

FINITE ELEMENT APPROXIMATIONS OF BURGERS' EQUATION

by

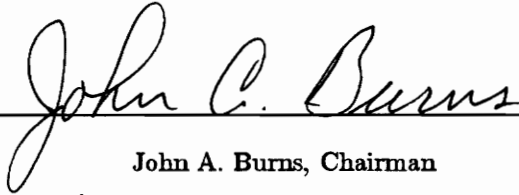
Steven M. Pugh

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

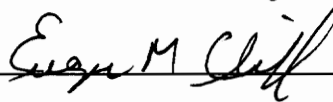
in

Mathematics

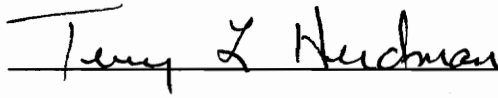
APPROVED:



John A. Burns, Chairman



Eugene M. Cliff



Terry L. Herdman

September, 1995
Blacksburg, Virginia

C.2

LD
565
V855
1995
T814
C.2

FINITE ELEMENT APPROXIMATIONS OF BURGERS' EQUATION

by

Steven M. Pugh

Committee Chairman: John A. Burns

Mathematics

(ABSTRACT)

This work is a numerical study of Burgers' equation with Neumann boundary conditions. The goal is to determine the long term behavior of solutions. We develop and test two separate finite element and Galerkin schemes and then use those schemes to compute the response to various initial conditions and Reynolds numbers.

It is known that for sufficiently small initial data, all steady state solutions of Burgers' equation with Neumann boundary conditions are constant. The goal here is to investigate the case where initial data is large. Our numerical results indicate that for certain initial data the solution of Burgers' equation can approach non-constant functions as time goes to infinity. In addition, the numerical results raise some interesting questions about steady state solutions of nonlinear systems.

ACKNOWLEDGEMENTS

I would like to express my earnest appreciation to my friend, advisor, and thesis committee chairman Dr. John Burns for his support, motivation, and guidance which made this research possible. Also, I want to thank him for his invaluable career insight which helped me choose a very profitable and rewarding one. I would also like to thank Dr. Terry Herdman and Dr. Eugene Cliff for serving on my committee and for offering their guidance throughout this endeavor.

I wish to acknowledge the financial support I received from the Air Force Office of Scientific Research under Grant F49620-93-1-0280.

I would like to give many thanks to my parents, Don and Thelma, my brother, Michael, and my grandmother, Ruth, for their constant encouragement, immeasurable love, and daily devotion to my intellectual, religious, and financial welfare. This paper is dedicated to the memory of my grandfather, Arnold, who instilled in me the ambition and endurance to humbly step over life's many obstacles.

I would like to thank my closest buddies Vic, Rusty, Kevin, and especially Doug for always being there and for helping me maintain a social and intellectual peace of mind. Also, I wish to express my sincere appreciation to my fiancée, Cherie, for her endless love, devotion, and companionship. Without her, this work would have never been possible.

Finally, I thank our Heavenly Father for his many blessings and for teaching me the meanings of happiness and security. May God bless all of you.

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	Introduction	1
1.2	Finite Dimensional Approximations	5
2	TWO APPROXIMATING SYSTEMS	10
2.1	Piecewise Linear Finite Element Method	10
2.2	Finite Element Method and Conservation Form of Burgers' Equation	14
3	NUMERICAL EXPERIMENTS	17
3.1	Setting Up the Problem	17
3.2	A Comparison to Exact and Other Known Numerical Solutions	20
3.3	Comparison of the Galerkin and Galerkin/Conservation Results	32
3.4	Asymptotic Behavior of Finite Element Solutions	43
4	CONCLUSIONS	72
4.1	Overview of Results	72
4.2	Conclusions	73
4.3	Open Problems	74

LIST OF FIGURES

1.1	A typical solution of $u(t, x)u_x(t, x) = \epsilon u_{xx}(t, x)$	3
3.1	Numerical solutions of the homogeneous initial/boundary value problem using $N=4,8,16,32$ elements	19
3.2	$\phi(x) = x^2(1 - x)^2$, $Re=10$, a) Galerkin method, b) Galerkin/Conservation method	29
3.3	$\phi(x) = \cos(\pi x)$, $Re=60$, a) Galerkin method, b) Galerkin/Conservation method	30
3.4	$\phi(x) = -\cos(\pi x)$, $Re=60$, a) Galerkin method, b) Galerkin/Conservation method	31
3.5	$\phi(x) = \cos(\pi x)$, $Re=60$, Galerkin method	34
3.6	$\phi(x) = \cos(\pi x)$, $Re=60$, Galerkin/Conservation method	35
3.7	$\phi(x) = \cos(\pi x)$, $Re=120$, Galerkin method	36
3.8	$\phi(x) = \cos(\pi x)$, $Re=120$, Galerkin/Conservation method	37
3.9	$\phi(x) = \cos(\pi x)$, $Re=240$, Galerkin method	38
3.10	$\phi(x) = \cos(\pi x)$, $Re=240$, Galerkin/Conservation method	39
3.11	$\phi(x) = 2\cos(\pi x)$, a) $Re=60$, Galerkin method, b) $Re=60$, G/C method, c) $Re=120$, Galerkin method, d) $Re=120$, G/C method	40
3.12	$\phi(x) = 4\cos(\pi x)$, a) $Re=60$, Galerkin method, b) $Re=60$, G/C method, c) $Re=120$, Galerkin method, d) $Re=120$, G/C method	41
3.13	$\phi(x) = \frac{1}{16}\cos(\pi x)$ ($Re=60$): a) Galerkin method, b) G/C method	47
3.14	$\phi(x) = \frac{1}{16}\cos(\pi x)$ ($Re=120$): a) Galerkin method, b) G/C method	48
3.15	$\phi(x) = \frac{1}{16}\cos(\pi x)$ ($Re=240$): a) Galerkin method, b) G/C method	49
3.16	$\phi(x) = \frac{1}{2}\cos(\pi x)$ ($Re=60$): a) Galerkin method, b) G/C method	50

LIST OF FIGURES

3.17	$\phi(x) = \frac{1}{2}\cos(\pi x)$ (Re=120): a) Galerkin method, b) G/C method	51
3.18	$\phi(x) = \frac{1}{2}\cos(\pi x)$ (Re=240): a) Galerkin method, b) G/C method	52
3.19	$\phi(x) = \frac{1}{4}\cos(\pi x)$ (Re=60): a) Galerkin method, b) G/C method	53
3.20	$\phi(x) = \frac{1}{4}\cos(\pi x)$ (Re=120): a) Galerkin method, b) G/C method	54
3.21	$\phi(x) = \frac{1}{4}\cos(\pi x)$ (Re=240): a) Galerkin method, b) G/C method	55
3.22	$\phi(x) = \frac{1}{8}\cos(\pi x)$ (Re=60): a) Galerkin method, b) G/C method	56
3.23	$\phi(x) = \frac{1}{8}\cos(\pi x)$ (Re=120): a) Galerkin method, b) G/C method	57
3.24	$\phi(x) = \frac{1}{8}\cos(\pi x)$ (Re=240): a) Galerkin method, b) G/C method	58
3.25	$\phi(x) = \sin(\pi x)$ (Re=60): a) Galerkin method, b) G/C method	59
3.26	$\phi(x) = 10\sin(\pi x)$ (Re=60): a) Galerkin method, b) G/C method	60
3.27	$\phi(x) = 50\sin(\pi x)$ (Re=60): a) Galerkin method, b) G/C method	61
3.28	a) $\phi(x) = 100x^2(1-x)^2$, Galerkin method, b) $\phi(x) = 100x^2(1-x)^2$, G/C method, c) $\phi(x) = 1000x^2(1-x)^2$, Galerkin method, d) $\phi(x) = 1000x^2(1-x)^2$, G/C method	62
3.29	a) $\phi(x) = 100x^4(1-x)^2$, Galerkin method, b) $\phi(x) = 100x^4(1-x)^2$, G/C method, c) $\phi(x) = 1000x^4(1-x)^2$, Galerkin method, d) $\phi(x) = 1000x^4(1-x)^2$, G/C method	63
3.30	G/C approximations (Re=60): a) $\phi(x) = \frac{1}{4}\cos(\pi x) + \frac{1}{4}\sin(\pi x)$, b) $\phi(x) = \frac{1}{4}\cos(\pi x) + \sin(\pi x)$	64
3.31	G/C approximations (Re=60): a) $\phi(x) = \frac{1}{2}\cos(\pi x) + \frac{1}{2}\sin(\pi x)$, b) $\phi(x) = \frac{1}{2}\cos(\pi x) + \sin(\pi x)$	65
3.32	G/C approximations (Re=60): $\phi(x) = \cos(\pi x) + \frac{1}{2}\sin(\pi x)$	66
3.33	G/C approximations (Re=60): $\phi(x) = \cos(\pi x) + \frac{1}{4}\sin(\pi x)$	67
3.34	G/C approximations, $\phi(x) = \frac{1}{4}\cos(2\pi x)$: a) Re=60, b) Re=120	68
3.35	G/C approximations, $\phi(x) = \frac{1}{2}\cos(2\pi x)$: a) Re=60, b) Re=120	69
3.36	G/C approximations, $\phi(x) = \frac{1}{4}\cos(3\pi x)$: a) Re=60, b) Re=120	70
3.37	G/C approximations, $\phi(x) = \frac{1}{2}\cos(3\pi x)$: a) Re=60, b) Re=120	71

LIST OF TABLES

3.1	Solution of Burgers' equation by Galerkin method, Galerkin/Conservation method (Re=60)	23
3.2	Solution of Burgers' equation by Galerkin method, Galerkin/Conservation method (Re=120)	24
3.3	Solution of Burgers' equation by Galerkin method, Galerkin/Conservation method (Re=240)	25
3.4	Comparison of Galerkin solution, Galerkin/Conservation solution, and exact solution $u(t, x) = \frac{1}{4}\cos(t)\cos(\pi x)$ (Re=60)	26
3.5	Comparison of Galerkin solution, Galerkin/Conservation solution, and exact solution $u(t, x) = \frac{1}{4}\cos(t)\cos(\pi x)$ (Re=120)	27
3.6	Comparison of Galerkin solution, Galerkin/Conservation solution, and exact solution $u(t, x) = \frac{1}{4}\cos(t)\cos(\pi x)$ (Re=240)	28
3.7	Comparison of Galerkin Elapsed Time and Galerkin/Conservation Elapsed Time in Seconds	42

Chapter 1

INTRODUCTION

1.1 Introduction

Burgers' equation has been considered by a wide range of researchers to be a useful model for many physically interesting problems involving nonlinear wave propagation subject to some sort of dissipation. In particular, Burgers' equation serves as a model for fluid flow problems and closely related problems of shock flow, traffic flow, etc. Depending on the nature of the problem, the associated dissipation may be a result of viscosity, heat conduction, thermal radiation, chemical reaction, or other source. Burgers' equation captures these and other important phenomena associated with problems of a fluid flow nature, yet it is simple enough to provide insight into more complex problems. For this reason, Burgers' equation is often used as a first case study for testing and comparing computational techniques and controlling more complex nonlinear partial differential equations.

Burgers' equation

$$u_t(t, x) + u(t, x)u_x(t, x) = \epsilon u_{xx}(t, x) + f(t, x) \quad (1.1.1)$$

was first introduced by Burgers as a simple one-dimensional model for turbulent fluid flow [1, 2, 3]. Lighthill [11] later derived it as a second order approximation to the one-dimensional unsteady Navier-Stokes equation. In most practical applications, $\epsilon > 0$ is a viscosity coefficient and $u(t, x)$ is a velocity-like dependent variable, where u_t and u_x indicate the first time and spatial derivatives, respectively. The equation itself is a quasi-linear parabolic partial differential equation which describes the evolution of u with respect to time.

CHAPTER 1. INTRODUCTION

The difficulty of computing fluid dynamic problems arises mainly from the inability to efficiently balance the nonlinear convective term, uu_x , and the diffusive term, ϵu_{xx} . Fletcher [8] illustrates these effects by simplifying equation (1.1.1). If the nonlinear convective term is dropped from equation (1.1.1) the result is simply the linear, parabolic partial differential equation

$$u_t(t, x) = \epsilon u_{xx}(t, x) \quad (1.1.2)$$

which is more commonly known as the heat equation. However, by dropping the diffusive term, equation (1.1.1) becomes

$$u_t(t, x) + u(t, x)u_x(t, x) = 0 \quad (1.1.3)$$

which is a hyperbolic partial differential equation modeling the convection of disturbances in inviscid flow. On a typical convecting wave, points with larger u are shown to convect faster than points on the wave with smaller u , causing u to take on more than one value at a future time \hat{t} . In order to have a unique solution, it is necessary to postulate a shock at which u is discontinuous.

Fletcher also considers the processes of convection and diffusion together and illustrates the evolution of the solution. As time progresses, the maximum amplitude of u becomes smaller and the profile steepens. Since the ϵu_{xx} term becomes larger as the steepening occurs, this dissipative term does not allow a multivalued solution to develop. Omitting the u_t term from equation (1.1.1), the result is

$$u(t, x)u_x(t, x) = \epsilon u_{xx}(t, x) \quad (1.1.4)$$

which is an elliptic partial differential equation representing the convective/dissipative balance in the steady-state sense. If u is finite at $x = +\infty$ and $x = -\infty$, the solution is of the form in Figure 1.1 and changes discontinuously as $\epsilon \rightarrow 0$. Based on these simplifications of equation (1.1.1), Burgers' equation can be categorized, under various conditions, as an elliptic, hyperbolic, or parabolic partial differential equation, furnishing a simple nonlinear model for convection/diffusion interactions.

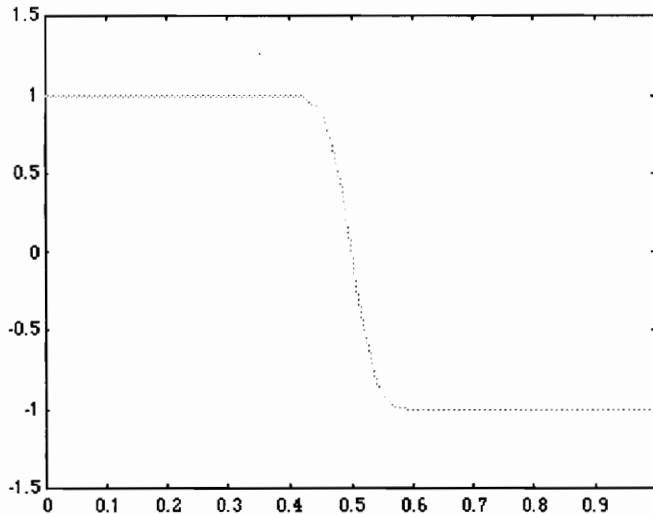


Figure 1.1: A typical solution of $u(t, x)u_x(t, x) = \epsilon u_{xx}(t, x)$.

