Nonoscillatory second-order procedures
for partial differential equations of nonsmooth data

By

Philku Lee

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Science
in Mathematical Sciences
in the Department of Mathematics and Statistics

Mississippi State, Mississippi

August 2020
Nonoscillatory second-order procedures
for partial differential equations of nonsmooth data

By

Philku Lee

Approved:

Seongjai Kim
(Major Professor)

George V. Popescu
(Co-Major Professor)

Mohsen Razzaghi
(Department Head)

Xiangsheng Xu
(Committee Member)

Jonathan R. Woody
(Committee Member)

Mohammad Sepehrifar
(Graduate Coordinator)

Rick Travis
Dean
College of Arts & Sciences
Elliptic obstacle problems are formulated to find either superharmonic solutions or minimal surfaces that lie on or over the obstacles, by incorporating inequality constraints. This dissertation investigates simple iterative algorithms based on the successive over-relaxation (SOR) method. It introduces subgrid methods to reduce accuracy deterioration occurring near the free boundary when the mesh grid does not match with the free boundary. For nonlinear obstacle problems, a method of gradient-weighting is introduced to solve the problem more conveniently and efficiently. The iterative algorithm is analyzed for convergence for both linear and nonlinear obstacle problems.

Parabolic initial-boundary value problems with nonsmooth data show either rapid transitions or reduced smoothness in its solution. For those problems, specific numerical methods are required to avoid spurious oscillations as well as unrealistic smoothing of steep changes in the numerical solution. This dissertation investigates characteristics of the $\theta$-method and introduces a variable-$\theta$ method as a synergistic combination of the Crank-
Nicolson (CN) method and the implicit method. It suppresses spurious oscillations, by evolving the solution implicitly at points where the solution shows a certain portent of oscillations or reduced smoothness, and maintains as a similar accuracy as the CN method with smooth data. An effective strategy is suggested for the detection of points where the solution may introduce spurious oscillations (the wobble set); the resulting variable-\(\theta\) method is analyzed for its accuracy and stability.

After a theory of morphogenesis in chemical cells was introduced in 1950s, much attention had been devoted to the numerical solution of reaction-diffusion (RD) equations. This dissertation studies a nonoscillatory second-order time-stepping procedure for RD equations incorporating with variable-\(\theta\) method, as a perturbation of the CN method. We also perform a sensitivity analysis for the numerical solution of RD systems to conclude that it is much more sensitive to the spatial mesh resolution than the temporal one. Moreover, to enhance the spatial approximation of RD equations, this dissertation investigates the averaging scheme, that is, an interpolation of the standard and skewed discrete Laplacian operator and introduce the simple optimizing strategy to minimize the leading truncation error of the scheme.

Key words: Partial differential equations (PDEs), Nonlinear PDEs, Relaxation method, Time-stepping method, \(\theta\)-method, Nonsmooth data, Obstacle problem, Reaction-diffusion problem, Pattern formation, Mehrstellen, Averaging scheme.
DEDICATION

I dedicate my dissertation to

my loving father, Soochan Lee,

my loving mother, Chunwha Park

and my loving brother, Hongkyu Lee

for their loves and supports.

I will always appreciate all they have done.
ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisor Prof. Seongjai Kim. He held me up from my desperation, and gave me invaluable opportunities and continuous supports of my doctoral study and related research with his supervision, motivation, and immense knowledge. Without his support, it would be impossible to conduct and complete this research.

My sincere appreciation also goes toward my co-advisor Prof. George V. Popescu, who has provided me the valuable opportunity to join Institute for Genomics, Biocomputing & Biotechnology as research assistant and inspirations in the research of Computational Biological. His guidance helped me in all the time of my research and doctoral study.

Besides my advisors, I would like to thank the rest of my research committee: Prof. Mohsen Razzaghi, Prof. Xiangsheng Xu, Prof. Jonathan R. Woody and Prof. Mohammad Sepehrifar for their insightful comments and encouragement, which incented me to widen my research from various perspectives.

I thank my fellow graduate students, Sungkeun Jo, Junggyo Jung, Dr. Jinyeop Lee, Dr. Hwamog Kim, Dr. Chartese Jones, Xiao Wang, Derrick Jones, Dr. Rachel Barber, Ghodsieh Ghanbari, Tobias Oketch, and countless others, for their support and friendship.
I thank my precious friends, Dr. Kwonse Kim, Punchaphol Chotivittayathanin, Pattarapon Trachu, Oluwaseun Kyle Johnson, Robert Nieh, Dr. Amirhamed Bakhtiarydavijani, and countless others, for their support and friendship.

I would like to express my sincere gratitude to my family for their endless support and unconditional love.

This research was supported by NSF-MCB 1714157 awarded to Prof. George V. Popescu and my thankful gratitude also goes toward NSF.
TABLE OF CONTENTS

DEDICATION ........................................ ii
ACKNOWLEDGEMENTS ............................... iii
LIST OF TABLES .................................... viii
LIST OF FIGURES ................................... x
LIST OF SYMBOLS AND ABBREVIATIONS ........ xiv

CHAPTER

1. INTRODUCTION ..................................... 1
   1.1 Elliptic Obstacle Problem ...................... 1
   1.2 Parabolic Obstacle Problem ..................... 2
   1.3 Reaction-Diffusion Problem in Biology ........ 5
   1.4 Outline of Dissertation ....................... 7

2. PRELIMINARY .................................... 10
   2.1 Obstacle Problem ............................... 10
      2.1.1 Elliptic and obstacle problem .............. 10
      2.1.2 Accuracy issue for obstacle problems ...... 11
      2.1.3 Parabolic obstacle problem ............... 14
   2.2 Time-Stepping Method ........................ 15
      2.2.1 The \( \theta \)-method ....................... 15
      2.2.2 An example: numerical oscillations of the CN method ... 17
   2.3 Reaction-Diffusion Problem ................... 19

3. ELLIPTIC OBSTACLE PROBLEM ................... 21
   3.1 Obstacle Relaxation for Elliptic Obstacle Problems .... 21
      3.1.1 The linear obstacle problem ............... 21
      3.1.2 The nonlinear obstacle problem ............ 24
5.3.2 FD schemes for the second spatial derivatives . . . . . . . 113
5.3.3 Averaging compact scheme . . . . . . . . . . . . . . . . . . 116
5.4 The Optimized Averaging Parameter . . . . . . . . . . . . . . 117
5.5 Numerical Experiments . . . . . . . . . . . . . . . . . . . . . . 119
5.5.1 Gray-Scott model . . . . . . . . . . . . . . . . . . . . . . . 120
5.5.2 Gierer-Meinhardt model . . . . . . . . . . . . . . . . . . . 123

6. CONCLUSIONS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 127

REFERENCES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 130
3.1 The number of iterations and CPU for the linear problem of the non-smooth obstacle $\varphi_1$ (3.41). For PDL1P [68], set $\mu = 10^8$, $r_1 = 0.01$, and $r_2 = 12.5$. 36

3.2 $L^\infty$-errors, the number of iterations, and the CPU for linear obstacle problem with $\varphi_2$ (3.42): The PDL1P uses the parameters suggested in [68] ($\mu = 0.1$, $r_1 = 0.008$, $r_2 = 15.625$). 39

3.3 The number of iterations and the CPU time for the elastic-plastic torsion problem (3.45). For the PDL1P, we use the parameters suggested in [68] ($\mu = 0.1$, $r_1 = 0.008$, $r_2 = 15.625$). 41

3.4 The performance of the obstacle SOR applied for nonlinear obstacle problems with $\varphi = \varphi_i$, $i = 1, 2, 3$. 43

3.5 CPU time and iteration comparisons for the suggested post-process, applied to the linear problem with $\varphi = \varphi_2$. 47

3.6 $L^\infty$ and $L^2$ error comparisons for the suggested post-process, applied to the linear problem with $\varphi = \varphi_2$. 47

4.1 The accumulated relative $L^2$-error $E_2[0, T]$ for the numerical solution of (4.20), when $T = 1.0$ and $\Delta x = 2\Delta t$. 63

4.2 The accumulated relative $L^2$-errors $E_2[0, T]$ and the average count of wobble points for the numerical solutions of (2.22) and (4.20) when $T = 1.0$, $\Delta x = 0.08$ ($n_x = 25$), and $\Delta t = 0.04$ ($n_t = 25$). The wobble set is assigned either artificially by 0%, 8%, 16%, and 32% largest values of $|\Delta u^{n-1}_i|$ or by the variable-$\theta$ method. 64

4.3 The accumulated relative $L^2$-error $E_2[0, T]$ and the CPU (in second), solving (4.40), when $T = 0.25$ and $\Delta x = \Delta y = 2\Delta t$. 75

4.4 $E_2[0, T]$ and $E_\infty[T]$ for the obstacle problem associated with $\varphi_i$, $i = 1, 2, 3$, in (4.42), when $T = 0.1$, $\Delta x = 0.04$ ($n_x = 25$), and $\Delta t = 0.01$ ($n_t = 10$). 77
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>$E_2[0,T]$, $E_\infty[T]$, and CPU for the obstacle problems associated with $\varphi_4$ and $\varphi_5$, when $T = 0.4$ and $\Delta x = \Delta y = 2\Delta t$.</td>
<td>79</td>
</tr>
<tr>
<td>4.6</td>
<td>$E_2[0,T]$, $E_\infty[T]$, and CPU for the obstacle problem associated with $\varphi_6$, when $T = 0.4$ and $\Delta x = 2\Delta x = 2\Delta t$.</td>
<td>82</td>
</tr>
<tr>
<td>5.1</td>
<td>$L^\infty$-error $E_\infty[T]$ and the elapsed time (CPU) for the numerical solution of the Gray-Scott model in the 2D space at $T = 1.0$.</td>
<td>102</td>
</tr>
<tr>
<td>5.2</td>
<td>Accumulated $L^2$-errors $E[0,T]$, and CPU-time for the Gray-Scott model associated with (5.27)-(5.29) at $T = 1.0$ and variable resolutions and the averaging parameter $\alpha$. Here, $e_1$ and $e_2$ are the $L^2$-errors for $u_1$ and $u_2$, respectively and we summarize the errors with $L^2$-norm $\sqrt{e_1^2 + e_2^2}$.</td>
<td>118</td>
</tr>
<tr>
<td>5.3</td>
<td>Accumulated $L^2$-errors $E[0,T]$, and CPU-time for the Gray-Scott model defined in Section 5.2.2.2 at $T = 1.0$ and variable resolutions by each method. Here, $e_1$ and $e_2$ are the $L^2$-errors for $u_1$ and $u_2$, respectively and we summarize the errors with $L^2$-norm $\sqrt{e_1^2 + e_2^2}$.</td>
<td>121</td>
</tr>
<tr>
<td>5.4</td>
<td>Accumulated $L^2$-errors $E[0,T]$, and CPU-time for the Gray-Scott model defined in Section 5.2.2.2 at $T = 1.0$ and variable temporal resolutions ($h = 0.05$ is fixed) by $A_5$-VT. Here, $e_1$ and $e_2$ are the $L^2$-errors for $u_1$ and $u_2$, respectively and we summarize the errors with $L^2$-norm $\sqrt{e_1^2 + e_2^2}$.</td>
<td>122</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

2.1 A non-matching grid: The true solution \( u \) (red solid curve) and the numerical solution on the non-matching grid \( u_h \) (blue dashed curve). Here the obstacle is the shaded region, \( u(x_0) = u_h(x_0) = \varphi(x_0) = 0 \), and \( u(x_5) = u_h(x_5) = \varphi(x_5) = 1 \). ................................. 12

2.2 The evolution of (a) analytic solution and numerical solution by (b) the implicit method and (c) the CN method for (2.22), for \( 0 \leq t \leq T = 0.5 \), when \( \Delta x = 0.04 \) (\( n_x = 50 \)) and \( \Delta t = 0.02 \) (\( n_t = 25 \)). ................................. 18

3.1 Contact points (red solid circle) and neighboring points (blue open circle). The red dashed curve indicates a possible free boundary. ................................. 31

3.2 Solutions to the linear problem for the obstacle \( \varphi_1 \) (3.41) at resolution \( 64 \times 64 \):
(a) the numerical solution by the PDL1P, (b) its contour plot, (c) the numerical solution by the obstacle SOR, and (d) its contour plot. ................................. 37

3.3 (a) The obstacle \( \varphi_2 \) (3.42) and (b) the true solution \( u^* \) at the \( 64 \times 64 \) resolution. ................................. 38

3.4 Numerical solutions \( u_h \) and errors \( u_h - u^* \) at the \( 64 \times 64 \) resolution. (a)-(b) by the PDL1P and (c)-(d) by the obstacle SOR. ................................. 39

3.5 Elastic-plastic torsion problem: (a) the obstacles (\( \psi_3 \) and \( \varphi_3 \)) and (b) the force \( v \). ................................. 40

3.6 The numerical solutions and the contact sets for the elastic-plastic torsion problem at the \( 64 \times 64 \) resolution: (a)-(b) by the PDL1P and (c)-(d) by the obstacle SOR. ................................. 42

3.7 Nonlinear obstacle problem with \( \varphi = \varphi_1 \) at resolution \( 64 \times 64 \): (a) The nonlinear numerical solution and (b) its contour plot. ................................. 44

3.8 The difference between the linear solution and the nonlinear solution, at the \( 64 \times 64 \) resolution, for the obstacle problem with \( \varphi = \varphi_1 \). (a) \( (u_{h,\text{linear}} - u_{h,\text{nonlinear}}) \) and (b) its density plot. ................................. 45

3.9 The post-processing algorithm applied for an obstacle problem in 1D at resolution 16.
(a) the computed solutions without the post-process (blue solid curve) and with the post-process (red dotted curve with \( \times \) marks) and (b) their errors. Here the subscript \( pp \) indicates the post-process. ................................. 46
3.10 Plots of the error ($u_h - u^*$) at the $50 \times 50$ resolution for the linear obstacle problem with $\varphi = \varphi_2$. (a) by the PDL1P, (b) by the obstacle SOR, and (c) by the obstacle SOR+PP. ................................. 48

3.11 The relaxation parameter $\omega$ (horizontal axis) vs. the number of iterations (vertical axis) for solving the linear obstacle problem with $\varphi = \varphi_3$ by the obstacle SOR:
(a) when $h = 1/64$ and (b) when $h = 1/128$. ................................. 50

4.1 The eight vicinal points of $x_{ij}$ and four partial directions. .................. 54

4.2 The numerical solutions of (2.22) for $0 \leq t \leq T = 0.5$, when $\Delta x = 0.04$ ($n_x = 50$) and $\Delta t = 0.02$ ($n_t = 25$): (a) the numerical solution by variable-$\theta$ method and (b) the numerical solutions at $t = T = 0.5$. ................................. 59

4.3 Propagation of numerical errors for (2.22) by (a) the implicit method, (b) the CN method, and (c) the variable-$\theta$ method, for $0 \leq t \leq T = 0.5$, when $\Delta x = 0.04$ ($n_x = 50$) and $\Delta t = 0.02$ ($n_t = 25$). ................................. 60

4.4 The analytic and numerical solutions of (4.20) for $0 \leq t \leq T = 1.0$, when $\Delta x = 0.02$ ($n_x = 100$) and $\Delta t = 0.01$ ($n_t = 100$): (a) the analytic solution, (b) the CN method, and (c) the variable-$\theta$ method. ................................. 62

4.5 The numerical solutions of (2.22) with the nonsmooth source term (4.22), when $\Delta x = 0.04$ ($n_x = 50$) and $\Delta t = 0.02$ ($n_t = 25$): (a) the implicit method, (b) the CN method, and (c) the variable-$\theta$ method. ................................. 66

4.6 Numerical solutions in Figure 4.5 on the center cross line, $\{0\} \times J$. ........ 66

4.7 Numerical solutions to the 2D heat equation (4.40) at $t = 0.125$, when $T = 0.25$, $\Delta x = \Delta y = 0.025$ ($n_x = n_y = 40$), and $\Delta t = 0.0125$ ($n_t = 20$): (a) the solution of the variable-$\theta$ method over the whole domain and (b) the solutions of the three $\theta$-methods on the horizontal half cross line, $[0, 0.5] \times \{0.5\}$. ................................. 74

4.8 Evolution of numerical solutions for the obstacle problem associated with $\varphi_3$ over $0 \leq t \leq T = 0.1$, when $\Delta x = 0.04$ ($n_x = 25$), $\Delta t = 0.01$ ($n_t = 10$): (a) the implicit method, (b) the CN method, (c) the SINT method, and (d) the variable-$\theta$ method. ................................. 76

4.9 Numerical solutions of the four methods for the obstacle problem associated with $\varphi_4$ at $t = 0.2$, when $T = 0.4$, $\Delta x = \Delta y = 0.08$ ($n_x = n_y = 50$), and $\Delta t = 0.04$ ($n_t = 10$): (a) over the whole domain and (b) on the horizontal half cross line $[-2, 0] \times \{0\}$. ................................. 80

4.10 Numerical solutions for the obstacle problem associated with $\varphi_5$ at $t = 0.2$, when $T = 0.4$, $\Delta x = \Delta y = 0.08$ ($n_x = n_y = 50$), and $\Delta t = 0.04$ ($n_t = 10$):
(a) the obstacle $\varphi_5$ (the solid body inside) and the numerical solution (transparent membrane) by the CN method and (b) the solutions of the four methods on the horizontal half cross line $[-2, 0] \times \{0\}$. ................................. 80
4.11 The obstacle \( \varphi_6 \) and numerical solutions of the obstacle problem associated to \( \varphi_6 \), when \( t = T = 0.4, \Delta x = 0.08, \Delta y = 0.04 \) \( (n_x = n_y = 50) \), and \( \Delta t = 0.04 \) \( (n_t = 10) \): (a) the obstacle \( \varphi_6 \), (b) the CN solution, (c) the SINT solution, and (d) the variable-\( \theta \) solution.

4.12 Elastic-plastic torsion problem with double obstacles: (a) the obstacles \( \phi_7 \) and \( \psi_7 \), (b) the force \( v \), and (c) the solution of the variable-\( \theta \) method, when \( T = 0.05, \Delta x = \Delta y = 0.01 \) \( (n_x = n_y = 100) \), and \( \Delta t = 0.01 \) \( (n_t = 5) \).

4.13 Misfits \( (u_{\text{computed}} - u_{\text{desired}}) \) of the double obstacle problem associated with \( \phi_7 \) and \( \psi_7 \), when \( t = T = 0.05, \Delta x = \Delta y = 0.01 \) \( (n_x = n_y = 100) \), and \( \Delta t = 0.01 \) \( (n_t = 5) \): (a) the implicit method, (b) the CN method, and (c) the variable-\( \theta \) method.

4.14 Contact sets associated with the elastic-plastic torsion problem at \( t = 0.05 \), when \( \Delta x = \Delta y = 0.01 \) \( (n_x = n_y = 100) \) and \( \Delta t = 0.01 \) \( (n_t = 5) \): (a) a desired contact set, (b) by the implicit method, (c) by the CN method, and (d) by the variable-\( \theta \) method.

4.15 Pixel differences between the desired contact set and contact sets by the three \( \theta \)-methods in Figure 4.14: (a) the implicit method (131), (b) the CN method (90), and (c) the variable-\( \theta \) method (55).

4.16 The evolution of numerical solutions for the American put option pricing (4.49), when \( T = 10 \) \( (\frac{\sigma^2 T}{2} = 1.0125) \), \( \Delta x = 0.125 \) \( (n_x = 100) \), and \( \Delta \tau = 0.10125 \) \( (n_\tau = 10) \): (a) the CN method and (b) the variable-\( \theta \) method.

5.1 Propagation of the numerical solutions for (5.23): (a) the implicit method, (b) the CN method, and (c) the variable-\( \theta \) method, for \( 0 \leq t \leq T = 0.5 \), when \( \Delta t = 0.02 \) and \( \Delta x = 0.04 \).

5.2 Propagation of the numerical solution of \( u_2 \) for the 1D Gray-Scott model by the variable-\( \theta \) method: (a) the wave-splitting (self-replication of the pulse) and (b) its aerial view over \( 0 \leq t \leq T = 2000 \), when \( \Delta t = 0.01 \) and \( \Delta x = 0.004 \).

5.3 Propagation of the numerical solution of \( u_2 \) for the 1D Gray-Scott model by the variable-\( \theta \) method: (a) the pulse traveling and reflecting and (b) its aerial view over \( 0 \leq t \leq T = 5000 \), when \( \Delta t = 0.01 \) and \( \Delta x = 0.004 \).

5.4 The wave-splitting (self-replication of the pulse) by the variable-\( \theta \) method over \( 0 \leq t \leq 2000 \) with various \( (\Delta t, \Delta x) \). The image \( I_{k\ell} \) represents the numerical solution obtained with the mesh resolution \( (\Delta t, \Delta x) = (10^{-k}, 0.01/2^{\ell-1}). \)

5.5 The numerical solution of the variable-\( \theta \) method and its error at \( T = 1.0 \) for the Gray-Scott model, when \( \Delta t = 0.025 \) and \( \Delta x = \Delta y = 0.05 \): (a) \( u_1 \), (b) \( u_2 \), (c) \( \hat{e}_1 = u_1 - \tilde{u}_1 \), and (d) \( \hat{e}_2 = u_2 - \tilde{u}_2 \).

5.6 The initial values in (5.31) for the Gierer-Meinhardt model with \( \epsilon = 0.04 \) at the mesh resolution \( \Delta x = \Delta y = 1/32 \): (a) \( u_1(x, y, 0) \) and (b) \( u_2(x, y, 0) \).

5.7 Numerical solutions for \( u_1 \) of the Gierer-Meinhardt model at the spatial resolution \( \Delta x = \Delta y = 1/32 \): (a) the variable-\( \theta \) method with \( \Delta t = 0.05 \), (b) the CN method with \( \Delta t = 0.05 \), and (c) the CN method with \( \Delta t = 0.005 \); (d), (e), and (f) are the aerial views of (a), (b) and (c), respectively.

xii
Numerical solutions for \( u_1 \) of the Gray-Scott model at \( T = 1.0 \) by variable-\( \theta \) method varying \( \Delta x = \Delta y \) with fixed \( \Delta t = 0.25 \). The image \( I_\ell \) represents the numerical solution obtained with the mesh resolution \( (\Delta t, \Delta x = \Delta y) = (0.25, 0.02/2^{\ell-1}) \) and \( I'_\ell \) represents the error of \( I_\ell \).

Numerical solutions for \( u_1 \) of the Gierer-Meinhardt model at \( T = 340 \) by the variable-\( \theta \) method varying \( \Delta x = \Delta y \) with fixed \( \Delta t = 0.1 \). The image \( I_\ell \) represents the numerical solution obtained with the mesh resolution \( (\Delta t, \Delta x = \Delta y) = (0.1, 0.02/2^{\ell-1}) \) and \( I'_\ell \) represents the aerial view of \( I_\ell \).

Numerical solutions for Gierer-Meinhardt model (5.30) and (5.31) of the steady-state \( (T = 500) \) approximated by the standard 5-point FD scheme at fixed time-step \( \Delta t = 0.01 \) and various spatial resolutions. The image \( I_\ell \) represents the numerical solution obtained with the mesh resolution \( (\Delta t, n_x = n_y) = (0.01, 50 \cdot 2^{\ell-1}) \).

Numerical solutions for Gierer-Meinhardt model (5.30) and (5.31) of the steady-state \( (T = 500) \) approximated by the spline collocation method computed with \( r = 3 \) at fixed time-step \( \Delta t = 0.01 \) and various spatial resolutions. The image \( J_\ell \) represents the numerical solution obtained with the mesh resolution \( (\Delta t, n_x = n_y) = (0.01, 50 \cdot 2^{\ell-1}) \).

Numerical solutions for Gray-Scott model defined in Section 5.2.2.2 at \( T = 1 \) and \( (\tau, h) = (0.001, 0.1) \) approximated by (a) the CNOSC \( (r = 3) \), and (b) the averaging scheme \( (\tilde{\alpha} = 2/3) \).

Numerical solutions for Gierer-Meinhardt model (5.30) and (5.31) of the steady-state \( (T = 500) \) approximated by the heuristic-optimal averaging scheme at fixed time-step \( \tau = 0.01 \) and various spatial resolutions. The image \( I_\ell \) represents the numerical solution obtained with the mesh resolution \( (\tau, h) = (0.01, 0.04/2^{\ell-1}) \).

Numerical solutions for Gierer-Meinhardt model (5.30) and (5.31) at \( T = 80 \) and \( (\tau, n_x = n_y) = (0.01, 50) \) approximated by (a) the standard 5-point scheme, (b) the CNOSC \( (r = 4) \), and (c) the averaging scheme \( (\tilde{\alpha} = 2/3) \).
LIST OF SYMBOLS AND ABBREVIATIONS

Ω  Bounded domain in $\mathbb{R}^n$
Ω _d Set of the discrete grid points in Ω
$H^k(\Omega)$ Sobolev space with $p = 2$ on Ω
$H^k_0(\Omega)$ Space of functions in $H^k(\Omega)$ that vanish at the boundary
W Wobble set
D Diffusion coefficient
A Finite difference (FD) approximation of Laplacian operator $(-\Delta)$
$L_J (N_J)$ Jacobi iteration for (non)linear obstacle problem
$L_{GS}(N_{GS})$ Gauss-Seidel (GS) iteration for (non)linear obstacle problem
$L_{SOR}(\omega) (N_{SOR}(\omega))$ SOR iteration for (non)linear obstacle problem
ω Over-relaxation parameter for SOR, $1 < \omega < 2$
$O\mathcal{R}$ $(DO\mathcal{R})$ (Double) Obstacle relaxation algorithm
$\mathcal{R}A$ Reaction-diffusion (RD) system relaxation algorithm
idxt(·, ·, ·) Indicator function for local extrema
iswb(·, ·, ·) Indicator function for wobble point
tridiag(·, ·, ·) Block tridiagonal matrix
CN Crank-Nicolson
SOR Successive over-relaxation
ADI Alternating direction implicit
PDL1P Fast primal-dual method in [68]
SINT Semi-iterative Newton-type method in [4]
CNOSC ADI extrapolated CN orthogonal spline collocation method in [17]
CHAPTER 1

INTRODUCTION

1.1 Elliptic Obstacle Problem

Variational inequalities initially developed to deal with equilibrium problems and have been studied as one of the key issues in the calculation of variations and applied sciences. The classic motivating example of such inequalities is the so-called (elliptic) obstacle problem, in which a minimization problem is often solved. In other word, the obstacle problem is to find the equilibrium position of an elastic membrane whose boundary is held fixed, with an added constraint that the membrane lies above a given obstacle in the interior of the domain.

In modern computational mathematics and engineering, the obstacle problems are not extremely difficult to solve numerically any more, as shown in numerous publications; see [1, 3, 4, 20, 23, 56, 68], for example. However, most of those known methods are either computationally expensive or yet to be improved for higher accuracy and efficiency of the numerical solution. In this dissertation, we consider accuracy-efficiency issues and their remedies for the numerical solution of elliptic obstacle problems. This dissertation makes following contributions for the elliptic obstacle problem, refer to [33]:

- *Accuracy improvement through subgrid finite differencing of the free boundary*: It can be verified either numerically or theoretically that the numerical solution easily involve a large error near the free boundary (the edges of obstacles), particularly
when the grid mesh does not match with the obstacle edges. We suggest a post-processing algorithm which can reduce the error (by about a digit) by detecting accurate free boundary in subgrid level and introducing nonuniform finite difference (FD) method. The main goal of the subgrid FD algorithm is to produce a numerical solution of a higher accuracy \( u_h \) which guarantees \( u_h(x) \geq \varphi(x) \) for all points \( x \in \Omega \), where \( \varphi \) is an obstacle.

• **Obstacle SOR**: The iterative algorithm for solving the linear system of the obstacle problem is implemented based on one of simplest iterative algorithms, the successive over-relaxation (SOR) method. Convergence of the obstacle SOR method is analyzed and compared with modern sophisticated methods. We also suggest an effective way to set the optimal relaxation parameter \( \omega \). Our simple obstacle SOR method with the optimal parameter performs better than state-of-the-art methods in both accuracy and efficiency.

• **Effective numerical methods for nonlinear problems**: For the nonlinear obstacle problem, a method of gradient-weighting is introduced to solve the problem more conveniently and efficiently. In particular, the suggested numerical schemes for the gradient-weighting problem produce an algebraic system of a symmetric and diagonally-dominant \( M \)-matrix of which the main diagonal entries are all the same positive constant. Thus the resulting system is easy to implement and presumably converges fast; as one can see from Section 3.3, the obstacle SOR algorithm for nonlinear problems converges in a similar number of iterations as for linear problems.

