau_P, \tau_U, \omega, \epsilon$ s.t. U_k converges linearly to U_*,

$$\|U_k - U_*\|_2^2 \leq \mathrm{Constant} \cdot \left(\frac{2}{\gamma + \sqrt{\gamma^2 + 4}} \right)^{k+1},$$

where $\gamma > 0$ is **independent** of the space discretization.

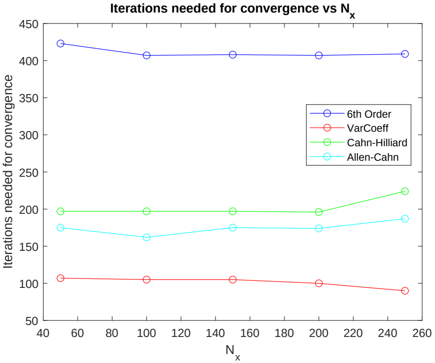
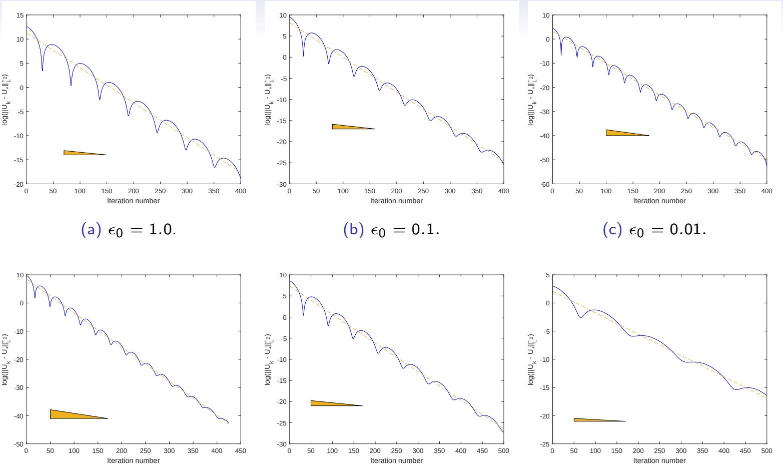


Figure: Relation between the number of iterations needed for convergence and space discretization N_x. We verify on four different equations with $N_x = 50, 100, 150, 200, 250$. We set $\epsilon_0 = 0.01$ for Allen-Cahn equation and $\epsilon_0 = 0.1$ for Cahn-Hilliard equation.



(a) $\epsilon_0 = 1.0$.

(b) $\epsilon_0 = 0.1$.

(c) $\epsilon_0 = 0.01$.

(d) $\epsilon = 10$.

(e) $\epsilon = 1.0$.

(f) $\epsilon = 0.1$.

Figure: Plot of $\log \|U_k - U_*\|^2$ vs iteration k when using hyperparameters specified in the next table to solve Allen-Cahn (AC) (upper figures, $1 \leq k \leq 400$) and Cahn-Hilliard (CH1) (down figures, $1 \leq k \leq 500$) equations with different ϵ_0 on 128×128 grid.

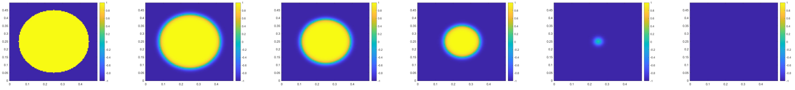
		h_t	N_t	u	τ_P	τ_U	ω	ϵ	$\widetilde{\theta}$	Φ	Actual rate
AC	$\epsilon_0 = 1.0$	0.005 < 0.1381	20 ≤ 27	0.5 $u \in (0.2250, 1)$	0.0498	0.0996	5.0181	1.0	0.3000	0.0112	0.0723
	$\epsilon_0 = 0.1$	0.001 < 0.0138	7 ≤ 13	0.5 $u \in (0.0760, 1)$	0.0574	0.1147	4.3587	1.0	0.2100	0.0141	0.0821
	$\epsilon_0 = 0.01$	0.0005 (< 0.0014)	1 (≤ 2)	0.5 $(u \in (0.0321, 1))$	0.0936	0.1872	2.6702	1.0	0.1500	0.0307	0.1325
CH	$\epsilon_0 = 10$	0.005 (< 1.4553)	10 (≤ 12)	0.5 $(u \in (0.04, 1))$	0.0842	0.1684	2.9695	1.0	0.1640	0.0260	0.0537
	$\epsilon_0 = 1.0$	0.001 (< 0.1455)	5 (≤ 9)	0.5 $(u \in (0.0978, 1))$	0.0475	0.0949	5.2662	1.0	0.2874	0.0103	0.0301
	$\epsilon_0 = 0.1$	0.0005 (< 0.0015)	1 (≤ 1)	0.5 $(u \in (0.1663, 1))$	0.0286	0.0572	8.7392	1.0	0.2741	0.0043	0.0169