Another noteworthy feature of Burgers' equation is that exact solutions exist for many combinations of initial and boundary conditions. Hopf [10] and Cole [7] developed a transformation which reduces equation (1.1.1) to the linear heat equation and transforms initial conditions with little computation. Unfortunately, most of these exact solutions exist only for x in the interval $(-\infty, +\infty)$ and many computational difficulties arise when a boundary condition is applied for a finite x (see Fletcher [8]). For Neumann boundary conditions in particular, the Hopf-Cole transformation introduces quadratic nonlinearities at the boundary of the finite interval and presents a nonlinear term which undermines the ability to obtain a priori estimates from which the convergence of Galerkin approximations can be established.

Burgers' equation has provided a test case for several papers on the problem of active control of fluid flows. Burns and Kang [4, 5] first considered a feedback control problem for Burgers' equation and later addressed a stabilization problem with unbounded input and output operators. Both problems involved applying boundary conditions to finite x locations. In the first paper, a finite dimensional nonlinear approximating system was

CHAPTER 1. INTRODUCTION

formulated for Burgers' equation, with Dirichlet boundary condition given by

$$u(t, 0) = u(t, l) = 0 \quad (1.1.5)$$

on a finite interval $[0, l]$. In [5] however, the model was formulated for Burgers' equation with boundary condition given by

$$u(t, 0) = 0, u(t, l) = v(t). \quad (1.1.6)$$

For $[0, l]$ divided into $N + 1$ equal subintervals, the two approximating systems developed in [4] and [5] were of dimensions N and $N + 1$, respectively. Both nonlinear ODE systems were solved using a 4th order Runge-Kutta method [13].

Another boundary control problem governed by Burgers' equation was studied by Marrekchi [12]. In this problem, Marrekchi considered the Burgers' equation with Neumann boundary condition given by

$$u_x(t, 0) = u_x(t, l) = 0. \quad (1.1.7)$$

The resulting nonlinear ODE system was of dimension $N + 2$ and was solved using the ODE45 4th and 5th order Runge-Kutta method in MATLAB. This model is of particular interest because for $l = 1$ and Reynolds number $Re = \frac{1}{\epsilon} = 60$, the solution converged to a constant steady state when given the initial condition

$$u(0, x) = -\cos(\pi x) \quad (1.1.8)$$

but failed to do so for

$$u(0, x) = \cos(\pi x). \quad (1.1.9)$$

In [6], Byrnes, Gilliam, and Shubov proved that for $\epsilon > 0$, there is a constant k such that if $\|\phi\|_{L_2} \leq k\epsilon$, then the solution to the homogeneous Burgers' equation with Neumann boundary conditions must converge to a constant solution as t approaches infinity. They conjectured that this result would hold for any initial data as long as the steady state limit existed. The goal of this effort is to test the conjecture by numerical experimentation.

CHAPTER 1. INTRODUCTION

In this paper we present two methods for approximating solutions to Burgers' equation, with initial condition and Neumann boundary data given by

$$u(0, x) = \phi(x) \quad (1.1.10)$$

$$u_x(t, 0) = 0, u_x(t, 1) = 0. \quad (1.1.11)$$

Given some initial function $\phi(x)$ each method (when applied to the homogeneous Burgers' equation, i.e. $f(t, x) = 0$) will yield a finite dimensional initial value problem of the form

$$\frac{d}{dt}\alpha^N(t) = \epsilon[A^N]\alpha(t)^N - \tilde{F}^N(\alpha^N(t)) \quad (1.1.12)$$

$$\alpha^N(0) = [M^N]^{-1}v \quad (1.1.13)$$

which we will solve using the ODE45 ordinary differential equation solver in MATLAB. (When either method is applied to the nonhomogeneous Burgers' equation, equation (1.1.12) will have an $f^N(t)$ term added to the right hand side.) We will use some polynomials and various multiples of $\sin(\pi x)$ and $\cos(\pi x)$ for our initial data and compare our results to those previously obtained by other researchers. Moreover, for each initial condition we will use $\epsilon = \frac{1}{60}, \frac{1}{120}, \frac{1}{240}$ and explore the effect this viscosity coefficient has on the convergence of the two finite dimensional approximations of Burgers' equation. Ultimately we will compare the two approximating systems on robustness, speed, and accuracy.

1.2 Finite Dimensional Approximations

In this section we introduce the two finite dimensional approximation schemes which will be formulated in the next chapter. The first is the conventional Finite Element/Galerkin Method which was applied to Burgers' equation by Marrekchi (see [12]). The second (alternate) method entails applying the Galerkin method to the Burgers' equation in its conservation form

$$u_t(t, x) + \frac{1}{2}(u^2(t, x))_x = \epsilon u_{xx}(t, x) + f(t, x). \quad (1.2.1)$$

CHAPTER 1. INTRODUCTION

For both methods, the unit interval $[0, 1]$ is divided into $(N + 1)$ subintervals $[x_i, x_{i+1}]$, each of length $h = \frac{1}{N+1}$, where $x_i = \frac{i}{N+1}$ for $i = 0, 1, \dots, N + 1$. For each i , let $h_i^N(x)$ denote the linear basis function defined as follows:

$$h_0^N(x) = \begin{cases} -(N + 1)(x - x_1), & \text{for } x_0 \leq x \leq x_1 \\ 0, & \text{otherwise} \end{cases} \quad (1.2.2)$$

$$h_i^N(x) = \begin{cases} (N + 1)(x - x_{i-1}), & \text{for } x_{i-1} \leq x \leq x_i \\ -(N + 1)(x - x_{i+1}), & \text{for } x_i \leq x \leq x_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (1.2.3)$$

for $1 \leq i \leq N$ and

$$h_{N+1}^N(x) = \begin{cases} (N + 1)(x - x_N), & \text{for } x_N \leq x \leq x_{N+1} \\ 0, & \text{otherwise.} \end{cases} \quad (1.2.4)$$

The basis functions clearly depend on the value of N , but the superscript will be left off when referring to the basis functions throughout this paper.

The trial solution can be written

$$u^N(t, x) = \sum_{i=0}^{N+1} \alpha_i(t) h_i(x) \quad (1.2.5)$$

where each α_i is a nodal unknown and $h_i(x)$ is the i^{th} linear basis (or 'hat') function defined on $[0, 1]$. Application of standard Finite Element/Galerkin procedures (to the homogeneous problem) yields the following finite dimensional ODE system:

$$[M^N] \frac{d}{dt} \alpha^N(t) + F_E^N(\alpha^N(t)) = \epsilon [K^N] \alpha^N(t) \quad (1.2.6)$$

where

$$\alpha^N(t) = \begin{bmatrix} \alpha_0(t) \\ \alpha_1(t) \\ \vdots \\ \alpha_{N+1}(t) \end{bmatrix}, \quad (1.2.7)$$

CHAPTER 1. INTRODUCTION

$$F_E^N(\alpha(t)) = \frac{1}{6} \begin{bmatrix} -2(\alpha_0(t))^2 + \alpha_0(t)\alpha_1(t) + (\alpha_1(t))^2 \\ -(\alpha_0(t))^2 - \alpha_0(t)\alpha_1(t) + \alpha_1(t)\alpha_2(t) + (\alpha_2(t))^2 \\ \vdots \\ -(\alpha_{N-1}(t))^2 - \alpha_{N-1}(t)\alpha_N(t) + \alpha_N(t)\alpha_{N+1}(t) + (\alpha_{N+1}(t))^2 \\ -(\alpha_N(t))^2 - \alpha_N(t)\alpha_{N+1}(t) + 2(\alpha_{N+1}(t))^2 \end{bmatrix} \quad (1.2.8)$$

and

$$[M^N] = \frac{1}{6(N+1)} \begin{bmatrix} 2 & 1 & 0 & 0 & \dots & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ & & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & & & 1 & 4 & 1 \\ 0 & \dots & & 0 & 1 & 2 \end{bmatrix}_{(N+2) \times (N+2)} \quad (1.2.9)$$

$$[K^N] = (N+1) \begin{bmatrix} -1 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ & & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & & & 1 & -2 & 1 \\ 0 & \dots & & 0 & 1 & -1 \end{bmatrix}_{(N+2) \times (N+2)} \quad (1.2.10)$$

are the mass and stiffness matrices, respectively. The entries for $F_E^N(\alpha(t))$, $[M^N]$, and $[K^N]$ will be derived in the next chapter.

The $F_E^N(\alpha(t))$ term which appears on the lefthand side of (1.2.6) is associated with the nonlinear term in equation (1.1.1) and can complicate the computational solution for large values of N . For this reason, we seek an alternative treatment of the nonlinear term which would lessen the amount of computation needed to approximate the solution. Such an alternative treatment is possible and is introduced in [9]. Here Burgers' equation is written

CHAPTER 1. INTRODUCTION

in its conservation form and a second trial solution is introduced for $u^2(t, x)$:

$$[u^N(t, x)]^2 = \sum_{i=0}^{N+1} (\alpha_i(t))^2 h_i(x). \quad (1.2.11)$$

When the Galerkin method is applied to equation (1.2.1) subject to $f(t, x) = 0$, (1.2.5), and (1.2.11), the resulting ODE system is given by

$$[M^N] \frac{d}{dt} \alpha^N(t) + F_D^N(\alpha^N(t)) = \epsilon [K^N] \alpha^N(t) \quad (1.2.12)$$

where $[M^N]$ and $[K^N]$ are as before and

$$F_D^N(\alpha^N(t)) = \frac{1}{4} \begin{bmatrix} (\alpha_1(t))^2 - (\alpha_0(t))^2 \\ (\alpha_2(t))^2 - (\alpha_0(t))^2 \\ (\alpha_3(t))^2 - (\alpha_1(t))^2 \\ \vdots \\ (\alpha_{N+1}(t))^2 - (\alpha_{N-1}(t))^2 \\ (\alpha_{N+1}(t))^2 - (\alpha_N(t))^2 \end{bmatrix} \quad (1.2.13)$$

corresponds to the nonlinear term appearing in equation (1.2.1). (Note that the only difference between (1.2.6) and (1.2.12) is the $F^N(\alpha(t))$ term.) Based on our numerical experiments this alternative method, which we will refer to as the Galerkin/Conservation Method, is the faster and more accurate of the two.

Since $[M^N]$ is known to be invertible, we can multiply (from the left) both sides of

$$[M^N] \frac{d}{dt} \alpha^N(t) + F^N(\alpha^N(t)) = \epsilon [K^N] \alpha^N(t) \quad (1.2.14)$$

by $[M^N]^{-1}$ to obtain

$$\frac{d}{dt} \alpha^N(t) + \tilde{F}^N(\alpha^N(t)) = \epsilon [A^N] \alpha^N(t). \quad (1.2.15)$$

For a specified initial condition

$$u(0, x) = \phi(x) \quad (1.2.16)$$

this procedure will yield the corresponding initial condition for (1.2.15):

$$\alpha^N(0) = [M^N]^{-1} v \quad (1.2.17)$$

CHAPTER 1. INTRODUCTION

where $v_j = \langle \phi(x), h_j(x) \rangle = \int_0^1 \phi(x) h_j(x) dx$ for $j = 0, 1, \dots, N + 1$. Again, these computations will be carried out in full detail in Chapter 2.

Chapter 2

TWO APPROXIMATING SYSTEMS

2.1 Piecewise Linear Finite Element Method

Consider the homogeneous Burgers' equation defined on the interval $[0,1]$, with initial condition and Neumann boundary data given by

$$u_t(t, x) + u(t, x)u_x(t, x) = \epsilon u_{xx}(t, x) \quad (2.1.1)$$

$$u(0, x) = \phi(x)$$

$$u_x(t, 0) = 0, u_x(t, 1) = 0$$

where $\epsilon > 0$ is a viscosity coefficient. The approximate solution can be written

$$u^N(t, x) = \sum_{i=0}^{N+1} \alpha_i(t) h_i(x) \quad (2.1.2)$$

where the α_i are unknowns and the h_i are the linear basis functions defined on $[0, 1]$. If $u(t, x)$ solves the initial/boundary value problem given by (2.1.1), then

$$u_t(t, x) + u(t, x)u_x(t, x) - \epsilon u_{xx}(t, x) = 0 \quad (2.1.3)$$

for all $(t, x) \in \mathfrak{R}^2$. Therefore, for $j = 0, 1, \dots, N + 1$,

$$[u_t(t, x) + u(t, x)u_x(t, x) - \epsilon u_{xx}(t, x)]h_j(x) = 0 \quad (2.1.4)$$

and on $[0,1]$

$$\int_0^1 [u_t(t, x) + u(t, x)u_x(t, x)]h_j(x) - \int_0^1 \epsilon u_{xx}(t, x)h_j(x)dx = 0. \quad (2.1.5)$$

CHAPTER 2. TWO APPROXIMATING SYSTEMS

Integrating by parts we obtain:

$$\int_0^1 [u_t(t, x) + u(t, x)u_x(t, x)]h_j(x)dx + \int_0^1 \epsilon u_x(t, x)h'_j(x)dx = 0. \quad (2.1.6)$$

We now substitute our approximate solution $u^N(t, x)$ for $u(t, x)$ to get

$$\int_0^1 \{ [u_t^N(t, x) + u^N(t, x)u_x^N(t, x)]h_j(x) + \epsilon u_x^N(t, x)h'_j(x) \} dx = 0. \quad (2.1.7)$$

Note that $u_t^N(t, x) = \sum_{i=0}^{N+1} \dot{\alpha}_i(t)h_i(x)$ and $u_x^N(t, x) = \sum_{i=0}^{N+1} \alpha_i(t)h'_i(x)$. Hence,

$$\begin{aligned} \int_0^1 \left[\left(\sum_{i=0}^{N+1} \dot{\alpha}_i(t)h_i(x) \right) + \left(\sum_{i=0}^{N+1} \alpha_i(t)h_i(x) \right) \left(\sum_{k=0}^{N+1} \alpha_k(t)h'_k(x) \right) \right] h_j(x)dx \\ = -\epsilon \int_0^1 \left(\sum_{i=0}^{N+1} \alpha_i(t)h'_i(x) \right) h'_j(x)dx. \end{aligned} \quad (2.1.8)$$

Rearranging terms then gives

$$\begin{aligned} \sum_{i=0}^{N+1} \left\{ \int_0^1 h_i(x)h_j(x)dx \right\} \dot{\alpha}_i(t) + \sum_{i=0}^{N+1} \sum_{k=0}^{N+1} \left\{ \int_0^1 h_i(x)h'_k(x)h_j(x)dx \right\} \alpha_i(t)\alpha_k(t) \\ = -\epsilon \sum_{i=0}^{N+1} \left\{ \int_0^1 h'_i(x)h'_j(x)dx \right\} \alpha_i(t). \end{aligned} \quad (2.1.9)$$