1.2 Parabolic Obstacle Problem

We can easily extend the elliptic obstacle problem above to parabolic obstacle problem which is regraded as an evolutionary problem of the elliptic problem. In particular, parabolic obstacle problem with nonsmooth initial or boundary conditions has important applications for the elastic-plastic torsion problems [63], the reaction-diffusion systems [25], and the American option pricing [60, 61]. Brugnano and Alessandra [4] considered parabolic obstacle problems by means of piecewise linear systems to solve the resulting problem using a Newton-type iteration. Yang et al. [64] employed the piecewise linear finite element discretization in space and the implicit method to find the numerical solution of a parabolic obstacle problem with a nonsmooth initial data.
Through the history of numerical methods, various time-stepping procedures have been developed for the numerical solution of evolutionary PDEs in applied sciences and industrial fields, e.g., constrained heating, optimal control, elasto-plasticity, and financial mathematics. The most common time-stepping methods for solving diffusion-like PDEs are the explicit method, the implicit method, and the Crank-Nicolson (CN) method. Among these, the CN method is unconditionally stable and of second-order accuracy in both spatial and temporal directions. However, the method applied for nonsmooth data may introduce spurious oscillations to the numerical solution unless the algorithm parameters satisfy the maximum principle, which has been recognized as early as 1947 by Crank and Nicolson themselves [11]. More than 30 years later, Lawson and Morris [30] found that these oscillations would not bring out disastrous consequences provided that high-frequency components decayed to zero faster than low-frequency components, which could be achieved by selecting smaller time steps or lower spatial resolutions satisfying a constraint, the maximum principle. For this reason, whenever a larger time step or a higher spatial resolution is desirable/necessary, the less-accurate implicit method has been used at least for several initial time steps for nonsmooth initial data; which is both unconditionally stable and immune to oscillations [30]. The implicit method, however, is not an advantageous remedy for the CN method since its temporal accuracy is first-order and the numerical error may become large unless the time step size is considerably small.

To overcome the oscillation problem of the CN method and restore the desired order of accuracy, numerous numerical methods have been suggested in the literature. For example, Pearson [42] examined analytically and experimentally the error of time-stepping methods
for diffusion problems with impulsively-changed boundary conditions. Then, Lawson and Morris [30] studied conditions for the decay of oscillations and introduced an extrapolation method based on the implicit method. In [19], Gourlay and Morris generalized the algorithms of Lawson and Morris [30] for higher orders of accuracy. Rannacher [46] discovered for higher order diagonal Padé schemes, which significantly enhanced the performance of the Padé schemes with using subdiagonal schemes at the initial step. The use of a few substeps of low-order smoothing schemes (e.g., the implicit method) for nonsmooth data has taken prominence in a wide range of applications [26, 53, 60]. Here, we newly introduce a novel algorithm for the parabolic problems, refer to [32]:

- **Variable-θ method**: In order to reduce oscillations in the second-order numerical solution of parabolic problems of nonsmooth data, we suggest a novel variable-θ method in which the time-stepping parameter θ is determined based on local oscillatory characteristics of the solution and the data. Recall that nonsmooth initial or boundary data can introduce oscillations into the evolution of the CN numerical solution. To overcome the drawback, we first detect nonsmooth or discontinuous grid points and then set θ = 1 there (the implicit method), while keeping θ = 1/2 for all other grid points (the CN method). Such a strategy can suppress possible spurious oscillations originated from nonsmooth data and result in numerical solutions of second-order accuracy at most grid points. The new method can restore numerical solutions of near second-order accuracy; details are presented in Section 4.1.

- **Maximum principle**: In the analytical point of view, the CN method applied for nonsmooth data may not satisfy the maximum principle and it causes spurious oscillations numerically so that it requires to choose a small time step size to satisfy the principle. The variable-θ method, however, is designed to satisfy the maximum principle unconditionally and it allows us to choose the time step without limitations. Various examples have been considered to show the maximum principle of our new method and it is proved mathematically as well.

A parabolic obstacle problem is a typical example for parabolic problems of nonsmooth data, when the obstacle shows edges and nonsmooth surface regions. In this dissertation, the algorithm for elliptic obstacle problems studied in [33] is generalized for parabolic ob-
stacle problems. However, its main development is the aforementioned variable-θ method, which is new in the literature of numerical analysis.

For a variant of the CN method for the simulation of convection-dominated flows, see [31] where the convection term in the explicit half step is approximated by the second-order essentially non-oscillatory (ENO) scheme [39, 40].

1.3 Reaction-Diffusion Problem in Biology

As molecular imaging and single cell analysis is advancing our understanding of spatial processes shaping the cellular dynamics, new models of nonlinear dynamics are necessary. Originating in study of organism development, spatial pattern formation has received a large amount of research over the past decade. Among the most studied, the reaction-diffusion (RD) systems are generating patterns that have been shown to represent well morphogenesis. A theory of morphogenesis based on a RD model was initially proposed by Turing [57]. Gierer and Meinhardt [18] were the first to explore pattern formation in biological systems using the RD model, which often contains nonlinear reaction terms.

After Turing proposed a theory of morphogenesis in chemical cells in 1952 [57], much attention has been devoted to the numerical solution of RD problems; see [8, 16, 17, 51, 52] and references therein. Most of the numerical methods studied employ finite difference or finite element approximations for the spatial discretization, while some researchers use finite volume and collocation methods. Once the nonlinear reaction terms are treated (linearized or extrapolated), the CN method can be applied as a second-order time-stepping procedure. Time-stepping procedures require at each time step to solve a system of linear
algebraic equations, which, although sparse, is compute-intensive for multi-dimensional problems. In order to enhance efficiency of time-stepping procedures, one can adopt the alternating direction implicit (ADI) method as in [8, 16, 17, 51]. In particular, [17] introduces an ADI extrapolated CN orthogonal spline collocation method for RD problems.

ADI was invented as a perturbation of the CN method by Douglas, Peaceman, and Rachford in 1955 [12, 15, 41] and has been employed effectively for the calculation of numerical solution of various time-dependent multi-dimensional problems, either parabolic or hyperbolic [14, 36]. ADI reduces a multi-dimensional problem to multiple easy-to-solve one-dimensional problems, for an extra cost of a splitting error in $O(\Delta t^2)$, where $\Delta t$ is the time step size. However, the splitting error can be much larger than the sum of spatial and temporal discretization errors, unless the time step size is sufficiently small [13].

On the other hand, the CN method applied for nonsmooth data may introduce spurious oscillations to the numerical solution unless the time step size is sufficiently small to satisfy the maximum principle, which has been recognized in the original paper as well [11].

For this reason, the CN method and its perturbations (such as ADI) must be applied with care when the solution involves fast transitions or sharp edges; in particular, the time step size should be set very small, e.g., $\Delta t = O(\Delta x^2)$, where $\Delta x$ is the spatial grid size. In order to overcome the oscillation problem of the CN method associated with nonsmooth data, we apply a variable-$\theta$ method to RD problems which is suggested for the parabolic problem above. Precisely, this dissertation makes following contributions for the RD problem, refer to [34]:

- **Application of Variable-$\theta$ method**: We apply the variable-$\theta$ method for the numerical solution of two-component nonlinear RD equations. The variable-$\theta$ method is a per-
turbation of the CN method which evolves the solution implicitly at points where the solution shows a certain portent of oscillations, and maintains as a similar accuracy as the CN method with smooth data. The proposed method would be an adequate choice of time-stepping procedure for the numerical solution of RD partial differential equations (PDEs) when a larger time step or a higher spatial mesh resolution is desirable.

- **Sensitivity Analysis**: We have performed a sensitivity analysis for the numerical solution of biological pattern forming models, to the spatial and temporal grid sizes. It has been observed from various examples that accuracy of the numerical solution is much more sensitive to the spatial mesh resolution than the temporal one. When the spatial mesh resolution is set high for a higher accuracy, the method allows to keep the temporal resolution moderate or low. The suggested variable-\(\theta\) method can results in a smooth/stable numerical solution by suppressing possible oscillations, unlike the CN method.

- **Averaging Compact Scheme**: As a possible solution for the sensitivity issue, this dissertation suggests the averaging scheme, as an interpolation of the standard 5-point scheme and the skewed 5-point scheme, to enhance the spatial approximation of RD equations. Together with combining the averaging scheme with the variable-\(\theta\) method, the simple optimizing strategy is also introduced which minimizes the leading truncation error of the scheme. We could elaborate the averaging scheme to simply add the inexpensive optimizing step but it is still necessary to improve the performance of the optimizing.

### 1.4 Outline of Dissertation

This dissertation is organized as follows. As preliminaries, Chapter 2 includes brief review on the *elliptic obstacle problem* and accuracy deterioration of the numerical solution (underestimation) is discussed by exemplifying an obstacle problem in 1D where the mesh grid does not match with edges of the free boundary. Also, the *parabolic obstacle problem* and the \(\theta\)-method for solving time-dependent problems are reviewed, and spurious oscillatory behaviors of the CN method is discussed by exemplifying a heat equation in 1D. We add the definition of the RD problem in this chapter, as our last concern.
In Chapter 3, we present the main results on the elliptic obstacle problem. First, the successive-over-relaxation (SOR) is applied for both linear and nonlinear problems and analyzed for convergence; the limits of iterates are proved to satisfy discrete obstacle problems. A method of gradient-weighting and second-order finite difference (FD) schemes are introduced for nonlinear problems. An effective strategy is suggested to find the optimal relaxation parameter for the SOR. Finally, we introduce subgrid FD schemes near the free boundary in order to reduce accuracy deterioration of the numerical solution. Various numerical examples are included to verify the claims we just made.

Chapter 4 gives the main results on the parabolic obstacle problem. First, we describe in detail the variable-θ method for the heat equation together with introducing the effective strategy detecting nonsmoothness or discontinuities, the wobble set. Then, we apply the new method for the parabolic obstacle problem; this chapter also introduces an effective obstacle relaxation algorithm for the numerical solution of the parabolic obstacle problem. The last part of chapter gives various numerical examples which reveal effectiveness of our new algorithm and an option pricing application of our method is added.

In Chapter 5, we present the results on the RD problem. A variable-θ method is presented for the numerical solution of two-component nonlinear RD equations. This chapter also considers a heuristic technique for the choice of the optimal relaxation parameter for SOR. Numerical section gives numerical examples that show effectiveness of the variable-θ method applied to RD problems in 1D and 2D spaces. In numerical experiments, we performed the sensitivity analysis and it has been observed from various examples that accuracy of the numerical solution is much more sensitive to the spatial mesh resolution.
than the temporal one. Also, as a possible solution for the sensitivity issue, we briefly introduced an averaging scheme for accurate spatial approximation as an interpolation of two different 5-point schemes. Finally, we discuss the combination between the averaging scheme and the variable-\(\theta\) method and suggest a simple optimizing strategy to minimize the leading truncation error.
CHAPTER 2
PRELIMINARY

2.1 Obstacle Problem

In this section, we review the obstacle problems which would be mainly dealt in this dissertation and examine an accuracy issue related to the free boundary.

2.1.1 Elliptic and obstacle problem

The obstacle problem is, for example, to find the equilibrium position $u$ of an elastic membrane whose boundary is held fixed, with an added constraint that the membrane lies above a given obstacle $\varphi$ in the interior of the domain $\Omega \subset \mathbb{R}^d$:

$$
\min_{u \in K} \left\{ \int_{\Omega} \sqrt{1 + |\nabla u|^2} \, dx - \int_{\Omega} f u \, dx \right\},
$$

(2.1)

$$
K = \{ u \in H^1(\Omega) : u \geq \varphi \text{ a.e. in } \Omega, \ u = g \text{ on } \Gamma \},
$$

where $\Gamma = \partial \Omega$ denotes the boundary of $\Omega$, $f$ is an external force, and $g$ is the fixed value of $u$ on the boundary. The problem is deeply related to the study of minimal surfaces and the capacity of a set in potential theory as well. Other classical applications of the obstacle problem include the study of fluid filtration in porous media, constrained heating, elasto-plasticity, optimal control, financial mathematics, and surface reconstruction [2, 5, 6, 27, 28, 43, 47].
The problem in (2.1) can be linearized in the case of small perturbations by expanding the energy functional in terms of its Taylor series and taking the first term, in which case the energy to be minimized is the standard Dirichlet energy:

$$
\min_{u \in K} \left\{ \int_{\Omega} |\nabla u|^2 \, dx - \int_{\Omega} f u \, dx \right\},
$$

where

$$
K = \{ u \in H^1(\Omega) : u \geq \varphi \text{ a.e. in } \Omega, \ u = g \text{ on } \Gamma \}.
$$

A variational argument [5] shows that, away from the contact set $\{ x| u(x) = \varphi(x) \}$, the solution to the obstacle problem (2.2) is harmonic. A similar argument (which restricts itself to variations that are positive) shows that the solution is superharmonic on the contact set. Thus both arguments imply that the solution is a superharmonic function. As a matter of fact, it follows from an application of the maximum principle that the solution to the obstacle problem (2.2) is the least superharmonic function in the set of admissible functions. The Euler-Lagrange equation for (2.2) reads

$$
\begin{aligned}
-\Delta u &\geq f, \\
u &\geq \varphi, \\
(-\Delta u - f) \cdot (u - \varphi) &= 0,
\end{aligned}
$$

in $\Omega$,

$$
\begin{aligned}
u &= g,
\end{aligned}
$$
on $\Gamma$.

2.1.2 Accuracy issue for obstacle problems

Now, we point out certain accuracy issues related to the free boundary for obstacle problems. The solution of obstacle problems must lie on or over the obstacle ($u \geq \varphi$), which is also one of requirements for numerical solutions. For finite difference (FD) methods and finite element (FE) methods for the obstacle problem (2.3), for example, this requirement can be easily violated when edges of the free boundary does not match with
A non-matching grid: The true solution $u$ (red solid curve) and the numerical solution on the non-matching grid $u_h$ (blue dashed curve). Here the obstacle is the shaded region, $u(x_0) = u_h(x_0) = \varphi(x_0) = 0$, and $u(x_5) = u_h(x_5) = \varphi(x_5) = 1$.

mesh grids. See Figure 2.1, where the shaded rectangle indicates the obstacle defined on one-dimensional (1D) interval $[x_0, x_5]$:

$$\varphi(x) = \begin{cases} 0 & \text{if } x_0 \leq x < p, \\ 1 & \text{if } p \leq x \leq x_5, \end{cases}$$

(2.4)

which is not matching with the mesh grids $\{x_i : x_i = i \cdot h_x, \ i = 0 \cdots 5\}$. The figure shows the true solution $u$ (red solid curve) and a numerical solution $u_h$ (blue dashed curve) of the linear obstacle problem (2.3) in 1D. The numerical solution is clearly underestimated and the magnitude of the error $|u_h - u|$ is maximized at $x = p$:

$$\max_x |u_h(x) - u(x)| = |u_h(p) - u(p)| = \frac{x_3 - p}{x_3 - x_0},$$

(2.5)

which is $O(h_x)$.

Let $C_h$ denote the numerical contact set:

$$C_h \equiv \{x \in \Omega^0_d : u_h(x) = \varphi(x)\},$$

(2.6)

where $\Omega^0_d$ is the set of interior grid points. Define an interior grid point is a neighboring point if it is not in the contact set but one of its adjacent grid points is in the contact set. Let the set of neighboring points be called the neighboring set $N_h$. Then, or the example in Figure 2.1, $C_h = \{x_3, x_4\}$ and $N_h = \{x_2\}$.
The accuracy of the numerical solution $u_h$ can be improved by applying a post processing in which a subgrid FD method is applied at grid points in the neighboring set. For example, at $x = x_2$, $-u_{xx}$ can be approximated by employing nonuniform FD schemes over the grid points $[x_1, x_2, p]$, given as

$$-u_{xx}(x_2) \approx \frac{2}{h_x^2} \left( -\frac{u_1}{1 + r} + \frac{u_2}{r} - \frac{\varphi(p)}{r(1 + r)} \right), \quad (2.7)$$

where $r = (p - x_2)/h_x \in (0, 1]$, and therefore numerical solution of $-u_{xx} = 0$ at $x = x_2$ must satisfy

$$u_2 = \frac{ru_1 + \varphi(p)}{1 + r}. \quad (2.8)$$

As $r$ is approaching 0 (i.e., $(p - x_2)$ becomes smaller proportionally), the obstacle value $\varphi(p)$ is more weighted. On the other hand, when $r = 1$, $\varphi(p) = u_3$ and the scheme in (2.7) becomes the standard second-order FD scheme. Let the numerical solution $\tilde{u}$ be obtained from

$$\varphi(x_j) \leq \tilde{u}_j = \begin{cases} 
(r \tilde{u}_{j-1} + \varphi(p))/(1 + r) & \text{if } j = 2, \\
(\tilde{u}_{j-1} + \tilde{u}_{j+1})/2 & \text{if } j = 1, 3, 4,
\end{cases} \quad (2.9)$$

where $\tilde{u}_0 = 0$ and $\tilde{u}_5 = 1$. Then it is not difficult to prove that $\tilde{u}$ is exactly the same as the true solution $u$ at all grid points (except numerical rounding error), regardless of the grid size $h_x$.

The above example has motivated the authors to develop an effective numerical algorithm for elliptic obstacle problems in 2D which detects the neighboring set of the free boundary, determines the subgrid proportions ($r$'s), and updates the solution for an improved accuracy using subgrid FD schemes. Here the main goal is to try to guarantee $u(x) \geq \varphi(x)$ for all $x \in \Omega$ (whether $x$ is a grid point or not). Since it is often the case
that the free boundary is determined only after solving the problem, the algorithm must be a post-process. Details are presented in Section 3.2.

2.1.3 Parabolic obstacle problem

We can easily extend the elliptic obstacle problem (2.3) above to parabolic case. Let \( J = (0, T] \) for some \( T > 0 \). The following is the standard parabolic equation (heat equation):

\[
\begin{align*}
\partial_t u - \Delta u &= f, \quad \Omega \times J, \\
\quad u &= g, \quad \Gamma \times J, \\
\quad u &= u_0, \quad \Omega \times \{t = 0\},
\end{align*}
\]

(2.10)

where \( u_0 \) is the prescribed initial value of the solution at \( t = 0 \). As the parabolic extension of the elliptic obstacle problem, we can modify the problem (2.10) in same manner of (2.3) as follows:

\[
\begin{align*}
\partial_t u - \Delta u &\geq f, \\
\quad u &\geq \varphi, \\
(\partial_t u - \Delta u - f) \:(u - \varphi) &= 0, \\
\quad u &= g, \quad \Gamma \times J, \\
\quad u &= u_0 = \varphi, \quad \Omega \times \{t = 0\},
\end{align*}
\]

(2.11)

where \( \varphi \) represents the obstacle.

Through the history of numerical methods, various time-stepping procedures have been developed for the numerical solution of the evolutionary partial differential equations (PDEs) above. The most common time-stepping methods for solving diffusion-like PDEs are the \( \theta \)-method, which will be detailed in the next section.
2.2 Time-Stepping Method

This section presents a brief review of time-stepping procedures for the numerical solution of the heat equation (2.10) in terms of the \(\theta\)-method, \(0 \leq \theta \leq 1\). In particular, we will focus on oscillation issues associated with the Crank-Nicolson (CN) method (\(\theta = 1/2\)).

2.2.1 The \(\theta\)-method

We begin with the finite difference (FD) schemes for heat equation:

\[
\frac{\partial u}{\partial t} - \Delta u = f.
\] (2.12)

For convenience, we consider a rectangular domain in \(\mathbb{R}^2\): \(\Omega = (a_x, b_x) \times (a_y, b_y)\).

By partitioning the space-time domain \(\Omega \times J\), we obtain the space-time grid points:

\[
(x_{ij}, t^n) := (x_i, y_j, t^n);
\] (2.13)

where \(n_x, n_y, n_t\) are positive integers and

\[
x_i = a_x + i \cdot \Delta x, \quad y_j = a_y + j \cdot \Delta y, \quad t^n = n \cdot \Delta t;
\]

\[
\Delta x = \frac{b_x - a_x}{n_x}, \quad \Delta y = \frac{b_y - a_y}{n_y}, \quad \Delta t = \frac{T}{n_t}.
\]

Define the discrete domain, the collection of the grid points, by

\[
\Omega_d = \{(x_i, y_j) : 0 \leq i \leq n_x, \ 0 \leq j \leq n_y\},
\] (2.14)

and denote \(\Gamma_d = \Omega_d \cap \Gamma\) by the set of boundary grid points and \(\Omega_d^0 = \Omega_d \setminus \Gamma_d\) by the set of interior grid points. Let \(g^n_{ij} = g(x_{ij}, t^n)\) for functions \(g\) defined in \((x, t)\). Then the second-order 5-point finite difference (FD) approximation \(\mathcal{A}\) of \(-\Delta\) reads

\[
\mathcal{A} u^n_{ij} := -(\mathcal{A}_x + \mathcal{A}_y) u^n_{ij},
\] (2.15)
where the FD operators are defined as

\begin{align*}
A_x u_{ij}^n &= \frac{u_{i-1,j}^n - 2u_{ij}^n + u_{i+1,j}^n}{\Delta x^2}, \quad A_y u_{ij}^n = \frac{u_{ij-1}^n - 2u_{ij}^n + u_{ij+1}^n}{\Delta y^2}.
\end{align*}

(2.16)

For the temporal derivative \( \partial_t u \), a convenient FD approximation can give

\[ \frac{u_{ij}^n - u_{ij}^{n-1}}{\Delta t}. \]

(2.17)

Expressing the spatial derivative by a weighted average \( \theta \in [0, 1] \) of previous and current time values, we can formulate the \( \theta \)-method for (2.12) as

\[ \frac{u_{ij}^n - u_{ij}^{n-1}}{\Delta t} + A[\theta u_{ij}^n + (1 - \theta) u_{ij}^{n-1}] = f_{ij}^{n-1+\theta}, \]

(2.18)

where \( f_{ij}^{n-1+\theta} \) is either \( f(x_{ij}, t^{n-1+\theta}) \) or \( \theta f_{ij}^n + (1 - \theta) f_{ij}^{n-1} \). A simple algebraic rearrangement of (2.18) in a vector form becomes

\[ (I + \theta \Delta t A) u^n = [I - (1 - \theta)\Delta t A] u^{n-1} + \Delta t f^{n-1+\theta}, \]

(2.19)

where \( u^n = [u_{ij}^n]_{0 \leq i \leq nx, 0 \leq j \leq ny} \) and \( f^{n-1+\theta} = [f_{ij}^{n-1+\theta}]_{0 \leq i \leq nx, 0 \leq j \leq ny} \), considered as column vectors. Popular choices of \( \theta \in [0, 1] \) are 0, 1, and 1/2, which are respectively the explicit method (the forward Euler method), the implicit method (the backward Euler method), and the semi-implicit method (the CN method).

When \( \theta = 0 \), the algorithm (2.19) is convergent when it satisfies the following stability condition

\[ \mu_x + \mu_y \leq \frac{1}{2}, \quad \mu_x = \frac{\Delta t}{\Delta x^2}, \quad \mu_y = \frac{\Delta t}{\Delta y^2}. \]

(2.20)

Although the explicit method is efficient for each time step, its stability condition enforces to choosing a small time-step size \( \Delta t \) so as possibly to become less efficient compared with
other implicit methods, particularly the CN method. It is elementary in numerical analysis that when $\theta \geq 1/2$, the $\theta$-method (2.18) is unconditionally stable and convergent. For a convenient use, here we write the CN method in a vector form:

$$\left( I + \frac{\Delta t}{2} A \right) u^n = \left( I - \frac{\Delta t}{2} A \right) u^{n-1} + \Delta t f^{n-\frac{1}{2}}.$$ (2.21)

The method has been the most popular time-stepping procedure for the numerical solution of the parabolic equation, because it is stable and of second-order accuracy in both spatial and temporal directions.

2.2.2 An example: numerical oscillations of the CN method

Although the CN method is unconditionally stable and of second-order accuracy in both spatial and temporal directions, it may produce poor numerical results when it introduces spurious oscillations into the numerical solution. In [30], Lawson and Morris pointed out that numerical oscillations would happen provided that high-frequency components of the numerical solution decay to zero slower than low-frequency components. Since jump discontinuities tend to cause large high-frequency components, numerical oscillations may occur in the neighborhood of discontinuities.

In order to examine the behavior of the $\theta$-methods numerically, we consider the following one-dimensional (1D) parabolic equation of a discontinuous initial condition on $[-1, 1]$:

$$\partial_t u - \partial_{xx} u = 0, \quad (x, t) \in (-1, 1) \times [0, T],$$

$$u(x, 0) = u_0(x) = \begin{cases} 
1 & \text{if } |x| < 0.5, \\
0.5 & \text{if } |x| = 0.5, \\
0 & \text{if } |x| > 0.5.
\end{cases}$$ (2.22)
The Dirichlet boundary condition is set to satisfy the analytic solution given in [54]:

\[
    u(x, t) = \frac{1}{2} + 2 \sum_{k=0}^{\infty} (-1)^k \frac{\cos \pi (2k+1) x}{\pi (2k+1)} e^{-\pi^2 (2k+1)^2 t}, \quad (x, t) \in [-1, 1] \times [0, T]. \tag{2.23}
\]

When the θ-method (2.18) is applied to (2.22), the stability condition reads

\[
    (1 - 2\theta) \mu_x \leq \frac{1}{2}, \tag{2.24}
\]

and therefore both the implicit method and the CN method are stable.

Figure 2.2

The evolution of (a) analytic solution and numerical solution by (b) the implicit method and (c) the CN method for (2.22), for \(0 \leq t \leq T = 0.5\), when \(\Delta x = 0.04\) \((n_x = 50)\) and \(\Delta t = 0.02\) \((n_t = 25)\).

Figure 2.2 exhibits the evolution of the analytic solution and numerical solutions obtained by the implicit method and the CN method, for \(0 \leq t \leq T = 0.5\), \(\Delta t = 0.02\) \((n_t = 25)\), and \(\Delta x = 0.04\) \((n_x = 50)\). It is clear to see that the CN method introduces numerical oscillations in the vicinity of discontinuities \(|x| = 0.5\). It is well known that
the CN method may produce oscillatory numerical solutions for nonsmooth data when it violates the maximum principle:

\[(1 - \theta) \mu_x \leq \frac{1}{2}. \quad (2.25)\]

The implicit method \((\theta = 1)\) satisfies the above inequality for all choices of \(\Delta t\), while the CN method \((\theta = 1/2)\) needs to choose \(\Delta t \leq \Delta x^2 = 0.0016\); for the example in Figure 2.2, the temporal grid must be refined more than 10 times for the CN method not to invoke spurious oscillations.

Here the main point is that although the CN method is stable and of second-order accuracy in both spatial and temporal directions, it must be carefully applied for parabolic problems of nonsmooth data in order not to introduce numerical oscillations. A simple remedy is to choose small time-step sizes \(\Delta t\), which may make the method expensive computationally. This dissertation suggests a more reasonable method called the \textit{variable-}\(\theta\) \textit{method}, which chooses the parameter \(\theta\) as a function of point and time in Section 4.1. For example, at the \(n\)-th time level, we may set \(\theta^n = 1/2\) everywhere, as default, and assign \(\theta^n_{ij} = 1\) at the points \(x_{ij}\) where the solution shows a certain portent of oscillations.

### 2.3 Reaction-Diffusion Problem

The RD system for biological pattern formation is defined as follows [10]: let \(\Omega\) be a bounded domain in \(\mathbb{R}^d\), \(d = 1, 2, 3\), \(\Gamma = \partial \Omega\) denote the boundary of \(\Omega\), and \(J = (0, T]\) for some \(T > 0\).

\[
\begin{align*}
\frac{\partial u}{\partial t} - D\Delta u &= f(u), \quad \Omega \times J, \\
\frac{\partial u}{\partial \nu} &= 0, \quad \Gamma \times J, \\
u &= u^0, \quad \Omega \times \{t = 0\},
\end{align*}
\]
where \( u = [u_1, u_2]^T \), \( D = \text{diag}[D_1, D_2]^T \) is the diffusion tensor, \( \Delta \) denotes the Laplace operator, \( \partial / \partial \nu \) is the outward normal derivative on the boundary \( \Gamma \), and \( f(u) \) is the reaction kinetics of the system given as

\[
f(u) = \begin{bmatrix} f_1(u_1, u_2) \\ f_2(u_1, u_2) \end{bmatrix}.
\]  

(2.27)

Several equations of RD type have been studied to understand patterning in developmental biology. Some were derived from phenomenological models (Gierer-Meinhardt) while other modeled simple reaction schemes (Schnackenberg trimolecular autocatalytic reactions model [50], Gray-Scott model [21], Brusselator model [44], chlorite-iodide-malonic acid – CIMA model [35]). Recent work on RD systems demonstrate that it can be used to understand biological patterns formation [29], while [37] and [55] reviewed the RD systems can be used to investigate spatial patterning in developmental systems.
CHAPTER 3

ELLIPTIC OBSTACLE PROBLEM

3.1 Obstacle Relaxation for Elliptic Obstacle Problems

This section introduces and analyzes effective relaxation methods for solving (2.3):

\[
\begin{align*}
-\Delta u & \geq f, \\
u & \geq \varphi, \\
(-\Delta u) \cdot (u - \varphi) & = 0, \\
u & = g,
\end{align*}
\]

in $\Omega$, on $\Gamma$, and its nonlinear problem as shown in (3.10) below.

, and spurious oscillatory behaviors of the CN method is discussed by exemplifying a heat equation in 1D.

and its accuracy issue related to the free boundary when edges of the free boundary does not match with mesh grids.