Theoretical convergence rate $\Phi/2$ vs actual convergence rate of $\|U_k - U_*\|_2^2$. The constraints in the parentheses in the columns of h_t, N_t, and u are derived from the conditions in Theorem 4. The actual rate r is solved from the linear regression model $r \cdot k + b$ given the numerical data $\{k, \log(\|U_{k+1} - U_*\|^2 / \|U_k - U_*\|^2)\}$.

Allen-Cahn equation

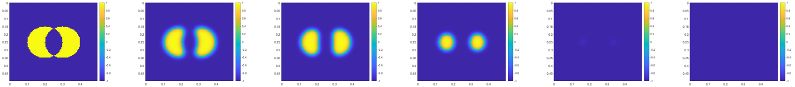
$$\frac{\partial u}{\partial t} = \epsilon_0 \Delta u - \frac{1}{\epsilon_0} W'(u), \quad \text{on } [0, 0.5]^2 \times [0, T], \quad u(x, 0) = u_0(x). \quad (AC)$$

We set $\epsilon_0 = 0.01$. We set the initial condition as $u_0 = 2\chi_{B(x_*, r)} - 1$ with $x_* = (0.25, 0.25), r = 0.2$. We impose the equation with periodic b.c.



(a) $t = 0.0$ (b) $t = 0.5$ (c) $t = 1.0$ (d) $t = 1.5$ (e) $t = 2.0$ (f) $t = 2.5$

Figure: Numerical solution at different times with initial condition $u_0 = 2\chi_B - 1$.



(a) $t = 0.0$ (b) $t = 0.05$ (c) $t = 0.1$ (d) $t = 0.2$ (e) $t = 0.3$ (f) $t = 0.4$

Figure: Numerical solution at different times with a new initial condition. Notice that in the last plot, we have almost converged to the equilibrium solution $u = -1$.

Allen-Cahn equation

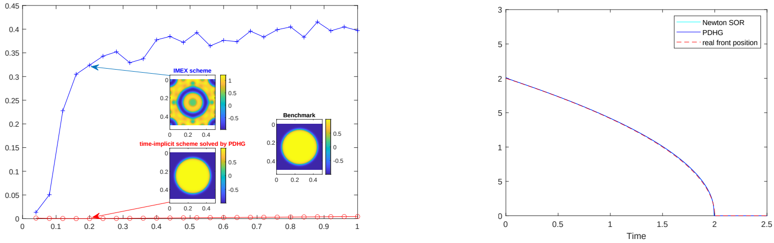
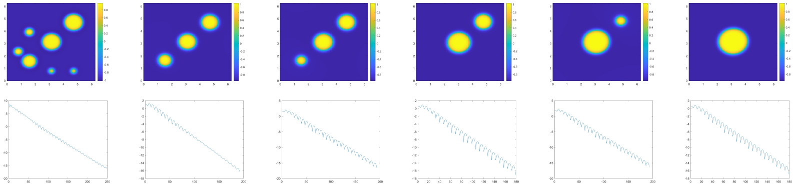


Figure: We solve the Allen-Cahn equation with $\epsilon_0 = 0.01$. (Left) Comparison between our method and the IMEX scheme. We use grid size 128×128. We compute both schemes with large time stepsize $h_t = 0.02$ and compare with the benchmark solution solved from the same IMEX scheme with $h_t = 0.001$. Blue curve indicates the L^1 error of the IMEX solution on the coarser time grid; Red curve indicates the L^1 error between the time-implicit solution. (Right) Comparison between the front position of the numerical solution solved via our PDHG method and the Nonlinear SOR method, as well as the real front position.