Let $m_{ij} = \int_0^1 h_i(x)h_j(x)dx$ and $k_{ij} = -\int_0^1 h'_i(x)h'_j(x)dx$ for $i = 0, 1, \dots, N + 1, j = 0, 1, \dots, N + 1$. Then

$$\sum_{i=0}^{N+1} m_{ij} \dot{\alpha}_i(t) + \sum_{i=0}^{N+1} \sum_{k=0}^{N+1} \left\{ \int_0^1 h_i(x)h'_k(x)h_j(x)dx \right\} \alpha_i(t)\alpha_k(t) = \epsilon \sum_{i=0}^{N+1} k_{ij}\alpha_i(t). \quad (2.1.10)$$

Written as a matrix equation,

$$[M^N] \begin{bmatrix} \dot{\alpha}_0(t) \\ \dot{\alpha}_1(t) \\ \vdots \\ \dot{\alpha}_{N+1}(t) \end{bmatrix} + F_E^N(\alpha(t)) = \epsilon [K^N] \begin{bmatrix} \alpha_0(t) \\ \alpha_1(t) \\ \vdots \\ \alpha_{N+1}(t) \end{bmatrix} \quad (2.1.11)$$

where $[M^N] = [m_{ij}]$, $[K^N] = [k_{ij}]$, and

CHAPTER 2. TWO APPROXIMATING SYSTEMS

$$F_E^N(\alpha(t)) = \frac{1}{6} \begin{bmatrix} -2(\alpha_0(t))^2 + \alpha_0(t)\alpha_1(t) + (\alpha_1(t))^2 \\ -(\alpha_0(t))^2 - \alpha_0(t)\alpha_1(t) + \alpha_1(t)\alpha_2(t) + (\alpha_2(t))^2 \\ \vdots \\ -(\alpha_{N-1}(t))^2 - \alpha_{N-1}(t)\alpha_N(t) + \alpha_N(t)\alpha_{N+1}(t) + (\alpha_{N+1}(t))^2 \\ -(\alpha_N(t))^2 - \alpha_N(t)\alpha_{N+1}(t) + 2(\alpha_{N+1}(t))^2 \end{bmatrix}.$$

Since the mass matrix $[M^N]$ is known to be nonsingular

$$\begin{bmatrix} \dot{\alpha}_0(t) \\ \dot{\alpha}_1(t) \\ \vdots \\ \dot{\alpha}_{N+1}(t) \end{bmatrix} + \tilde{F}_E^N(\alpha(t)) = \epsilon A^N \begin{bmatrix} \alpha_0(t) \\ \alpha_1(t) \\ \vdots \\ \alpha_{N+1}(t) \end{bmatrix} \quad (2.1.12)$$

where $\tilde{F}_E^N(\alpha(t)) = [M^N]^{-1}F_E^N(\alpha(t))$ and $A^N = [M^N]^{-1}[K^N]$.

We now give a similar treatment for the initial condition $u^N(0, x) = \phi(x)$. Multiplying both sides by $h_j(x)$ and integrating on the interval $[0, 1]$ yields

$$\int_0^1 u^N(0, x)h_j(x)dx = \int_0^1 \phi(x)h_j(x)dx. \quad (2.1.13)$$

Therefore,

$$\int_0^1 \left[\sum_{i=0}^{N+1} \alpha_i(0)h_i(x) \right] h_j(x)dx = \int_0^1 \phi(x)h_j(x)dx. \quad (2.1.14)$$

As before, we can rearrange the left hand side of equation (2.1.14) to obtain

$$\sum_{i=0}^{N+1} \left[\int_0^1 h_i(x)h_j(x)dx \right] \alpha_i(0) = \int_0^1 \phi(x)h_j(x)dx. \quad (2.1.15)$$

But $\int_0^1 h_i(x)h_j(x)dx = m_{ij}$ as defined previously. Hence

$$\sum_{i=0}^{N+1} m_{ij}\alpha_i(0) = \int_0^1 \phi(x)h_j(x)dx \quad (2.1.16)$$

CHAPTER 2. TWO APPROXIMATING SYSTEMS

or in matrix form

$$[M^N] \begin{bmatrix} \alpha_0(0) \\ \alpha_1(0) \\ \vdots \\ \alpha_{N+1}(0) \end{bmatrix} = \begin{bmatrix} \langle \phi(x), h_0(x) \rangle \\ \langle \phi(x), h_1(x) \rangle \\ \vdots \\ \langle \phi(x), h_{N+1}(x) \rangle \end{bmatrix} \quad (2.1.17)$$

where $\langle \phi(x), h_j(x) \rangle = \int_0^1 \phi(x)h_j(x)dx$ for $j = 0, 1, \dots, N+1$. Again, since the mass matrix is known to be nonsingular, we can multiply (from the left) both sides of equation (2.1.17) by $[M^N]^{-1}$ to obtain

$$\begin{bmatrix} \alpha_0(0) \\ \alpha_1(0) \\ \vdots \\ \alpha_{N+1}(0) \end{bmatrix} = [M^N]^{-1} \begin{bmatrix} \langle \phi(x), h_0(x) \rangle \\ \langle \phi(x), h_1(x) \rangle \\ \vdots \\ \langle \phi(x), h_{N+1}(x) \rangle \end{bmatrix}. \quad (2.1.18)$$

Note that

$$v = \begin{bmatrix} \langle \phi(x), h_0(x) \rangle \\ \langle \phi(x), h_1(x) \rangle \\ \vdots \\ \langle \phi(x), h_{N+1}(x) \rangle \end{bmatrix}$$

is simply an $(N+2) \times 1$ vector of scalars and therefore $[M^N]^{-1}v$ is as well. So the problem has reduced to solving the $(N+2)$ dimensional initial value problem

$$\dot{\alpha}^N(t) = \epsilon A^N \alpha^N(t) - \tilde{F}_E^N(\alpha^N(t)) \quad (2.1.19)$$

$$\alpha^N(0) = [M^N]^{-1}v. \quad (2.1.20)$$

Note that if $f(t, x) \neq 0$, then equation (2.1.19) is modified by adding

$$f^N(t) = [M^N]^{-1} \begin{bmatrix} \langle f(t, x), h_0(x) \rangle \\ \langle f(t, x), h_1(x) \rangle \\ \vdots \\ \langle f(t, x), h_{N+1}(x) \rangle \end{bmatrix}$$

CHAPTER 2. TWO APPROXIMATING SYSTEMS

to the right hand side. The above IVP was initially set up in MATLAB by Marrekchi (see [12]) and solved using the MATLAB ordinary differential equation solver ODE45. The solution was found to converge to zero for initial condition

$$u^N(0, x) = -\cos(\pi x) \quad (2.1.21)$$

but failed to reach a constant steady state for

$$u^N(0, x) = \cos(\pi x). \quad (2.1.22)$$

To test the validity of these results, we completely regenerated the ODE system using newly coded MATLAB subroutines. We then solved the system subject to various initial conditions and Reynolds numbers using ODE45. The numerical results are given in Chapter 3.

2.2 Finite Element Method and Conservation Form of Burgers' Equation

Consider the conservation form of the homogeneous Burgers' equation [9] given by

$$u_t(t, x) + \frac{1}{2}(u^2(t, x))_x = \epsilon u_{xx}(t, x). \quad (2.2.1)$$

Then for $j = 0, 1, \dots, N + 1$,

$$\int_0^1 \left[u_t(t, x) + \frac{1}{2}(u^2(t, x))_x \right] h_j(x) dx = \epsilon \int_0^1 u_{xx}(t, x) h_j(x) dx. \quad (2.2.2)$$

Integrating the term on the right by parts, we obtain

$$\int_0^1 \left[u_t(t, x) + \frac{1}{2}(u^2(t, x))_x \right] h_j(x) dx = -\epsilon \int_0^1 u_x(t, x) h'_j(x) dx. \quad (2.2.3)$$

Suppose in addition to the trial solution

$$u^N(t, x) = \sum_{i=0}^{N+1} \alpha_i(t) h_i(x) \quad (2.2.4)$$

for $u(t, x)$, we introduce the following trial solution for $u^2(t, x)$:

$$[u^N(t, x)]^2 = \sum_{i=0}^{N+1} (\alpha_i(t))^2 h_i(x). \quad (2.2.5)$$

CHAPTER 2. TWO APPROXIMATING SYSTEMS

Substituting into equation (2.2.3) gives

$$\int_0^1 \left\{ [u_t^N(t, x) + \frac{1}{2}[(u^N(t, x))]_x^2] \right\} h_j(x) = -\epsilon \int_0^1 u_x^N(t, x) h_j'(x) dx. \quad (2.2.6)$$

Note that $[(u^N(t, x))]_x^2 = \sum_{i=0}^{N+1} (\alpha_i(t))^2 h_i'(x)$ and as before $u_t^N(t, x) = \sum_{i=0}^{N+1} \dot{\alpha}_i(t) h_i(x)$ and $u_x^N(t, x) = \sum_{i=0}^{N+1} \alpha_i(t) h_i'(x)$. We make the appropriate substitutions to obtain

$$\begin{aligned} \int_0^1 \left\{ \sum_{i=0}^{N+1} \dot{\alpha}_i(t) h_i(x) + \frac{1}{2} \sum_{i=0}^{N+1} (\alpha_i(t))^2 h_i'(x) \right\} h_j(x) dx \\ = -\epsilon \int_0^1 \sum_{i=0}^{N+1} \alpha_i(t) h_i'(x) h_j'(x) dx. \end{aligned} \quad (2.2.7)$$

If we rearrange terms as in section 2.1, we get

$$\begin{aligned} \sum_{i=0}^{N+1} \left\{ \int_0^1 h_i(x) h_j(x) dx \right\} \dot{\alpha}_i(t) + \frac{1}{2} \sum_{i=0}^{N+1} \left\{ \int_0^1 h_i'(x) h_j(x) dx \right\} (\alpha_i(t))^2 \\ = -\epsilon \sum_{i=0}^{N+1} \left\{ \int_0^1 h_i'(x) h_j'(x) dx \right\} \alpha_i(t) \end{aligned} \quad (2.2.8)$$

which can be written

$$\sum_{i=0}^{N+1} m_{ij} \dot{\alpha}_i(t) + \sum_{i=0}^{N+1} b_{ij} (\alpha_i(t))^2 = -\epsilon \sum_{i=0}^{N+1} k_{ij} \alpha_i(t) \quad (2.2.9)$$

where $b_{ij} = \frac{1}{2} \int_0^1 h_i'(x) h_j(x) dx$ and m_{ij}, k_{ij} are as defined in section 2.1. In matrix form,

$$[M^N] \begin{bmatrix} \dot{\alpha}_0(t) \\ \dot{\alpha}_1(t) \\ \vdots \\ \dot{\alpha}_{N+1}(t) \end{bmatrix} + [B^N] \begin{bmatrix} \alpha_0(t)^2 \\ \alpha_1(t)^2 \\ \vdots \\ \alpha_{N+1}(t)^2 \end{bmatrix} = \epsilon [K^N] \begin{bmatrix} \alpha_0(t) \\ \alpha_1(t) \\ \vdots \\ \alpha_{N+1}(t) \end{bmatrix} \quad (2.2.10)$$

CHAPTER 2. TWO APPROXIMATING SYSTEMS

where

$$[B^N] = [b_{ij}] = \frac{1}{4} \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 0 & 1 & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & & & -1 & 0 & 1 \\ 0 & \cdots & & 0 & -1 & 1 \end{bmatrix}_{(N+2) \times (N+2)} \quad (2.2.11)$$

and $[M^N]$, $[K^N]$ are as in section 1.2. Multiplying $[B^N](\alpha^N(t))^2$ gives

$$F_D^N(\alpha^N(t)) = \frac{1}{4} \begin{bmatrix} (\alpha_1(t))^2 - (\alpha_0(t))^2 \\ (\alpha_2(t))^2 - (\alpha_0(t))^2 \\ (\alpha_3(t))^2 - (\alpha_1(t))^2 \\ \vdots \\ (\alpha_{N+1}(t))^2 - (\alpha_{N-1}(t))^2 \\ (\alpha_{N+1}(t))^2 - (\alpha_N(t))^2 \end{bmatrix} \quad (2.2.12)$$

which can be compared to the $F_E^N(\alpha^N(t))$ term in the previous section. Thus, equation (2.2.10) becomes

$$[M^N] \begin{bmatrix} \dot{\alpha}_0(t) \\ \dot{\alpha}_1(t) \\ \vdots \\ \dot{\alpha}_{N+1}(t) \end{bmatrix} + F_D^N(\alpha(t)) = \epsilon [K^N] \begin{bmatrix} \alpha_0(t) \\ \alpha_1(t) \\ \vdots \\ \alpha_{N+1}(t) \end{bmatrix}. \quad (2.2.13)$$

Finally, since $[M^N]$ is invertible, we obtain

$$\alpha^N(t) = \epsilon A^N \alpha^N(t) - \tilde{F}_D^N(\alpha^N(t)) \quad (2.2.14)$$

where A^N is defined as before and $\tilde{F}_D^N(\alpha^N(t)) = [M^N]^{-1} F_D^N(\alpha(t))$. Again, if $f(t, x) \neq 0$ then $f^N(t)$ (as defined in the previous section) must be added to the right hand side of equation (2.2.14).

Chapter 3

NUMERICAL EXPERIMENTS

3.1 Setting Up the Problem

We consider the nonhomogeneous Burgers' equation

$$u_t(t, x) + u(t, x)u_x(t, x) = \epsilon u_{xx}(t, x) + f(t, x) \quad (3.1.1)$$

with initial condition

$$u(0, x) = \phi(x) \quad (3.1.2)$$

and Neumann boundary condition

$$u_x(t, 0) = u_x(t, 1) = 0. \quad (3.1.3)$$

We are interested in obtaining approximate solutions of equation (3.1.1) subject to (3.1.2) and (3.1.3) with $f(t, x) = 0$. The Galerkin model and the Galerkin/Conservation model are set up in MATLAB and solved using the ODE45 ordinary differential equation solver.

In nearly all of the examples in this paper we employ the approximation routines using $N=16$ elements. This partitions the x -interval $[0,1]$ into $N+1=17$ subintervals of equal length using $N+2=18$ nodes. A typical solution of the above initial/boundary value problem (with $f(t, x) = 0$) is plotted in Figure 3.1 using $N=4,8,16,32$ elements. Here the initial function is given by $\phi(x) = \frac{1}{2}\cos(\pi x)$ and the value of the Reynolds number and final time are $Re=60$ and $t_f = 10$, respectively. With t fixed at $t = 10$, it is evident that the numerical solution is approaching some limit as we increase the number of elements used. This is certainly consistent with the finite element theory found in [9]. The important issue to note is that the solution obtained using $N=16$ elements is so 'close' to that obtained using $N=32$

CHAPTER 3. NUMERICAL EXPERIMENTS

elements that it is more practical to use $N=16$ elements due to the increased amount of execution time incurred by the ODE solver when going from 16 to 32 elements.