3.1.1 The linear obstacle problem

We will begin with second-order approximation schemes for $-\Delta u$. For simplicity, we consider a rectangular domain in $\mathbb{R}^2$, $\Omega = (a_x, b_x) \times (a_y, b_y)$. Then the second-order FD scheme can be formulated on the grid points

\[
x_{pq} := (x_p, y_q), \quad p = 0, 1, \cdots, n_x, \quad q = 0, 1, \cdots, n_y,
\]

(3.1)
where for some positive integers \( n_x \) and \( n_y \),

\[
x_p = a_x + p \cdot h_x, \quad y_q = a_y + q \cdot h_y; \quad h_x = \frac{b_x - a_x}{n_x}, \quad h_y = \frac{b_y - a_y}{n_y}.
\]

Let \( u_{pq} = u(x_p, y_q) \). Then, at each of interior points \( x_{pq} \), the five-point FD approximation of \(-\Delta u\) reads

\[
-\Delta_h u_{pq} = \frac{-u_{p-1,q} + 2u_{pq} - u_{p+1,q}}{h_x^2} + \frac{-u_{p,q-1} + 2u_{pq} - u_{p,q+1}}{h_y^2}.
\] (3.2)

Multiply both sides of (3.2) by \( h_x^2 \) to have

\[
(-\Delta_h u_{pq}) h_x^2 = (2 + 2r_{xy}^2)u_{pq} - u_{p-1,q} - u_{p+1,q} - r_{xy}^2 u_{p,q-1} - r_{xy}^2 u_{p,q+1},
\] (3.3)

where \( r_{xy} = h_x/h_y \) and \( u_{st} = f_{st} \) at boundary grid points \((x_s, y_t)\).

Now, consider the following Jacobi iteration for simplicity. Given an initialization \( u^0 \), find \( u^n \) iteratively as follows.

**Algorithm \( L_J \):**

For \( n = 1, 2, \ldots \)

For \( q = 1 : n_y - 1 \)

For \( p = 1 : n_x - 1 \)

(a) \( u_{J,pq} = \frac{1}{2 + 2r_{xy}^2} \left( u_{p-1,q}^{n-1} + u_{p+1,q}^{n-1} + r_{xy}^2 u_{p,q-1}^{n-1} + r_{xy}^2 u_{p,q+1}^{n-1} \right) \); (3.4)

(b) \( u_{pq}^n = \max(u_{J,pq}, \varphi_{pq}) \);

end

end

end

where \( u_{st}^{n-1} = f_{st} \) at boundary grid points \((x_s, y_t)\)

Note that Algorithm \( L_J \) produces a solution \( u \) of which the function value at a point is a simple average of four neighboring values, satisfying the constraint \( u \geq \varphi \).
Theorem 3.1.1 Let \( \hat{u} \) be the limit of the iterates \( u^n \) of Algorithm \( \mathcal{L}_J \). Then, \( \hat{u} \) satisfies the FD discretization of (2.3). That is,

\[
\begin{align*}
-\Delta_h \hat{u}_{pq} & \geq 0, \\
\hat{u}_{pq} & \geq \varphi_{pq}, \\
(-\Delta_h \hat{u}_{pq}) \cdot (\hat{u}_{pq} - \varphi_{pq}) & = 0, \\
\hat{u}_{st} & = f_{st},
\end{align*}
\]

(3.5)

where \( \Omega^0_d \) denotes the set of interior grid points and \( \Gamma_d \) is the set of boundary grid points.

**Proof.** It is clear to see from Algorithm \( \mathcal{L}_J \) that

\[
\hat{u}_{pq} \geq \varphi_{pq} \text{ for } (x_p, y_q) \in \Omega^0_d \text{ and } \hat{u}_{st} = f_{st} \text{ for } (x_s, y_t) \in \Gamma_d.
\]

Let \( \hat{u}_{pq} = \varphi_{pq} \) at an interior point \((x_p, y_q)\). Then it follows from (3.4.b) that

\[
\hat{u}_{J,pq} = \frac{1}{2 + 2r^2_{xy}} (\hat{u}_{p-1,q} + \hat{u}_{p+1,q} + r^2_{xy} \hat{u}_{p,q-1} + r^2_{xy} \hat{u}_{p,q+1}) \leq \varphi_{pq} = \hat{u}_{pq},
\]

(3.6)

which implies that

\[
0 \leq (2 + 2r^2_{xy}) \hat{u}_{pq} - \hat{u}_{p-1,q} - \hat{u}_{p+1,q} - r^2_{xy} \hat{u}_{p,q-1} - r^2_{xy} \hat{u}_{p,q+1} = (-\Delta_h \hat{u}_{pq}) \cdot h_x^2.
\]

(3.7)

On the other hand, let \( \hat{u}_{pq} > \varphi_{pq} \) at \((x_p, y_q)\). Then, since \( \hat{u}_{pq} = \max(\hat{u}_{J,pq}, \varphi_{pq}) \), we must have

\[
\hat{u}_{pq} = \hat{u}_{J,pq},
\]

(3.8)

which implies that \(-\Delta_h \hat{u}_{pq} = 0\). This completes the proof.

One can easily prove that the algebraic system obtained from (3.3) is irreducibly diagonally dominant and symmetric positive definite. Since its off-diagonal entries are all nonpositive, the matrix must be a Stieltjes matrix and therefore an M-matrix [58]. Thus relaxation methods of regular splitting (such as the Jacobi, the Gauss-Seidel (GS), and
the successive over-relaxation (SOR) iterations) are all convergent and their limits are the same as \( \hat{u} \) and therefore satisfy (3.5). In this dissertation, variants of Algorithm \( \mathcal{L}_J \) for the GS and the SOR would be denoted respectively by \( \mathcal{L}_{GS} \) and \( \mathcal{L}_{SOR}(\omega) \), where \( \omega \) is an over-relaxation parameter for the SOR, \( 1 < \omega < 2 \). For example, \( \mathcal{L}_{SOR}(\omega) \) is formulated as

<table>
<thead>
<tr>
<th>Algorithm ( \mathcal{L}_{S O R}(\omega) ):</th>
</tr>
</thead>
<tbody>
<tr>
<td>For ( n = 1, 2, \ldots )</td>
</tr>
<tr>
<td>For ( q = 1 : n_y - 1 )</td>
</tr>
<tr>
<td>For ( p = 1 : n_x - 1 )</td>
</tr>
<tr>
<td>(a) ( u_{GS,pq} = \frac{1}{2 + 2 r_{xy}^2} \left( u_{p-1,q}^{n-1} + u_{p+1,q}^{n-1} + r_{xy}^2 u_{p,q-1}^n + r_{xy}^2 u_{p,q+1}^{n-1} \right) );</td>
</tr>
<tr>
<td>(b) ( u_{SOR,pq} = \omega \cdot u_{GS,pq} + (1 - \omega) \cdot u_{p,q}^{n-1} ; )</td>
</tr>
<tr>
<td>(c) ( u_{pq}^n = \max(u_{SOR,pq}, \varphi_{pq}) ; )</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

where \( u_{st}^{n-1} = u_{st}^n = f_{st} \) at boundary grid points \((x_s, y_t)\)

Note that the right side of (3.9.a) involves updated values wherever available. When \( \omega = 1 \), Algorithm \( \mathcal{L}_{S O R}(\omega) \) becomes Algorithm \( \mathcal{L}_{GS} \); that is, \( \mathcal{L}_{SOR}(1) = \mathcal{L}_{GS} \).

3.1.2 The nonlinear obstacle problem

Applying the same arguments for the linear problem (2.3), the Euler-Lagrange equation for the nonlinear minimization problem (2.1) can be formulated as

\[
\begin{align*}
\mathcal{N}(u) &\geq 0, \\
u &\geq \varphi, \\
\mathcal{N}(u) \cdot (u - \varphi) &= 0, \\
u &= f, \\
\end{align*}
\]

in \( \Omega \), on \( \Gamma \),

\[
\text{(3.10)}
\]
where
\[ \mathcal{N}(u) = -\nabla \cdot \left( \frac{\nabla u}{\sqrt{1 + |\nabla u|^2}} \right). \] (3.11)

Thus the solution to the nonlinear problem (3.10) can be considered as a minimal surface satisfying the constraint given by the obstacle function \( \varphi \).

Since \( \sqrt{1 + |\nabla u|^2} \geq 1 \), the nonlinear obstacle problem (3.10) can equivalently be formulated as
\[
\begin{aligned}
\mathcal{M}(u) &\geq 0, \\
u &\geq \varphi, \\
\mathcal{M}(u) \cdot (u - \varphi) &= 0,
\end{aligned}
\] in \( \Omega \),
\[
\begin{aligned}
u &= f, &\text{on } \Gamma,
\end{aligned}
\] (3.12)

where
\[
\mathcal{M}(u) = -\sqrt{1 + |\nabla u|^2} \nabla \cdot \left( \frac{\nabla u}{\sqrt{1 + |\nabla u|^2}} \right).
\] (3.13)

Such a method of gradient-weighting will make algebraic systems simpler and better conditioned, as to be seen below. In order to introduce effective FD schemes for \( \mathcal{M}(u) \), we first rewrite \( \mathcal{M}(u) \) as
\[
\mathcal{M}(u) = -\left( \sqrt{1 + |\nabla u|^2} \right)_1 \left( \frac{u_x}{\sqrt{1 + |\nabla u|^2}} \right)_x
- \left( \sqrt{1 + |\nabla u|^2} \right)_2 \left( \frac{u_y}{\sqrt{1 + |\nabla u|^2}} \right)_y,
\] (3.14)

where both \( \left( \sqrt{1 + |\nabla u|^2} \right)_1 \) and \( \left( \sqrt{1 + |\nabla u|^2} \right)_2 \) are the same as \( \sqrt{1 + |\nabla u|^2} \); however, they will be approximated in a slightly different way. The following numerical schemes are of second-order accuracy and specifically designed for the resulting algebraic system to be simpler and better conditioned.
For the FD scheme at the \((p, q)\)-th pixel, we first compute second-order FD approximations of \(\sqrt{1 + |\nabla u|^2}\) at \(x_{p-1/2,q}(W), x_{p+1/2,q}(E), x_{p,q-1/2}(S), \) and \(x_{p,q+1/2}(N)\):

\[
d_{pq,W} = \left[ 1 + \frac{(u_{pq} - u_{p-1,q})^2}{h_x^2} + \frac{(u_{p-1,q-1} + u_{p,q+1} - u_{p-1,q} - u_{p,q-1})^2}{(16h_y^2)} \right]^{1/2},
\]

\[
d_{pq,E} = d_{p+1,q,W},
\]

\[
d_{pq,S} = \left[ 1 + \frac{(u_{pq} - u_{p,q-1})^2}{h_y^2} + \frac{(u_{p+1,q} + u_{p+1,q-1} - u_{p,q} - u_{p,q-1})^2}{(16h_x^2)} \right]^{1/2},
\]

\[
d_{pq,N} = d_{p,q+1,S}.
\]

(3.15)

Then, the directional-derivative terms at the pixel point \(x_{pq}\) can be approximated as

\[
\left( \sqrt{1 + |\nabla u|^2} \right)_{x}^1 (x_{pq}) \approx \frac{1}{h_x} \left[ \frac{1}{d_{pq,W}} u_{p-1,q} + \frac{1}{d_{pq,E}} u_{p+1,q} - \left( \frac{1}{d_{pq,W}} + \frac{1}{d_{pq,E}} \right) u_{pq} \right],
\]

\[
\left( \sqrt{1 + |\nabla u|^2} \right)_{y}^1 (x_{pq}) \approx \frac{1}{h_y} \left[ \frac{1}{d_{pq,S}} u_{p,q-1} + \frac{1}{d_{pq,N}} u_{p,q+1} - \left( \frac{1}{d_{pq,S}} + \frac{1}{d_{pq,N}} \right) u_{pq} \right].
\]

(3.16)

Now, we discretize the surface element as follows:

\[
\left( \sqrt{1 + |\nabla u|^2} \right)_{x} (x_{pq}) \approx \frac{1}{2} \left( \frac{1}{d_{pq,W}} + \frac{1}{d_{pq,E}} \right)^{-1} = \frac{2d_{pq,W}d_{pq,E}}{d_{pq,W} + d_{pq,E}},
\]

\[
\left( \sqrt{1 + |\nabla u|^2} \right)_{y} (x_{pq}) \approx \frac{1}{2} \left( \frac{1}{d_{pq,S}} + \frac{1}{d_{pq,N}} \right)^{-1} = \frac{2d_{pq,S}d_{pq,N}}{d_{pq,S} + d_{pq,N}},
\]

(3.17)

where the right-hand sides are harmonic averages of FD approximations of \(\sqrt{1 + |\nabla u|^2}\) in \(x\)- and \(y\)-coordinate directions, respectively. Then, it follows from (3.14), (3.16), and (3.17) that

\[
\mathcal{M}(u)(x_{pq}) \cdot h_x^2 \approx (2 + 2r_{xy}^2) u_{pq} - a_{pq,W} u_{p-1,q} - a_{pq,E} u_{p+1,q} - r_{xy}^2 a_{pq,S} u_{p,q-1} - r_{xy}^2 a_{pq,N} u_{p,q+1},
\]

(3.18)

where

\[
a_{pq,W} = \frac{2d_{pq,E}}{d_{pq,W} + d_{pq,E}}, \quad a_{pq,E} = \frac{2d_{pq,W}}{d_{pq,W} + d_{pq,E}}, \quad a_{pq,S} = \frac{2d_{pq,N}}{d_{pq,S} + d_{pq,N}}, \quad a_{pq,N} = \frac{2d_{pq,S}}{d_{pq,S} + d_{pq,N}}.
\]

(3.19)
Note that $a_{pq,W} + a_{pq,E} = a_{pq,S} + a_{pq,N} = 2$. As for the linear problem, it is easy to prove that the algebraic system obtained from (3.18) is an M-matrix.

Given FD schemes for $\mathcal{M}(u)$ as in (3.18), the nonlinear obstacle problem (3.12) can be solved iteratively by the Jacobi iteration.

<table>
<thead>
<tr>
<th>Algorithm $\mathcal{N}_J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $n = 1, 2, \ldots$</td>
</tr>
<tr>
<td>For $q = 1 : n_y - 1$</td>
</tr>
<tr>
<td>For $p = 1 : n_x - 1$</td>
</tr>
<tr>
<td>(a) $u_{J,pq}^{n} = \frac{1}{2 + 2r^2_{xy}} \left( a_{pq,W}^{n-1} u_{p-1,q}^{n-1} + a_{pq,E}^{n-1} u_{p+1,q}^{n-1}</td>
</tr>
</tbody>
</table><p>ight.$ |
| $\quad + r^2_{xy} a_{pq,S}^{n-1} u_{p,q-1}^{n-1} + r^2_{xy} a_{pq,N}^{n-1} u_{p,q+1}^{n-1} \right)$; |
| (b) $u_{pq}^{n} = \max(u_{J,pq}, \varphi_{pq})$; |
| end |
| end |
| end |
| where $u_{st}^{n-1} = f_{st}$ at boundary grid points $(x_s, y_t)$ |</p>

The superscript $(n - 1)$ on the coefficients $a_{pq,D}$, $D = W$, $E$, $S$, $N$, indicate that they are obtained using the last iterate $u^{n-1}$. Algorithm $\mathcal{N}_J$ produces a solution $u$ of which the function value at a point is a weighted average of four neighboring values, satisfying the constraint $u \geq \varphi$. One can prove the following corollary, using the same arguments introduced in the proof of Theorem 3.1.1.
Corollary 3.1.2 Let \( \hat{u} \) be the limit of the iterates \( u^n \) of Algorithm \( \mathcal{N}_J \). Then, \( \hat{u} \) satisfies the FD discretization of (3.12). That is,

\[
\begin{align*}
M_h(\hat{u})_{pq} & \geq 0, \\
\hat{u}_{pq} & \geq \varphi_{pq}, \\
M_h(\hat{u})_{pq} \cdot (\hat{u}_{pq} - \varphi_{pq}) & = 0, \\
\hat{u}_{st} &= f_{st}, \\
(x_p, y_q) & \in \Omega_0^d, \\
(x_s, y_t) & \in \Gamma_d,
\end{align*}
\] (3.21)

where \( M_h(\hat{u})_{pq} \) denotes the FD scheme of \( M(u)(x_{pq}) \) as defined in (3.18) with \( u = \hat{u} \).

Variants of Algorithm \( \mathcal{N}_J \) for the GS and the SOR can be formulated similarly as for the linear obstacle problem; they would be denoted respectively by \( \mathcal{N}_{GS} \) and \( \mathcal{N}_{SOR}(\omega) \). In practice, such symmetric coercive optimization problems, the SOR methods are much more efficient than the Jacobi and Gauss-Seidel methods. We will exploit \( L_{SOR}(\omega) \) and \( \mathcal{N}_{SOR}(\omega) \) for numerical comparisons with state-of-the-art methods, by setting the relaxation parameter \( \omega \) optimal.

3.1.3 The optimal relaxation parameter \( \hat{\omega} \)

Consider the standard Poisson equation with a Dirichlet boundary condition

\[
\begin{align*}
-\Delta u &= g & \text{in } \Omega, \\
u &= f & \text{on } \Gamma = \partial \Omega,
\end{align*}
\] (3.22)

for prescribed functions \( f \) and \( g \). Let \( \Omega = [0, 1]^2 \), for simplicity, and apply the second-order FD method for the second derivatives on a uniform grid: \( h = h_x = h_y = 1/(m + 1) \), for some positive integer. The its algebraic system can be written as

\[
Au = b \in \mathbb{R}^{m^2}.
\] (3.23)
Then the theoretically optimal relaxation parameter for the SOR method can be determined as [58, § 4.3]
\[
\hat{\omega} = \frac{2}{1 + \sqrt{1 - \rho(T_J)^2}},
\]  
(3.24)
where \(\rho(T_J)\) is the spectral radius of the iteration matrix of the Jacobi method \(T_J\). The iteration matrix \(T_J\) can be explicitly presented as a block tridiagonal matrix
\[
T_J = \frac{1}{4} \text{tridiag}(I_m, B_m, I_m),
\]  
(3.25)
where \(I_m\) is the \(m\)-dimensional identity matrix and
\[
B = \text{tridiag}(1, 0, 1) = \begin{bmatrix} 0 & 1 & & \cdots & \cdots & \cdots \\ 1 & 0 & 1 & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & 1 & 0 & 1 & \\ & & & 1 & 0 & \end{bmatrix} \in \mathbb{R}^{m \times m}.
\]
For such a matrix \(T_J\), it is known that
\[
\rho(T_J) = 1 - c h^2, \quad \text{for some } c > 0.
\]  
(3.26)
Thus it follows from (3.24) and (3.26) that the optimal SOR parameter corresponding to the mesh size \(h\), \(\hat{\omega}_h\), can be expressed as
\[
\hat{\omega}_h = \frac{2}{1 + \sqrt{1 - (1 - c h^2)^2}} \approx \frac{2}{1 + c_0 h},
\]  
(3.27)
where \(c_0 = \sqrt{2c}\). Hence, for general mesh size \(h\), the corresponding optimal SOR parameter \(\hat{\omega}_h\) can be found as follows.

\[
\begin{cases}
\text{(a) Determine } \hat{\omega}_{h_0} \text{ for a prescribed mesh size } h = h_0, \text{ heuristically.} \\
\text{(b) Find } c_0 \text{ by solving (3.27) for } c_0:\ \ \\
\qquad c_0 = \frac{(2/\hat{\omega}_{h_0} - 1)}{h_0}. & \quad (3.28) \\
\text{(c) Use (3.27) with the above } c_0 \text{ to determine } \hat{\omega}_h \text{ for general } h. \\
\end{cases}
\]
It is often the case that the calibration (3.28.a)-(3.28.b) can be carried out with a small problem, i.e., with $h_0$ of a very low resolution.

### 3.2 Subgrid FD Schemes for the Free Boundary: a Post-Process

This section describes subgrid FD schemes for the free boundary, focusing on the linear obstacle problem; the arguments to be presented can be applied the same way for nonlinear problems. Again, we assume for simplicity that $h = h_x = h_y$.

Let $\hat{u}$ be the numerical solution of an obstacle problem. Then it would satisfy the discrete obstacle problem (3.5), particularly $\hat{u}_{pq} \geq \varphi_{pq}$ at all (interior) grid points $x_{pq} \in \Omega^0_d$. However, when the mesh grid is not matching with the free boundary, the obstacle constraint $\hat{u} \geq \varphi$ may not be satisfied at all points $x \in \Omega$. This implies that when the mesh is not fine enough, the numerical solution can be underestimated near the free boundary, as shown in Figure 2.1 in Section 2.1.2. Note that the error introduced by non-matching grids is in $\mathcal{O}(h)$, while the numerical truncation error is in $\mathcal{O}(h^2)$ for second-order FD schemes. That is, the underestimation is in $\mathcal{O}(h)$ which can be much larger than the truncation error.

The strategy below can be considered as a post-processing algorithm designed in order to reduce the underestimation without introducing a mesh refinement. The post-processing algorithm consists of three steps: (a) finding the numerical contact set and the neighboring set, (b) subgrid determination of the free boundary, and (c) nonuniform FD schemes on the neighboring set.
3.2.1 The contact set and the neighboring set

Finding the numerical contact set is an easy task. Let $\hat{u}$ and $\varphi$ be the numerical solution and the lower obstacle, respectively. Then, for example, the characteristic set of contact points $C_h$ can be determined as follows.

$$
C_h = \hat{u} - \varphi;
$$

if $C_h(x_{pq}) > 0$, then $C_h(x_{pq}) = 1$; for all points $x_{pq}$; \hfill (3.29)

$$
C_h = 1 - C_h;
$$

As defined in Section 2.1.2, an interior grid point is a neighboring point when it is not in the contact set but one of its adjacent grid points is in the contact set. Thus the neighboring points can be found more effectively as follows. Visit each point in the contact set; if any one of its four adjacent points is not in the contact set, then the non-contacting point is a neighboring point. The set of all neighboring points is the neighboring set $N_h$.

Contact points (red solid circle) and neighboring points (blue open circle). The red dashed curve indicates a possible free boundary.

Figure 3.1
3.2.2 Subgrid determination of the free boundary

Let \( x_{pq} \) be a neighboring point with two of its adjacent points are contact points \( (C_h(p+1,q) = C_h(p,q-1) = 1) \), as in Figure 3.1. Then we may assume that the *real* free boundary passes somewhere between the contact points and the neighboring points. We will suggest an effective strategy for the determination of the free boundary in subgrid level.

We first focus on the horizontal line segment connecting \( x_{pq} \) and \( x_{p+1,q} \) in the east (E) direction. Define

\[
x_E(r) = (1 - r)x_{pq} + rx_{p+1,q}, \quad r \in [0, 1].
\]  

(3.30)

Then the corresponding linear interpolation between \( u_{pq} \) and \( u_{p+1,q} \) over the line segment is formulated as

\[
L_E(r) = (1 - r)u_{pq} + ru_{p+1,q}, \quad r \in [0, 1].
\]  

(3.31)

Let

\[
F_E(r) = \varphi(x_E(r)) - L_E(r), \quad r \in [0, 1].
\]  

(3.32)

Since \( x_{pq} \) and \( x_{p+1,q} \) are a neighboring point and a contact point, respectively, we have

\[
F_E(0) < 0 \quad \text{and} \quad F_E(1) = 0.
\]  

(3.33)

If the free boundary passes between \( x_{pq} \) and \( x_{p+1,q} \), then there must exist \( r \in (0, 1) \) such that \( F_E(r) > 0 \). Let \( r_E \) be such that \( x_E(r_E) \) represents the intersection between the line segment \( x_E(\cdot) \) and the free boundary. Then it can be approximated as follows.

\[
r_E = \max_{r \in [0,1]} F_E(r).
\]  

(3.34)
The maximization problem in (3.34) can be solved easily (using the Newton method, for example), when the obstacle is defined as a smooth function. A more robust method can be formulated as a combination of a line search algorithm and the bisection method.

\[
\begin{align*}
\text{set } k_0, k_1; \\
 r_E = 1; \quad F_{\text{max}} = 0; \\
\text{for } k = 1 : k_0 - 1 \quad \% \text{line search} \\
  & \quad \text{if } F_E(k/k_0) > F_{\text{max}} \\
  & \quad \quad r_E = k/k_0; \quad F_{\text{max}} = F_E(k/k_0); \\
  & \quad \text{end} \\
\text{end} \\
\text{if } r_E < 1 \quad \% \text{refine it through bisection} \\
 & \quad r_b = 1/k_0; \\
\text{for } k = 1 : k_1 \\
  & \quad r_b = r_b/2; \\
  & \quad \text{if } F_E(r_E - r_b) > F_{\text{max}} \\
  & \quad \quad r_E = r_E - r_b; \quad F_{\text{max}} = F_E(r_E - r_b); \\
  & \quad \text{end} \\
  & \quad \text{if } F_E(r_E + r_b) > F_{\text{max}} \\
  & \quad \quad r_E = r_E + r_b; \quad F_{\text{max}} = F_E(r_E + r_b); \\
  & \quad \text{end} \\
\text{end} \\
B_E = \varphi(x_E(r_E)); \\
\end{align*}
\]

(3.35)

\textbf{Remark 3.2.1}  
\begin{itemize}
  \item The last evaluation of \( \varphi \) (and saving) is necessary for the nonuniform FD schemes on the neighboring set, which will be discussed in Section 3.2.3. The quantity \( B_E \) will be used as the Dirichlet value on the free boundary.
  \item For other directions \( D \) (=\( W \), \( S \), or \( N \)), one can define corresponding difference functions \( F_D \) as shown in (3.30)–(3.32) for \( D = E \). Then, \( r_D \) can be obtained by applying (3.35) with \( F_E \) being replaced with \( F_D \). When the adjacent point \( x_D(1) \) is not a contact point, you may simply set \( r_D = 1 \). Thus each neighboring point produces an array of four values \([r_W, r_E, r_S, r_N]\) and free boundary values for the directions \( D \) where \( r_D < 1 \).
\end{itemize}

33
Assuming that (3.34) has a unique solution and the obstacle is given as a smooth function, the maximum error for the detection of the free boundary using (3.35) is

\[
\left( \frac{1}{k_0} \cdot \frac{1}{2k_1} \right) h,
\]

(3.36)

where \( h \) is mesh size. It has been numerically verified that the choice \((k_0, k_1) = (10, 4)\) is enough for an accurate detection of the free boundary, for which the upper bound of the error becomes \( h/160 = 0.00625 h \).

3.2.3 Nonuniform FD schemes on the neighboring set

Let \( x_{pq} = (x_p, y_q) \) be a neighboring point. Then, \( r_{pq,D} \in (0, 1] \) would be available for each \( D \in \{W, E, S, N\} \); \( B_{pq,D} \) is also available for \( r_D < 1 \). Thus the FD scheme for \(-u_{xx}(x_{pq})\) can be formulated over three points \( \{(x_p - r_{pq,W} h_x, y_q), (x_p, y_q), (x_p + r_{pq,E} h_x, y_q)\} \) as follows.

\[
-u_{xx}(x_{pq}) \approx \frac{2}{h_x^2} \left( -\frac{u_{pq,W}}{r_{pq,W}(r_{pq,W} + r_{pq,E})} + \frac{u_{pq}}{r_{pq,W} \cdot r_{pq,E}} - \frac{u_{pq,E}}{r_{pq,E}(r_{pq,W} + r_{pq,E})} \right),
\]

(3.37)

where

\[
u_{pq,W} = \begin{cases} u_{p-1,q} & \text{if } r_{pq,W} = 1, \\ B_{pq,W} & \text{if } r_{pq,W} < 1, \end{cases} \quad u_{pq,E} = \begin{cases} u_{p+1,q} & \text{if } r_{pq,E} = 1, \\ B_{pq,E} & \text{if } r_{pq,E} < 1. \end{cases}
\]

Similarly, the FD scheme for \(-u_{yy}(x_{pq})\) can be formulated over three points in the \( y \)-direction \( \{(x_p, y_q - r_{pq,S} h_y), (x_p, y_q), (x_p, y_q + r_{pq,N} h_y)\} \):

\[
-u_{yy}(x_{pq}) \approx \frac{2}{h_y^2} \left( -\frac{u_{pq,S}}{r_{pq,S}(r_{pq,S} + r_{pq,N})} + \frac{u_{pq}}{r_{pq,S} \cdot r_{pq,N}} - \frac{u_{pq,N}}{r_{pq,N}(r_{pq,S} + r_{pq,N})} \right),
\]

(3.38)

where

\[
u_{pq,S} = \begin{cases} u_{p,q-1} & \text{if } r_{pq,S} = 1, \\ B_{pq,S} & \text{if } r_{pq,S} < 1, \end{cases} \quad u_{pq,N} = \begin{cases} u_{p,q+1} & \text{if } r_{pq,N} = 1, \\ B_{pq,N} & \text{if } r_{pq,N} < 1. \end{cases}
\]

Thus, the post-processing algorithm of the obstacle SOR (3.9), \( L_{SOR}(\omega) \), can be formulated by replacing the two terms in the right side of (3.2) with the right sides of (3.37) and (3.38), and computing \( u_{GS,pq} \) in (3.9.a) correspondingly at all neighboring points.
3.3 Numerical Experiments

In this section, we apply the obstacle SOR method and the post-processing schemes to various obstacles to verify their effectiveness and accuracy. We mainly concern 2-D obstacle problems of Dirichlet boundary conditions. The algorithms are implemented, for both one and double obstacles, in Matlab and carried out on a Desktop computer of an Intel i5-3450S 2.80 GHz processor. The optimal relaxation parameter is calibrated with the lowest resolution to find a constant $c_0$ (3.28) and the constant is used for all other cases. For a comparison purpose, we implemented a state-of-the-art method, PDL1P [68], and its parameters are found heuristically for cases where the parameters are not suggested in [68].