Cahn-Hilliard equation

$$\frac{\partial u(x,t)}{\partial t} = -\epsilon_0^2 \Delta\Delta u(x,t) + \Delta W'(u(x,t)), \text{ on } [0,2\pi]^2, \quad u(\cdot,0) = u_0. \quad (CH1)$$

We set $\epsilon_0 = 0.1$. We set the initial condition as indicator function of seven circles. We impose the equation with periodic b.c.

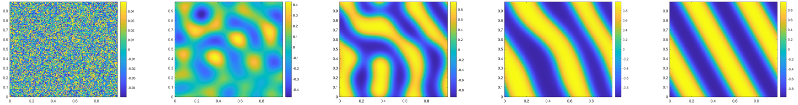


(a) $t = 0.0$ (b) $t = 1.0$ (c) $t = 5.0$ (d) $t = 15.0$ (e) $t = 25.0$ (f) $t = 30.0$

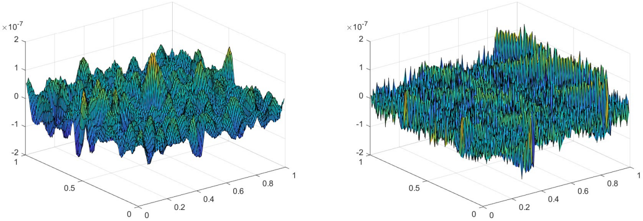
Figure: Numerical solution and $\log_{10} \mathrm{Res}(U_n)$ plot at different time stages for (CH1). The residual plots verify the linear convergence of the PDHG method.

Cahn-Hilliard equation

We consider the same Cahn-Hilliard equation (CH2) with random initial condition and periodic boundary condition.



(a) $t = 0.0$ (b) $t = 0.001$ (c) $t = 0.003$ (d) $t = 0.01$ (e) $t = 1.0$



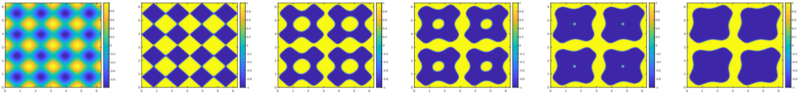
(a) Plot of the residual $\mathrm{Res}(U)$ at $t = 0.01$ (b) Plot of the residual $\mathrm{Res}(U)$ at $t = 1.0$

A reaction-diffusion equation with mobility term

$$\frac{\partial u}{\partial t} = a\nabla \cdot (\sigma(x)\nabla u) - bW'(u), \quad \text{on } [0, 2\pi]^2 \times [0, T], \quad u(x, 0) = u_0(x).$$

We choose $a = \epsilon_0, b = \frac{1}{\epsilon_0}$. We set $\epsilon_0 = 0.01$. The mobility term (variable coefficient) $\sigma(x, y) = 1 + \frac{\mu}{2}(\sin^2 x + \sin^2 y)$ with $\mu = 5.0$. The initial condition is $u_0(x, y) = \frac{1}{2}(\cos(4x) + \cos(4y))$. The equation is imposed with periodic boundary condition.

When we set up the precondition matrix $\mathscr{M}$, we replace the original $\mathcal{L}_h = \nabla_h \cdot (\sigma \nabla_h)$ by $\widetilde{\mathcal{L}}_h = \overline{\sigma}\Delta_h$ where $\overline{\sigma}$ denotes the average value of σ over Ω.



$t = 0.0$ $t = 0.2$ $t = 1.0$ $t = 2.4$ $t = 3.6$ $t = 5.0$

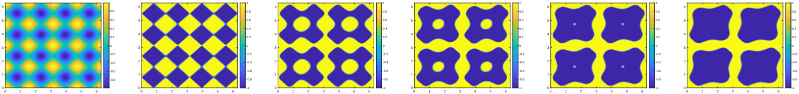
Figure: Numerical solution of the time-implicit scheme solved via our PDHG method on a 256×256 grid at different time stages $t = 0.0, 0.2, 1.0, 3.6, 5.0$.