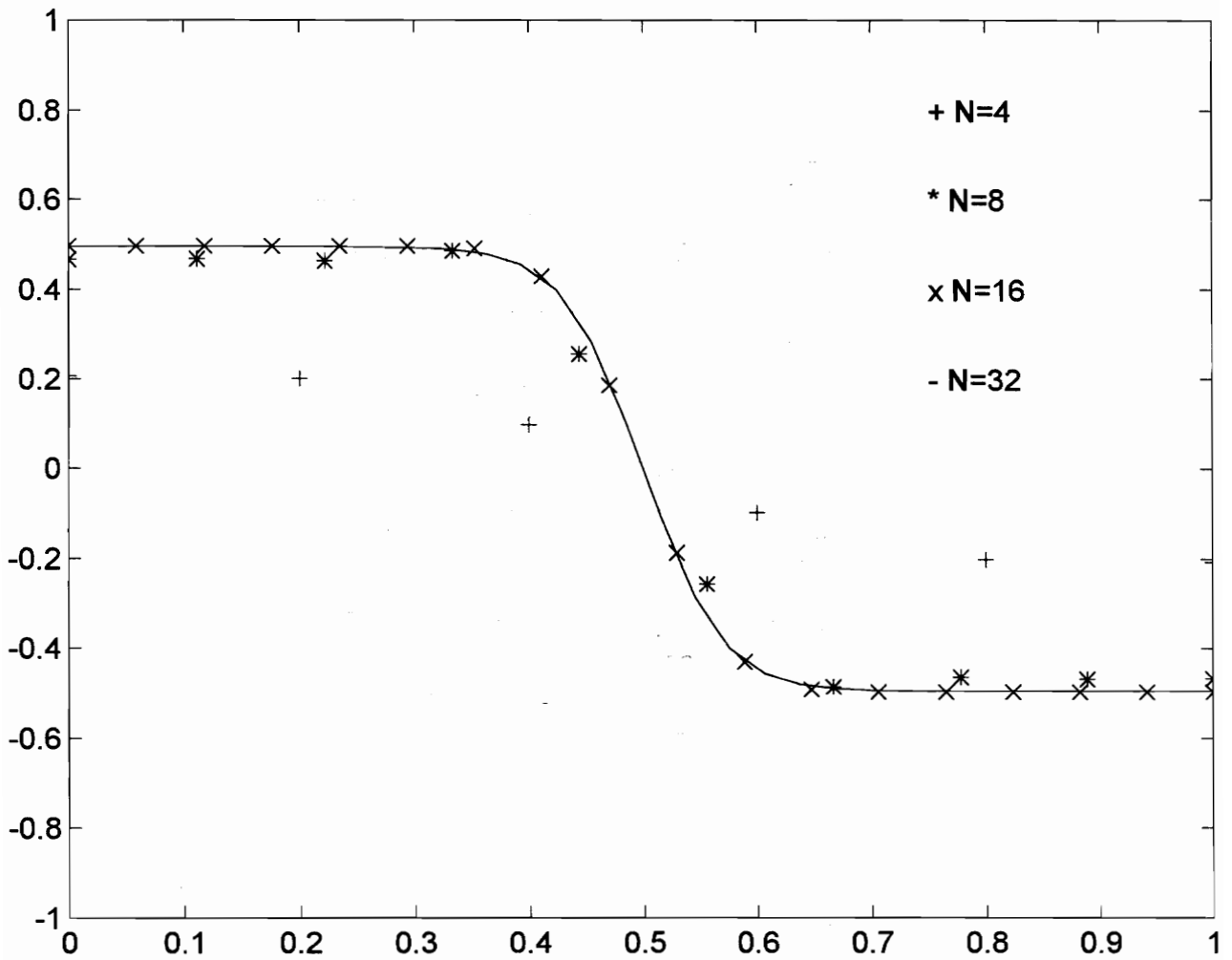


Figure 3.1: Numerical solutions of the homogeneous initial/boundary value problem using $N=4,8,16,32$ elements

CHAPTER 3. NUMERICAL EXPERIMENTS

3.2 A Comparison to Exact and Other Known Numerical Solutions

To measure the numerical error introduced by the two approximation schemes, we compute the exact solution of the above initial/boundary value problem by choosing $f(t, x)$ appropriately. The solution can be written

$$u(t, x) = h(t)\phi(x) \quad (3.2.1)$$

where $h(t)$ is some differentiable function with the property that $h(0) = 1$. Clearly $u(0, x) = h(0)\phi(x) = \phi(x)$, satisfying the initial condition. The following are also observed:

$$u_t(t, x) = \frac{d}{dt}h(t)\phi(x), \quad (3.2.2)$$

$$u_x(t, x) = h(t)\phi'(x), \quad (3.2.3)$$

and

$$u_{xx}(t, x) = h(t)\phi''(x). \quad (3.2.4)$$

Therefore

$$u_t(t, x) + u(t, x)u_x(t, x) - \epsilon u_{xx}(t, x) = \frac{d}{dt}h(t)\phi(x) + h^2(t)\phi(x)\phi'(x) - \epsilon h(t)\phi''(x). \quad (3.2.5)$$

Letting

$$f(t, x) = \frac{d}{dt}h(t)\phi(x) + h^2(t)\phi(x)\phi'(x) - \epsilon h(t)\phi''(x), \quad (3.2.6)$$

$u(t, x)$ as defined by equation (3.2.1) above furnishes the exact solution for equation (3.1.1).

Example 3.2.1

Let

$$h(t) = e^{-\epsilon t} \quad (3.2.7)$$

and

$$\phi(x) = \frac{1}{4}\cos(\pi x). \quad (3.2.8)$$

Then

$$f(t, x) = -\frac{\epsilon}{4}e^{-\epsilon t}\cos(\pi x) - \frac{\pi}{16}e^{-2\epsilon t}\cos(\pi x)\sin(\pi x) + \frac{\epsilon\pi^2}{4}e^{-\epsilon t}\cos(\pi x)$$

CHAPTER 3. NUMERICAL EXPERIMENTS

$$= -\frac{1}{4}e^{-\epsilon t} \cos(\pi x) \left[\epsilon + \frac{\pi}{4}e^{-\epsilon t} \sin(\pi x) - \epsilon \pi^2 \right] \quad (3.2.9)$$

and the exact solution to equation (3.1.1) subject to (3.1.2) and (3.1.3) is given by

$$u(t, x) = h(t)\phi(x) = \frac{1}{4}e^{-\epsilon t} \cos(\pi x). \quad (3.2.10)$$

Tables 3.1, 3.2, and 3.3 compare the exact solution to the Galerkin and Galerkin/Conservation approximations at final time $t = \frac{1}{2}$. The finite dimensional models were set up using $N=16$ elements and the equations were solved with $Re=60,120,240$. The numbers in the bottom row of these tables are the error norms $\|u_G(t, x) - u_e(t, x)\|$ and $\|u_{G/C}(t, x) - u_e(t, x)\|$ for the Galerkin and Galerkin/Conservation approximations, respectively. As the Reynolds number increases these error norms become larger, indicating reduction in accuracy. This is to be expected since $\epsilon = \frac{1}{Re} \rightarrow 0$ corresponds to decreasing viscosity. In all three cases, the Galerkin/Conservation method is the more accurate of the two approximations. This appears to be true in general as we shall see in further examples.

Example 3.2.2

Let

$$\phi(x) = \frac{1}{4} \cos(\pi x) \quad (3.2.11)$$

as in Example 3.2.1 and let

$$h(t) = \cos(t). \quad (3.2.12)$$

Then

$$\begin{aligned} f(t, x) &= -\frac{1}{4} \sin(t) \cos(\pi x) - \frac{\pi}{16} \cos^2(t) \cos(\pi x) \sin(\pi x) + \frac{\epsilon \pi^2}{4} \cos(t) \cos(\pi x) \\ &= -\frac{1}{4} \cos(\pi x) \left[\sin(t) + \frac{\pi}{4} \cos^2(t) \sin(\pi x) - \epsilon \pi^2 \cos(t) \right] \end{aligned} \quad (3.2.13)$$

and the exact solution to equation (3.1.1) subject to (3.1.2) and (3.1.3) is given by

$$u(t, x) = \frac{1}{4} \cos(t) \cos(\pi x). \quad (3.2.14)$$

For $N=16$ and $Re=60,120,240$, Tables 3.4, 3.5, and 3.6 compare the exact solution to the Galerkin and Galerkin/Conservation approximations at final time $t = \frac{1}{2}$. Note that the

CHAPTER 3. NUMERICAL EXPERIMENTS

Galerkin/Conservation method is once again the more accurate of the two approximations. The normal errors associated with the two approximation schemes are smaller than those obtained in Example 3.2.1 for $Re=60,120,240$. One would expect this trend to reverse itself for final time large enough since $u(t, x) = \frac{1}{4}e^{-\epsilon t} \cos(\pi x)$ approaches zero exponentially in time and $u(t, x) = \frac{1}{4} \cos(t) \cos(\pi x)$ is periodic in both t and x .

Example 3.2.3

In this example we compare our solutions to the homogeneous initial/boundary value problem to those obtained by Byrnes, Gilliam, and Shubov [6]. Here the initial function is $\phi(x) = x^2(1-x)^2$ and the Reynolds number is $Re=10$. We obtained (by both the Galerkin method and the Galerkin/Conservation method) the exact results found on page 44 in [6]. Our results are plotted in Figure 3.2 for comparison.

Example 3.2.4

For $\phi(x) = \cos(\pi x)$, $f(t, x) = 0$, and $Re=60$, Marrekchi's solution converged to a non-constant steady state and is plotted in Figure 4.3.1 on page 76 in [12]. Changing only the sign of the initial function (i.e. $\phi(x) = -\cos(\pi x)$) caused the solution to evolve to a constant in less than 10 seconds, as shown in Figure 4.3.2 on page 77 in [12]. We observed the exact same phenomena using both the Galerkin and the Galerkin/Conservation methods. Our results are plotted in Figures 3.3 and 3.4. Note that our solutions are consistent with those obtained by Marrekchi.

CHAPTER 3. NUMERICAL EXPERIMENTS

Table 3.1: Solution of Burgers' equation by Galerkin method, Galerkin/Conservation method (Re=60)

x	Galerkin	G/C	exact
0	0.2505	0.2504	0.2479
0.0588	0.2457	0.2456	0.2437
0.1176	0.2326	0.2325	0.2312
0.1765	0.2118	0.2116	0.2108
0.2353	0.1839	0.1836	0.1832
0.2941	0.1499	0.1496	0.1494
0.3529	0.1108	0.1106	0.1105
0.4118	0.068	0.0679	0.0678
0.4706	0.0229	0.0229	0.0229
0.5294	-0.0229	-0.0229	-0.0229
0.5882	-0.068	-0.0679	-0.0678
0.6471	-0.1108	-0.1106	-0.1105
0.7059	-0.1499	-0.1496	-0.1494
0.7647	-0.1839	-0.1836	-0.1832
0.8235	-0.2118	-0.2116	-0.2108
0.8824	-0.2326	-0.2325	-0.2312
0.9412	-0.2457	-0.2456	-0.2437
1	-0.2505	-0.2504	-0.2479
	0.0053	0.0049	

CHAPTER 3. NUMERICAL EXPERIMENTS

Table 3.2: Solution of Burgers' equation by Galerkin method, Galerkin/Conservation method (Re=120)

x	Galerkin	G/C	exact
0	0.2529	0.2528	0.249
0.0588	0.2474	0.2474	0.2447
0.1176	0.2338	0.2337	0.2321
0.1765	0.2127	0.2125	0.2117
0.2353	0.1846	0.1843	0.184
0.2941	0.1504	0.1502	0.15
0.3529	0.1113	0.111	0.111
0.4118	0.0683	0.0681	0.0681
0.4706	0.023	0.023	0.023
0.5294	-0.023	-0.023	-0.023
0.5882	-0.0683	-0.0681	-0.0681
0.6471	-0.1113	-0.111	-0.111
0.7059	-0.1504	-0.1502	-0.15
0.7647	-0.1846	-0.1843	-0.184
0.8235	-0.2127	-0.2125	-0.2117
0.8824	-0.2338	-0.2337	-0.2321
0.9412	-0.2474	-0.2474	-0.2447
1	-0.2529	-0.2528	-0.249
	0.0073	0.0071	

CHAPTER 3. NUMERICAL EXPERIMENTS

Table 3.3: Solution of Burgers' equation by Galerkin method, Galerkin/Conservation method (Re=240)

x	Galerkin	G/C	exact
0	0.2556	0.2556	0.2495
0.0588	0.249	0.2489	0.2452
0.1176	0.2345	0.2344	0.2326
0.1765	0.213	0.2128	0.2121
0.2353	0.1849	0.1846	0.1844
0.2941	0.1508	0.1504	0.1503
0.3529	0.1115	0.1112	0.1112
0.4118	0.0685	0.0683	0.0683
0.4706	0.0231	0.023	0.023
0.5294	-0.0231	-0.023	-0.023
0.5882	-0.0685	-0.0683	-0.0683
0.6471	-0.1115	-0.1112	-0.1112
0.7059	-0.1508	-0.1504	-0.1503
0.7647	-0.1849	-0.1846	-0.1844
0.8235	-0.213	-0.2128	-0.2121
0.8824	-0.2345	-0.2344	-0.2326
0.9412	-0.249	-0.2489	-0.2452
1	-0.2556	-0.2556	-0.2495
	0.0107	0.0105	

CHAPTER 3. NUMERICAL EXPERIMENTS

Table 3.4: Comparison of Galerkin solution, Galerkin/Conservation solution, and exact solution $u(t, x) = \frac{1}{4}\cos(t)\cos(\pi x)$ (Re=60)

x	Galerkin	G/C	exact
0	0.2217	0.2216	0.2194
0.0588	0.2174	0.2173	0.2157
0.1176	0.2059	0.2057	0.2046
0.1765	0.1874	0.1872	0.1865
0.2353	0.1627	0.1625	0.1621
0.2941	0.1326	0.1324	0.1322
0.3529	0.0981	0.0978	0.0978
0.4118	0.0602	0.06	0.06
0.4706	0.0203	0.0202	0.0202
0.5294	-0.0203	-0.0202	-0.0202
0.5882	-0.0602	-0.06	-0.06
0.6471	-0.0981	-0.0978	-0.0978
0.7059	-0.1326	-0.1324	-0.1322
0.7647	-0.1627	-0.1625	-0.1621
0.8235	-0.1874	-0.1872	-0.1865
0.8824	-0.2059	-0.2057	-0.2046
0.9412	-0.2174	-0.2173	-0.2157
1	-0.2217	-0.2216	-0.2194
	0.0047	0.0044	