The iterations are stopped when the maximum difference of consecutive iterates becomes smaller than the tolerance $\varepsilon$:

$$\|u^n - u^{n-1}\|_{\infty} < \varepsilon,$$  \hspace{1cm} (3.39)

where $\varepsilon = 10^{-6}$ mostly; Section 3.3.3 uses $\varepsilon = 10^{-7}$ for an accurate estimation of the error. For all examples, the numerical solution is initialized from $\varphi$ (the lower obstacle) and the boundary condition $f$.

$$u^0(x) = \begin{cases} \varphi(x) & \text{if } x \in \Omega^0_d, \\ f(x) & \text{if } x \in \Gamma_d. \end{cases}$$  \hspace{1cm} (3.40)

3.3.1 Linear obstacle problems

We first consider a non-smooth obstacle $\varphi_1 : \Omega \rightarrow \mathbb{R}$ with $\Omega = [0, 1]^2$, defined by

$$\varphi_1(x,y) = \begin{cases} 5 & \text{if } |x - 0.6| + |y - 0.6| < 0.04, \\ 4.5 & \text{if } (x - 0.6)^2 + (y - 0.25)^2 < 0.001, \\ 4.5 & \text{if } y = 0.57 \text{ and } 0.075 < x < 0.13, \\ 0 & \text{otherwise}. \end{cases}$$  \hspace{1cm} (3.41)
Table 3.1

The number of iterations and CPU for the linear problem of the non-smooth obstacle $\varphi_1$ (3.41). For PDL1P [68], set $\mu = 10^8$, $r_1 = 0.01$, and $r_2 = 12.5$.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>PDL1P</th>
<th>Obstacle SOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>iter</td>
<td>CPU</td>
</tr>
<tr>
<td>32 \times 32</td>
<td>1284</td>
<td>0.13</td>
</tr>
<tr>
<td>64 \times 64</td>
<td>1744</td>
<td>0.71</td>
</tr>
<tr>
<td>128 \times 128</td>
<td>2111</td>
<td>3.58</td>
</tr>
<tr>
<td>256 \times 256</td>
<td>2097</td>
<td>14.09</td>
</tr>
</tbody>
</table>

We solve the linear obstacle problem varying resolutions. The tolerance is set $\varepsilon = 10^{-6}$ hereafter except for examples in Section 3.3.3. Table 3.1 presents the number of iterations and CPU (the elapsed time, measured in second) for the linear problem of the non-smooth obstacle (3.41). One can see from the table, our suggested method requires less iterations and converges about one order faster in the computation time than the PDL1P, a state-of-the-art method. We have also implemented the primal-dual hybrid gradient (PDHD) algorithm in [7, 66, 67] for obstacle problems. The PDL1P turns out to be a simple adaptation of the PDHD and their performances are about the same, particularly when $\mu$ is set large.

For the resolution 64 \times 64, Figure 3.2 depicts the numerical solutions of the PDL1P and the obstacle SOR and their contour lines. For this example, both the PDL1P and the obstacle SOR resulted in almost identical solutions.
Figure 3.2

Solutions to the linear problem for the obstacle $\varphi_1$ (3.41) at resolution $64 \times 64$: (a) the numerical solution by the PDL1P, (b) its contour plot, (c) the numerical solution by the obstacle SOR, and (d) its contour plot.
As the second example, we consider the radially-symmetric obstacle \( \varphi_2 : \Omega \rightarrow \mathbb{R} \) with \( \Omega = [-2, 2]^2 \) defined by

\[
\varphi_2(r) = \begin{cases} 
\sqrt{1 - r^2} & \text{if } r \leq r^*, \\
-1 & \text{otherwise},
\end{cases}
\]

(3.42)

where \( r^* = 0.6979651482233 \ldots \), the solution of

\[
(r^*)^2(1 - \log(r^*/2)) = 1.
\]

(3.43)

For the obstacle \( \varphi_2 \), the analytic solution to the linear obstacle problem can be defined as

\[
u^*(r) = \begin{cases} 
\sqrt{1 - r^2} & \text{if } r \leq r^*, \\
-(r^*)^2 \ln(r/2)/\sqrt{1 - (r^*)^2} & \text{otherwise},
\end{cases}
\]

(3.44)

when the boundary condition is set appropriately using \( u^* \). See Figure 3.3, in which we give plots of \( \varphi_2 \) and the true solution \( u^* \).

In Table 3.2, we compare performances of the PDL1P and the obstacle SOR applied for the linear obstacle problem with (3.42). The PDL1P uses the parameters suggested in [68]
Table 3.2

$L^\infty$-errors, the number of iterations, and the CPU for linear obstacle problem with $\varphi_2$ (3.42): The PDL1P uses the parameters suggested in [68] ($\mu = 0.1$, $r_1 = 0.008$, $r_2 = 15.625$).

<table>
<thead>
<tr>
<th>resolution</th>
<th>PDL1P</th>
<th>Obstacle SOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L^\infty$-error</td>
<td>iter(CPU)</td>
</tr>
<tr>
<td>$32 \times 32$</td>
<td>$8.91 \cdot 10^{-3}$</td>
<td>715 (0.09)</td>
</tr>
<tr>
<td>$64 \times 64$</td>
<td>$3.01 \cdot 10^{-3}$</td>
<td>1340 (0.60)</td>
</tr>
<tr>
<td>$128 \times 128$</td>
<td>$7.66 \cdot 10^{-4}$</td>
<td>1971 (3.51)</td>
</tr>
<tr>
<td>$256 \times 256$</td>
<td>$1.86 \cdot 10^{-4}$</td>
<td>2072 (14.85)</td>
</tr>
</tbody>
</table>

Figure 3.4

Numerical solutions $u_h$ and errors $u_h - u^*$ at the $64 \times 64$ resolution. (a)-(b) by the PDL1P and (c)-(d) by the obstacle SOR.
(µ = 0.1, r₁ = 0.008, r₂ = 15.625). As one can see from the table, our suggested method takes about one order less CPU time than the PDL1P for the computation of the numerical solution. In Figure 3.4, we show the numerical solutions \( u_h \) and the errors \( u_h - u^* \) produced by the PDL1P and the obstacle SOR at the 64 \( \times \) 64 resolution. The solutions are almost identical and the errors are nonpositive. This implies that the numerical solutions of the obstacle problem are underestimated.

![Figure 3.5](image)

**Figure 3.5**

Elastic-plastic torsion problem: (a) the obstacles \( \psi_3 \) and \( \varphi_3 \) and (b) the force \( v \).

As a more general obstacle problem, we consider the elastic-plastic torsion problem in [62]. The problem is to find the equilibrium position of the membrane between two obstacles \( \varphi, \psi \) when a force \( v \) is acting on:

\[
\min_{u} \int_{\Omega} |\nabla u|^2 \, dx - \int_{\Omega} uv \, dx, \quad \text{s.t. } \psi \geq u \geq \varphi \text{ in } \Omega, \; u = f \text{ on } \Gamma. \quad (3.45)
\]
Let $\Omega = [0,1]^2$ and the problem consist of two obstacles $\varphi_3 : \Omega \to \mathbb{R}$, $\psi_3 : \Omega \to \mathbb{R}$ and the force $v : \Omega \to \mathbb{R}$ defined by $\varphi_3(x,y) = -\text{dist}(x,\partial\Omega)$, $\psi_3(x,y) = 0.2$ and

$$v(x,y) = \begin{cases} 
300 & \text{if } (x,y) \in S = \{(x,y) : |x-y| \leq 0.1 \land x \leq 0.3\}, \\
-70e^y g(x) & \text{if } x \leq 1 - y \text{ and } (x,y) \notin S, \\
15e^y g(x) & \text{if } x > 1 - y \text{ and } (x,y) \notin S,
\end{cases} \quad (3.46)$$

where

$$g(x) = \begin{cases} 
6x & \text{if } 0 \leq x \leq 1/6, \\
2(1-3x) & \text{if } 1/6 < x \leq 1/3, \\
6(x-1/3) & \text{if } 1/3 < x \leq 1/2, \\
2(1-3(x-1/3)) & \text{if } 1/2 < x \leq 2/3, \\
6(x-2/3) & \text{if } 2/3 < x \leq 5/6, \\
2(1-3(x-2/3)) & \text{if } 5/6 < x \leq 1. 
\end{cases} \quad (3.47)$$

See Figure 3.5, where the obstacles and the force are depicted.

### Table 3.3

The number of iterations and the CPU time for the elastic-plastic torsion problem (3.45).

For the PDL1P, we use the parameters suggested in [68] ($\mu = 0.1$, $r_1 = 0.008$, $r_2 = 15.625$).

<table>
<thead>
<tr>
<th>$(\varepsilon = 10^{-6})$</th>
<th>PDL1P</th>
<th>Obstacle SOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>resolution</td>
<td>iter</td>
<td>CPU</td>
</tr>
<tr>
<td>32 $\times$ 32</td>
<td>887</td>
<td>0.13</td>
</tr>
<tr>
<td>64 $\times$ 64</td>
<td>1287</td>
<td>0.68</td>
</tr>
<tr>
<td>128 $\times$ 128</td>
<td>1609</td>
<td>3.43</td>
</tr>
<tr>
<td>256 $\times$ 256</td>
<td>1866</td>
<td>17.27</td>
</tr>
</tbody>
</table>

In Table 3.3, we present performances of the PDL1P and the obstacle SOR applied for the elastic-plastic torsion problem (3.45). For the PDL1P, we use the parameters suggested in [68] ($\mu = 0.1$, $r_1 = 0.008$, $r_2 = 15.625$). As one can see from the table, our suggested method again resulted in the numerical solution about one order faster than the PDL1P.
Figure 3.6

The numerical solutions and the contact sets for the elastic-plastic torsion problem at the $64 \times 64$ resolution: (a)-(b) by the PDL1P and (c)-(d) by the obstacle SOR.
measured in the CPU time. In Figure 3.6, we illustrate the simulated membranes in the
equilibrium satisfying (3.45) and their contact sets at resolution $64 \times 64$. In Figures 3.6(b)
and 3.6(d), the upper and lower contact sets are colored in yellow (brightest in gray scale)
and blue (darkest in gray scale), respectively. The results produced by the two methods are
apparently the same.

3.3.2 Nonlinear obstacle problems

The obstacle SOR is implemented for nonlinear obstacle problems as described in Sec-
tion 3.1.2.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>$\varphi_1$</th>
<th>$\varphi_2$</th>
<th>$\varphi_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>iter</td>
<td>CPU</td>
<td>iter</td>
</tr>
<tr>
<td>$32 \times 32$</td>
<td>79</td>
<td>0.02</td>
<td>62</td>
</tr>
<tr>
<td>$64 \times 64$</td>
<td>133</td>
<td>0.16</td>
<td>121</td>
</tr>
<tr>
<td>$128 \times 128$</td>
<td>257</td>
<td>1.25</td>
<td>239</td>
</tr>
<tr>
<td>$256 \times 256$</td>
<td>513</td>
<td>10.19</td>
<td>477</td>
</tr>
</tbody>
</table>

In Table 3.4, we present experiments for which the obstacle SOR is applied for non-
linear obstacle problems with $\varphi = \varphi_i$, $i = 1, 2, 3$. From a comparison with linear cases
presented in Tables 3.1, 3.2, and 3.3, we can see for each of the obstacles that the obsta-
cle SOR iteration for the nonlinear problem converges in a similar number of iterations
as for the linear problem. Only the apparent difference is the CPU time; an iteration of
the nonlinear solver is about as six time expensive as that of the linear solver, due to the computation of coefficients as in (3.19). For $\varphi = \varphi_1$, the nonlinear solution is plotted in Figure 3.7. Compared with the linear solutions in Figure 3.2, the nonlinear solution shows slightly lower function values, which is expected. As the grid point approaches the obstacles, the solution shows an increasing gradient magnitude. This may enlarge weights for far-away grid values as shown in (3.19), which in return acts as a force to reduce function values. The difference between the linear solution and the nonlinear solution, at the $64 \times 64$ resolution, is depicted in Figure 3.8.

### 3.3.3 Post-processing algorithm

In Figure 3.4, one have seen that the error, the difference between the numerical solution and the analytic solution, shows its highest values near the free boundary. The larger error is due to the result of mismatch between the mesh grid and the obstacle edges. In
The difference between the linear solution and the nonlinear solution, at the 64 × 64 resolution, for the obstacle problem with \( \phi = \phi_1 \). (a) \((u_{h,\text{linear}} - u_{h,\text{nonlinear}})\) and (b) its density plot.

In order to eliminate the error effectively, we apply the subgrid FD schemes in Section 3.2 as a post-processing (PP) algorithm. For the examples presented in this subsection, the numerical solutions are solved as follows: (a) The problem is solved with \( \varepsilon = 10^{-5} \) (pre-processing), (b) the free boundary is estimated with \((k_0, k_1) = (10, 4)\) and subgrid FD schemes are applied at neighboring grid points as in Section 3.2, and (c) another round of iterations is applied to satisfy the tolerance \( \varepsilon = 10^{-7} \).

First, we consider a step function for an one-dimensional (1D) obstacle, as in Section 2.1.2. Let \( \Omega = [0, 1] \) and \( \phi_4 : \Omega \to \mathbb{R} \) defined by

\[
\phi_4(x) = \begin{cases} 
0 & \text{if } 0 \leq x < \pi/6, \\
1 & \text{if } \pi/6 \leq x \leq 1.
\end{cases}
\]  

The analytic solution to the linear problem is given as

\[
u_{4,\text{true}}(x) = \begin{cases} 
6x/\pi & \text{if } 0 \leq x < \pi/6, \\
1 & \text{if } \pi/6 \leq x \leq 1.
\end{cases}
\]
The post-processing algorithm applied for an obstacle problem in 1D at resolution 16. (a) the computed solutions without the post-process (blue solid curve) and with the post-process (red dotted curve with × marks) and (b) their errors. Here the subscript $pp$ indicates the post-process.

Figure 3.9 shows the numerical solutions to the linear problem associated to (3.48) with and without the post-process, and their errors. The numerical solutions without and with the post-process are obtained iteratively satisfying the tolerance $\varepsilon = 10^{-7}$. Notice that the solution without post-process is underestimated and shows a relatively high error: $\|u - u_{4,\text{true}}\|_\infty = 0.066$. The error is reduced to $\|u_{pp} - u_{4,\text{true}}\|_\infty = 8.57 \times 10^{-7}$ after the post-process.

The post-processing algorithm is applied to the linear obstacle problem in 2-D involving $\varphi = \varphi_2$. Table 3.5 contains efficiency results that compare performances of the PDL1P, the obstacle SOR (without post-process), and the obstacle SOR with the post-process (Obstacle SOR+PP) at various resolutions; while Table 3.6 presents an accuracy comparison for those methods. According to Table 3.5, the post-processed solution requires about 40% more iterations than the non-post-processed one; the incorporation of the post-process
Table 3.5

CPU time and iteration comparisons for the suggested post-process, applied to the linear problem with $\varphi = \varphi_2$.

<table>
<thead>
<tr>
<th>($\varepsilon = 10^{-7}$)</th>
<th>PDL1P</th>
<th>Obstacle SOR</th>
<th>Obstacle SOR+PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>resolution</td>
<td>iter</td>
<td>CPU</td>
<td>iter</td>
</tr>
<tr>
<td>25 x 25</td>
<td>814</td>
<td>0.07</td>
<td>56</td>
</tr>
<tr>
<td>50 x 50</td>
<td>1486</td>
<td>0.40</td>
<td>108</td>
</tr>
<tr>
<td>100 x 100</td>
<td>2092</td>
<td>2.26</td>
<td>213</td>
</tr>
<tr>
<td>200 x 200</td>
<td>2482</td>
<td>10.47</td>
<td>416</td>
</tr>
</tbody>
</table>

Table 3.6

$L^\infty$ and $L^2$ error comparisons for the suggested post-process, applied to the linear problem with $\varphi = \varphi_2$.

<table>
<thead>
<tr>
<th>($\varepsilon = 10^{-7}$)</th>
<th>PDL1P</th>
<th>Obstacle SOR</th>
<th>Obstacle SOR+PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>resolution</td>
<td>$L^\infty$ error</td>
<td>$L^2$ error</td>
<td>$L^\infty$ error</td>
</tr>
<tr>
<td>25 x 25</td>
<td>1.94 $\cdot 10^{-2}$</td>
<td>4.35 $\cdot 10^{-3}$</td>
<td>1.94 $\cdot 10^{-2}$</td>
</tr>
<tr>
<td>50 x 50</td>
<td>4.38 $\cdot 10^{-3}$</td>
<td>8.10 $\cdot 10^{-4}$</td>
<td>4.39 $\cdot 10^{-3}$</td>
</tr>
<tr>
<td>100 x 100</td>
<td>1.25 $\cdot 10^{-3}$</td>
<td>2.73 $\cdot 10^{-4}$</td>
<td>1.25 $\cdot 10^{-3}$</td>
</tr>
<tr>
<td>200 x 200</td>
<td>5.45 $\cdot 10^{-4}$</td>
<td>7.29 $\cdot 10^{-5}$</td>
<td>5.46 $\cdot 10^{-4}$</td>
</tr>
</tbody>
</table>
makes the iterative algorithm as twice expensive measured in CPU time as the original iteration. However, one can see from Table 3.6 that the post-process makes the error reduced by a factor of $15 \sim 20$. Thus in order to achieve a three-digit accuracy in the maximum-norm, for example, the PDL1P requires 10.47 seconds and the obstacle SOR completes the task in 0.84 seconds; when the obstacle SOR+PP takes only 0.1 second.

![Figure 3.10](image)

Figure 3.10

Plots of the error $(u_h - u^*)$ at the $50 \times 50$ resolution for the linear obstacle problem with $\varphi = \varphi_2$. (a) by the PDL1P, (b) by the obstacle SOR, and (c) by the obstacle SOR+PP.

Figure 3.10 includes plots of the error $(u_h - u^*)$ at the $50 \times 50$ resolution for the linear obstacle problem with $\varphi = \varphi_2$, produced by the PDL1P, the obstacle SOR, and the obstacle SOR+PP. The numerical solutions of the PDL1P and the obstacle SOR are almost identical to each other and clearly underestimated, with the maximum discrepancy occurring around the free boundary due to the misfit between the mesh grid and the free boundary. It can be seen from Figure 3.10(c) that the post-process can eliminate the misfit error very effectively; the remained error is the truncation error introduced by the second-order FD schemes.
3.3.4 Parameter choices

Finally, we present experimental results for parameter choices, when the obstacle SOR is applied for the linear problem with $\varphi = \varphi_3$. For an effective calibration of the optimal relaxation parameter as suggested in (3.28), we first select $h_0 = 1/25$. Then by using a trial-by-error method, we found the calibrated optimal relaxation parameter $\tilde{\omega}_h = 1.61$, which results in the following calibrated constant

$$c_0 \approx 6.0559. \quad (3.50)$$

Thus it follows from (3.27) that the calibrated optimal relaxation parameter reads

$$\tilde{\omega}_{\text{cal},h} \approx \begin{cases} 
1.6817 & \text{when } h = 1/32, \\
1.8371 & \text{when } h = 1/64, \\
1.9097 & \text{when } h = 1/128, \\
1.9538 & \text{when } h = 1/256,
\end{cases} \quad (3.51)$$

which is used for the results of the obstacle SOR included in Table 3.3.

In order to verify effectiveness of the calibration, we implement a line search algorithm to find a relaxation parameter $\tilde{\omega}$ that converges in the smallest number of iterations with $\epsilon = 10^{-6}$, the same tolerance as for the results in Table 3.3. For $h = 1/64$ and $h = 1/128$, the line search algorithm returned the curves as shown in Figure 3.11 with

$$[\tilde{\omega}, \text{iter}] \approx \begin{cases} 
[1.6732, 45] & \text{when } h = 1/32, \\
[1.8217, 95] & \text{when } h = 1/64, \\
[1.9009, 189] & \text{when } h = 1/128.
\end{cases} \quad (3.52)$$

Note that when the calibrated parameters are used, the iteration counts of the obstacle SOR presented in Table 3.3 are 47, 98, and 193 respectively for $h = 1/32$, $h = 1/64$, and $h = 1/128$. Thus the calibrated optimal parameters in (3.51) are quite accurate for the optimal convergence.
Figure 3.11

The relaxation parameter $\omega$ (horizontal axis) vs. the number of iterations (vertical axis) for solving the linear obstacle problem with $\varphi = \varphi_3$ by the obstacle SOR:

(a) when $h = 1/64$ and (b) when $h = 1/128$. 
4.1 The Variable-$\theta$ Method

This section introduces the variable-$\theta$ method for the numerical solution of the heat equation (2.10) for simplicity; it can be applied straightforwardly for parabolic obstacle problems (2.11), as shown in Section 4.2 below. The new method, briefly speaking, takes the advantage of the CN method (in particular, the second-order accuracy in time) and makes up for oscillations by employing the implicit method locally near discontinuities.

As we have seen in Section 2.2.2, the CN method may produce spurious oscillations near discontinuities of the data. In order to remove the oscillations from the numerical solution of the CN method, we will simply allow the parameter $\theta$ in (2.18) to become 1 in the vicinity of discontinuities. Thus the resulting method will be a hybrid procedure combining the CN method and the implicit method. This strategy will prevent the numerical solution from developing spurious oscillations. Since the variable-$\theta$ method assigns $1/2$ to the parameter $\theta$ at most of the grid points and 1 only when necessary, the new method can produce a numerical solution of near second-order accuracy. It has been numerically verified that the proposed method is of second-order accuracy for smooth data, while its accuracy is deteriorated (but, not much) for nonsmooth data; see Tables 4.1, 4.3, and 4.5 below. Now, we begin with an effective strategy for the detection of oscillations.
4.1.1 The wobble set

We define the wobble set, the set of wobble points, as a collection of the grid points where the solution has high fluctuations so that the implicit method ($\theta = 1$) should be applied for the numerical solution not to develop oscillations. Consider the following claim.

**Claim 4.1.1** *Numerical oscillations of the CN method occur when its explicit half step produces spurious non-physical oscillations.*

This claim can be easily verified. The implicit method ($\theta \equiv 1$) introduces no spurious oscillations, so it is also true for the implicit half step of the CN method. When the explicit scheme ($\theta \equiv 0$) is applied with a time-step size larger than the stability limit, it may add spurious oscillations to the numerical solution. Thus we can conclude that when the CN method is applied without satisfying the maximum principle, it may introduce numerical oscillations in its first explicit half step. It must be further noticed that non-physical oscillations are a form of non-physical local extrema. Thus the wobble set may include points where the explicit step of the CN method introduces undesired local extrema.

Recall that $(I - \frac{\Delta t}{2} \mathcal{A}) u^{n-1}$ is the explicit half step of the CN method (2.21). Let

$$u^{n,*} \equiv \left(I - \frac{\Delta t}{2} \mathcal{A}\right) u^{n-1}. \quad (4.1)$$
4.1.1.1 The wobble set in 1D

We start with the determination of the wobble set for 1D cases, for simplicity. Define an indicator function for local extrema \( \text{idxt} \) as

\[
\text{idxt}(a, b, c) := \begin{cases} 
0, & \text{if } \min(a, c) < b < \max(a, c), \\
1, & \text{if } b = \max(a, c), \\
-1, & \text{if } b = \min(a, c), \\
2, & \text{if } \max(a, c) < b, \\
-2, & \text{if } b < \min(a, c).
\end{cases}
\] (4.2)

Then, we can define the wobble set of \( u^{n-1} \), to be used for the computation of \( u^n \), as

\[
W^{n-1}_{1D} = \{ x_i \in (-1, 1) \mid \text{idxt}(u^{n,*}_{i-1}, u^n_{i,*}, u^{n,*}_i) \neq 0 \text{ and } |\text{idxt}(u^{n,*}_{i-1}, u^n_{i,*}, u^{n,*}_i) + \text{idxt}(u^{n-1}_{i-1}, u^{n-1}_i, u^{n-1}_{i+1})| < 4 \}.
\] (4.3)

where \( u^{n,*} \) is the result of the explicit half step of the CN method given in (4.1). Thus the wobble set in 1D is a collection of interior points \( x_i \) where \( u^n_{i,*} \) becomes a local extremum while \( u^{n-1}_{i} \) is either non-extreme or extreme in the opposite sense. The wobble set in (4.3) excludes cases where a strict extremum in \( u^{n-1} \) becomes a strict extremum in the same sense in \( u^{n,*} \); that is,

\[
|\text{idxt}(u^{n,*}_{i-1}, u^n_{i,*}, u^{n,*}_i) + \text{idxt}(u^{n-1}_{i-1}, u^n_i, u^{n-1}_{i+1})| = 4.
\] (4.4)

4.1.1.2 The wobble set in 2D

The above scheme of determining the wobble set to 1D problems can be expanded to 2D cases, in many ways. We suggest the following scheme. Consider an interior grid point \( x_{ij} \) and the four partial directions made with its eight vicinal points. See Figure 4.1. We apply (4.3) for the four partial directions; if at least one of them wobbles in the corresponding
Figure 4.1

The eight vicinal points of $x_{ij}$ and four partial directions.

direction, then we select the point $x_{ij}$ as a wobble point. Let $P$, $Q$, and $R$ be point indices
and define an indicator function

$$
iswb(P, Q, R, n-1) := \begin{cases} 1, & \text{if } \text{idxt}(u_{P}^{n}, u_{Q}^{n}, u_{R}^{n}) \neq 0 \text{ and } \left| \text{idxt}(u_{P}^{n}, u_{Q}^{n}, u_{R}^{n}) + \text{idxt}(u_{P}^{n-1}, u_{Q}^{n-1}, u_{R}^{n-1}) \right| < 4, \\ 0, & \text{otherwise}. \end{cases}$$  \tag{4.5}

Then, the wobble set (for the computation of $u^n = \{u_{ij}^n\}$) is defined as

$$W^{n-1} = \left\{ x_{ij} \in \Omega^0_d \mid \text{iswb}[(i-1, j), (i, j), (i+1, j), n-1] = 1 \right\}$$

or $\text{iswb}[(i-1, j-1), (i, j), (i+1, j+1), n-1] = 1$

or $\text{iswb}[(i, j-1), (i, j), (i, j+1), n-1] = 1$

or $\text{iswb}[(i+1, j-1), (i, j), (i-1, j+1), n-1] = 1$. \tag{4.6}

4.1.2 The variable-$\theta$ method and its qualitative analysis

Having the wobble set, the parameter $\theta$ for the computation of $u^n$ can be assigned
pointwisely

$$\theta_{ij}^n := \theta(x_{ij}, t^n) = \begin{cases} 1, & \text{if } x_{ij} \in W^{n-1}, \\ 1/2, & \text{otherwise}. \end{cases}$$  \tag{4.7}
Thus, the variable-$\theta$ method for (2.18) can be formulated as

$$\frac{u_{ij}^n - u_{ij}^{n-1}}{\Delta t} + A[\theta_{ij}^n u_{ij}^n + (1 - \theta_{ij}^n)u_{ij}^{n-1}] = f_{ij}^{n-1+\theta_{ij}^n}. \quad (4.8)$$

**Remark 4.1.2** The wobble set (4.6) is the set of interior grid points $x_{ij}$ where $u_{ij}^{n,*}$ becomes a local extremum while $u_{ij}^{n-1}$ is either a non-extreme or an extreme in the opposite sense, for at least one of four partial directions. Thus it can be a collection of isolated points and the corresponding $\{\theta_{ij}^n\}$ is most likely discontinuous over the domain, particularly for first several time steps after encountering discontinuities in data. Here the reader may concern stability issues of the numerical solution. We have tested numerical stability and accuracy of the variable-$\theta$ method for various examples, with variants of the wobble set. When the wobble set is dilated, for example, some of isolated wobble points are connected. However, it has been observed numerically that the change in numerical accuracy is negligible for all tested cases and the variable-$\theta$ method introduces no stability issues.

**Truncation error:** It is a simple matter to check that at each of the grid point $x_{ij}$ the truncation error of the variable-$\theta$ method gives

$$\mathcal{O} \left[ (1 - 2\theta_{ij}) \Delta t + \Delta t^2 + \Delta x^2 + \Delta y^2 \right]. \quad (4.9)$$

Thus it is almost trivial to say that once the variable-$\theta$ method (4.8) is stable, it would converge with a numerical error given above. That is,

$$|u(x_{ij}, t^n) - u_{ij}^n| = \mathcal{O} \left[ (1 - 2\theta_{ij}) \Delta t + \Delta t^2 + \Delta x^2 + \Delta y^2 \right], \text{ as } \Delta t, \Delta x, \Delta y \to 0.$$
Stability: For simplicity, we may consider the 1D case. Setting $f \equiv 0$, the 1D version of (4.8) reads pointwisely (at the grid point $x_p$)

$$
-\theta_p \mu u_{p-1}^n + (1 + 2\theta_p \mu)u_p^n - \theta_p \mu u_{p+1}^n = (1 - \theta_p)\mu u_{p-1}^{n-1} + [1 - 2(1 - \theta_p)\mu]u_p^{n-1} + (1 - \theta_p)\mu u_{p+1}^{n-1},
$$

(4.10)

where $\theta_p = \theta^n$ and $\mu = \Delta t / \Delta x^2$. Employing the von Neumann analysis (substituting $(g_p)^n e^{i\xi}$ for $u_p^n$, where $i$ is the imaginary unit and $\xi$ denotes the frequency), we have

$$
g_p \left[ -\theta_p \mu e^{-i\xi} + (1 + 2\theta_p \mu) - \theta_p \mu e^{i\xi} \right] = (1 - \theta_p)\mu e^{-i\xi} + [1 - 2(1 - \theta_p)\mu] + (1 - \theta_p)\mu e^{i\xi},
$$

or, equivalently,

$$
g_p = \frac{1 - 2(1 - \theta_p)\mu (1 - \cos \xi)}{1 + 2\theta_p \mu (1 - \cos \xi)} = \frac{1 - 4(1 - \theta_p)\mu \sin^2(\xi/2)}{1 + 4\theta_p \mu \sin^2(\xi/2)}.
$$

(4.11)

Therefore, $|g_p| \leq 1$ when $\theta_p \in [1/2, 1]$, for all values of $\mu > 0$. For higher-dimensional problems, stability of the variable-$\theta$ method can be proved similarly. The point is that the variable-$\theta$ method introduces no extra mathematical issues on stability and convergence.