A reaction-diffusion equation with mobility term

$$\frac{\partial u}{\partial t} = a\nabla\cdot(\sigma(x)\nabla u) - bW'(u), \quad \text{on } [0,2\pi]^2\times[0,T], \quad u(x,0)=u_0(x).$$

We choose $a=\epsilon_0, b=\frac{1}{\epsilon_0}$. We set $\epsilon_0 = 0.01$. The mobility term (variable coefficient) $\sigma(x,y) = 1+\frac{5}{2}(\sin^2 x + \sin^2 y)$ with $\mu = 5.0$. The initial condition is $u_0(x,y)=\frac{1}{2}(\cos(4x)+\cos(4y))$. The equation is imposed with periodic boundary condition.

When we set up the precondition matrix $\mathscr{M}$, we replace the original $\mathcal{L}_h = \nabla_h\cdot(\sigma\nabla_h)$ by $\widetilde{\mathcal{L}}_h = \overline{\sigma}\Delta_h$ where $\overline{\sigma}$ denotes the average value of σ over Ω.



$t=0.0$ $t=0.2$ $t=1.0$ $t=2.4$ $t=3.6$ $t=5.0$

Figure: Numerical solution of the time-implicit scheme solved via our PDHG method on a 256×256 grid at different time stages $t=0.0, 0.2, 1.0, 3.6, 5.0$.

A reaction-diffusion equation with mobility term

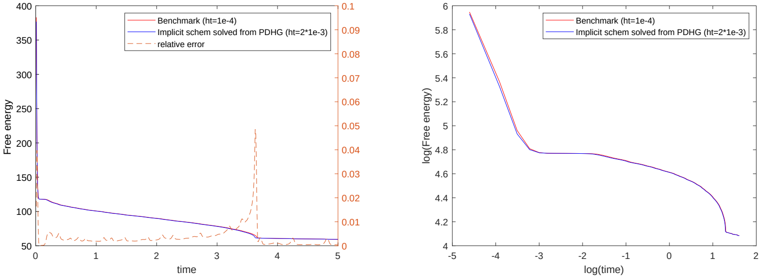


Figure: (Left) Free energy decay (blue) of the implicit scheme (solved by PDHG method) with $h_t = 2 \cdot 10^{-3}$, and the reference energy decay (red) solved from IMEX scheme with $h_t = 10^{-4}$. The relative error between them is plotted in orange. (Right) $\log - \log$ plots of free energy.

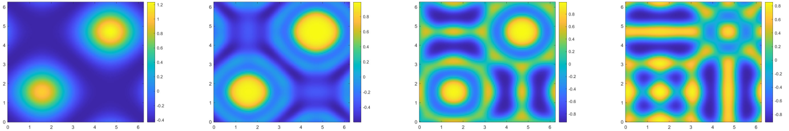
A 6th order equation

We consider the 6th-order Cahn-Hilliard-type equation depicting pore formation in functionalized polymers [Gavish et al. 2012].

$$\frac{\partial u(x,t)}{\partial t} = \Delta(\epsilon_0^2\Delta - W''(u) + \epsilon_0^2)(\epsilon_0^2\Delta u - W'(u)) \text{ on } [0,2\pi]^2,\ u(\cdot,0) = u_0.$$

Set $\epsilon = 0.18$ with initial condition $u_0 = 2e^{\sin x + \sin y - 2} + 2.2e^{-\sin x - \sin y - 2} - 1$.

When we set up the precondition matrix $\mathscr{M}$, we replace original $\mathcal{G}_h = \Delta_h(\epsilon_0^2\Delta_h - \mathrm{diag}(W''(U)) + \epsilon_0^2 I)$ by $\widetilde{\mathcal{G}_h} = \Delta_h(\epsilon_0^2\Delta_h - W''(\bar{u})I + \epsilon_0^2 I)$, where $W''(\bar{u}) = W''(\pm 1) = 2$.



(a) $t = 0.0$ (b) $t = 0.1$ (c) $t = 2.0$ (d) $t = 20.0$

Figure: Numerical solution at different time stages.