CHAPTER 3. NUMERICAL EXPERIMENTS

Table 3.5: Comparison of Galerkin solution, Galerkin/Conservation solution, and exact solution $u(t, x) = \frac{1}{4}\cos(t)\cos(\pi x)$ ($\text{Re}=120$)

x	Galerkin	G/C	exact
0	0.2228	0.2228	0.2194
0.0588	0.2181	0.218	0.2157
0.1176	0.2061	0.206	0.2046
0.1765	0.1874	0.1872	0.1865
0.2353	0.1627	0.1624	0.1621
0.2941	0.1326	0.1323	0.1322
0.3529	0.0981	0.0978	0.0978
0.4118	0.0602	0.06	0.06
0.4706	0.0203	0.0202	0.0202
0.5294	-0.0203	-0.0202	-0.0202
0.5882	-0.0602	-0.06	-0.06
0.6471	-0.0981	-0.0978	-0.0978
0.7059	-0.1326	-0.1323	-0.1322
0.7647	-0.1627	-0.1624	-0.1621
0.8235	-0.1874	-0.1872	-0.1865
0.8824	-0.2061	-0.206	-0.2046
0.9412	-0.2181	-0.218	-0.2157
1	-0.2228	-0.2228	-0.2194
	0.0065	0.0063	

CHAPTER 3. NUMERICAL EXPERIMENTS

Table 3.6: Comparison of Galerkin solution, Galerkin/Conservation solution, and exact solution $u(t, x) = \frac{1}{4}\cos(t)\cos(\pi x)$ ($\text{Re}=240$)

x	Galerkin	G/C	exact
0	0.2249	0.2248	0.2194
0.0588	0.219	0.219	0.2157
0.1176	0.2063	0.2062	0.2046
0.1765	0.1873	0.1871	0.1865
0.2353	0.1626	0.1623	0.1621
0.2941	0.1326	0.1323	0.1322
0.3529	0.0981	0.0978	0.0978
0.4118	0.0602	0.06	0.06
0.4706	0.0203	0.0202	0.0202
0.5294	-0.0203	-0.0202	-0.0202
0.5882	-0.0602	-0.06	-0.06
0.6471	-0.0981	-0.0978	-0.0978
0.7059	-0.1326	-0.1323	-0.1322
0.7647	-0.1626	-0.1623	-0.1621
0.8235	-0.1873	-0.1871	-0.1865
0.8824	-0.2063	-0.2062	-0.2046
0.9412	-0.219	-0.219	-0.2157
1	-0.2249	-0.2248	-0.2194
	0.0095	0.0093	

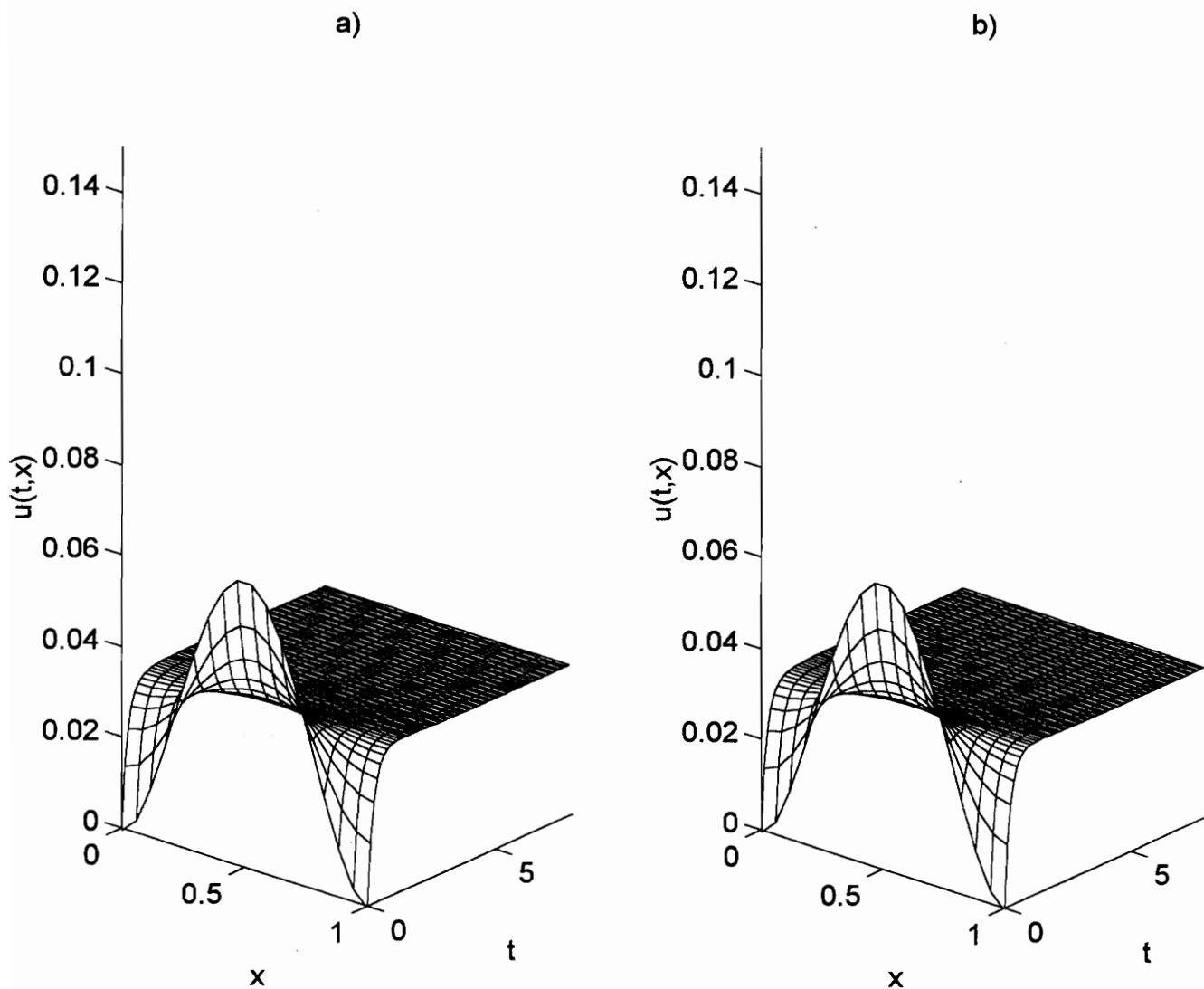
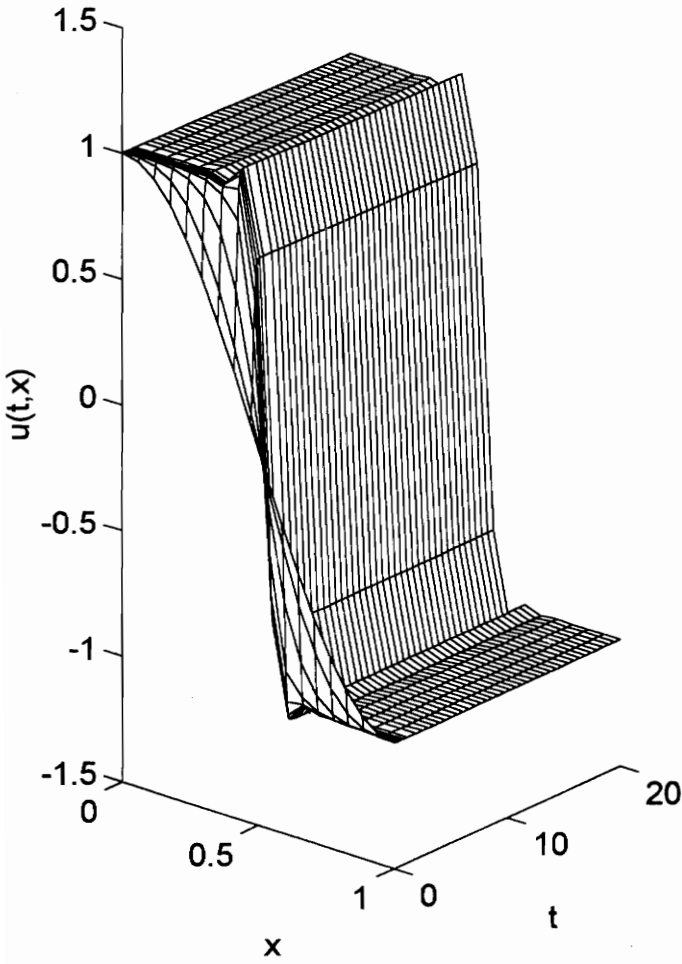


Figure 3.2: $\phi(x) = x^2(1 - x)^2$, $Re=10$, a) Galerkin method, b) Galerkin/Conservation method

a)



b)

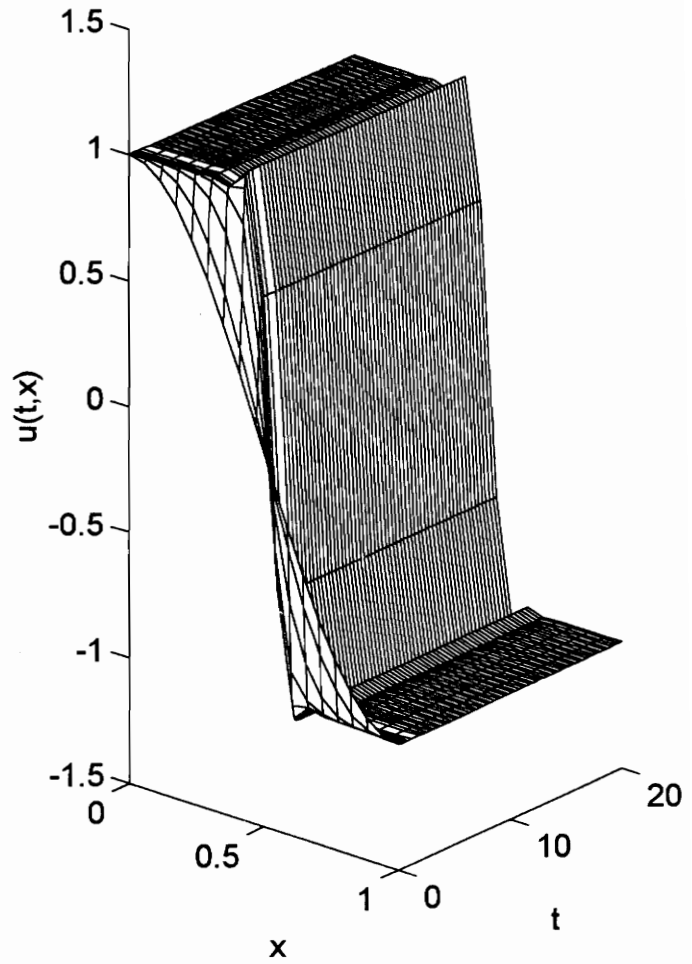
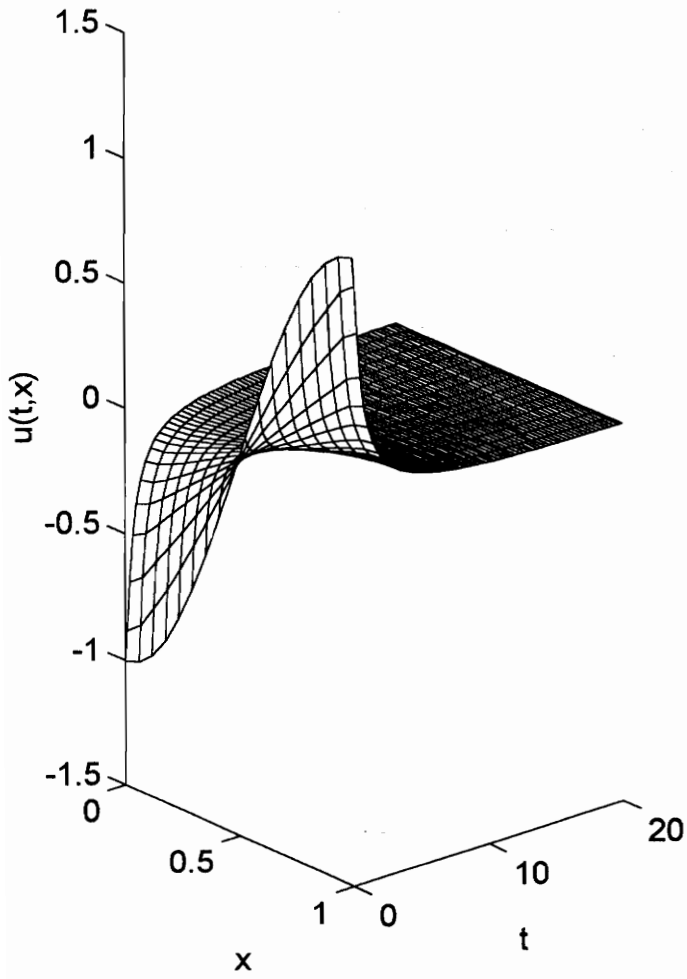


Figure 3.3: $\phi(x) = \cos(\pi x)$, $Re=60$, a) Galerkin method, b) Galerkin/Conservation method

a)



b)

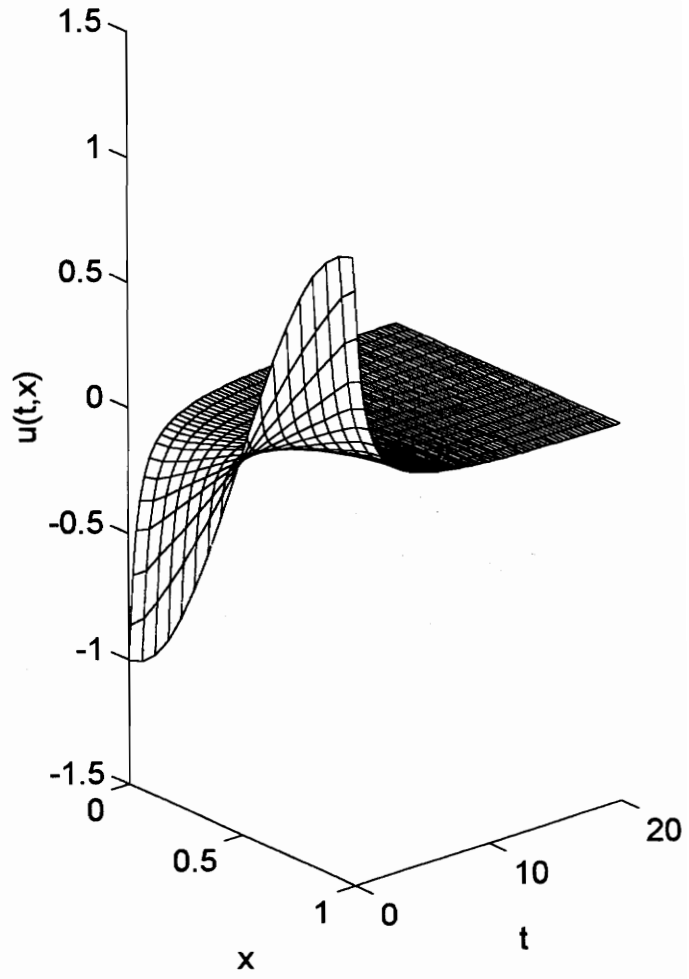


Figure 3.4: $\phi(x) = -\cos(\pi x)$, $\text{Re}=60$, a) Galerkin method, b) Galerkin/Conservation method

CHAPTER 3. NUMERICAL EXPERIMENTS

3.3 Comparison of the Galerkin and Galerkin/Conservation Results

In this section we compare our results obtained by the two approximating systems developed in the preceding chapter. For many combinations of initial conditions and Reynolds numbers, the Galerkin and Galerkin/Conservation methods produce nearly identical results. However, in several examples, the ODE45 ordinary differential equation solver in MATLAB appears to capture the numerical solution of the initial value problem resulting from the Galerkin/Conservation method but seems to break down numerically when solving the IVP associated with the conventional Galerkin method, producing solutions which grow exponentially in time. Moreover, we have seen that the Galerkin/Conservation method is (practically) more accurate than the standard Galerkin method. We will soon demonstrate that the Galerkin/Conservation method is generally faster and therefore yields a greater computational efficiency.