**Remark 4.1.3** When $\theta$ is constant, the $\theta$-method is stable for $\theta \in [1/2, 1]$. The above von Neumann analysis shows that the variable-$\theta$ method is stable for $\theta \in [1/2, 1]$, although $\theta$ is discontinuous. In particular, it does not introduce any spurious local extrema by setting $\theta = 1$, because it satisfies the maximum principle locally; see (2.25).

Furthermore, we can prove the unconditional maximum principle for the variable-$\theta$ method.
Theorem 4.1.4 The numerical solution of the variable-\( \theta \) method to heat equation:

\[
\partial_t u - \Delta u = 0
\]  

(4.12)

unconditionally satisfies the maximum principle.

Proof. First, we consider the simple 1D heat equation with \( f = 0 \):

\[
\partial_t u - \partial_{xx} u = 0.
\]  

(4.13)

Let \( \mu = \Delta t / \Delta x^2 > 0 \). For fixed \( 0 \leq i \leq n_x, 0 \leq n \leq n_t \) and \( \theta \in [0, 1] \), the \( \theta \)-method of (4.13) can be expressed pointwisely as

\[
(1 + 2\theta \mu)u^n_i = \mu (u^n_{i-1} + u^n_{i+1}) + \mu (u^{n-1}_{i-1} + u^{n-1}_{i+1}) + [1 - 2(1 - \theta)\mu]u^{n-1}_i. \]  

(4.14)

If \( \theta^n_i = 1 \) then the variable-\( \theta \) method is formulated as

\[
(1 + 2\mu)u^n_i = \mu (u^n_{i-1} + u^n_{i+1}) + u^{n-1}_i
\]  

(4.15)

and all coefficients in the right side of the equation (4.15) are positive and sum to \((1 + 2\mu)\).

Hence this leads to the conclusion that \( u^n_i \) can be a local maximum or minimum only if all three neighboring points: \( u^{n-1}_i, u^n_{i-1}, u^n_{i+1} \), related to the right side of (4.15), have the same maximum or minimum value.

If \( \theta^n_i = 1/2 \), we have

\[
(1 + \mu)u^n_i = \frac{\mu}{2} (u^n_{i-1} + u^n_{i+1}) + u^*_i,
\]  

(4.16)

where

\[
u^*_i = \frac{\mu}{2} (u^{n-1}_{i-1} + u^{n-1}_{i+1}) + (1 - \mu)u^{n-1}_i.\]  

(4.17)
Clearly, \( u^n_i \) can be a maximum or minimum of the values in (4.16) only if all three neighboring points: \( u^n_i, u^n_{i-1}, u^n_{i+1} \), related to the right side of (4.16), have the same maximum or minimum value since all coefficients in the right side of (4.16) are positive and sum to \((1 + \mu)\).

Let \( N^{n-1}_i := \{u^{n-1}_{i-2}, u^{n-1}_{i-1}, u^{n-1}_i, u^{n-1}_{i+1}, u^{n-1}_{i+2}\} \) be the set of two-step neighboring points of \( u^{n-1}_i \). Suppose that \( u^n_i \) has a maximum of \( N^{n-1}_i \). Then,

\[
\max\{u^n_{i-1}, u^n_{i+1}\} \leq \frac{\mu}{2} (u^n_i + u^n_i) + (1 - \mu)u^n_i = u^n_i. \tag{4.18}
\]

From the wobble process hypothesis (4.2) and (4.3), we have

\[
\min\{u^n_{i-1}, u^n_{i+1}\} < u^n_i < \max\{u^n_{i-1}, u^n_{i+1}\}. \tag{4.19}
\]

This is contradiction, thus \( u^n_i \) can not be a maximum of \( N^{n-1}_i \). Similarly, we can prove that \( u^n_i \) also can not be a minimum of \( N^{n-1}_i \).

Combining all the facts for \( \theta^n_i = 1/2 \), we conclude that the interior point \( u^n_i \) can be a local maximum or minimum only if all seven neighboring points \( N^{n-1}_i \cup \{u^n_{i-1}, u^n_{i+1}\} \) have the same maximum or minimum.

Thus, for both \( \theta^n_i = 1 \) and \( \theta^n_i = 1/2 \), the interior point \( u^n_i \) can be a local maximum or minimum only if all neighboring points have the same maximum or minimum. It implies that the variable-\( \theta \) method for 1D heat equation (4.13) satisfies the maximum principle for all \( \mu > 0 \). Moreover, we can easily extend the proof above to multi-dimensional cases.

This completes the proof. \[ \square \]
The numerical solutions of (2.22) for $0 \leq t \leq T = 0.5$, when $\Delta x = 0.04$ ($n_x = 50$) and $\Delta t = 0.02$ ($n_t = 25$): (a) the numerical solution by variable-$\theta$ method and (b) the numerical solutions at $t = T = 0.5$.

4.1.3 Revisited: the 1D heat equation

In order to investigate the effectiveness of the variable-$\theta$ method, we first apply the new method to the 1D heat equation (2.22) with the same space-grid and time-step sizes as in Figure 2.2. Figure 4.2(a) presents the numerical solution evolved by the variable-$\theta$ method, while Figure 4.2(b) depicts the numerical solutions at $t = T = 0.5$ by the implicit method, the CN method, and the variable-$\theta$ method. As one can see from the figure, the new method has produced an accurate numerical solution without spurious oscillations. The implicit method does not introduce numerical oscillations, but its error is considerable due to the first-order approximation of the time-derivative. On the other hand, the CN method reveals numerical oscillations, although it is quite accurate far from the discontinuities.

The variable-$\theta$ method combines advantages from the two conventional methods, which can be verified numerically. See Figure 4.3, where the propagation of numerical er-
Figure 4.3

Propagation of numerical errors for (2.22) by (a) the implicit method, (b) the CN method, and (c) the variable-$\theta$ method, for $0 \leq t \leq T = 0.5$, when $\Delta x = 0.04$ ($n_x = 50$) and $\Delta t = 0.02$ ($n_t = 25$).

ror is depicted for the implicit method, the CN method, and the variable-$\theta$ method, for $0 \leq t \leq T = 0.5$. Each of the three methods produces an observable error in first few steps of time marching, with the error of the CN method being largest. (It should be noticed that Figure 4.3(b) is in a different scale.) The error of the implicit method is vivid, although it decreases as the time marches; the error of the CN method is small in smooth regions. However, the variable-$\theta$ method introduces a small error (similar to that of the implicit method) in the beginning and produces an accurate numerical solution, like the CN method in smooth regions. The $L^\infty$-errors $E_{\infty}[T]$ at $T = 0.5$ are $1.799 \cdot 10^{-2}$, $4.505 \cdot 10^{-2}$, and $7.297 \cdot 10^{-4}$ respectively for the implicit method, the CN method, and the variable-$\theta$ method.

Remark 4.1.5 As one can see from Figure 4.3(c), the error of the variable-$\theta$ method shows slight oscillations at the early time. However, it does not imply that the corresponding solution involves spurious oscillations; see Figure 4.2(a), in which the solution involves
no undesired local extrema. We believe that the oscillation on the error of the variable-\(\theta\) method is due to discontinuities on \(\theta\). Nonetheless it does not introduce any spurious oscillations to the solution.

When the CN method chooses the time step size small enough, the oscillations should disappear. However, this will make the method expensive computationally. For the example presented in Figures 4.2 and 4.3, the CN method requires at least 313 time steps to satisfy the maximum principle (2.25); with the choice \(\Delta t = T/313\), it takes 0.107 seconds. For the same accuracy level of the CN method (error \(\approx 3.3 \cdot 10^{-4}\)), the proposed variable-\(\theta\) method requires 30 time steps and a slightly more number of grid points \((n_x = 64)\); the new method takes 0.015 seconds, which is more than 7 times faster than the CN method for the problem in 1D. It has been numerically verified that the variable-\(\theta\) method is at least 4 times faster than the CN method for all cases we have tested (including 2D problems), when the methods are to obtain nonoscillatory solutions in the same accuracy level from problems of nonsmooth data. The higher accuracy levels, the more gains in computational costs for the variable-\(\theta\) method over the CN method.

### 4.1.4 The variable-\(\theta\) method for smooth solutions

Although the variable-\(\theta\) method can remove the oscillations, the reader may have a question concerning the variable-\(\theta\) method: *How effective is the method for smooth data?* In order to answer the question, we examine the method applied for the following problem of smooth data, for which the CN method would not produce spurious oscillations:

\[
\begin{align*}
\partial_t u - \partial_{xx} u &= f(x, t), \quad (x, t) \in (-1, 1) \times (0, T], \\
\quad u(x, 0) &= \sin 2\pi x, \quad x \in [-1, 1],
\end{align*}
\]  

\text{equation (4.20)}
where the source term $f$ and the boundary condition are set corresponding to the analytic solution given by

$$u(x, t) = \sin 2\pi(x - t), \quad (x, t) \in [-1, 1] \times [0, T],$$

(4.21)
of which the oscillatory pattern travels to the northeast direction in the $xt$-plane.

![Figure 4.4](image)

Figure 4.4

The analytic and numerical solutions of (4.20) for $0 \leq t \leq T = 1.0$, when $\Delta x = 0.02$ ($n_x = 100$) and $\Delta t = 0.01$ ($n_t = 100$): (a) the analytic solution, (b) the CN method, and (c) the variable-$\theta$ method.

Figure 4.4 depicts the evolution of the analytic and numerical solutions for (4.20) by the CN method and the variable-$\theta$ method for $0 \leq t \leq T = 1.0$, when $\Delta x = 0.02$ ($n_x = 100$) and $\Delta t = 0.01$ ($n_t = 100$). The $L^\infty$-errors $E^\infty[T]$ at $T = 1.0$ are $2.435 \cdot 10^{-3}$ and $2.272 \cdot 10^{-3}$ respectively for the CN method and the variable-$\theta$ method. One can see from Figure 4.4 and the $L^\infty$-errors $E^\infty[T]$ that the variable-$\theta$ method shows a quite similar accuracy to the CN method (occasionally, better). This implies that in the case of the smooth data the new method assigns implicit grids ($\theta = 1$) as few as possible so that it would not lose the accuracy of the CN method.
The accumulated relative $L^2$-error $E_2[0, T]$ for the numerical solution of (4.20), when $T = 1.0$ and $\Delta x = 2\Delta t$.

<table>
<thead>
<tr>
<th>$(n_t, n_x)$</th>
<th>Implicit</th>
<th>CN</th>
<th>Variable-$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(100, 100)</td>
<td>$3.89 \cdot 10^{-3}$</td>
<td>$1.93 \cdot 10^{-3}$</td>
<td>$1.83 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>(200, 200)</td>
<td>$2.30 \cdot 10^{-4}$</td>
<td>$4.83 \cdot 10^{-4}$</td>
<td>$4.70 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>(400, 400)</td>
<td>$1.24 \cdot 10^{-3}$</td>
<td>$1.21 \cdot 10^{-4}$</td>
<td>$1.18 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>order</td>
<td>0.82</td>
<td>2.00</td>
<td>1.98</td>
</tr>
</tbody>
</table>

Table 4.1 summarizes the accumulated relative $L^2$-error $E_2[0, T]$ for the numerical solution of (4.20), when $T = 1.0$ and $\Delta x = 2\Delta t$. We can see from the table that for the case of smooth data, the variable-$\theta$ method achieves a second-order accuracy, similarly to the CN method. The new method is effective in the selection of the wobble set so that it tries to maintain the second-order accuracy of the CN method for smooth data. It should be noticed that the variable-$\theta$ method shows the lowest error among the three methods. When the solution is smooth but highly oscillatory, both the CN method and the variable-$\theta$ method result in smooth solutions. However, it is often the case that the variable-$\theta$ method produces more accurate solutions, as shown in Table 4.1 and Table 4.3 below.

4.1.5 Effectiveness of the wobble set scheme

In order to investigate effectiveness of the wobble set scheme suggested in Section 4.1.1, Table 4.2 compares it with the cases that the wobble set is assigned artificially by 0%, 8%, 16%, and 32% largest values of $\{|Au^n_i|\}$ (after sorting). The number in parentheses indicates the size or average size (per time level) of the wobble set either artificially assigned or determined by the variable-$\theta$ method. For the nonsmooth data (the problem in (2.22)), the
Table 4.2

The accumulated relative $L^2$-errors $E_2[0, T]$ and the average count of wobble points for the numerical solutions of (2.22) and (4.20) when $T = 1.0$, $\Delta x = 0.08$ ($n_x = 25$), and $\Delta t = 0.04$ ($n_t = 25$). The wobble set is assigned either artificially by 0%, 8%, 16%, and 32% largest values of $\{|Au_{i}^{n-1}|\}$ or by the variable-$\theta$ method.

<table>
<thead>
<tr>
<th></th>
<th>0% (0; CN)</th>
<th>8% (2)</th>
<th>16% (4)</th>
<th>32% (8)</th>
<th>Variable-$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2.22)</td>
<td>$4.49 \cdot 10^{-2}$</td>
<td>$2.46 \cdot 10^{-2}$</td>
<td>$2.30 \cdot 10^{-2}$</td>
<td>$2.60 \cdot 10^{-2}$</td>
<td>$7.61 \cdot 10^{-3}$ (6.0)</td>
</tr>
<tr>
<td>(4.20)</td>
<td>$3.14 \cdot 10^{-2}$</td>
<td>$2.82 \cdot 10^{-2}$</td>
<td>$2.41 \cdot 10^{-2}$</td>
<td>$1.60 \cdot 10^{-2}$</td>
<td>$2.44 \cdot 10^{-2}$ (3.9)</td>
</tr>
</tbody>
</table>

accumulated relative $L^2$-error becomes smallest among the cases of artificially-assigned wobble sets, when the algorithm chooses 16% largest values of $\{|Au_{i}^{n-1}|\}$. However, the variable-$\theta$ method results in a much more accurate solution than the artificial-assignment cases; for the proposed method, the size of the wobble set turns out to be 6.0 in average.

In the case of smooth data (the problem in (4.20)), the error decreases continuously as the artificially-assigned wobble set grows up to 32%. For the smooth problem, the variable-$\theta$ method results in a numerical solution whose accuracy is a bit worse than the best solution, selecting wobble points of size 3.9 in average. By adapting the $\text{iswb}$ function (4.5), we may allow the algorithm to choose more wobble points so as to improve accuracy for this problem; however, the resulting variable-$\theta$ method might introduce excessive smoothing of steep changes in the numerical solution, for most other cases. The suggested strategy for the choice of the wobble set (in Section 4.1.1) works well for most cases to produce an accurate numerical solution which is near optimal for nonsmooth data and showing a similar accuracy to that of the CN method for smooth data.
Remark 4.1.6 The source term $f$ in (4.20) incorporates oscillatory patterns of the analytic solution (4.21) which push summits and valleys of the solution to rapidly travel to the positive $x$-direction. The application of certain smoothing operations can improve accuracy of the numerical solution as in the case of smooth data in Table 4.2. The variable-$\theta$ method must be useful whenever the numerical solution is expected to involve steep changes, due to either smooth or nonsmooth data.

Here, we summarize our observations for the variable-$\theta$ method.

- (For smooth data): Accuracy of the numerical solution is not much sensitive to a little excessive assignment of the wobble set. The wobble set scheme in Section 4.1.1 makes the variable-$\theta$ method resolve its numerical solution with a similar accuracy to the CN method.

- (For nonsmooth data): The proposed method can produce an accurate numerical solution by selecting a wobble set which is a collection of a small number of points but sufficient for suppressing spurious oscillations.

4.1.6 Multiple nonsmooth sources

Besides nonsmooth initial data, a nonsmooth source term may introduce spurious oscillations into the numerical solution of the CN method. The nonsmooth source generates sudden high-frequency components at points where the source is applied. Consider a source term $f : [-1, 1] \times J \to \mathbb{R}$ to be applied to (2.22):

$$f(x, t) = \begin{cases} -100, & \text{if } (x, t) \in \{0\} \times \left[\frac{T}{2}, T\right], \\ 0, & \text{otherwise.} \end{cases} \quad (4.22)$$

Figure 4.5 presents the evolution of the numerical solutions of (2.22) with (4.22), by the implicit method, the CN method, and the variable-$\theta$ method. Note that at a glance all the methods resolve their numerical solutions quite similarly for the nonsmooth source; in particular, the CN method has responded to the nonsmooth source relatively well. To
The numerical solutions of (2.22) with the nonsmooth source term (4.22), when \( \Delta x = 0.04 \) \((n_x = 50)\) and \( \Delta t = 0.02 \) \((n_t = 25)\): (a) the implicit method, (b) the CN method, and (c) the variable-\( \theta \) method.

Numerical solutions in Figure 4.5 on the center cross line, \( \{0\} \times J \).
see details of the solution evolution, Figure 4.6 depicts the numerical solution of the three methods on the center cross line, \( \{0\} \times J \). The CN method introduces spurious oscillations triggered from the nonsmooth source, although the oscillations are damped out as time marches. On the other hand, the suggested variable-\( \theta \) method can suppress spurious oscillations effectively. Note that in Section 4.1.1, the wobble set is selected using \( u^{n-1} \) and \( u^{n,*} \) only, but not incorporating the source term explicitly. It implies that our wobble set is determined based on solution behaviors, turns out effective enough to detect portents of spurious oscillations, and makes the resulting variable-\( \theta \) method produce an accurate solution without spurious oscillations for all kinds of nonsmooth data.

### 4.2 Obstacle Relaxation for Parabolic Obstacle Problems

This section introduces an effective numerical method for solving parabolic obstacle problems of the form (2.11). The new method can easily incorporate the variable-\( \theta \) method proposed in the previous section.

Let us begin with the \( \theta \)-method for (2.11), which is formulated as follows.

\[
\begin{align*}
\frac{u_{ij}^n - u_{ij}^{n-1}}{\Delta t} + A[\theta u_{ij}^n + (1 - \theta)u_{ij}^{n-1}] & \geq f_{ij}^{n-1+\theta}, \\
\frac{u_{ij}^n - u_{ij}^{n-1}}{\Delta t} + A[\theta u_{ij}^n + (1 - \theta)u_{ij}^{n-1}] - f_{ij}^{n-1+\theta} & = 0,
\end{align*}
\]

\( in \Omega_0^d \times J_d, \) (4.23)

\[
\begin{align*}
u_{ij}^n & = g_{ij}, \quad on \; \Gamma_d \times J_d, \\
u_{ij}^0 & = \varphi_{ij}, \quad x_{ij} \in \Omega_d, \; t = 0,
\end{align*}
\]

where \( \Omega_0^d \) denotes the set of interior grid points and \( \Gamma_d \) is the set of boundary grid points. Since the variable-\( \theta \) method can be applied for parabolic problems without introducing
difficulties on convergence and stability, as shown in the previous section, the reader may consider (4.23) as a variable-$\theta$ method by considering $\theta$ as $\theta_{ij}^{n-1}$.

Each time step of (4.23) can be seen as an elliptic obstacle problem that finds $u^n$ satisfying all the conditions. In order to introduce an effective numerical method that solves it efficiently, we first rewrite (4.23) as follows.

$$
\begin{align*}
\mathcal{B}u_{ij}^n &\geq r_{ij}^{n-1}, \\
u_{ij}^n &\geq \varphi_{ij}, \\
(\mathcal{B}u_{ij}^n - r_{ij}^{n-1})(u_{ij}^n - \varphi_{ij}) &= 0,
\end{align*}
$$

in $\Omega_0^d \times J_d$, \hspace{1cm} (4.24)

where $u_{ij}^0 = \varphi_{ij}, \hspace{0.2cm} x_{ij} \in \Omega_d$, and

$$
\mathcal{B}u_{ij}^n = (I + \theta \Delta t A)u_{ij}^n, \hspace{0.2cm} r_{ij}^{n-1} = [I + (1 - \theta)\Delta t A]u_{ij}^{n-1} + \Delta t f_{ij}^{n-1+\theta}.
$$

(4.25)

Then, as in [33], we can apply the so-called obstacle relaxation (OR), which is a point relaxation method incorporating an adjustment of values to satisfy all the obstacle conditions.

It is often the case that relaxation methods solving an algebraic system begin with a regular splitting of the coefficient matrix. The splitting $\mathcal{B} = \mathcal{M} - \mathcal{N}$ is called a regular splitting of $\mathcal{B}$ if $\mathcal{M}$ is nonsingular with $\mathcal{M}^{-1} \geq 0$ and $\mathcal{N} \geq 0$. It has been well known [58] that if $\mathcal{B} = \mathcal{M} - \mathcal{N}$ is a regular splitting and $\mathcal{B}^{-1} \geq 0$, then the spectral radius of the iteration matrix ($\mathcal{M}^{-1}\mathcal{N} = I - \mathcal{M}^{-1}\mathcal{B}$) is strictly less than 1. That is,

$$
\rho(\mathcal{M}^{-1}\mathcal{N}) = \frac{\rho(\mathcal{B}^{-1}\mathcal{N})}{1 + \rho(\mathcal{B}^{-1}\mathcal{N})} < 1.
$$

(4.26)
Now, given an initialization $u^{n,0} \equiv \{u^{n,0}_{ij}\}$, let

$$u^{n,k}_{ij} = [u^{n,k-1} + M^{-1}(r^{n-1} - Bu^{n,k-1})]_{ij} \quad (4.27)$$

be a point relaxation method solving $Bu^n = r^{n-1}$ for $u^n = \{u^n_{ij}\}$. Then the $\mathcal{OR}$ procedure for solving (4.24) can be formulated as

\begin{algorithm}
for $k = 1, 2, \ldots$
  for $x_{ij} \in \Omega^0_d$
    (a) $u^{n,k*}_{ij} = [u^{n,k-1} + M^{-1}(r^{n-1} - Bu^{n,k-1})]_{ij}$;
    (b) $u^{n,k}_{ij} = \max(u^{n,k*}_{ij}, \varphi_{ij})$;
  end
end
\end{algorithm}

\textbf{Theorem 4.2.1} Suppose that the point relaxation step in (4.28.a) is based on a regular splitting of $B = M - \mathcal{N}$. Then, the $\mathcal{OR}$ is convergent and its limit satisfies (4.24).

\textbf{Proof.} Since $B$ is an $M$-matrix, relaxation methods incorporating a regular splitting must be convergent. Let $\hat{u}^n$ be the limit of the iterates $u^{n,k}$, $k = 1, 2, \ldots$, in $\mathcal{OR}$. Then it is clear to see from $\mathcal{OR}$ that

$$\hat{u}^n_{ij} \geq \varphi_{ij}, \quad \forall x_{ij} \in \Omega^0_d,$$

and

$$\hat{u}^n_{ij} = \max\left( [\hat{u}^n + M^{-1}(r^{n-1} - B\hat{u}^n)]_{ij} , \varphi_{ij} \right), \quad \forall x_{ij} \in \Omega^0_d. \quad (4.29)$$

Let $\hat{u}^n_{pq} = \varphi_{pq}$ at a point $x_{pq} \in \Omega^0_d$. Then (4.29) implies

$$[\hat{u}^n + M^{-1}(r^{n-1} - B\hat{u}^n)]_{pq} \leq \varphi_{pq} = \hat{u}^n_{pq}. \quad (4.30)$$
Note that $\mathcal{M}^{-1} \geq 0$ (because the relaxation incorporates a regular splitting). Thus, since (4.30) implies $[\mathcal{M}^{-1}(\mathbf{r}^{n-1} - B\mathbf{\hat{u}}^n)]_{pq} \leq 0$, we have

$$B\mathbf{\hat{u}}^n_{pq} \geq \mathbf{r}^{n-1}_{pq}. \quad (4.31)$$

Finally, let $\mathbf{\hat{u}}^n_{pq} > \varphi_{pq}$ at a point $\mathbf{x}_{pq} \in \Omega^0$. Then (4.29) implies

$$\mathbf{\hat{u}}^n_{pq} = \left[\mathbf{\hat{u}}^n + \mathcal{M}^{-1}(\mathbf{r}^{n-1} - B\mathbf{\hat{u}}^n)\right]_{pq}.$$

from which we conclude $B\mathbf{\hat{u}}^n_{pq} = \mathbf{r}^{n-1}_{pq}$. This completes the proof. \Box

Popular relaxation methods are the Jacobi method, the Gauss-Seidel method, and the successive over-relaxation (SOR) method; each of which is formulated with a specific regular splitting. In this dissertation, we will employ the SOR method for the relaxation, which converges fast when the relaxation parameter is set appropriately. The optimal SOR parameter can be found theoretically as [59]

$$\tilde{\omega} = \frac{2}{1 + \sqrt{1 - \rho(J_T)}}, \quad (4.33)$$

where $J_T$ is the associated iteration matrix for the Jacobi method; see also [58, p. 123].

When the algebraic system is obtained from second-order elliptic problems by applying second-order discretization schemes, we may assume $\rho(J_T) = 1 - c_0^2 h^2$, where $h$ is the mesh parameter. Thus the optimal parameter for a given mesh size $h$, $\tilde{\omega}_h$, can be obtained as

$$\tilde{\omega}_h = \frac{2}{1 + c_0 h}. \quad (4.34)$$

For a specific mesh size $h_0$, one can select $c_0$ which makes the SOR converge fastest (calibration); the same $c_0$ can be used to produce near-optimal SOR parameters for algebraic systems of various $h$’s.
Remark 4.2.2  

- The authors first introduced the OR for the numerical solution of elliptic obstacle problems [33]; it has been numerically verified that the OR is about one-order more efficient than the primal-dual method for elliptic obstacle problems, one of state-of-the-art algorithms studied in [68].

- Theorem 4.2.1 is a generalization of the convergence theory developed for (simpler) elliptic obstacle problems in [33].

The main idea of OR is applicable with little modification for the numerical solution of more general obstacle problems, linear or nonlinear, with or without external force terms. For example, a general form of obstacle problem can be formulated by minimizing the surface area of a membrane that is bounded from below ($\varphi$) and from above ($\psi$). The obstacle problem (2.11) can be reformulated for double obstacles as follows.

$$
\begin{align*}
\partial_t u - \Delta u &\geq f \quad \text{when } u = \varphi, \\
\partial_t u - \Delta u &\leq f \quad \text{when } u = \psi, \\
\varphi &\leq u \leq \psi, \\
(\partial_t u - \Delta u - f)(u - \varphi)(u - \psi) &= 0,
\end{align*}
$$

in $\Omega \times J$,

$$
(\partial_t u - \Delta u - f)(u - \varphi)(u - \psi) = 0,
$$

in $\Omega \times J$,

$$
u = g, \quad \text{on } \Gamma \times J,$$

where $u(\cdot, t = 0) = u_0$ is given. Its numerical approximation in the algebraic form reads

$$
\begin{align*}
Bu_{ij}^n &\geq r_{ij}^{n-1} \quad \text{when } u_{ij}^n = \varphi_{ij}, \\
Bu_{ij}^n &\leq r_{ij}^{n-1} \quad \text{when } u_{ij}^n = \psi_{ij}, \\
\varphi_{ij} &\leq u_{ij}^n \leq \psi_{ij}, \\
(Bu_{ij}^n - r_{ij}^{n-1})(u_{ij}^n - \varphi_{ij})(u_{ij}^n - \psi_{ij}) &= 0,
\end{align*}
$$

in $\Omega_d^0 \times J_d$,

$$
u_{ij}^n = g_{ij}, \quad \text{on } \Gamma_d \times J_d,$$

where $Bu_{ij}^n$ and $r_{ij}^{n-1}$ are as defined as in (4.25). Then the double obstacle relaxation (DOR) procedure for each time step of (4.36) can be formulated as a simple modification of (4.28):
Algorithm $\mathcal{DOR}$:

for $k = 1, 2, \ldots$
  for $x_{ij} \in \Omega_d^0$
    (a) $u^{n,k*}_{ij} = \left[ u^{n,k-1}_{ij} + \mathcal{M}^{-1}(r^{n-1} - \mathcal{B}u^{n,k-1}_{ij}) \right]_{ij}$;
    (b) $u^{n,k}_{ij} = \min(\max(u^{n,k*}_{ij}, \varphi_{ij}), \psi_{ij})$;
  end
end

(4.37)

Theorem 4.2.3 Suppose that the point relaxation step in (4.37.a) is based on a regular splitting of $\mathcal{B} = \mathcal{M} - \mathcal{N}$. Then, the $\mathcal{DOR}$ is convergent and its limit satisfies (4.36).

The above theorem can be proved, following the same arguments introduced in the proof of Theorem 4.2.1.

4.3 Numerical Experiments

In this section, we present numerical experiments using the variable-$\theta$ method to verify its effectiveness and accuracy. The experiments are implemented in Matlab and carried out on a Desktop computer of an Intel i7-7700K 4.20GHz processor with 16GB RAM. For a comparison purpose, besides the implicit method and the CN method, we also implemented the semi-iterative Newton-type method (SINT) in [4], which has utilized the QMR solver with an optimized setting in Matlab. (It is a standard iterative solver for nonsymmetric systems; see [48].)