Hyperparameter selection

Given h_x, h_t of the implicit scheme, there are 5 hyperparameters to be determined, $N_t, \tau_U, \tau_P, \omega, \epsilon$. How to choose them properly?

- (Choosing N_t) Suppose we are to solve an equation on $[0, T_{\text{total}}]$, we divide the time interval into $M \cdot N_t$ small intervals, i.e.,

$$[0, T_{\text{total}}] = \bigcup_{k=1}^{M} I_k = \bigcup_{k=1}^{M} \left(\bigcup_{j=1}^{N_t} I_{k,j} \right), \quad \text{where each } I_{k,j} = [(k-1)T + (j-1)h_t, (k-1)T + jh_t].$$

$$\text{with } T = T_{\text{total}}/M, h_t = T/N_t.$$

We then apply our proposed method to each subinterval I_k in order to obtain the entire numerical solution on $[0, T_{\text{total}}]$. It is usually the most efficient to pick the hyperparameter $N_t \leq 3$.

Equation Name $[\tau_U, \tau_P, T]$	$M \times N_t$								
	1×100	2×50	4×25	10×10	20×5	25×4	$33 \times 3 + 1$	50×2	100×1
AC($\epsilon_0 = 0.01$) $[0.5, 0.5, 1.0]$	–	–	1198.41	219.52	137.71	138.65	**88.53**	106.41	92.72
AC($\epsilon_0 = 0.1$) $[0.5, 0.5, 1.0]$	–	–	90.28	57.73	34.37	50.43	41.37	26.62	**24.20**
AC($\epsilon_0 = 1$) $[0.5, 0.5, 1.0]$	64.28	38.11	23.42	24.24	13.05	13.29	12.51	10.89	**10.72**
CH $[0.5, 0.5, 1.0]$	775.15	208.93	170.77	252.99	148.96	183.34	101.41	**77.35**	86.37
6th Order $[0.8, 0.8, 0.1]$	–	–	374.82	389.90	285.12	384.52	199.11	**188.58**	208.30
Varcoeff $[0.95, 0.5, 1.0]$	–	–	305.73	206.72	204.34	153.88	144.67	142.22	**61.46**

Figure: Comparison of CPU time (s) with different N_ts (All problems are solved on 256×256 grids).

Hyperparameter selection

- (Choosing τ_U, τ_P) In practice, we pick a larger τ_U, τ_P to achieve faster convergence. The optimal stepsize τ_P is around 0.9, and the optimal ratio $k_\tau = \frac{\tau_P}{\tau_U}$ should be slightly less than 2.

$\epsilon = 0.1$ for all problems		$\tau_U = 0.9, \tau_P = 0.5$	$\tau_U = 0.65, \tau_P = 0.65$	$\tau_U = 0.5, \tau_P = 0.9$
6thOrder $[T = 0.5]$	$N_x = 256, N_t = 50$	62.28	47.92	**30.53**
	$N_x = 128, N_t = 50$	12.31	9.47	**8.54**
VarCoeff $[T = 0.5]$	$N_x = 256, N_t = 50$	103.23	109.38	**82.38**
	$N_x = 128, N_t = 50$	15.92	13.35	**9.54**

Figure: Comparing speeds among different ratios $k_\tau = \frac{\tau_P}{\tau_U}$ for different equations.

The intuition is that we expect to treat the inner optimization of $\min_u \max_Q \widehat{L}(U, Q)$ w.r.t. the dual variable Q more thoroughly.

- (Choosing ω) We pick $\omega = 1$ in our experiments.
- (Choosing ϵ) We set ϵ around 0.1. The method experiences stronger oscillations when ϵ approaches 0; The method gets slower when ϵ increases beyond 0.1.

Comparison with classical root-finding algorithms

We compare our PDHG method with three classical root-finding algorithms, Nonlinear SOR, fixed point, and Newton's method. We always set $\tau_U = 0.5, \tau_P = 0.95$ for our PDHG method. We solve step by step ($N_t = 1$) in our PDHG method.

- (Nonlinear SOR) We solve the Allen-Cahn equation (AC) with $\epsilon_0 = 0.1$ and $h_t = 0.005$. We solve the equation on 128×128 grid.