Example 3.3.1

Consider the initial condition given by

$$\phi(x) = A \cos(\pi x) \tag{3.3.1}$$

where A is some constant. We implemented our Galerkin and Galerkin/Conservation methods for all combinations of $A=4,2,1$ and $Re=60,120,240$. Figures 3.5 and 3.6 are MATLAB plots of the results obtained by the Galerkin and Galerkin/Conservation methods, respectively, for initial condition $\phi(x) = \cos(\pi x)$ and Reynolds number $Re=60$. The corresponding equations were solved from $t = 0$ seconds to $t = 20$ seconds. It is apparent that, in this case, both methods produce the same results on the finite x -interval $[0, 1]$. However for $Re=120$ and $Re=240$, the Galerkin solution grows exponentially in time, producing a plot very much different from that obtained by the Galerkin/Conservation method (see Figures 3.7, 3.8, 3.9, and 3.10.) One would expect, based on the results for $Re=60$ and the error estimates in the previous section, that the Galerkin/Conservation solutions (Figures 3.8 and 3.10) are more likely to represent the exact solutions for $Re=120$ and $Re=240$. For initial condition (3.3.1) with $A = 2$ and $A = 4$ we observed the same radical differences in the results of the two

CHAPTER 3. NUMERICAL EXPERIMENTS

approximating schemes. In these cases however, we were unable to obtain similar solutions for any $Re=60,120,240$. Our results (with $Re=60,120$) are plotted in Figures 3.11 and 3.12 for $A = 2$ and $A = 4$, respectively.

It is not known whether the approximate solutions obtained by the Galerkin/Conservation method are indeed true representations of the exact solution, but there is more evidence to support this claim than there is for the case that the solution grows exponentially with time. Note that the Galerkin/Conservation solution evolves in time from its initial state to a shape which is not everywhere constant. This seems to be the case in general for $A \geq \frac{1}{2}$ as we shall see in the next section.

Example 3.3.2

For this example, we selected eleven different initial functions (some differing only by a scalar) and recorded the execution time associated with solving (via ODE45) the resulting Galerkin and Galerkin/Conservation models. We solved the ODE systems on short, medium, and long time intervals and used various Reynolds numbers. Our results are located in Table 3.7. Note that the Galerkin/Conservation method is computationally faster than the Galerkin Method in all of these scenarios except the one tabulated in the second row of data (highlighted.) Although these conditions were not chosen at random, they are constant multiples of every initial condition considered in this paper. Therefore, for all practical purposes, we conclude (without any sort of proof) that the Galerkin/Conservation method is generally faster than the conventional Galerkin method. This means that the computational efficiency (i.e. the achieved accuracy per unit execution time) associated with the Galerkin/Conservation method should be generally better than that of the conventional Galerkin method.