For the algebraic solver for the solution of each time level, the SOR with the near-optimal parameter in (4.34), calibrated with a small problem, is employed and the iteration
is stopped when the maximum difference of consecutive iterates becomes smaller than a tolerance $\varepsilon = 10^{-6}$:

$$\|u^{n,k} - u^{n,k-1}\|_{\infty} < \varepsilon. \quad (4.38)$$

Parabolic obstacle problems are initialized using the obstacle $\varphi$ and the boundary condition $g$ as follows:

$$u_0(x) = \begin{cases} 
\varphi(x) & \text{if } x \in \Omega_d, \\
g(x,0) & \text{if } x \in \Gamma_d.
\end{cases} \quad (4.39)$$

For error analysis, the computed solution is compared with either the analytic solution or an accurate solution $\hat{u}$ which is constructed using the CN method with a sufficiently fine time grid, which satisfies the maximum principle.

### 4.3.1 Heat equations in 2D with smooth data

We begin with the numerical solution of the heat equation of smooth data defined in the two-dimensional (2D) space. Consider

$$\begin{align*}
\partial_t u - \Delta u &= f(x,t), \quad (x,y,t) \in (0,1)^2 \times (0,T], \\
u(x,y,0) &= \sin \pi x \sin \pi y, \quad (x,y) \in [0,1]^2,
\end{align*} \quad (4.40)$$

where the source term $f$ and the boundary condition are set corresponding to the analytic solution given by

$$u(x,y,t) = e^{-2\pi t} \sin \pi x \sin \pi y, \quad (x,t) \in [0,1]^2 \times [0,T]. \quad (4.41)$$

In Figure 4.7, we present the numerical solution for the variable-$\theta$ method over the whole domain and the solutions of the three $\theta$-methods (implicit, CN, and variable-$\theta$) on the horizontal half cross line, $[0,0.5] \times \{0.5\}$, at $t = 0.125$, when $T = 0.25$, $\Delta x = \Delta y = 0.025$ ($n_x = n_y = 40$), and $\Delta t = 0.0125$ ($n_t = 20$). As the same as the 1D case for smooth
Numerical solutions to the 2D heat equation (4.40) at $t = 0.125$, when $T = 0.25$, $\Delta x = \Delta y = 0.025$ ($n_x = n_y = 40$), and $\Delta t = 0.0125$ ($n_t = 20$): (a) the solution of the variable-$\theta$ method over the whole domain and (b) the solutions of the three $\theta$-methods on the horizontal half cross line, $[0, 0.5] \times \{0.5\}$.

data in Section 4.1.4, we can see that the proposed method shows a similar performance to the CN method in accuracy for the 2D case of smooth data. In Figure 4.7(b), however, the solution of the implicit method shows a much larger difference from the analytic solution than other two methods, due to its first-order convergence in the temporal direction.

In Table 4.3, in order to investigate their accuracy quantitatively, we present the accumulated relative $L^2$-errors $E_2[0, T]$ and the elapsed time (CPU, in second) for the three methods solving (4.40). We choose $T = 0.25$ and three different levels of numerical grids are selected satisfying $\Delta x = \Delta y = 2\Delta t$.

As one can see from the table, expected orders of accuracy are achieved for the implicit method and the CN method, while the variable-$\theta$ method shows an accuracy order of 2.75 and a smaller error than the CN method on the finest grid. Such a unusual accuracy of
Table 4.3

The accumulated relative $L^2$-error $E_2[0, T]$ and the CPU (in second), solving (4.40), when $T = 0.25$ and $\Delta x = \Delta y = 2\Delta t$.

<table>
<thead>
<tr>
<th>$(n_t, n_x = n_y)$</th>
<th>Implicit (CPU)</th>
<th>CN (CPU)</th>
<th>Variable-$\theta$ (CPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(20, 40)</td>
<td>$8.34 \cdot 10^{-2}$ (0.066)</td>
<td>$2.96 \cdot 10^{-3}$ (0.068)</td>
<td>$6.76 \cdot 10^{-3}$ (0.080)</td>
</tr>
<tr>
<td>(40, 80)</td>
<td>$3.98 \cdot 10^{-2}$ (0.322)</td>
<td>$7.21 \cdot 10^{-4}$ (0.342)</td>
<td>$1.02 \cdot 10^{-4}$ (0.431)</td>
</tr>
<tr>
<td>(80, 160)</td>
<td>$1.95 \cdot 10^{-2}$ (3.707)</td>
<td>$1.78 \cdot 10^{-4}$ (4.062)</td>
<td>$1.49 \cdot 10^{-4}$ (4.728)</td>
</tr>
<tr>
<td>order</td>
<td>1.05</td>
<td>2.03</td>
<td>2.75</td>
</tr>
</tbody>
</table>

the proposed method is partially due to a tendency of improved accuracy for finer grids. That is, when time-space grids are refined (with the same ratio for all of $\Delta x$, $\Delta y$, and $\Delta t$), our method will allocate relatively less portions of the grid points for $\theta = 1$ (implicit grid points) than when the grid is coarse. However, when the size of the implicit grid points is appropriate, its smoothing property may help the algorithm result in an improved accuracy, as mentioned in Remark 4.1.6 for 1D cases. Thus, the unusual accuracy of the variable-$\theta$ method is not really unusual for smooth data. For an efficiency concern, we can see that the wobble process is not expensive computationally; on the finest grid in the table, the proposed method takes just 16% more CPU time than the CN method.
4.3.2 Parabolic obstacle problems in 1D

Now, we consider parabolic obstacles problems (2.11) defined on the unit interval \([0, 1]\), with one smooth obstacle and two nonsmooth obstacles:

\[
\varphi_1(x) = \begin{cases} 
100x^2, & \text{if } 0 \leq x \leq 0.25, \\
100x(1 - x) - 12.5, & \text{if } 0.25 \leq x \leq 0.5, \\
\varphi_1(1 - x), & \text{if } 0.5 \leq x \leq 1;
\end{cases}
\]

\[
\varphi_2(x) = \begin{cases} 
1, & \text{if } x = 0.5, \\
0, & \text{otherwise};
\end{cases} \quad \varphi_3(x) = \begin{cases} 
1, & \text{if } 0 \leq x < 0.5, \\
0, & \text{if } 0.5 \leq x \leq 1.
\end{cases}
\]

Figure 4.8

Evolution of numerical solutions for the obstacle problem associated with \(\varphi_3\) over \(0 \leq t \leq T = 0.1\), when \(\Delta x = 0.04\) \((n_x = 25)\), \(\Delta t = 0.01\) \((n_t = 10)\): (a) the implicit method, (b) the CN method, (c) the SINT method, and (d) the variable-\(\theta\) method.

Table 4.4 exhibits the accumulated relative \(L^2\)-error \(E_2[0, T]\) and the relative \(L^\infty\)-error \(E_\infty[T]\), for each of the implicit method, the CN method, the SINT method, and the variable-\(\theta\) method, when \(T = 0.1\), \(\Delta x = 0.04\) \((n_x = 25)\), and \(\Delta t = 0.01\) \((n_t = 10)\). In most cases, the variable-\(\theta\) method shows smallest errors and the SINT method performs very similarly to the implicit method in accuracy. Particularly when the obstacle is nonsmooth \((\varphi_2 \text{ and } \varphi_3)\), the variable-\(\theta\) method produces smallest errors for all cases; the
Table 4.4

$E_2[0, T]$ and $E_\infty[T]$ for the obstacle problem associated with $\varphi_i$, $i = 1, 2, 3$, in (4.42), when $T = 0.1$, $\Delta x = 0.04$ ($n_x = 25$), and $\Delta t = 0.01$ ($n_t = 10$).

| Obstacle | Implicit |  |  |     |     |     |     |  |  |
|----------|----------|-----------|------------|------|------|------|---|---|
|          | $E_2[0, T]$ | $E_\infty[T]$ | $E_2[0, T]$ | $E_\infty[T]$ | $E_2[0, T]$ | $E_\infty[T]$ | $E_2[0, T]$ | $E_\infty[T]$ |
| $\varphi_1$ | $6.87 \cdot 10^{-4}$ | $4.36 \cdot 10^{-4}$ | $2.38 \cdot 10^{-4}$ | $7.84 \cdot 10^{-4}$ | $8.33 \cdot 10^{-4}$ | $8.96 \cdot 10^{-4}$ | $2.43 \cdot 10^{-4}$ | $3.13 \cdot 10^{-5}$ |
| $\varphi_2$ | $5.05 \cdot 10^{-3}$ | $1.17 \cdot 10^{-2}$ | $7.71 \cdot 10^{-3}$ | $2.06 \cdot 10^{-2}$ | $5.05 \cdot 10^{-3}$ | $1.17 \cdot 10^{-2}$ | $3.00 \cdot 10^{-3}$ | $3.49 \cdot 10^{-3}$ |
| $\varphi_3$ | $2.14 \cdot 10^{-3}$ | $1.17 \cdot 10^{-2}$ | $3.11 \cdot 10^{-3}$ | $2.06 \cdot 10^{-2}$ | $2.14 \cdot 10^{-3}$ | $1.17 \cdot 10^{-2}$ | $1.19 \cdot 10^{-3}$ | $3.49 \cdot 10^{-3}$ |
CN method introduces largest errors, while the errors of the variable-\( \theta \) method are much smaller than those of the implicit method and the SINT method.

In Figure 4.8, we depict the evolution of numerical solutions of the four methods for the obstacle problem associated with one of nonsmooth obstacles \( \varphi_3 \), when \( T = 0.1 \), \( \Delta x = 0.04 \) (\( n_x = 25 \)), and \( \Delta t = 0.01 \) (\( n_t = 10 \)). As one can see from the figure, only the CN method has introduced spurious oscillations to its numerical solution.

### 4.3.3 Parabolic obstacle problems in 2D

Moving to parabolic obstacle problems in 2D, we consider the radially-symmetric obstacles \( \varphi_4, \varphi_5 : \Omega \to \mathbb{R} \), where \( \Omega = [-2, 2]^2 \), defined by

\[
\varphi_4(r) = \begin{cases} 
1 + \sqrt{1 - r^2}, & \text{if } r \leq 1, \\
0, & \text{otherwise,}
\end{cases}
\]

and

\[
\varphi_5(r) = \begin{cases} 
2 - \frac{\sqrt{r}}{2}, & \text{if } r < 1, \\
0, & \text{otherwise.}
\end{cases}
\]

Table 4.5 shows \( E_2[0, T] \), \( E_\infty[T] \), and CPU for the obstacle problems associated with \( \varphi_4 \) and \( \varphi_5 \), when \( T = 0.4 \) and \( \Delta x = \Delta y = 2\Delta t \). As expected, the variable-\( \theta \) method achieves the highest accuracy for all cases and tends to recover a second-order convergence, while the CN method shows the lowest accuracy and a lower convergence order than the implicit method. We can see again that the wobble process is not expensive computationally so that the resulting variable-\( \theta \) method takes slightly more CPU time than the CN method. Notice that the SINT method exhibits a similar accuracy to the implicit method; however, its CPU time is a few hundred times larger than the \( \theta \)-methods.
Table 4.5

$E_2[0, T], E_\infty[T]$, and CPU for the obstacle problems associated with $\varphi_4$ and $\varphi_5$, when $T = 0.4$ and $\Delta x = \Delta y = 2\Delta t$.

<table>
<thead>
<tr>
<th>$(n_1, n_x = n_y)$</th>
<th>Implicit $E_2[0, T]$</th>
<th>$E_\infty[T]$</th>
<th>CPU</th>
<th>CN $E_2[0, T]$</th>
<th>$E_\infty[T]$</th>
<th>CPU</th>
<th>SINT $E_2[0, T]$</th>
<th>$E_\infty[T]$</th>
<th>CPU</th>
<th>Variable-θ $E_2[0, T]$</th>
<th>$E_\infty[T]$</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(5, 25)$</td>
<td>1.66 · 10^{-2}</td>
<td>2.40 · 10^{-2}</td>
<td>0.003</td>
<td>2.25 · 10^{-2}</td>
<td>1.72 · 10^{-2}</td>
<td>0.004</td>
<td>1.75 · 10^{-2}</td>
<td>2.79 · 10^{-2}</td>
<td>0.084</td>
<td>1.20 · 10^{-4}</td>
<td>8.83 · 10^{-4}</td>
<td>0.004</td>
</tr>
<tr>
<td>$(10, 50)$</td>
<td>9.88 · 10^{-3}</td>
<td>1.24 · 10^{-2}</td>
<td>0.012</td>
<td>1.82 · 10^{-2}</td>
<td>1.25 · 10^{-2}</td>
<td>0.015</td>
<td>1.29 · 10^{-2}</td>
<td>2.21 · 10^{-2}</td>
<td>2.903</td>
<td>4.91 · 10^{-3}</td>
<td>2.12 · 10^{-3}</td>
<td>0.020</td>
</tr>
<tr>
<td>$(20, 100)$</td>
<td>5.70 · 10^{-3}</td>
<td>6.34 · 10^{-3}</td>
<td>0.150</td>
<td>1.34 · 10^{-2}</td>
<td>8.96 · 10^{-3}</td>
<td>0.112</td>
<td>8.86 · 10^{-3}</td>
<td>1.52 · 10^{-2}</td>
<td>199.6</td>
<td>2.05 · 10^{-3}</td>
<td>5.99 · 10^{-3}</td>
<td>0.168</td>
</tr>
</tbody>
</table>

$\varphi_5$

<table>
<thead>
<tr>
<th>$(n_1, n_x = n_y)$</th>
<th>Implicit $E_2[0, T]$</th>
<th>$E_\infty[T]$</th>
<th>CPU</th>
<th>CN $E_2[0, T]$</th>
<th>$E_\infty[T]$</th>
<th>CPU</th>
<th>SINT $E_2[0, T]$</th>
<th>$E_\infty[T]$</th>
<th>CPU</th>
<th>Variable-θ $E_2[0, T]$</th>
<th>$E_\infty[T]$</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(5, 25)$</td>
<td>1.78 · 10^{-2}</td>
<td>2.51 · 10^{-2}</td>
<td>0.003</td>
<td>2.46 · 10^{-2}</td>
<td>1.56 · 10^{-2}</td>
<td>0.004</td>
<td>1.78 · 10^{-2}</td>
<td>2.51 · 10^{-2}</td>
<td>0.084</td>
<td>1.31 · 10^{-4}</td>
<td>1.05 · 10^{-4}</td>
<td>0.004</td>
</tr>
<tr>
<td>$(10, 50)$</td>
<td>1.08 · 10^{-2}</td>
<td>1.24 · 10^{-2}</td>
<td>0.011</td>
<td>2.11 · 10^{-2}</td>
<td>8.87 · 10^{-3}</td>
<td>0.015</td>
<td>1.08 · 10^{-2}</td>
<td>1.24 · 10^{-2}</td>
<td>2.083</td>
<td>5.88 · 10^{-3}</td>
<td>2.06 · 10^{-3}</td>
<td>0.020</td>
</tr>
<tr>
<td>$(20, 100)$</td>
<td>8.27 · 10^{-3}</td>
<td>6.24 · 10^{-3}</td>
<td>0.153</td>
<td>1.63 · 10^{-2}</td>
<td>5.98 · 10^{-3}</td>
<td>0.114</td>
<td>6.27 · 10^{-3}</td>
<td>6.22 · 10^{-3}</td>
<td>59.93</td>
<td>2.34 · 10^{-3}</td>
<td>4.93 · 10^{-3}</td>
<td>0.170</td>
</tr>
</tbody>
</table>

$\varphi_5$
Numerical solutions of the four methods for the obstacle problem associated with $\varphi_4$ at $t = 0.2$, when $T = 0.4$, $\Delta x = \Delta y = 0.08$ ($n_x = n_y = 50$), and $\Delta t = 0.04$ ($n_t = 10$): (a) over the whole domain and (b) on the horizontal half cross line $[-2, 0] \times \{0\}$.

Numerical solutions for the obstacle problem associated with $\varphi_5$ at $t = 0.2$, when $T = 0.4$, $\Delta x = \Delta y = 0.08$ ($n_x = n_y = 50$), and $\Delta t = 0.04$ ($n_t = 10$): (a) the obstacle $\varphi_5$ (the solid body inside) and the numerical solution (transparent membrane) by the CN method and (b) the solutions of the four methods on the horizontal half cross line $[-2, 0] \times \{0\}$.
To examine characteristics of the methods in detail, we depict the numerical solutions of the radially-symmetric obstacle problems associated with \( \varphi_4 \) (Figure 4.9) and \( \varphi_5 \) (Figure 4.10) at \( t = 0.2 \), when \( T = 0.4, \Delta x = \Delta y = 0.08 (n_x = n_y = 50) \), and \( \Delta t = 0.04 (n_t = 10) \). For both cases, the CN method introduces spurious oscillations near the obstacle edges, while the variable-\( \theta \) method results in accurate solutions without spurious oscillations. From Figure 4.9(b) and Figure 4.10(b), the reader will find that the solutions of the SINT method look the same as those of the implicit method, as shown in Table 4.5. It should be noticed that in smooth regions, the solutions of the proposed method are between the CN solutions and the implicit solutions. The variable-\( \theta \) method tries to keep a second-order accuracy without introducing spurious oscillations.

Next, we consider an extremely nonsmooth “tent-shaped” obstacle \([4]\) \( \varphi_6 : \Omega \to \mathbb{R} \), \( \Omega = [-2, 2] \times [-1, 1] \), defined by

\[
\varphi_6(x, y) = \begin{cases} 
\min(1 - |x|, 2 - |y|), & \text{if } (x, y) \in \Omega^0, \\
0.5, & \text{if } (x, y) \in \Gamma.
\end{cases}
\]  

(4.45)

In this example, we set the boundary condition \( g(x, t) = 0.5 \) on \( \Gamma \times J \); see Figure 4.11(a) for the obstacle.

In Table 4.6, we compare performances of the variable-\( \theta \) method with the SINT method and other \( \theta \)-methods, applied to the obstacle problem with \( \varphi_6 \). It contains the accumulated \( L^2 \)-error \( E_2[0, T] \), the \( L^\infty \)-error \( E_\infty[T] \), and the CPU time for the four methods, when \( T = 0.4, \Delta x = 2\Delta t, \) and \( \Delta y = \Delta t \). As we have seen earlier, the implicit method and the SINT method show a similar accuracy and the CN method reveals the worst accuracy with the \( L^\infty \)-error \( E_\infty[T] \) being not improved as the mesh is refined. On the other hand,
Table 4.6

$E_2[0, T]$, $E_\infty[T]$, and CPU for the obstacle problem associated with $\varphi_6$, when $T = 0.4$ and $\Delta x = 2\Delta x = 2\Delta t$.

<table>
<thead>
<tr>
<th>$(n_t, n_x = n_y)$</th>
<th>Implicit</th>
<th>CN</th>
<th>SINT</th>
<th>Variable-$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(10, 50)$</td>
<td>$1.13 \cdot 10^{-2}$</td>
<td>$8.60 \cdot 10^{-3}$</td>
<td>0.003</td>
<td>$2.93 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>$(20, 100)$</td>
<td>$7.56 \cdot 10^{-3}$</td>
<td>$3.79 \cdot 10^{-3}$</td>
<td>0.018</td>
<td>$2.41 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>order</td>
<td>0.64</td>
<td>1.11</td>
<td>0.32</td>
<td>-0.01</td>
</tr>
</tbody>
</table>
The obstacle $\varphi_6$ and numerical solutions of the obstacle problem associated to $\varphi_6$, when $t = T = 0.4$, $\Delta x = 0.08$, $\Delta y = 0.04$ ($n_x = n_y = 50$), and $\Delta t = 0.04$ ($n_t = 10$): (a) the obstacle $\varphi_6$, (b) the CN solution, (c) the SINT solution, and (d) the variable-\(\theta\) solution.

The variable-\(\theta\) method produces accurate solutions of a reasonable convergence property. Roughly speaking, for this extremely nonsmooth obstacle problem, the proposed method can achieve convergence of order 1 for the accumulated $L^2$-error $E_2[0,T]$ and of order 1.5 for the $L^\infty$-error $E_\infty[T]$ at the last moment (when the solution becomes smooth in most regions). Note that for the finest case in the table, the variable-\(\theta\) method takes a slightly smaller CPU time than the implicit method, although it performs the wobble process; which implies that the algebraic solver (SOR) converges faster for the proposed method than the implicit method.

Figures 4.11(b)-(d) depict the numerical solutions of the obstacle problem with $\varphi_6$ at $t = T = 0.4$ for the CN method, the SINT method, and the variable-\(\theta\) method. Due to an extremely nonsmooth obstacle, the CN method (even at $t = 0.4$) is not yet able to resolve the discontinuities introduced from the obstacle and the initial-boundary conditions, unlike the other methods.
Remark 4.3.1 The error analyses in Tables 4.4–4.6 are based on accurate numerical solutions that are constructed by the CN method with sufficiently fine time grids satisfying the maximum principle. Although the CN solutions are not the same as the analytic solutions, we believe that they are accurate enough to be used as desired solutions.

4.3.4 Double obstacle problems

The parabolic obstacle problems have been widely applied particularly to the literature of Mechanics and Finance problems, for example, elastic-plastic torsion problems and the American option pricing. We implemented the three $\theta$-methods (the implicit method, the CN method, and the variable-$\theta$ method) for those problems.

For a general (double) obstacle problem, we begin with the elastic-plastic torsion problem in [62], which admits a nonsmooth obstacle. Let $\Omega = [0, 1]^2$. Then the problem consists of two obstacles, $\varphi_7, \psi_7 : \Omega \to \mathbb{R}$,

$$\varphi_7(x, y) = -\text{dist}((x, y), \partial\Omega), \quad \psi_7(x, y) \equiv 0.2,$$

and the static force $v : \Omega \times J \to \mathbb{R}$ defined by

$$v(x, y, t) = \begin{cases} 
300, & \text{if } (x, y, t) \in S = \{(x, y) : |x-y| \leq 0.1 \land x \leq 0.3\} \times J, \\
-70e^y g(x), & \text{if } x \leq 1 - y, \ (x, y) \notin S \text{ and } t \in J, \\
15e^y g(x), & \text{if } x > 1 - y \ (x, y) \notin S \text{ and } t \in J, 
\end{cases}$$

(4.47)

where

$$g(x) = \begin{cases} 
6x, & \text{if } 0 \leq x \leq 1/6, \\
2(1 - 3x), & \text{if } 1/6 < x \leq 1/3, \\
6(x - 1/3), & \text{if } 1/3 < x \leq 1/2, \\
2(1 - 3(x - 1/3)), & \text{if } 1/2 < x \leq 2/3, \\
6(x - 2/3), & \text{if } 2/3 < x \leq 5/6, \\
2(1 - 3(x - 2/3)), & \text{if } 5/6 < x \leq 1. 
\end{cases}$$

(4.48)
The double parabolic obstacle problem to be considered here is of the form (4.35) with the choices $\varphi = \varphi_7$, $\psi = \psi_7$, and $f = v$. See Figure 4.12, which depicts the obstacles, the force, and an example solution produced by the variable-$\theta$ method, when $T = 0.05$, $\Delta x = \Delta y = 0.01 (n_x = n_y = 100)$, and $\Delta t = 0.01 (n_t = 5)$.

Figure 4.13 presents the misfits ($u_{\text{computed}} - u_{\text{desired}}$) at $t = 0.05$ for the three $\theta$-methods (the implicit method, the CN method, and the variable-$\theta$ method), where $u_{\text{desired}}$ is the CN solution obtained with a sufficiently fine time grid. For this double obstacle problem, we can also see that the CN method introduces spurious oscillations to its numerical solution, as shown in Figure 4.13(b), and the variable-$\theta$ method achieves the smallest error as in Figure 4.13(c). (It should be noticed that Figures 4.13(a)-(c) are in different scales.)

Figure 4.14 depicts the upper contact sets (in yellow, brightest) and lower contact sets (in blue, darkest) for the three $\theta$-methods, along with the desired contact set obtained by the CN method of a sufficiently fine time grid. One can easily see that the contact sets obtained by the variable-$\theta$ (Figure 4.14(d)) are closest to the desired ones (Figure 4.14(a)), while the contact sets from the CN method in Figure 4.14(c) show certain deformed shapes and unusual line segments caused by spurious oscillations. To examine the differences between contact sets precisely, we measure the pixel differences from the desired ones in Figure 4.14(a) and depict the results in Figure 4.15. The number of pixels is counted for each method. Although the CN method introduces spurious oscillations, its pixel difference in the contact sets turns out to be 90, which is smaller than that of the implicit method (131); the implicit method tends to smoothen steep changes excessively. However, the
Elastic-plastic torsion problem with double obstacles: (a) the obstacles \( \varphi_7 \) and \( \psi_7 \), (b) the force \( v \), and (c) the solution of the variable-\( \theta \) method, when \( T = 0.05 \), \( \Delta x = \Delta y = 0.01 \) (\( n_x = n_y = 100 \)), and \( \Delta t = 0.01 \) (\( n_t = 5 \)).

Misfits \((u_{\text{computed}} - u_{\text{desired}})\) of the double obstacle problem associated with \( \varphi_7 \) and \( \psi_7 \), when \( t = T = 0.05 \), \( \Delta x = \Delta y = 0.01 \) (\( n_x = n_y = 100 \)), and \( \Delta t = 0.01 \) (\( n_t = 5 \)): (a) the implicit method, (b) the CN method, and (c) the variable-\( \theta \) method.
Figure 4.14

Contact sets associated with the elastic-plastic torsion problem at $t = 0.05$, when $\Delta x = \Delta y = 0.01$ ($n_x = n_y = 100$) and $\Delta t = 0.01$ ($n_t = 5$): (a) a desired contact set, (b) by the implicit method, (c) by the CN method, and (d) by the variable-$\theta$ method.
Pixel differences between the desired contact set and contact sets by the three $\theta$-methods in Figure 4.14: (a) the implicit method (131), (b) the CN method (90), and (c) the variable-$\theta$ method (55).

Contact sets of the variable-$\theta$ method differ only at 55 pixels from the desired contact sets (out of 10,000 pixels), which is the smallest among all methods.

### 4.3.5 Option pricing

Finally, we consider the American put option problem and the Black-Scholes equation on $\Omega = \mathbb{R}_{>0} \times (0,T]$:

$$
\begin{align*}
\partial_t V + \mathcal{L}V &\leq 0 \\
V - (K - s)^+ &\geq 0 \\
(\partial_t V + \mathcal{L}V)(V - (K - s)^+) &= 0 \\
\lim_{s \to 0} V(s,t) &= K \\
\lim_{s \to \infty} V(s,t) &= 0 \\
V(s,T) &= V_T = (K - s)^+, \quad s \in \mathbb{R}_{>0},
\end{align*}
$$

(4.49)

where $(\cdot)^+ = \max(\cdot, 0)$ and the elliptic operator $\mathcal{L}$ is given by

$$
\mathcal{L}V = \frac{1}{2} \sigma^2 s^2 \frac{\partial^2 V}{\partial s^2} + rs \frac{\partial V}{\partial s} - rV.
$$

(4.50)
By introducing change of variables, we can reformulate the problem (4.49) to be a parabolic obstacle problem. Let

\[ x := \ln s + \left( r - \frac{\sigma^2}{2} \right) (T - t), \quad \tau := \frac{\sigma^2}{2} (T - t), \]

\[ u(x, \tau) := e^{-r(T-\frac{\tau}{\sigma^2})} V \left( e^{x-(\frac{2\tau}{\sigma^2}-1)\tau}, T - \frac{2\tau}{\sigma^2} \right), \tag{4.51} \]

where \( x \in \mathbb{R} \) and \( \tau \in \left[ 0, \frac{\sigma^2}{2} T \right] \). Then, some algebraic manipulation of (4.49) reads

\[
\begin{aligned}
&\partial_\tau u + \partial_{xx} u \leq 0 \\
&y - \varphi_8 \geq 0 \\
&(\partial_\tau u + \partial_{xx} u) (u - \varphi_8) = 0 \\
&\lim_{x \to -\infty} u(x, \tau) = Ke^{-r(T-\frac{\tau}{\sigma^2})} \\
&\lim_{x \to \infty} u(x, \tau) = 0, \\
&u(x, 0) = u_0 = \varphi_8(x, 0), \quad x \in \mathbb{R},
\end{aligned}
\tag{4.52}
\]

where

\[ \varphi_8(x, \tau) = e^{-r(T-\frac{\tau}{\sigma^2})} \left( K - e^{x-(\frac{2\tau}{\sigma^2}-1)\tau} \right)^+. \tag{4.53} \]

Now, we select the following parameters for the option pricing problem, (4.52)–(4.53):

\( T = 10, K = 45, r = 0.05, \) and \( \sigma = 0.45 \) \( (\frac{\sigma^2}{2} T = 1.0125) \). Since the computation must be carried out on a finite interval in the \( x \)-direction, we choose \([-8, 4.5]\) as our computation domain.