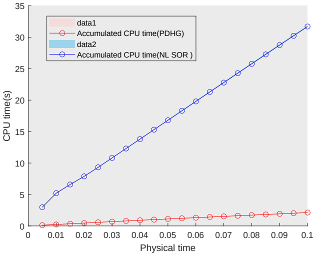


Figure: Accumulated CPU time comparison between our method (red) and Nonlinear SOR method (blue) applied to Allen-Cahn equation. The quantile plots are composed based on 40 independent runs of both algorithms.

- (Fixed point method) We solve the Cahn-Hilliard equation (CH1) with $\epsilon_0 = 0.1$ and $h_t = 0.01$. We solve the equation on 256 × 256 grid.

When applying the fixed point method, we need to solve a linear equation. We apply the preconditioned conjugate gradient (PCG) method with the same precondition used in our PDHG algorithm to solve such equations.

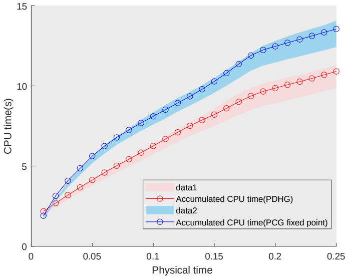


Figure: Accumulated CPU time comparison between our method (red) and PCG-fixed point iteration (blue). We solve the equation with mobility. These quantile plots are composed based on 40 independent runs of both algorithms.

- (Newton's method [Christlieb et al. 2014]) We solve the 6th order equation with on 256 × 256 grid.

When applying Newton's method, we need to solve linear equations involving Jacobian matrix. We apply the preconditioned conjugate gradient (PCG) method with the precondition matrix suggested in [Christlieb et al. 2014] to solve the linear equations.

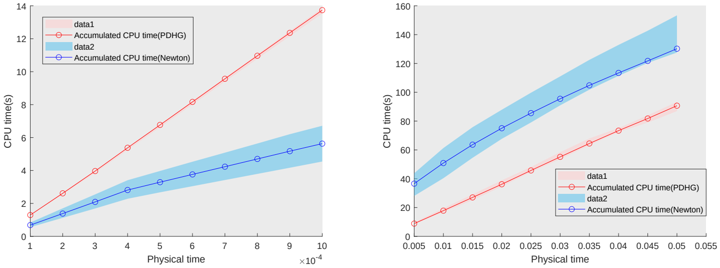


Figure: Accumulated CPU time comparison between our method (red) and Newton's method (blue). Solving the 6th order equation with $h_t = 0.001$ (Left) and $h_t = 0.005$ (Right). These quantile plots are composed based on 40 independent runs of both algorithms.

Table of Contents

Summary & Future research

In this research, we

- Apply PDHG algorithm to resolve implicit schemes of reaction-diffusion equations;
- Provide the convergence guarantee for both the PDHG flow (time-continuous version) and the PDHG algorithm;
- Justify our theoretical findings via numerical examples; Test our method on various types of reaction-diffusion equations;
- Verify the proper hyperparameters of our method;
- Compare our method with the classical root-finding algorithms.

Summary & Future research

Possible future directions

- Can we prove a sharper convergence rate (may need more sophisticated Lyapunov functional)?
- Better preconditioner $\mathscr{M}$? time-dependent preconditioner?
- Since we formulate the numerical PDE scheme as an optimization problem, can we apply parallel computing techniques to accelerate it? [Lions et al. 2001], [Lederman et al. 2018].
- Apply the (preconditioned) PDHG method to machine learning and design a saddle scheme for solving PDEs in high-dimensional spaces.

More details on our paper:

- Shu Liu, Siting Liu, Stanley Osher, and Wuchen Li. A first-order computational algorithm for reaction-diffusion type equations via primal-dual hybrid gradient method. JCP Volume 500, 1 March 2024. (Methodology & Numerical experiments.)
- Shu Liu, Xinzhe Zuo, Stanley Osher, Wuchen Li. Numerical analysis of a first-order computational algorithm for reaction-diffusion equations via the primal-dual hybrid gradient method. arXiv: 2401.14602. (Numerical analysis & Comparison with classical methods.)

Thank you!

Welcome to any comments & questions.