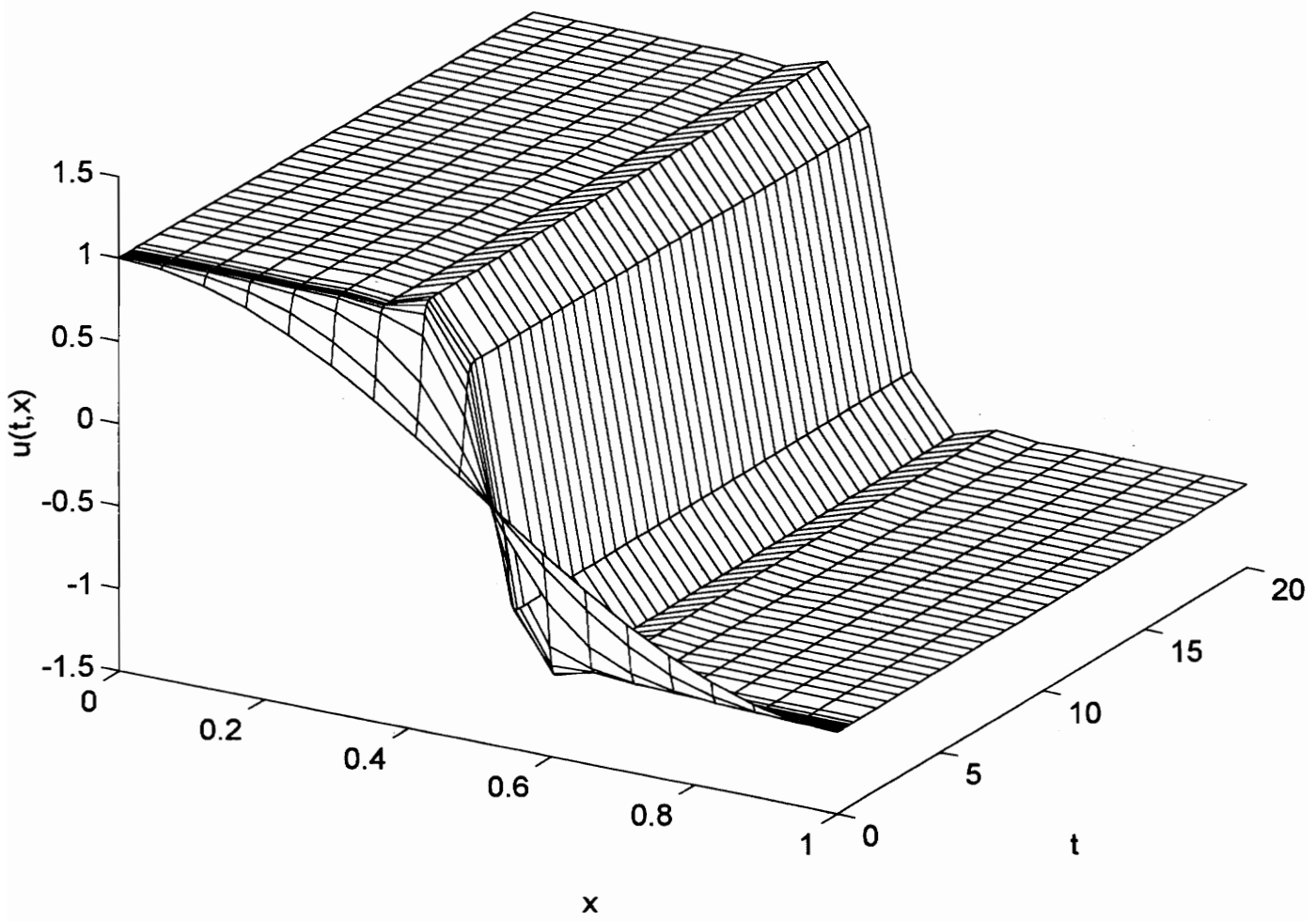


Figure 3.5: $\phi(x) = \cos(\pi x)$, $Re=60$, Galerkin method

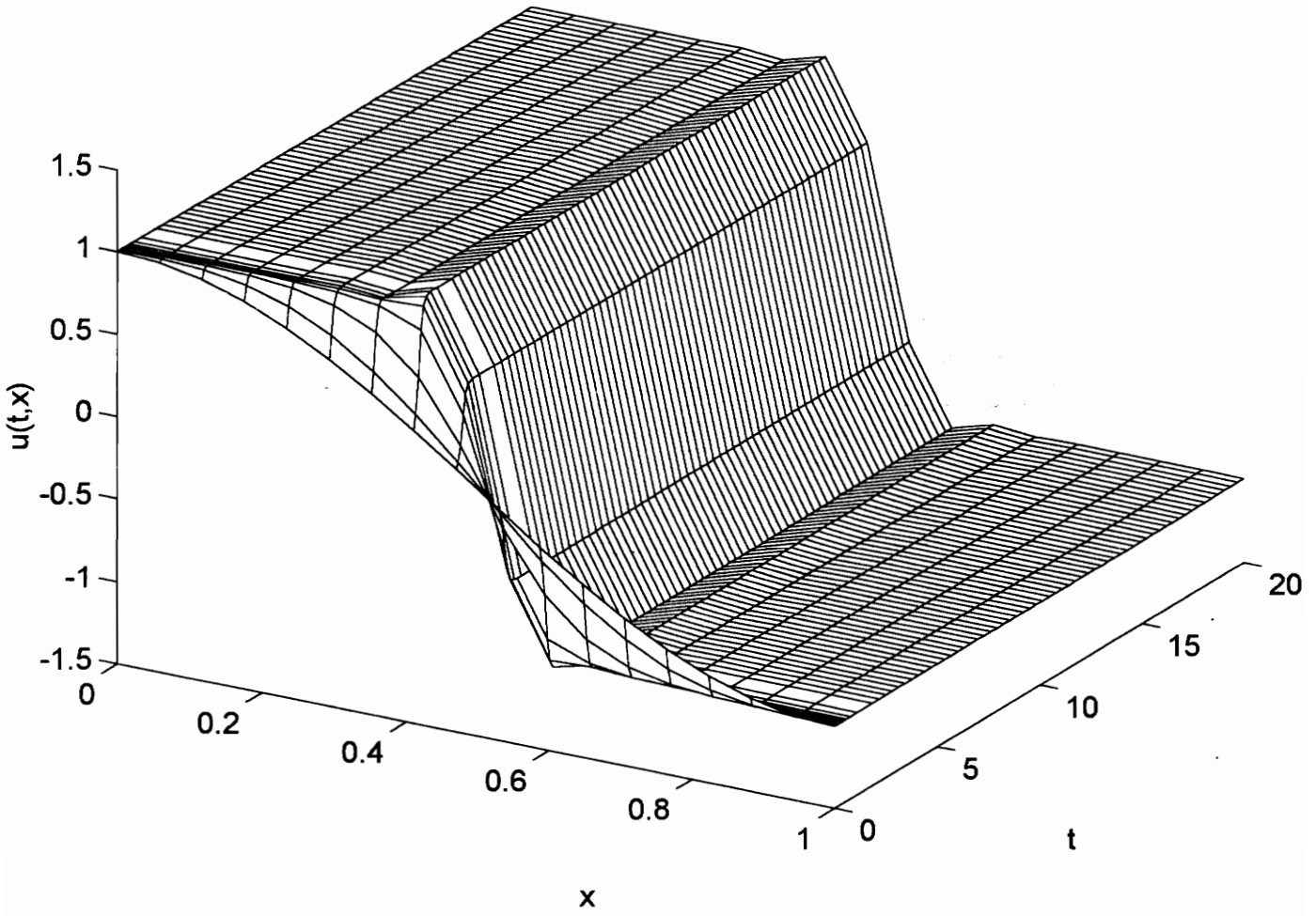


Figure 3.6: $\phi(x) = \cos(\pi x)$, $Re=60$, Galerkin/Conservation method

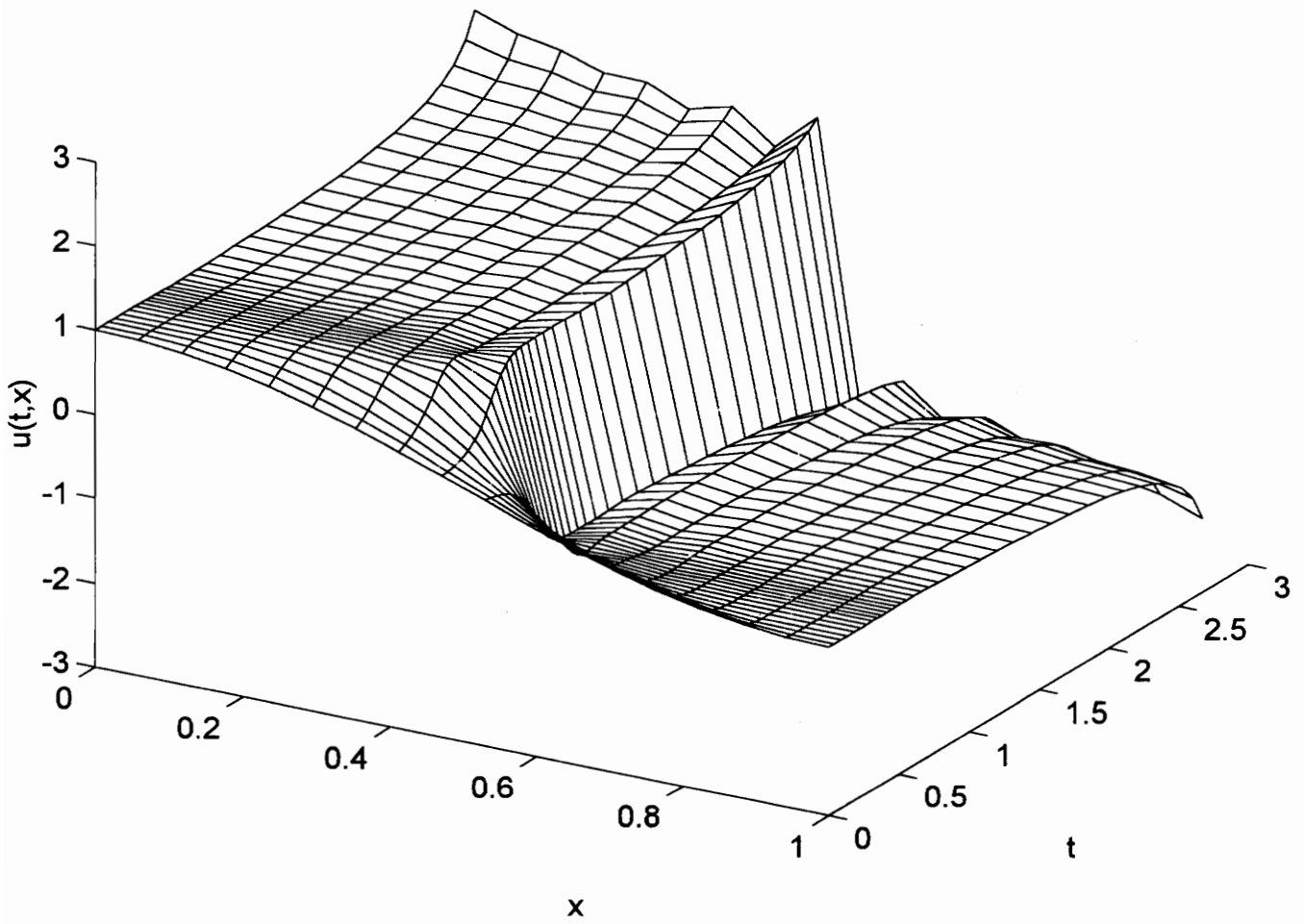


Figure 3.7: $\phi(x) = \cos(\pi x)$, $Re=120$, Galerkin method

CHAPTER 3. NUMERICAL EXPERIMENTS

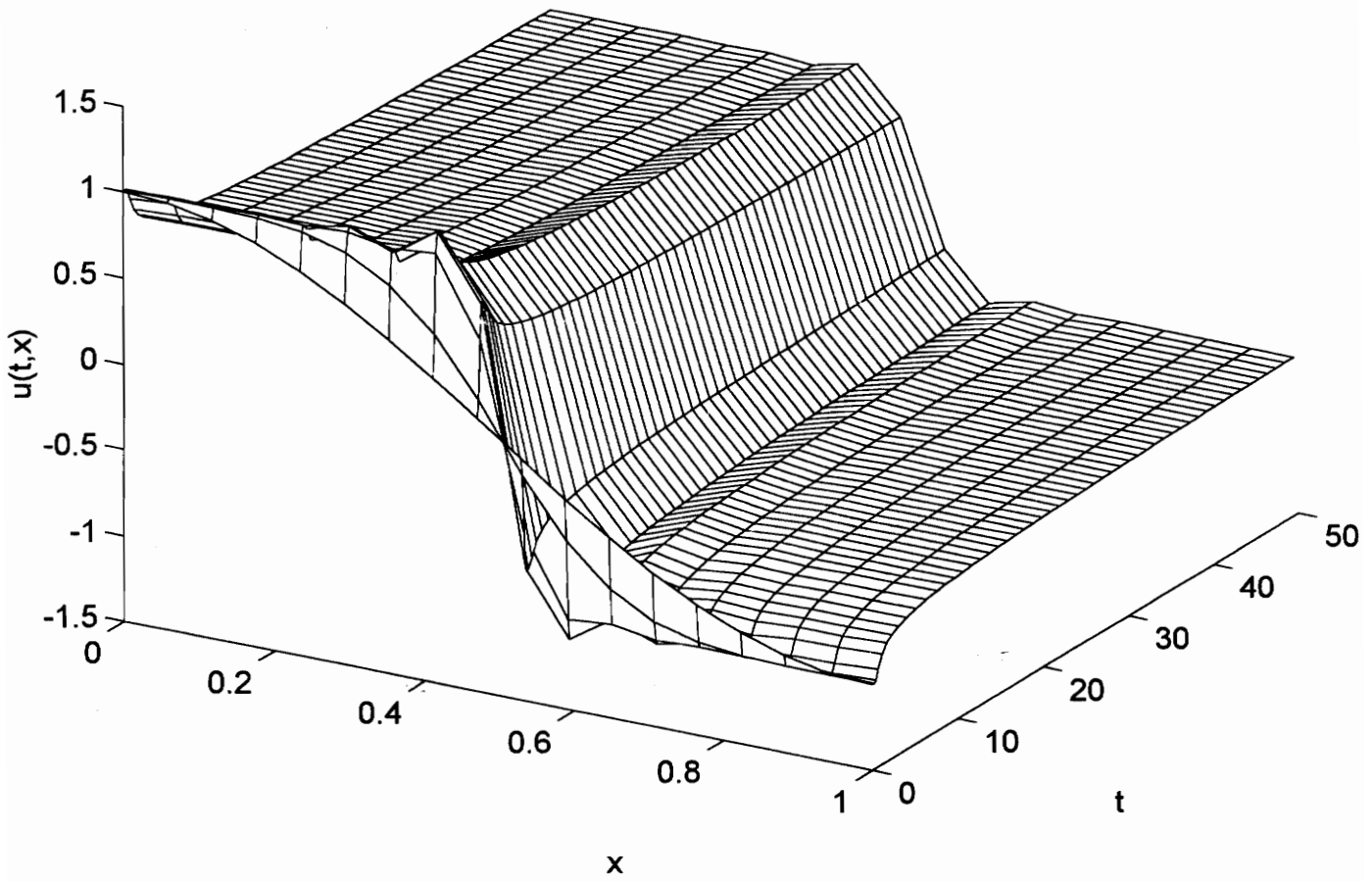


Figure 3.8: $\phi(x) = \cos(\pi x)$, $Re=120$, Galerkin/Conservation method

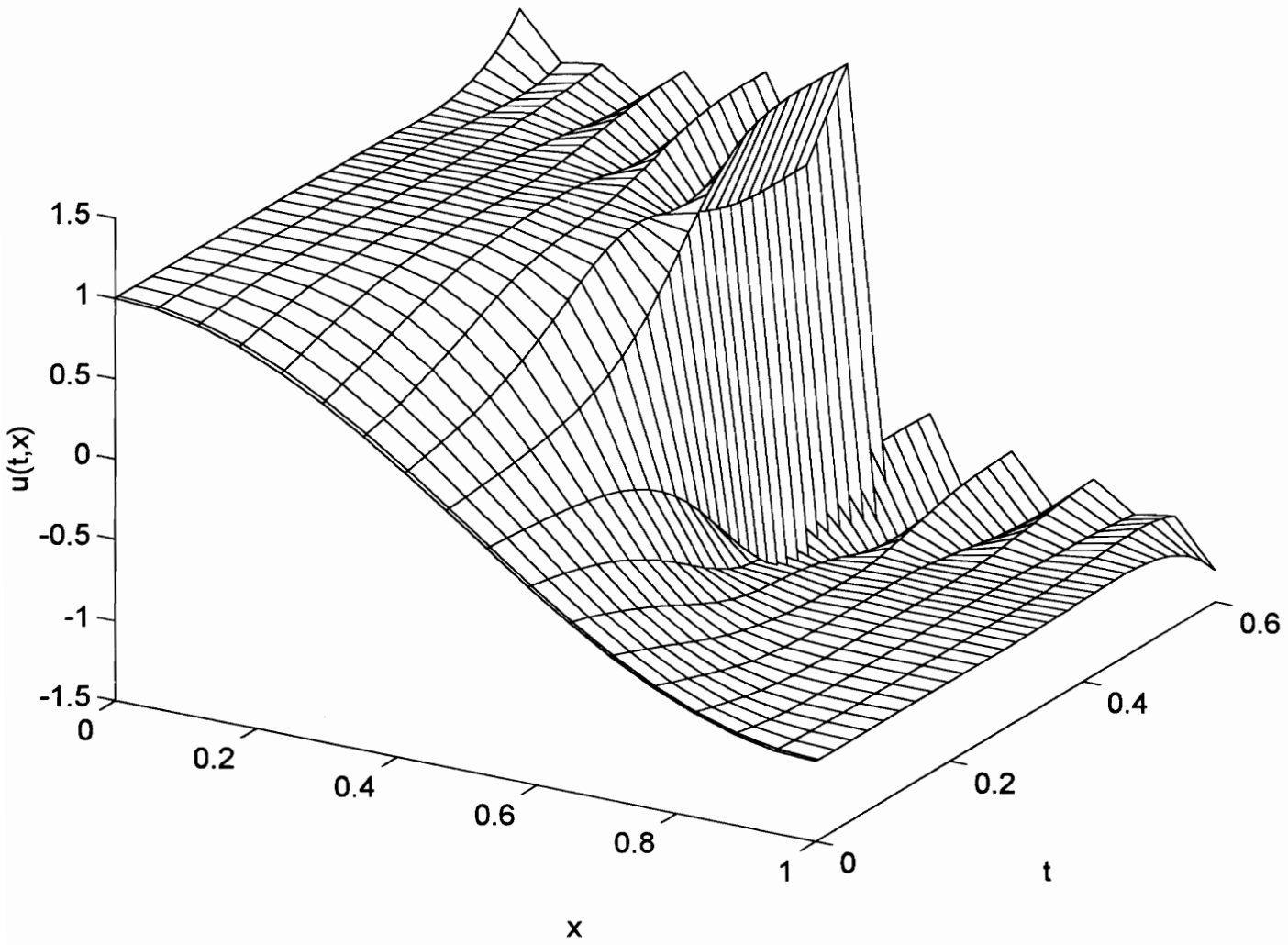


Figure 3.9: $\phi(x) = \cos(\pi x)$, $Re=240$, Galerkin method

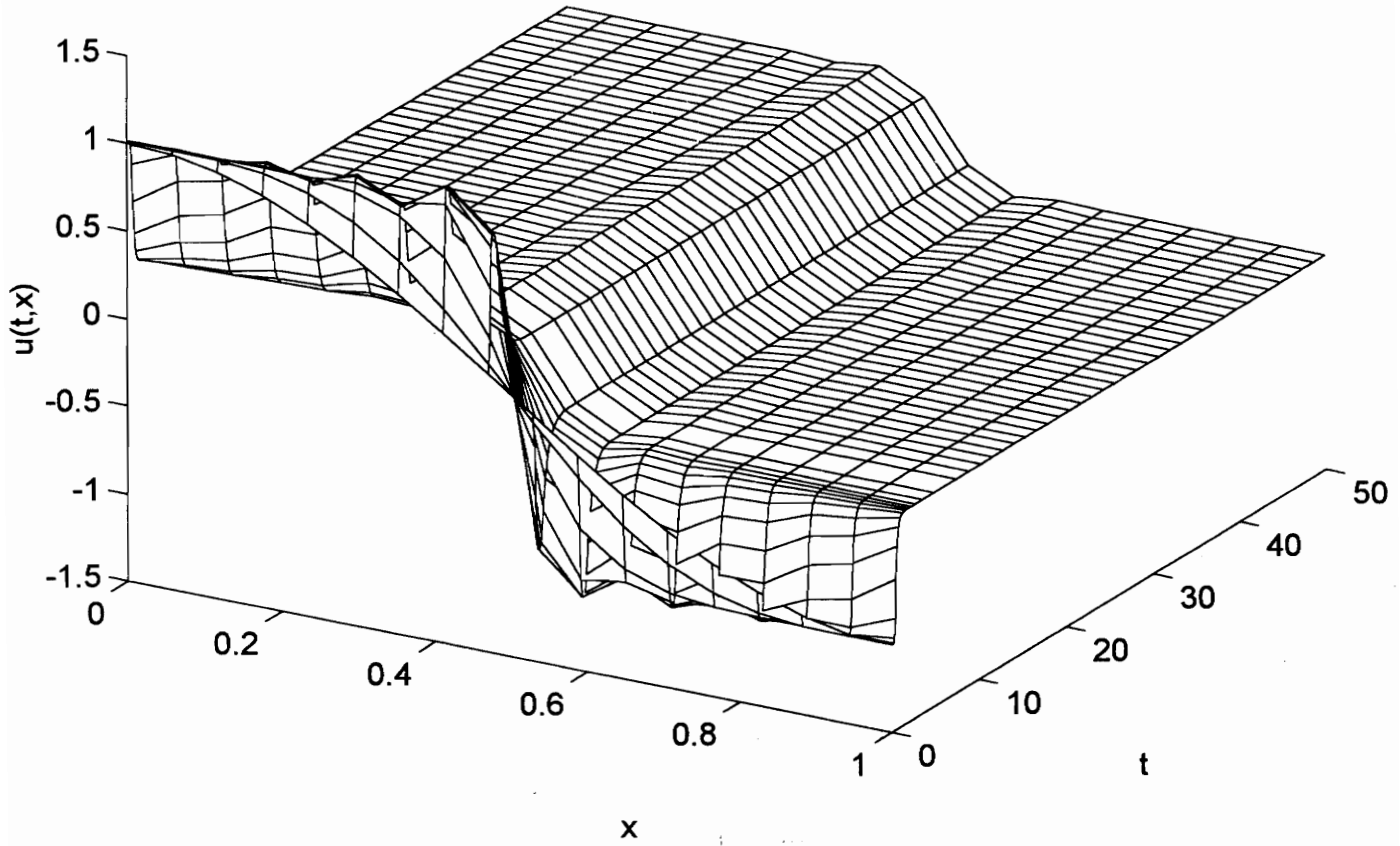


Figure 3.10: $\phi(x) = \cos(\pi x)$, $\text{Re}=240$, Galerkin/Conservation method

CHAPTER 3. NUMERICAL EXPERIMENTS

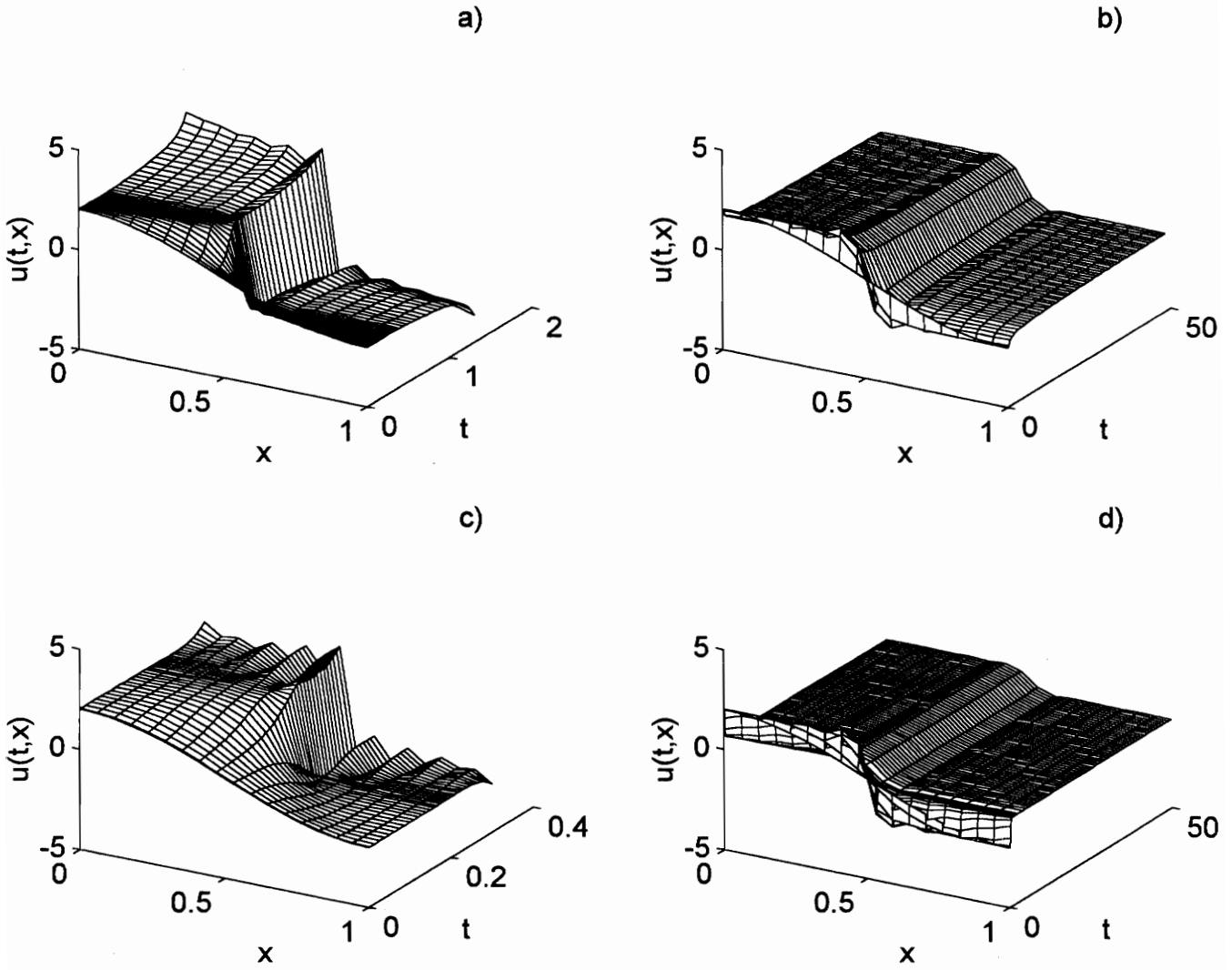


Figure 3.11: $\phi(x) = 2\cos(\pi x)$, a) $Re=60$, Galerkin method, b) $Re=60$, G/C method, c) $Re=120$, Galerkin method, d) $Re=120$, G/C method