In Figure 4.16, we depict the evolution of solutions for the American put option pricing (4.49), when \( T = 10 \) \( (\frac{\sigma^2}{2} T = 1.0125) \), \( \Delta x = 0.125 \) \( (n_x = 100) \), and \( \Delta \tau = 0.10125 \) \( (n_\tau = 10) \). The CN method again introduces spurious oscillations to its numerical solution in the beginning, due to the nonsmooth obstacle \( \varphi_8 \) (4.53), while the proposed method resolves the solution without oscillations. The \( L^\infty \)-error at the last moment of \( \tau \), measured
Figure 4.16

The evolution of numerical solutions for the American put option pricing (4.49), when $T = 10 \left( \sigma^2 T = 1.0125 \right)$, $\Delta x = 0.125 \left( n_x = 100 \right)$, and $\Delta \tau = 0.10125 \left( n_\tau = 10 \right)$: (a) the CN method and (b) the variable-$\theta$ method.

from a desired solution, reads $5.814 \cdot 10^{-2}$ and $1.520 \cdot 10^{-2}$ for the CN method and the variable-$\theta$ method, respectively.
CHAPTER 5
REACTION-DIFFUSION PROBLEM IN BIOLOGY

5.1 The Variable-\(\theta\) Method for Two-Component Nonlinear Problems

This section introduces an effective time-stepping procedure for the numerical solution of the two-component RD problems (2.26) incorporating with the variable-\(\theta\) method.

5.1.1 Linearization through extrapolation

Once the spatial derivatives are approximated by second-order finite difference schemes as in Section 2.2.1, the semi-discrete problem for (2.26) is formulated as

\[
\frac{\partial u}{\partial t} + \mathcal{D}Au = f(u), \quad t \in (0, T],
\]
\[
u = u^0, \quad t = 0.
\]

(5.1)

Let numerical solutions be obtained up to the \(n\)-th time level, \(n > 0\). For the numerical solution in the \((n + 1)\)-th level, we first extrapolate numerical solutions in the two previous levels to approximate the solution at the mid-point \(t^{n+1/2}\):

\[
\tilde{u}^{n+1/2} := \frac{3}{2}u^n - \frac{1}{2}u^{n-1}.
\]

(5.2)

(See [17] for details of second-order extrapolation schemes for \(n \geq 0\).) Then, the \(\theta\)-method for the two-component RD problem reads:

\[
\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} + \mathcal{D}A[\theta u_{ij}^{n+1} + (I - \theta)u_{ij}^n] = f(\tilde{u}^{n+1/2})_{ij},
\]

(5.3)
where \( \mathbf{u} = [u_1, u_2]^T, \mathcal{D} = \text{diag}[D_1, D_2], \) and \( \mathbf{\theta} = \text{diag}[\theta_1, \theta_2]. \)

The linearized problem (5.3) can be resolved by solving for two separate components \( \mathbf{u}^{n+1} = [u_1^{n+1}, u_2^{n+1}]^T. \) Each component in (5.3) can be formulated as follows.

\[
\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} + \mathcal{D}_A[\theta u_{ij}^{n+1} + (1 - \theta) u_{ij}^n] = f_{ij}^{n+1/2},
\]

where \( u, \mathcal{D}, \) and \( \theta \) denote respectively \( u_k, D_k, \) and \( \theta_k, \) for \( k = 1 \) or \( 2, \) and \( f^{n+1/2} \) is a known source term. The \( \theta \)-method (5.4) can be rewritten in a vector form as

\[
(I + \Delta t \mathcal{D}_A)\mathbf{u}^{n+1} = [I - (1 - \theta) \Delta t \mathcal{D}_A] \mathbf{u}^n + \Delta t f^{n+1/2}.
\]  

(5.5)

Having the wobble set as in Section 4.1.1, the parameter \( \theta \) for the computation of \( \mathbf{u}^{n+1} \) can be assigned pointwisely

\[
\theta_{ij}^{n+1} := \theta(x_{ij}, t^{n+1}) = \begin{cases} 
1, & \text{if } x_{ij} \in \mathcal{W}^n, \\
1/2, & \text{otherwise}. 
\end{cases}
\]  

(5.6)

Thus, the variable-\( \theta \) method for (5.4) can be formulated as

\[
\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} + \mathcal{D}_A[\theta_{ij}^{n+1} u_{ij}^{n+1} + (1 - \theta_{ij}^{n+1}) u_{ij}^n] = f_{ij}^{n+1/2},
\]

or, in a vector form after grouping variables,

\[
(I + \theta^{n+1} \Delta t \mathcal{D}_A)\mathbf{u}^{n+1} = [I - (1 - \theta^{n+1}) \Delta t \mathcal{D}_A] \mathbf{u}^n + \Delta t f^{n+1/2}.
\]  

(5.8)

The algebraic system in (5.8) will be solved by applying the SOR method, with its initial value at the time level \( t^{n+1} \) being set as

\[
\mathbf{u}^{n+1,0} = 2\mathbf{u}^n - \mathbf{u}^{n-1}.
\]  

(5.9)

In particular, SOR converges quite fast for an appropriate choice of the relaxation parameter \( \omega. \)
In the following, we will consider how to tune the optimal relaxation parameter $\hat{\omega}$ for SOR.

5.1.2 The optimal relaxation parameter $\hat{\omega}$

In this section, we will try to find a relaxation parameter which is *heuristically optimal*. Let us begin with the 2D algebraic system of (5.5) with $\theta = 1/2$:

$$L \mathbf{u} = \mathbf{r} \in \mathbb{R}^{m \times m}, \quad (5.10)$$

where $L = I + \frac{\Delta t}{2} DA$, $\mathbf{r} = (I - \frac{\Delta t}{2} DA) \mathbf{u} + \Delta t^{n+1/2}$, and $m$ is the dimension of the algebraic system. It is known that the optimal relaxation parameter for the SOR method can be determined as [58, § 4.3]

$$\hat{\omega} = \frac{2}{1 + \sqrt{1 - \rho(T_J)^2}}, \quad (5.11)$$

where $\rho(T_J)$ is the spectral radius of the Jacobi iteration matrix $T_J$.

For simplicity, assume that the problem is defined on the unit square with a Dirichlet boundary condition. We further assume that the domain is partitioned into $N \times N$ grids so that $h = \Delta x = \Delta y = 1/N$. Then the eigenvalues of the second-order 5-point FD coefficient matrix $A$ read [58, § 6.5]

$$\lambda_{k,\ell}(A) = \frac{1}{h^2} \left[ 4 - 2 \cos \left( \frac{k\pi}{N} \right) - 2 \cos \left( \frac{\ell\pi}{N} \right) \right], \quad 1 \leq k, \ell \leq N - 1, \quad (5.12)$$

and therefore the eigenvalues of $L$ can be formulated as

$$\lambda_{k,\ell}(L) = 1 + \frac{\Delta t}{2} D \lambda_{k,\ell}(A) = 1 + \frac{D \Delta t}{h^2} \left[ 2 - \cos \left( \frac{k\pi}{N} \right) - \cos \left( \frac{\ell\pi}{N} \right) \right], \quad (5.13)$$
for $1 \leq k, \ell \leq N - 1$. Note that the diagonal element of $L$ is

$$p := 1 + \frac{\Delta t}{2} D \frac{4}{h^2} = 1 + 2 \frac{D \Delta t}{h^2}. \quad (5.14)$$

So the eigenvalues of the Jacobi iteration matrix $T_J$ are given as

$$\lambda_{k,\ell}(T_J) = \frac{p - \lambda_{k,\ell}(L)}{p} = \frac{D \Delta t}{h^2} \left[ \cos \left( \frac{k \pi}{N} \right) + \cos \left( \frac{\ell \pi}{N} \right) \right] / \left( 1 + 2 \frac{D \Delta t}{h^2} \right). \quad (5.15)$$

In order to find the maximum of $|\lambda_{k,\ell}(T_J)|$, we first obtain

$$\max_k \left| \cos \left( \frac{k \pi}{N} \right) \right| = \cos \left( \frac{\pi}{N} \right) = 1 - c_1 h^2, \quad (5.16)$$

for some $c_1 > 0$. Here we have used $h = 1/N$ and the approximation $\cos(x) \approx 1 - x^2/2$.

Now the spectral radius of $T_J$ reads

$$\rho(T_J) = \max_{k,\ell} |\lambda_{k,\ell}(T_J)| = (1 - c_1 h^2) \left( 1 + \frac{1}{2} \frac{h^2}{D \Delta t} \right)^{-1}. \quad (5.17)$$

Assuming that $c_1 h^2 < h^2/(2D \Delta t) < 1$, we finally obtain

$$\rho(T_J) = 1 - c_2 \frac{h^2}{D \Delta t}, \quad (5.18)$$

for some $c_2 > 0$.

It follows from (5.11) and (5.18) that the optimal SOR parameter $\tilde{\omega}_{\Delta t, h}$, corresponding to the spatial grid size $h$ and the time step size $\Delta t$, can be written as

$$\tilde{\omega}_{h, \Delta t} = \frac{2}{1 + c_0 \frac{h}{\sqrt{D \Delta t}}}, \quad (5.19)$$

for some $c_0 > 0$. The constant $c_0$ can be found experimentally from a selected set of $(\Delta t, h)$, as summarized in the following.

\[
\begin{align*}
\text{(a) } & \text{Determine } \tilde{\omega}_{\Delta t_0, h_0} \text{ for prescribed grid sizes } (\Delta t_0, h_0), \text{ heuristically.} \\
\text{(b) } & \text{Solve (5.19) for } c_0 : \\
& c_0 = \frac{\sqrt{D \Delta t_0}}{h_0} \left( \tilde{\omega}_{\Delta t_0, h_0} - 1 \right). \quad (5.20)
\end{align*}
\]
Once $c_0$ is estimated as in (5.20), the parameter $\hat{\omega}_{\Delta t, h}$ in (3.27) is near-optimal for various choices of $(\Delta t, h)$.

### 5.2 Numerical Experiments

In this section, we present numerical experiments which show effectiveness of the variable-$\theta$ method. The algorithms are implemented in MATLAB and carried out on a desktop computer of Intel Xeon CPU E5-1620 3.60GHz processor.

To solve the algebraic system at each time level, the SOR method is employed with the near-optimal parameter $\hat{\omega}$ calculated as in (5.19), with $c_0$ being estimated with a small grid problem. The SOR iteration is stopped when the maximum difference of consecutive iterates becomes smaller than a tolerance $\varepsilon = 10^{-6}$:

$$\|u^{n,k} - u^{n,k-1}\|_{\infty} < \varepsilon.$$  \hfill (5.21)

The $L^\infty$-error $E_{\infty}[t^n]$, measured at $t = t^n$, is defined as follows:

$$E_{\infty}[t^n] := \|u^n - \hat{u}(t^n)\|_{\infty},$$  \hfill (5.22)

where $\hat{u}$ is the exact solution.

#### 5.2.1 One-component reaction-diffusion system

First, consider a nonlinear RD problem of the form

$$\frac{\partial u}{\partial t} - u_{xx} = (1 + u)^2, \quad (x, t) \in (-1, 1) \times [0, T],$$  \hfill (5.23)

with the boundary and initial values given as in (2.22) and (2.23).
Figure 5.1

Propagation of the numerical solutions for (5.23): (a) the implicit method, (b) the CN method, and (c) the variable-\(\theta\) method, for \(0 \leq t \leq T = 0.5\), when \(\Delta t = 0.02\) and \(\Delta x = 0.04\).

Figure 5.1 presents the numerical solutions evolved by the implicit method, the CN method, and the variable-\(\theta\) method, when \(\Delta t = 0.02\) and \(\Delta x = 0.04\) (The mesh is the same as the one selected in Figure 2.2). Similar to the linear problem in Figure 2.2, spurious oscillations are introduced into the numerical solution of the nonlinear problem by the CN method only. It should be noticed that spurious oscillations of the CN method are damped out faster for the nonlinear problem than the linear problem, which is due to the reaction kinetic term \(f(u) = (1 + u)^2\). For the nonlinear problem, it looks at a glance that the oscillations at early time steps do not affect the solution at later steps much. This observation explains a partial reason that the second-order CN method has been popular for the numerical solution of PDEs in mathematical biology. However, for other applications, such spurious oscillations at early moments may alter the numerical solution significantly so as for the CN method to be unstable; see Figure 5.7 below. It is important to develop an effective algorithm which can suppress spurious oscillations for convenient choices of algorithm parameters; the variable-\(\theta\) method is effective and stable.
5.2.2 Two-component nonlinear reaction-diffusion systems

Two-component RD systems enable to explain a much wider range of phenomena than their one-component counterparts. Many two-component models have been developed and numerically verified for dynamical patterning behaviors in biology and chemistry. In this subsection, we consider two-component models interested in the literature of biology and chemistry, to verify effectiveness of the variable-θ method.

5.2.2.1 Gray-Scott model in 1D

We apply the numerical methods for the numerical solution of the Gray-Scott model [21, 22] defined as (2.26) associated with the following reaction kinetics

\[
f(u) = [F(1 - u_1) - u_1u_2^2, u_1u_2^2 - (F + k)u_2]^T
\]  

(5.24)

for any constants \(F\) and \(k\). Let \(\Omega = (0, 1)\). We assign two sets of model constants and initial and boundary conditions as follows [65].

(a) \(D = [10^{-4}, 10^{-6}]^T, \ F = 0.035, \ k = 0.049\),

(b) \(u_1(x, 0) = 1 - \frac{1}{2} \sin^{100}(\pi x), \ u_2(x, 0) = \frac{1}{4} \sin^{100}(\pi x), \ x \in (0, 1)\), \ (5.25)

(c) \(u_1(0, t) = u_1(1, t) = 1, \ u_2(0, t) = u_2(1, t) = 0, \ t \in [0, T]\),

where (5.25.b) is a mid-pulse initial condition, and

(a) \(D = [10^{-4}, 5 \cdot 10^{-5}]^T, \ F = 0.025, \ k = 0.0544\),

(b) \(u_1(x, 0) = 1 - \frac{1}{2} \cos^{100}\left(\frac{\pi x}{2}\right), \ u_2(x, 0) = \frac{1}{4} \cos^{100}\left(\frac{\pi x}{2}\right), \ x \in (0, 1)\), \ (5.26)

(c) \(\frac{\partial u_1}{\partial x}(0, t) = \frac{\partial u_1}{\partial x}(1, t) = 0, \ \frac{\partial u_2}{\partial x}(0, t) = \frac{\partial u_2}{\partial x}(1, t) = 0, \ t \in [0, T]\),

where (5.26.b) is a left-pulse initial condition.

In Figure 5.2, we present the propagation of the numerical solution of \(u_2\) by the variable-θ method associated with (5.25) at the grid sizes \(\Delta t = 0.01\) and \(\Delta x = 0.004\) over
Propagation of the numerical solution of $u_2$ for the 1D Gray-Scott model by the variable-$\theta$ method: (a) the wave-splitting (self-replication of the pulse) and (b) its aerial view over $0 \leq t \leq T = 2000$, when $\Delta t = 0.01$ and $\Delta x = 0.004$.

$0 \leq t \leq T = 2000$. The initial mid-pulse splits in early moments to travel in both directions (self-replication of the pulse). As each of the pulses travels, it becomes thicker (bigger) up to a certain width and begins to replicate itself recursively.

Figure 5.3 depicts the propagation of the numerical solution of $u_2$ by the variable-$\theta$ method associated with (5.26) over $0 \leq t \leq T = 5000$, at the same resolution as in Figure 5.2. One can clearly observe a traveling pulse which begins from the left edge point and reflects whenever it hits the boundary, due to the no-flux boundary condition (5.26.c).

To investigate bifurcation in RD systems and our method numerical accuracy, we present numerical solutions of the wave-splitting problem (5.25) obtained with various spatial and temporal grid sizes, as shown in Figure 5.4. The image $I_{k\ell}$ represents the numerical solution obtained with the mesh resolution $(\Delta t, \Delta x) = (10^{-k}, 0.01/2^{\ell-1})$. For example, the image $I_{23}$ is associated with the mesh resolution $(\Delta t, \Delta x) = (1/100, 1/400)$. One can
Propagation of the numerical solution of $u_2$ for the 1D Gray-Scott model by the variable-$\theta$ method: (a) the pulse traveling and reflecting and (b) its aerial view over $0 \leq t \leq T = 5000$, when $\Delta t = 0.01$ and $\Delta x = 0.004$.

easily point out from the images that the spatial resolution alters the numerical solution dramatically even with halved spatial grid sizes (compare the images horizontally), while the temporal resolution affects little the numerical solution even with one-order smaller temporal step sizes (compare them vertically).

The main reason for such a sensitivity to the spatial resolution is that the RD does not have as much time before growing to reach the margins of the mesh in low spatial resolutions (of large $\Delta x$’s). When this happens, the RD pattern typically deteriorates and it does not travel in an appropriate speed nor reach a condition to replicate itself on time; see [49, § 4.2] for similar observations. We summarize the experiments with the Gray-Scott model in 1D as follows.
The wave-splitting (self-replication of the pulse) by the variable-θ method over $0 \leq t \leq 2000$ with various $(\Delta t, \Delta x)$. The image $I_{k\ell}$ represents the numerical solution obtained with the mesh resolution $(\Delta t, \Delta x) = (10^{-k}, 0.01/2^{\ell-1})$. 

Figure 5.4
• Accuracy of the numerical solution is much more sensitive to the spatial mesh resolution than the temporal one.

• Thus it is crucial to set a high spatial resolution (small $\Delta x$’s) for a desirable accuracy.

• As the spatial resolution becomes higher, the CN method may more likely produce spurious oscillations, while the variable-$\theta$ method results in stable solutions.

### 5.2.2.2 Gray-Scott model in 2D

Note that the two-component Gray-Scott model is formulated as in (2.26) with the reaction kinetics $f(u)$ given in (5.24). We choose problem coefficients as follows [17].

$$\Omega = (-1, 1) \times (-1, 1), \quad D = [0.001, 0.001]^T, \quad F = 1, \quad k = 0. \quad (5.27)$$

For the purpose of error analysis, we select a smooth solution $\hat{u} = [\hat{u}_1, \hat{u}_2]$ defined as

$$\begin{align*}
\hat{u}_1(x, y, t) &= \cos(2t) \cos(2\pi x) \cos(\pi y), \\
\hat{u}_2(x, y, t) &= \cos(2t) \cos(\pi x) \cos(2\pi y),
\end{align*} \quad (5.28)$$

and replace the reaction kinetics $f(u)$ with $f_\hat{u}(u)$:

$$f_\hat{u}(u) := f(u) + \frac{\partial \hat{u}}{\partial t} - D\Delta \hat{u} - f(\hat{u}). \quad (5.29)$$

Then $\hat{u} = [\hat{u}_1, \hat{u}_2]$ in (5.28) would be the exact solution of $\partial u / \partial t - D \Delta u - f(u) = f_\hat{u}(u)$ with the initial condition $u^0 = \hat{u}(x, y, 0)$.

Table 5.1 summarizes the $L^\infty$-error $E_\infty[T]$ with $T = 1.0$ and the elapsed time (CPU) for the implicit, CN, variable-$\theta$ methods for three different meshes refined by a factor of 2 in both spatial and temporal directions with $\Delta x = \Delta y$. Since the solution including the initial condition is smooth over the entire time interval $[0, T = 1]$, the CN method
Table 5.1

$L^\infty$-error $E_\infty[T]$ and the elapsed time (CPU) for the numerical solution of the Gray-Scott model in the 2D space at $T = 1.0$.

<table>
<thead>
<tr>
<th>$\varepsilon = 10^{-6}$</th>
<th>Implicit</th>
<th>CN</th>
<th>Variable-(\theta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\Delta t, \Delta x = \Delta y)$</td>
<td>$E_\infty[T]$</td>
<td>CPU</td>
<td>$E_\infty[T]$</td>
</tr>
<tr>
<td>$(0.05, 0.1)$</td>
<td>$4.3 \cdot 10^{-2}$</td>
<td>$0.036s$</td>
<td>$9.6 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>$(0.025, 0.05)$</td>
<td>$2.0 \cdot 10^{-2}$</td>
<td>$0.225s$</td>
<td>$2.5 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>$(0.0125, 0.025)$</td>
<td>$8.0 \cdot 10^{-3}$</td>
<td>$1.629s$</td>
<td>$6.4 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>conv.order</td>
<td>1.2</td>
<td>1.9</td>
<td>1.9</td>
</tr>
</tbody>
</table>

102
introduces no spurious oscillations into its numerical solution and proves a second-order accuracy for the two-component Gray-Scott model in the 2D space. One should notice that the variable-\( \theta \) has also proved its second-order accuracy, the same as the CN method. On the other hand, the implicit method involves considerable errors due to its first-order convergence in temporal direction. Figure 5.5 shows the numerical solution the variable-\( \theta \) method and its error at \( T = 1.0 \) for the Gray-Scott model, when \( \Delta t = 0.025 \) and \( \Delta x = \Delta y = 0.05 \).

For all the three methods, the algebraic system is solved by the SOR method with its optimal relaxation parameter being calibrated from the lowest resolution, \((\Delta t, \Delta x = \Delta y) = (0.05, 0.1)\). That is, the constant \( c_0 \) in (5.20) is evaluated using the experimentally optimal \( \hat{\omega}_{\Delta t_0, h_0} \) with \((\Delta t_0, h_0) = (0.05, 0.1)\) and then (5.19) is utilized to estimate \( \hat{\omega}_{\Delta t, h} \) for other grid sizes \((\Delta t, h = \Delta x = \Delta y)\). With the near-optimal parameter and an effective initialization scheme in (5.9), for both the CN and variable-\( \theta \) procedures, the SOR method has converged in 5.9 iterations in average for solving the two algebraic systems (for \( u_1 \) and \( u_2 \)) in a time level. SOR is comparable with ADI in efficiency when the parameter is set optimal and the initialization is carried out accurately; SOR has proven its efficiency for the numerical solution of elliptic obstacle problems [33]. For the Gray-Scott model in 2D, the variable-\( \theta \) method becomes about a third more expensive computationally than the CN method, due to the wobble set processing. We summarize the experiments with the Gray-Scott model in 2D as follows.
Figure 5.5

The numerical solution of the variable-θ method and its error at $T = 1.0$ for the Gray-Scott model, when $\Delta t = 0.025$ and $\Delta x = \Delta y = 0.05$: (a) $u_1$, (b) $u_2$, (c) $e_1 = u_1 - \hat{u}_1$, and (d) $e_2 = u_2 - \hat{u}_2$. 
5.2.2.3 Gierer-Meinhardt model

The Gierer-Meinhardt model [18] is (2.26) defined in \( \Omega = [-1, 1]^2 \subset \mathbb{R}^2 \) with the following reaction kinetics and parameters:

\[
\mathcal{D} = [\epsilon^2, \kappa/\mu]^T, \quad \mathbf{f}(\mathbf{u}) = \left[ \frac{u_1^2}{u_2} - u_1, \frac{1}{\mu} \left( \frac{u_1^2}{\epsilon} - u_2 \right) \right]^T, \tag{5.30}
\]

for which various numerical methods have been developed [16, 38, 45]. We cast the experiment employing coefficients and the initial condition used in [45]:

\[
\epsilon = 0.04, \quad \mu = 0.1, \quad \kappa = 0.0128, \\
\begin{align*}
    u_1(x, y, 0) &= \frac{1}{2} \left[ 1 + 0.001 \sum_{k=1}^{20} \cos \left( \frac{k\pi y}{2} \right) \right] \text{sech}^2 \left( \frac{\sqrt{x^2 + y^2}}{2\epsilon} \right), \\
    u_2(x, y, 0) &= \frac{\cosh \left( 1 - \sqrt{x^2 + y^2} \right)}{3\cosh(1)}. \tag{5.31}
\end{align*}
\]

The initial values are depicted in Figure 5.6. In this subsection, we restrict our attention to the dynamics of \( u_1 \) of the model.

5.2.2.3 Gierer-Meinhardt model

The initial values in (5.31) for the Gierer-Meinhardt model with \( \epsilon = 0.04 \) at the mesh resolution \( \Delta x = \Delta y = 1/32 \): (a) \( u_1(x, y, 0) \) and (b) \( u_2(x, y, 0) \).
Figure 5.7

Numerical solutions for $u_1$ of the Gierer-Meinhardt model at the spatial resolution $\Delta x = \Delta y = 1/32$: (a) the variable-$\theta$ method with $\Delta t = 0.05$, (b) the CN method with $\Delta t = 0.05$, and (c) the CN method with $\Delta t = 0.005$; (d), (e), and (f) are the aerial views of (a), (b) and (c), respectively.
In order to investigate effectiveness of the variable-$\theta$ method and oscillatory behaviors of the CN method as well, we have carried out numerical experiments for the Gierer-Meinhardt model with a relatively low spatial resolution. Figure 5.7 presents numerical solutions at four different times for $u_1$ of the Gierer-Meinhardt model with the spatial resolution $\Delta x = \Delta y = 1/32$. When the time step size is set $\Delta t = 0.05$, the variable-$\theta$ method evolves the numerical solution as shown in Figures 5.7(a) and (d), for which the final steady-state pattern is the same as that in [45]. On the other hand, with the same $\Delta t = 0.05$, the CN method has produced a quite different pattern as in Figures 5.7(b) and (e), due to the nonsmooth initial values (Figure 5.6). However, the CN method can recover the correct steady-state pattern when it runs with $\Delta t = 0.005$, as depicted in Figures 5.7(c) and (f). For a similar accuracy, the variable-$\theta$ method (taking 170 s) is about 7 times more efficient than the CN method (taking 1242 s).

We summarize our experiments with the Gray-Scott and Gierer-Meinhardt models in 2D as follows.

- The variable-$\theta$ method shows the same accuracy as the CN method for problems of smooth data.
- For nonsmooth data, the variable-$\theta$ method evolves a smooth solution for all choices of $\Delta t$, while the CN method introduces spurious oscillations to alter the solution unless the time step size is sufficiently small.
- When a large time step size is desirable, the suggested method is a few times more efficient than the CN method for a similar accuracy.

Remark 5.2.1 Although the variable-$\theta$ method can employ larger time step sizes than the CN method to get stable numerical solutions for problems of nonsmooth data, one may not set the time step size too large, due to an accuracy issue rather than the stability issue.
Furthermore, for nonlinear problems, the overall stability of the numerical algorithm can be determined by not only grid sizes but also numerical schemes including methods of dealing with the nonlinear terms.

Figure 5.8 presents numerical solutions and their errors of $u_1$ for the Gray-Scott model (5.27)-(5.29) at $T = 1.0$ by the variable-$\theta$ method varying $\Delta x = \Delta y$ with fixed $\Delta t = 0.25$. Compared with Figure 5.5, the solutions show stability and a good accuracy, although the time step size is as large as $\Delta t = 0.25$. As shown in the bottom line in Figure 5.8, all three spatially different cases show the same level of errors since the entire errors are dominated by temporal direction errors. We conclude from this example that grid sizes in both temporal and spatial directions would not significantly affect the stability of the proposed method when the initial condition is smooth and the nonlinearity is not severe.

As an example of nonsmooth data and severe nonlinearity, we select the Gierer-Meinhardt model (5.30)-(5.31) to simulate with large temporal step sizes. When $\Delta t \geq 0.2$, the proposed algorithm introduced a rapid decay of solution values (particularly, of $u_1$), independently of the spatial grid size, so that the pattern is not formed appropriately. We believe that it is due to the error incorporated with the reaction term (5.3) when $u^{n+1/2}$ is approximated by the extrapolation scheme (5.2). However, when $\Delta t \leq 0.1$, our method produces stable solutions for all choices of spatial grid sizes. Figure 5.9 presents numerical solutions of $u_1$ for the Gierer-Meinhardt model at $T = 340$ by the variable-$\theta$ method with fixed $\Delta t = 0.1$ and various $\Delta x = \Delta y$. Note that for Gierer-Meinhardt model, the pattern forming is slow down as the spatial grid size becomes smaller, as shown in $(I_2)$ and $(I_3)$; this tendency has been observed for all other choices of $\Delta t \leq 0.1$. This is another example that
Figure 5.8

Numerical solutions for $u_1$ of the Gray-Scott model at $T = 1.0$ by variable-$\theta$ method varying $\Delta x = \Delta y$ with fixed $\Delta t = 0.25$. The image $I_\ell$ represents the numerical solution obtained with the mesh resolution $(\Delta t, \Delta x = \Delta y) = (0.25, 0.02/2^{\ell-1})$ and $I'_\ell$ represents the error of $I_\ell$. 

109
Figure 5.9

Numerical solutions for $u_1$ of the Gierer-Meinhardt model at $T = 340$ by the variable-$\theta$ method varying $\Delta x = \Delta y$ with fixed $\Delta t = 0.1$. The image $I_\ell$ represents the numerical solution obtained with the mesh resolution $(\Delta t, \Delta x = \Delta y) = (0.1, 0.02/2^{\ell-1})$ and $I'_\ell$ represents the aerial view of $I_\ell$. 
accuracy of the numerical solution is much more sensitive to the spatial mesh resolution than the temporal one.

5.3 The Averaging Scheme for Reaction-Diffusion System

As pointed out in the previous Section 5.2.1, accuracy of the numerical solution is much more sensitive to the spatial mesh resolution than the temporal one. Since the spatial mesh resolution is mostly related to the approximation of diffusion (Laplacian operator), we implemented accuracy experiments for some approximations of diffusion in 2D and would present the accuracy issue of the RD problem related to the spatial approximations. Here, for convenience, we assume the uniform spatial grid for 2D ($\Delta x = \Delta y = h$).

5.3.1 Accuracy issue for RD problems

The Gierer-Meinhardt model [18] is (2.26) defined in $\Omega = [-1, 1]^2 \subset \mathbb{R}^2$ associated with (5.30) and (5.31). Here, we also restrict our attention to the dynamics of $u_1$ of the model.

Figure 5.10 and 5.11 present the numerical solutions associated with the Gierer-Meinhardt model (5.30) and (5.31) at the steady-state ($T = 500$) with the fixed time-step $\Delta t = 0.01$ and various spatial resolutions. The diffusion is spatially approximated by the standard 5-point FD scheme and the spline collocation method ($r = 3$) [17], respectively. For a fair comparison, we choose the same number of grid points for the standard 5-point scheme as the gaussian quadrature points for the orthogonal spline collocation method ($r = 3$).

In both figures, one can check that the low resolution case (Figure 5.10($I_1$)) and Figure 5.11($J_1$)) shows quite different steady-state pattern from the patterns of the high res-
Numerical solutions for Gierer-Meinhardt model (5.30) and (5.31) of the steady-state 
\((T = 500)\) approximated by the standard 5-point FD scheme at fixed time-step \(\Delta t = 0.01\) 
and various spatial resolutions. The image \(I_\ell\) represents the numerical solution obtained 
with the mesh resolution \((\Delta t, n_x = n_y) = (0.01, 50 \cdot 2^{\ell-1})\).

Numerical solutions for Gierer-Meinhardt model (5.30) and (5.31) of the steady-state 
\((T = 500)\) approximated by the spline collocation method computed with \(r = 3\) at fixed 
time-step \(\Delta t = 0.01\) and various spatial resolutions. The image \(J_\ell\) represents the 
numerical solution obtained with the mesh resolution \((\Delta t, n_x = n_y) = (0.01, 50 \cdot 2^{\ell-1})\).
olution cases (Figures 5.10(1, 3) and Figures 5.11(2, 3)). In particular, the spline collocation method, in spite of its fourth order accuracy in spatial direction, results a distinct pattern in the low resolution, as the standard 5-point FD scheme of spatially second order. It implies that the different evolutinal behaviors are not originated from the accuracy order of spatial approximation but from the concerning direction of that, since the both methods only deal with the derivatives of dependent variable direction. Thus, we will consider the FD approximations able to compactly reflect the influences of all vicinities.

5.3.2 FD schemes for the second spatial derivatives

First, for the convenience, we assume the uniform grid $\Delta x = \Delta y = h$, then Taylor series gives us the following finite difference (FD) approximation for vicinity of $(i, j)$-position:

$$u_{i-1,j} = u_{ij} - hu_x + \frac{h^2}{2}u_{xx} - \frac{h^3}{6}u_{xxx} + \frac{h^4}{24}u_{xxxx} + O(h^5) \quad (5.32)$$

$$u_{i+1,j} = u_{ij} + hu_x + \frac{h^2}{2}u_{xx} + \frac{h^3}{6}u_{xxx} + \frac{h^4}{24}u_{xxxx} + O(h^5) \quad (5.33)$$

$$u_{i,j-1} = u_{ij} - hu_y + \frac{h^2}{2}u_{yy} - \frac{h^3}{6}u_{yyy} + \frac{h^4}{24}u_{yyyy} + O(h^5) \quad (5.34)$$

$$u_{i,j+1} = u_{ij} + hu_y + \frac{h^2}{2}u_{yy} + \frac{h^3}{6}u_{yyy} + \frac{h^4}{24}u_{yyyy} + O(h^5). \quad (5.35)$$

Combining (5.32)-(5.35), we obtain the standard 5-point FD approximation $A_5$ of $-\Delta$ ($= \partial_x^2 + \partial_y^2$) at $x_{ij}$:

$$A_5 u_{ij} = \frac{-u_{i-1,j} - u_{i,j+1} + 4u_{ij} - u_{i+1,j} - u_{i,j-1}}{h^2}, \quad (5.36)$$

113
and the Laplacian operator applied to \( u \) at \( x_{ij} \) can be written as

\[
-\Delta u_{ij} = \frac{1}{h^2} \begin{pmatrix} -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} -1 \ 4 \ -1 \end{pmatrix} u_{ij} + \frac{h^2}{12} (u_{xxxx} + u_{yyyy}) + \mathcal{O}(h^4).
\] (5.37)

It is easy to see that the odd-order terms vanish during computation thus, we can get the fourth-order remainder \( \mathcal{O}(h^4) \) after the second-order dominant truncation error.

Again, Taylor series gives us the another finite difference (FD) approximation for cross vicinity of \((i,j)\)-position:

\[
u_{i-1,j-1} = u - h(u_x + u_y) + \frac{h^2}{2} (u_{xx} - 2u_{xy} + u_{yy}) - \frac{h^3}{6} (u_{xxx} + 3u_{xxy} + 3u_{xyy} + u_{yyy})
+ \frac{h^4}{24} (u_{xxxx} + 4u_{xxyy} + 6u_{xxyy} + 4u_{xyyy} + u_{yyyy}) + \mathcal{O}(h^5)
\] (5.38)

\[
u_{i-1,j+1} = u - h(u_x - u_y) + \frac{h^2}{2} (u_{xx} + 2u_{xy} + u_{yy}) - \frac{h^3}{6} (u_{xxx} + 3u_{xxy} + 3u_{xyy} - u_{yyy})
+ \frac{h^4}{24} (u_{xxxx} - 4u_{xxyy} + 6u_{xxyy} - 4u_{xyyy} + u_{yyyy}) + \mathcal{O}(h^5)
\] (5.39)

\[
u_{i+1,j-1} = u + h(u_x - u_y) + \frac{h^2}{2} (u_{xx} - 2u_{xy} + u_{yy}) + \frac{h^3}{6} (u_{xxx} + 3u_{xxy} + 3u_{xyy} - u_{yyy})
+ \frac{h^4}{24} (u_{xxxx} - 4u_{xxyy} + 6u_{xxyy} - 4u_{xyyy} + u_{yyyy}) + \mathcal{O}(h^5)
\] (5.40)

\[
u_{i+1,j+1} = u + h(u_x + u_y) + \frac{h^2}{2} (u_{xx} + 2u_{xy} + u_{yy}) + \frac{h^3}{6} (u_{xxx} + 3u_{xxy} + 3u_{xyy} + u_{yyy})
+ \frac{h^4}{24} (u_{xxxx} + 4u_{xxyy} + 6u_{xxyy} + 4u_{xyyy} + u_{yyyy}) + \mathcal{O}(h^5)
\] (5.41)

On the other hand, the skewed 5-point FD approximation \( A_s \) of \(-\Delta\) at \( x_{ij} \) can be obtained from (5.38)-(5.41):

\[
A_s u_{ij} = \frac{-u_{i-1,j-1} - u_{i-1,j+1} + 4u_{ij} - u_{i+1,j+1} - u_{i+1,j-1}}{h^2},
\] (5.42)
and the Laplacian operator is expressed as

\[ -\Delta u_{ij} = \frac{1}{2h^2} \begin{bmatrix} -1 & -1 \\ 4 & -1 \\ -1 & -1 \end{bmatrix} u_{ij} + \frac{h^2}{12} \left( u_{xxxx} + 6u_{xxyy} + u_{yyyy} \right) + O(h^4). \]  

(5.43)

As same as the above, the odd-order terms vanish during computation.

Furthermore, we can consider the following combination of \( A_5 \) and \( A_s \), which gives us a 9-point compact FD approximation \( A_9 \) of \(-\Delta \) at \( x_{ij} \):

\[ A_9 u_{ij} = \frac{2}{3} A_5 u_{ij} + \frac{1}{3} A_s u_{ij}. \]  

(5.44)

This formula is often called Mehrstellen and has been known for a long time, refer to [9].

The Laplacian approximation with its truncation error can be written as

\[ -\Delta u_{ij} = \frac{1}{6h^2} \begin{bmatrix} -1 & -4 & -1 \\ -4 & 20 & -4 \\ -1 & -4 & -1 \end{bmatrix} u_{ij} + \frac{h^2}{12} \left( u_{xxxx} + 2u_{xxyy} + u_{yyyy} \right) + O(h^4). \]  

(5.45)

In particular, the leading truncation error of \( A_9 \) coincides with a scaled biharmonic operator applied to \( u \) at \( x_{ij} \):

\[ \frac{h^2}{12} \left( u_{xxxx} + 2u_{xxyy} + u_{yyyy} \right) = \frac{h^2}{12} \Delta^2 u = \frac{h^2}{12} \Delta (\Delta u). \]  

(5.46)

Thus, the combination \( A_9 \) can achieve fourth-order accurate under some well-known conditions of \( u \) such as harmonic (\( \Delta u = 0 \)) and biharmonic (\( \Delta^2 u = 0 \)) condition, while the both approximations \( A_5 \) and \( A_s \) are second-order accurate in general.

From the fact that combinations of \( A_5 \) and \( A_s \) and properties of equations possibly result more accuracy, the authors in [24] suggested a generalized 9-point compact Laplacian formula by averaging \( A_5 \) and \( A_s \). In next section, we will introduce the generalized formula.
5.3.3 Averaging compact scheme

In this section, we would introduce a compact approximation $A_\alpha$ of the Laplacian operator suggested in [24], which is derived from averaging $A_5$ and $A_s$ as follows:

for $0 < \alpha < 1$,

$$A_\alpha u_{ij} = \alpha A_5 u_{ij} + (1 - \alpha) A_s u_{ij}$$

and the Laplacian operator can be written as

$$-\Delta u_{ij} = \frac{1}{2h^2} \left[ -1 + \alpha \right] \left[ -2\alpha \right] \left[ -1 + \alpha \right] u_{ij} + \frac{1}{4} \left[ -2\alpha \right] \left[ 4(1 + \alpha) \right] \left[ -2\alpha \right] u_{ij} + \frac{h^2}{12} \left[ u_{xxxx} + 6(1 - \alpha) u_{xxyy} + u_{yyyy} \right] + O(h^4).$$

Here, the averaging parameter $\alpha$ can be extended to the values greater than 1, in this case, the formula can be regarded as the extrapolation of $A_5$ and $A_s$.

In [24], Jo, Shin and Suh proposed the above averaging compact scheme for the Helmholtz equation. They also gave an optimized parameter by minimizing the numerical dispersion error of the phase velocity. Their optimal averaging compact scheme with 5 grid points per wavelength could achieve the same accuracy of the standard 5-point FD scheme with 10 grid points per wavelength. However, this optimal averaging compact scheme is yet to be developed for the nonlinear RD system. Next, we will discuss a strategy for optimizing the averaging scheme for the nonlinear RD system.
5.4 The Optimized Averaging Parameter

In this section, as an optimal averaging parameter, we will derive the averaging parameter $\alpha$ which minimizes the leading truncation error:

$$\frac{h^2}{12} [u_{xxxx} + 6(1 - \alpha)u_{xxyy} + u_{yyyy}] - \frac{\Delta t^2}{24}u_{ttt}. \quad (5.49)$$

Here, for convenience, $u$, $f$, $D$, and $\alpha$ represent respectively $u_l$, $f_l$, $D_l$, and $\alpha_l$ for fixed $l = 1, 2$.

Note that

$$\Delta^2 u = u_{xxxx} + 2u_{xxyy} + u_{yyyy} \quad (5.50)$$

then the leading truncation error (5.49) can be expressed by

$$\frac{h^2}{12} [\Delta^2 u + (4 - 6\alpha)u_{xxyy}] - \frac{\Delta t^2}{24}u_{ttt}. \quad (5.51)$$

To choose $\alpha$ which makes it possible to vanish the leading error, we obtain the following equation:

$$\Delta^2 u + (4 - 6\alpha)u_{xxyy} - \frac{\gamma^2}{2}u_{ttt} = 0, \quad (5.52)$$

where $\gamma = \Delta t/h$. Solving (5.52) for $\alpha$, we obtain

$$\alpha = \frac{2\Delta^2 u - \gamma^2 u_{ttt}}{12 u_{xxyy}} + \frac{2}{3}. \quad (5.53)$$

Note that

$$\partial_t u^n := \frac{u^n - u^{n-1}}{\Delta t} \quad \text{and} \quad \partial_t u^n \approx u_t + \frac{\Delta t^2}{24}u_{ttt}, \quad (5.54)$$

$$u_{ttt} \approx \frac{24}{\Delta t^2}(\partial_t u^n - u_t).$$
Then, from (2.26) and (5.54)
\[
\alpha \approx \frac{h^2 \Delta^2 u - 12(\overline{\partial_t} u^n - u_t)}{6h^2 u_{xxyy}} + \frac{2}{3} \tag{5.55}
\]
\[
= \frac{h^2 \Delta^2 u - 12(\overline{\partial_t} u^n - D\Delta u - f(u))}{6h^2 u_{xxyy}} + \frac{2}{3}
\]

In order to obtain a computable formula for (5.55), utilizing the discrete Laplacian operators \(A_x, A_y,\) and \(A_5\) (refer to (2.16)), we can have the computable optimal parameter at each \(n\)-th time-step for \(\alpha^n\) as follows
\[
\alpha^n \approx \frac{h^2 A_5^2 u^n - 12(\overline{\partial_t} u^n - D A_5 u^n - f(u^n))}{6h^2 A_x A_y u^n} + \frac{2}{3} \tag{5.56}
\]

Taking average to attain single constant parameter and give it the constraint \(0 < \alpha < 2/3,\) we can compute the optimal parameter \(\tilde{\alpha}^n\) as follows
\[
\alpha^n \approx \max \left\{ 0, \min \left\{ \frac{2}{3}, \text{average} \left( \frac{h^2 A_5^2 u^n - 12(\overline{\partial_t} u^n - D A_5 u^n - f(u^n))}{6h^2 A_x A_y u^n} + \frac{2}{3} \right) \right\} \right\}. \tag{5.57}
\]

Table 5.2

Accumulated \(L^2\)-errors \(E[0, T],\) and CPU-time for the Gray-Scott model associated with (5.27)-(5.29) at \(T = 1.0\) and variable resolutions and the averaging parameter \(\alpha.\) Here, \(e_1\) and \(e_2\) are the \(L^2\)-errors for \(u_1\) and \(u_2,\) respectively and we summarize the errors with \(L^2\)-norm \(\sqrt{e_1^2 + e_2^2}.\)

<table>
<thead>
<tr>
<th>((\tau, h))</th>
<th>(\alpha = 1) (CPU)</th>
<th>(\alpha = 2/3) (CPU)</th>
<th>(\tilde{\alpha}^n) (CPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.1, 0.1)</td>
<td>8.22 \cdot 10^{-4} (0.29s)</td>
<td>8.04 \cdot 10^{-4} (0.29s)</td>
<td>7.80 \cdot 10^{-4} (0.34s)</td>
</tr>
<tr>
<td>(0.05, 0.05)</td>
<td>9.71 \cdot 10^{-5} (0.67s)</td>
<td>9.49 \cdot 10^{-5} (0.69s)</td>
<td>9.12 \cdot 10^{-5} (0.79s)</td>
</tr>
<tr>
<td>(0.025, 0.025)</td>
<td>1.18 \cdot 10^{-5} (3.31s)</td>
<td>1.15 \cdot 10^{-5} (3.38s)</td>
<td>1.02 \cdot 10^{-5} (3.55s)</td>
</tr>
</tbody>
</table>

With the test problem of the Gray-Scott model (5.27)-(5.29), we will check the effectiveness of the optimal parameter. Table 5.2 summarize the accumulated \(L^2\)-errors \(E[0, T],\)
and CPU-time for the Gray-Scott model associated with (5.24)-(5.29) at $T = 1.0$ and resolution mesh $(\Delta t, h) = (0.05, 0.05)$ with the fixed $\alpha^n = 1, 2/3$ and $\alpha^n$.

We can check that the optimal case shows the smallest error amount among the three choices of $\alpha$ and the error of Mehrstellen ($\alpha = 2/3$) is smaller than that of the standard 5-point scheme ($\alpha = 1$). Also, we can achieve at most 15% more error advantage with the optimal parameter than the standard 5-point scheme. Furthermore, since the computation of the optimal parameter $\alpha^n$ is designed to utilize the pre-calculated values during the algorithm, it consumes a negligible amount of CPU-time comparing to other two schemes with the fixed parameter. We conclude that the averaging scheme can be elaborated to add the inexpensive optimizing step for $\alpha$ but it is still necessary to improve the performance of the optimizing.

5.5 Numerical Experiments

In this section, we present numerical experiments using the averaging scheme to verify its effectiveness and accuracy. The experiments are implemented in Matlab and carried out on a Desktop computer of an Intel i7-3520M CPU 2.90GHz processor with 8GB RAM. For a comparison purpose, besides the standard 5-point scheme, we also cast the ADI extrapolated Crank-Nicolson orthogonal spline collocation method (CNOSC) in [17], which reproduced by the author with MATLAB.
Numerical solutions $u_1$ for Gray-Scott model defined in Section 5.2.2.2 at $T = 1$ and $(\tau, h) = (0.001, 0.1)$ approximated by (a) the CNOSC ($r = 3$), and (b) the averaging scheme ($\bar{\alpha} = 2/3$).

For the algebraic solver for the solution of each time level, the SOR with the near-optimal parameter in [34, §4] is employed and the SOR iteration is stopped when the maximum difference of consecutive iterates becomes smaller than a tolerance $\varepsilon = 10^{-8}$:

$$\|u^{n,k} - u^{n,k-1}\|_{\infty} < \varepsilon.$$ (5.58)

### 5.5.1 Gray-Scott model

First, we will verify the effectiveness of the averaging scheme quantitatively. Consider the following two-component Gray-Scott model defined in Section 5.2.2.2.

Figure 5.12 presents the two numerical solutions of $u_1$ approximated by the CNOSC ($r = 3$) and the averaging scheme with $\alpha = 2/3$. In particular, for the numerical solution by the CNOSC, we choose the same parameters as Example 3 in [17], which are $n_x = n_y = 20$ (that is $h = 0.1$), $\tau = h^3$, and $r = 4$ and Figure 5.12(a) shows the same shapes.
displayed in Figure 2 in [17]. Due to the smooth initial condition and fine enough temporal step-size, we could achieve the stable patterns in each method.

Table 5.3

Accumulated \(L^2\)-errors \(E[0, T]\), and CPU-time for the Gray-Scott model defined in Section 5.2.2.2 at \(T = 1.0\) and variable resolutions by each method. Here, \(e_1\) and \(e_2\) are the \(L^2\)-errors for \(u_1\) and \(u_2\), respectively and we summarize the errors with \(L^2\)-norm \(\sqrt{e_1^2 + e_2^2}\).

<table>
<thead>
<tr>
<th>((\tau = h^3, h))</th>
<th>CNOSC</th>
<th>(A_5)-CN (CPU)</th>
<th>(A_5)-VT (CPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>((0.2^3, 0.2))</td>
<td>1.77 (\cdot) 10(^{-3}) (0.80s)</td>
<td>3.21 (\cdot) 10(^{-4}) (0.85s)</td>
<td>3.21 (\cdot) 10(^{-4}) (0.90s)</td>
</tr>
<tr>
<td>((0.1^3, 0.1))</td>
<td>9.92 (\cdot) 10(^{-6}) (18.62s)</td>
<td>3.78 (\cdot) 10(^{-5}) (9.40s)</td>
<td>3.78 (\cdot) 10(^{-5}) (10.10s)</td>
</tr>
<tr>
<td>((0.05^3, 0.05))</td>
<td>5.61 (\cdot) 10(^{-6}) (612.27s)</td>
<td>4.54 (\cdot) 10(^{-6}) (171.89s)</td>
<td>4.54 (\cdot) 10(^{-6}) (187.32s)</td>
</tr>
<tr>
<td>order</td>
<td>4.15</td>
<td>3.07</td>
<td>3.07</td>
</tr>
</tbody>
</table>

Also, we summarize the accumulated \(L^2\)-errors and CPU-time for the Gray-Scott model defined in Section 5.2.2.2 at \(T = 1.0\) and various resolutions in Table 5.1. We exhibit the errors for the CNOSC with the splines of degree \(r = 3\) and two standard 5-point scheme equipped with the CN and variable-\(\theta\) method (VT) of their time-stepping procedure. In particular, we choose the finer time step \(\tau = h^{(r+1)/2}\), since it is expected that the CNOSC is \(O(\tau^2 + h^{r+1})\) accurate in the \(L^2\)-norm.

First, the errors of the FD schemes are coincidence since the initial and boundary conditions are not associated with nonsmoothness. As we expected from [17], the CNOSC shows fourth-order convergence in spatial direction due to utilizing the splines of degree 3 while two FD schemes achieve third-order convergence in spatial direction which is one order higher than the order of the standard 5-point scheme since total errors are dominated
by temporal one. Beside the convergence order, we can point out that the $L^2$-error amounts ($\sqrt{e_1^2 + e_2^2}$) of the FD schemes are smaller than that of the CNOSC, in particular, the error differences between the CNOSC and the FD schemes are severe (5.5 times) in low spatial resolution $(\tau, h) = (0.2^3, 0.2)$. It implies that the CNOSC might introduce imperceptible oscillations spreading out to all over its numerical solutions, which are originated from its rough orthogonal basis constructed with less grid points of the low spatial resolution. Furthermore, even though the CNOSC employs the ADI method to fasten the calculation, it takes at most 3.5 times more CPU-time than the two FD methods since the calculation of the CNOSC deals with the coefficient matrices whose size is 82 by 82 ($2 \cdot 40 + 2$) when it utilizes the splines of degree $r = 3$ of 40 by 40 spatial grid points.

Table 5.4

Accumulated $L^2$-errors $E[0, T]$, and CPU-time for the Gray-Scott model defined in Section 5.2.2.2 at $T = 1.0$ and variable temporal resolutions ($h = 0.05$ is fixed) by $A_5$-VT. Here, $e_1$ and $e_2$ are the $L^2$-errors for $u_1$ and $u_2$, respectively and we summarize the errors with $L^2$-norm $\sqrt{e_1^2 + e_2^2}$.

<table>
<thead>
<tr>
<th>$(\tau, h)$</th>
<th>$A_5$-VT $3.92 \cdot 10^{-6}$ (2.14s)</th>
<th>$0.025, 0.05$ (CPU) $2.11 \cdot 10^{-5}$ (1.03s)</th>
<th>$0.05, 0.05$ (CPU) $9.71 \cdot 10^{-5}$ (0.51s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\tau, h)$</td>
<td>$(0.0125, 0.05)$ (CPU) $3.92 \cdot 10^{-6}$ (2.14s)</td>
<td>$(0.025, 0.05)$ (CPU) $2.11 \cdot 10^{-5}$ (1.03s)</td>
<td>$(0.05, 0.05)$ (CPU) $9.71 \cdot 10^{-5}$ (0.51s)</td>
</tr>
</tbody>
</table>

Furthermore, we present the accumulated $L^2$-errors $E[0, T]$ of $A_5$-VT and CPU-time for the Gray-Scott model defined in Section 5.2.2.2 at $T = 1.0$ and various temporal resolutions ($h = 0.05$ is fixed) in Table 5.4. From the first column in Table 5.4, one can see that at $(\tau, h) = (0.0125)$, $A_5$ with the variable-$\theta$ method can achieve the similar level of error to the error of the CNOSC at $(\tau, h) = (0.05^3, 0.05)$ (the third row in the first column
Numerical solutions for Gierer-Meinhardt model (5.30) and (5.31) of the steady-state ($T = 500$) approximated by the heuristic-optimal averaging scheme at fixed time-step $\tau = 0.01$ and various spatial resolutions. The image $I_\ell$ represents the numerical solution obtained with the mesh resolution $(\tau, h) = (0.01, 0.04/2^{\ell-1})$.

in Table 5.1), whose time-step is 100 times less than that of the CNOSC. We also can prove the effectiveness of our method by CPU-time of $\mathcal{A}_5$-VT (2.14s) which is 300 times less than that of the CNOSC (612.27s).

5.5.2 Gierer-Meinhardt model

In order to investigate the effectiveness of the averaging scheme qualitatively, again consider the Gierer-Meinhardt model associated with (5.30) and (5.31). In Figure 5.13, we depict the numerical solutions to the Gierer-Meinhardt model at the steady-state ($T = 500$) by the averaging scheme and its mesh resolutions are followed by Figures 5.10 and 5.11. The parameter $\alpha$ is heuristically chosen to be $2/3$. Different from the other low resolution cases: Figures 5.10$(I_1)$ and 5.11$(I_1)$, in Figure 5.13, the averaging scheme shows the same steady-state pattern as the high resolution cases even in low spatial resolution case. It notes that, due to its compactness of spatial approximation, the averaging scheme can achieve
the stable evolution even in low spatial resolution in which the standard 5-point scheme and higher order scheme showed a distinct evolution pattern.

To examine the initial states of pattern formation by each method in low spatial resolution, we present the numerical solutions for Gierer-Meinhardt model (5.30) and (5.31) at \( T = 80 \) and \((\tau, n_x = n_y) = (0.01, 50)\) approximated by three different schemes (the standard 5-point scheme, the CNOSC \((r = 3)\), and the averaging scheme \((\tilde{\alpha} = 2/3)\)) in Figure 5.14. The figures (a), (b), and (c) in Figure 5.14 correspond to the initial state \((t = 80)\) of Figures 5.10\((I_1)\), 5.11\((I_1)\), and 5.13\((I_1)\), respectively.

In Figure 5.14(a), we can check that the standard 5-point scheme shows the artifacts along with dependant variable directions, which is originated from the spatial approximation concerning only two dependant variable directions. The CNOSC shows a rotationally invariant pattern formation in Figure 5.14(b) but we can see an abnormal submit in the center of the figure which causes the bigger circular pattern than the others.

In particular, this submit is arisen from the rough spline polynomial generated by less gaussian quadrature points and we can suppress this artifact by adding more grid points and employing higher order polynomial but it does not mean an appropriate remedy for the artifact. Eventually, these artifacts of the standard 5-point scheme and CNOSC would produce abnormal steady-state patterns in low spatial resolution which are different from their steady-state patterns in the high spatial resolution.

Unlike the two schemes above, the averaging scheme shows a rotationally invariant pattern formation without any artifacts in Figure 5.14(c) and we can also achieve the same
Numerical solutions for Gierer-Meinhardt model (5.30) and (5.31) at $T = 80$ and $(\tau, n_x = n_y) = (0.01, 50)$ approximated by (a) the standard 5-point scheme, (b) the CNOSC ($r = 4$), and (c) the averaging scheme ($\tilde{\alpha} = 2/3$).
steady-state patterns in low spatial resolution as in high spatial resolution since the scheme approximates the diffusion compactly with considering every directions.
CHAPTER 6

CONCLUSIONS

Although various numerical algorithms have been suggested for solving elliptic obstacle problems effectively, most of the algorithms presented in the literature are yet to be improved for both accuracy and efficiency. In this dissertation, we have studied obstacle relaxation methods in order to get second-order finite difference (FD) solutions of obstacle problems more accurately and more efficiently. The suggested iterative algorithm is based on one of the simplest relaxation methods, the successive over-relaxation (SOR). The iterative algorithm is incorporated with subgrid FD methods to reduce accuracy deterioration occurring near the free boundary when the mesh grid does not match with the free boundary. For nonlinear obstacle problems, a method of gradient-weighting has been introduced to solve the problem more conveniently and efficiently. The iterative algorithm has been analyzed for convergence for both linear and nonlinear obstacle problems. An effective strategy is also presented to find the optimal relaxation parameter. The resulting obstacle SOR has converged about one-order faster than state-of-the-art methods and the subgrid FD methods could reduce numerical errors by one order magnitude.

For parabolic initial-boundary value problems with nonsmooth data, specific numerical methods are required to avoid spurious oscillations as well as unrealistic smoothing of
steep changes in the numerical solution. We have investigated characteristics of the \( \theta \)-method, \( 0 \leq \theta \leq 1 \), and introduced a \textit{variable-\( \theta \) method} as a variant of the second-order Crank-Nicolson (CN) time-stepping procedure. After introducing an effective strategy for detecting portents of oscillations, the resulting \textit{variable-\( \theta \) method} has been analyzed for its accuracy and stability. The new method has been applied for various parabolic problems in 1D and 2D, with data either smooth or nonsmooth, including obstacle problems and the Black-Scholes model in option pricing. It has been numerically verified that the new \textit{variable-\( \theta \) method} can suppress spurious oscillations at discontinuities of data and maintain as a similar accuracy to the CN method for smooth data. Also, it has been analytically proved that the \textit{variable-\( \theta \) method} satisfies the maximum principle unconditionally.

Finally, we have studied a nonoscillatory time-stepping procedure for the \textit{reaction-diffusion} (RD) equations incorporating with a \textit{variable-\( \theta \) method} which is a perturbation of the CN method. The proposed time-stepping procedure has proven nonoscillatory and give us stable patterning results, although the initial conditions are nonsmooth and the time-step is large. Various examples have been considered to show effectiveness of the method. We also have performed a sensitivity analysis for the numerical solution of biological pattern forming models to conclude that the numerical solution is much more sensitive to the spatial mesh resolution than the temporal one. As a possible solution for the sensitivity issue, this dissertation has suggested the averaging scheme, as an interpolation of the standard 5-point scheme and the skewed 5-point scheme, to enhance the spatial approximation of the RD equations, which has been combined with the \textit{variable-\( \theta \) method} and applied to the RD equations. Due to the 9-point compact approximation of the averaging scheme, the
scheme could achieve more accurate pattern formation than other conventional methods specially in low spatial resolution. We also have introduced the simple optimizing strategy to elaborate the averaging scheme with minimizing the leading truncation error of the scheme. In the experiment of the optimizing strategy, we could not achieve remarkable results but it can be improved with various ideas.
REFERENCES