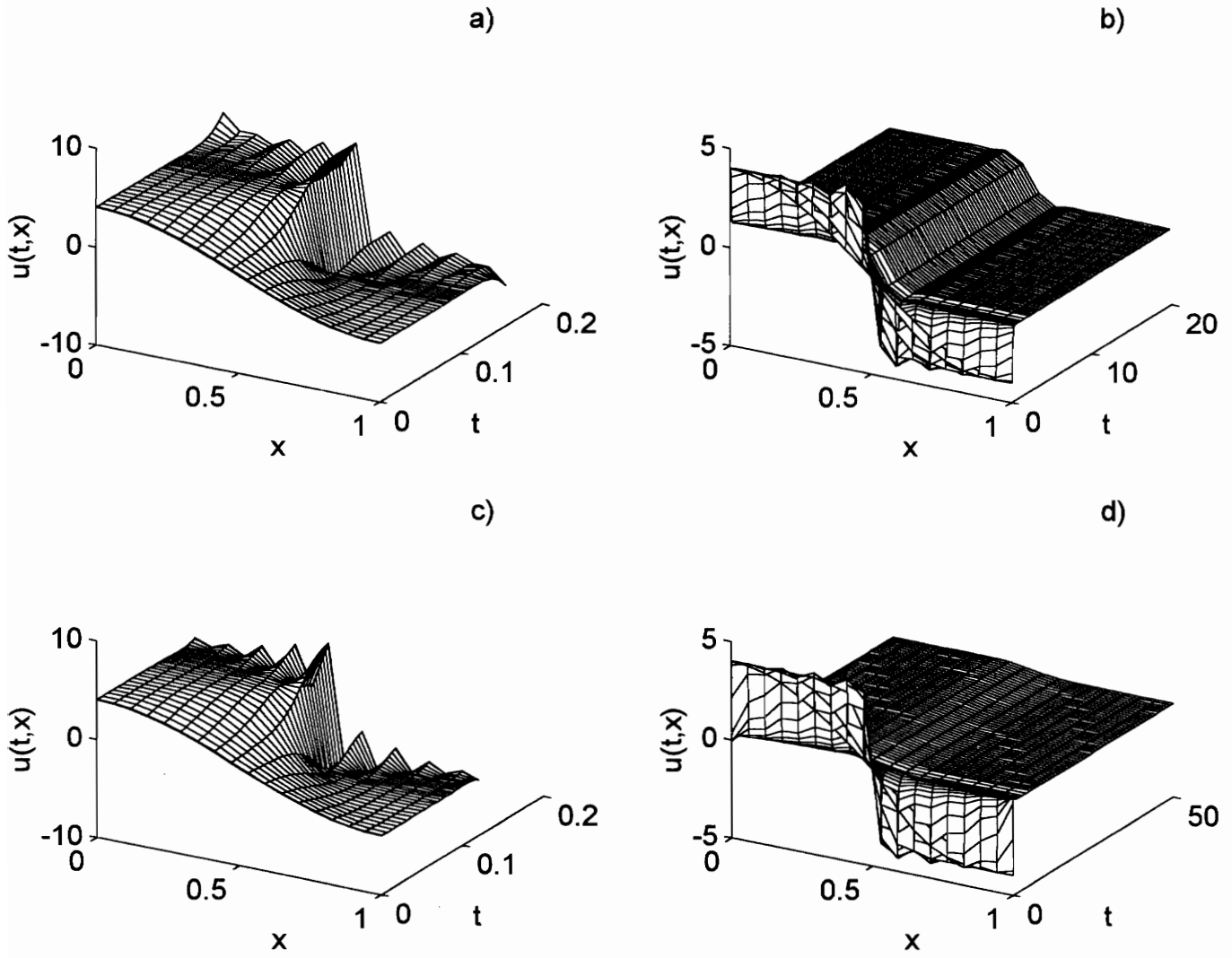


Figure 3.12: $\phi(x) = 4\cos(\pi x)$, a) $Re=60$, Galerkin method, b) $Re=60$, G/C method, c) $Re=120$, Galerkin method, d) $Re=120$, G/C method

CHAPTER 3. NUMERICAL EXPERIMENTS

Table 3.7: Comparison of Galerkin Elapsed Time and Galerkin/Conservation Elapsed Time in Seconds

Initial Function	Re	Final Time	Galerkin Elapsed Time	G/C Elapsed Time	Time Differential
1	60	20	42.68	37.73	4.95
1	120	20	15.49	18.34	-2.85
2	60	200	543.05	479.5	63.55
2	120	50	48.77	44.44	4.33
3	120	500	669.71	590.94	78.77
4	60	5	13.79	11.26	2.53
4	120	5	9.4	7.53	1.87
4	240	5	8.63	6.98	1.65
5	60	2	16.14	12.68	3.46
6	60	2	34.39	26.97	7.42
7	60	3	12.25	9.78	2.47
8	60	3	23.5	18.72	4.78
9	60	20	44.6	37.96	6.64
10	60	25	52.84	43.72	9.12
11	60	25	53.06	44.33	8.73

Initial functions by number:

1. $\phi(x) = \frac{1}{2}\cos(\pi x)$
2. $\phi(x) = \frac{1}{4}\cos(\pi x)$
3. $\phi(x) = \frac{1}{8}\cos(\pi x)$
4. $\phi(x) = \sin(\pi x)$
5. $\phi(x) = 100x^2(1-x)^2$
6. $\phi(x) = 1000x^2(1-x)^2$
7. $\phi(x) = 100x^4(1-x)^2$
8. $\phi(x) = 1000x^4(1-x)^2$
9. $\phi(x) = \frac{1}{4}\cos(\pi x) + \frac{1}{4}\sin(\pi x)$
10. $\phi(x) = \frac{1}{4}\cos(2\pi x)$
11. $\phi(x) = \frac{1}{4}\cos(3\pi x)$

CHAPTER 3. NUMERICAL EXPERIMENTS

3.4 Asymptotic Behavior of Finite Element Solutions

In this section we examine the different types of behavior of solutions to the initial/boundary value problem given various initial conditions and Reynolds numbers. Although the Galerkin and Galerkin/Conservation methods give similar solutions in many cases, we have concluded that the Galerkin/Conservation method is more likely to capture the basic evolution of the solution and that this approximation is generally more accurate. We would like to be able to categorize which combinations of initial conditions and Reynolds numbers yield solutions which converge to a constant in finite time. To actually do this is beyond the scope of this paper and is the subject of ongoing scientific research. Instead, we present data which supports existing basic theoretical findings and offer our results as a reference material for future research.

Example 3.4.1

Consider initial condition (3.3.1) with $A = \frac{1}{16}$. For Reynolds numbers $Re=60,120,240$, we solved the initial/boundary value problem using both the Galerkin method and the Galerkin/Conservation method. The two methods produced virtually identical results (see Figures 3.13, 3.14, 3.15), indicating that the approximations are sufficient for representing the exact solution. Contrary to Example 3.3.1, the solution evolved from its initial state to a constant for each value of Re . Moreover, it is apparent that for larger Reynolds numbers the solution takes more time to reach this constant state. We observed the same phenomena for $A = \frac{1}{32}$.

Examples 3.3.1 and 3.4.1 support the hypothesis that the solution will converge to a constant for $\phi(x)$ sufficiently small. This is consistent with findings that the 'size' of $\phi(x)$ is an important factor in categorizing initial conditions for which the solution will evolve to a constant (see [6]). The following example illustrates how the value of the Reynolds number is also a contributing factor for constant steady-state convergence.

Example 3.4.2

In this example we solve the homogeneous initial/boundary value problem with initial

CHAPTER 3. NUMERICAL EXPERIMENTS

condition (3.3.1) for the cases of $A = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$. We consider the three cases one at a time for convenience.

Case 1: $A = \frac{1}{2}$

Figures 3.16, 3.17, and 3.18 illustrate the results obtained by the two methods for $Re=60, 120, 240$, respectively. Note that for $Re=60$ and $Re=120$ the two approximating schemes produce very similar results and the solution again goes to a state which is not everywhere constant. For $Re=240$, however, the Galerkin solution once again grows exponentially with time and the Galerkin/Conservation method produces a more likely solution, very similar to those for $Re=60, 120$.

Case 2: $A = \frac{1}{4}$

In this case, both methods give the same results indicating that we have indeed obtained a close representation of the exact solution. For $Re=60$, the solution reached a constant state after about 100 seconds, as shown in Figure 3.19. This was not true for $Re=120$ and $Re=240$. For these values of Re , the solution reached states similar to those we have observed in previous examples (see Figures 3.20 and 3.21.)

Case 3: $A = \frac{1}{8}$

Here both methods again give the same results. This time the solution converges to a constant for $Re=60$ and $Re=120$ (see Figures 3.22 and 3.23) and fails to do so for $Re=240$ (Figure 3.24). With $Re=60$, the solution reaches this constant state in less than 50 seconds. When the Reynolds number is increased to $Re=120$, the solution takes nearly 350 seconds to reach the constant state. This phenomena was also observed in Example 3.4.1.

We have given examples which support Byrnes, Gilliam, and Shubov's assertion that the solution will converge to a constant if the initial data is smaller in norm than some bound which depends on the value of the Reynolds number. It is worthwhile to note, however, that this condition is not a necessary one. That is, it is not necessary for the initial data to be 'small' in order for the solution to converge to a constant.

Example 3.4.3

In this example we fix the Reynolds number at $Re=60$ and solve the homogeneous ini-

CHAPTER 3. NUMERICAL EXPERIMENTS

tial/boundary value problem subject to initial conditions $\phi(x) = \sin(\pi x)$, $\phi(x) = 10\sin(\pi x)$, and $\phi(x) = 50\sin(\pi x)$. The results are plotted in Figures 3.25, 3.26, and 3.27 and scaled for comparison. Notice that the solutions obtained by the two approximating systems are virtually identical and both converge to a constant from all three initial states. Hence, it is clearly not necessary that the initial data be 'small' in order for the solution to evolve to a constant.

Example 3.4.4

Recall Example 3.2.3 where we considered the Byrnes, Gilliam, and Shubov (BGS) initial function $\phi(x) = x^2(1-x)^2$. We solved the initial/boundary value problem subject to this initial condition (with $\text{Re}=10$) and found the solution to converge to a constant in finite time. We also solved the IBVP subject to scalar multiples of the BGS function and plotted our results in Figure 3.28. Here the Reynolds number was $\text{Re}=60$ and the scalars used were 100 and 1000. Note that in both cases the solution evolved to a constant in finite time.

Examples 3.4.3 and 3.4.4 demonstrate solutions of the IBVP which are fairly large and yet still converge to a constant. In both examples, however, the initial functions have the property that they are symmetric with respect to the perpendicular bisector of the x -interval $[0, 1]$. To check that this property has no direct correlation with constant steady-state convergence of solutions, we seek an initial function which behaves in a similar manner but is skewed about the x -interval.

Example 3.4.5

The sixth-order polynomial function $\phi(x) = x^4(1-x)^2$ behaves much like the fourth-order polynomial BGS function but is somewhat skewed. We implemented both approximating procedures for the IBVP subject to the initial data $\phi(x) = 100x^4(1-x)^2$ and $\phi(x) = 1000x^4(1-x)^2$ with Reynolds number $\text{Re}=60$. Although neither initial function is symmetric, both solutions evolved to a constant in finite time. Our plots can be found in Figure 3.29.

Example 3.4.6

CHAPTER 3. NUMERICAL EXPERIMENTS

In this example, we consider several initial functions of the form $\phi(x) = A\cos(\pi x) + B\sin(\pi x)$ where A and B take on the values $\frac{1}{4}$, $\frac{1}{2}$, and 1. For each run we fixed the Reynolds number at $Re=60$ and the two approximation methods gave virtually the same numerical solutions. Our results obtained by the Galerkin/Conservation procedure are plotted in Figures 3.30, 3.31, 3.32, and 3.33. The solutions converged to a constant for $A = \frac{1}{2}$ and $A = \frac{1}{4}$ but failed to do so for the (A, B) pairs $(1, \frac{1}{4})$ and $(1, \frac{1}{2})$. In the latter cases, it does not appear that the solutions will converge to a steady-state at all. We applied both approximation procedures and solved the IBVP from initial time $t_0 = 0$ to final time $t_f = 74$ for $(A, B) = (1, \frac{1}{4})$ and $(A, B) = (1, \frac{1}{2})$. Upon examination of the plots, it is quite evident that the mesh produced by the approximating procedures becomes much more dense at t -values less than but near final time $t_f = 74$. This phenomena most likely indicates that very rapid changes are occurring in the evolution of the solutions at these values of t and the ODE45 solver must use a smaller step-size in order to capture these changes. It is also apparent that, for both of these initial conditions, some sort of 'blow-up' occurs around $t = 74$. These 'blow-ups' could be appropriate representations of the actual solutions or they could be the result of a 'break-down' in the ordinary differential equation solver; we are unable to determine at this time.

Example 3.4.7

Consider initial functions of the form $\phi(x) = A\cos(C\pi x)$ where $0 < A < 1$ is a real number and C is a positive integer. For each combination of $A = \frac{1}{4}, \frac{1}{2}$, $C=2,3$, and $Re=60,120$, we approximated solutions to the corresponding initial/boundary value problem using both the Galerkin method and the Galerkin/Conservation method. Our Galerkin/Conservation results (which are nearly identical to the conventional Galerkin results) are plotted in Figures 3.34, 3.35, 3.36, and 3.37.

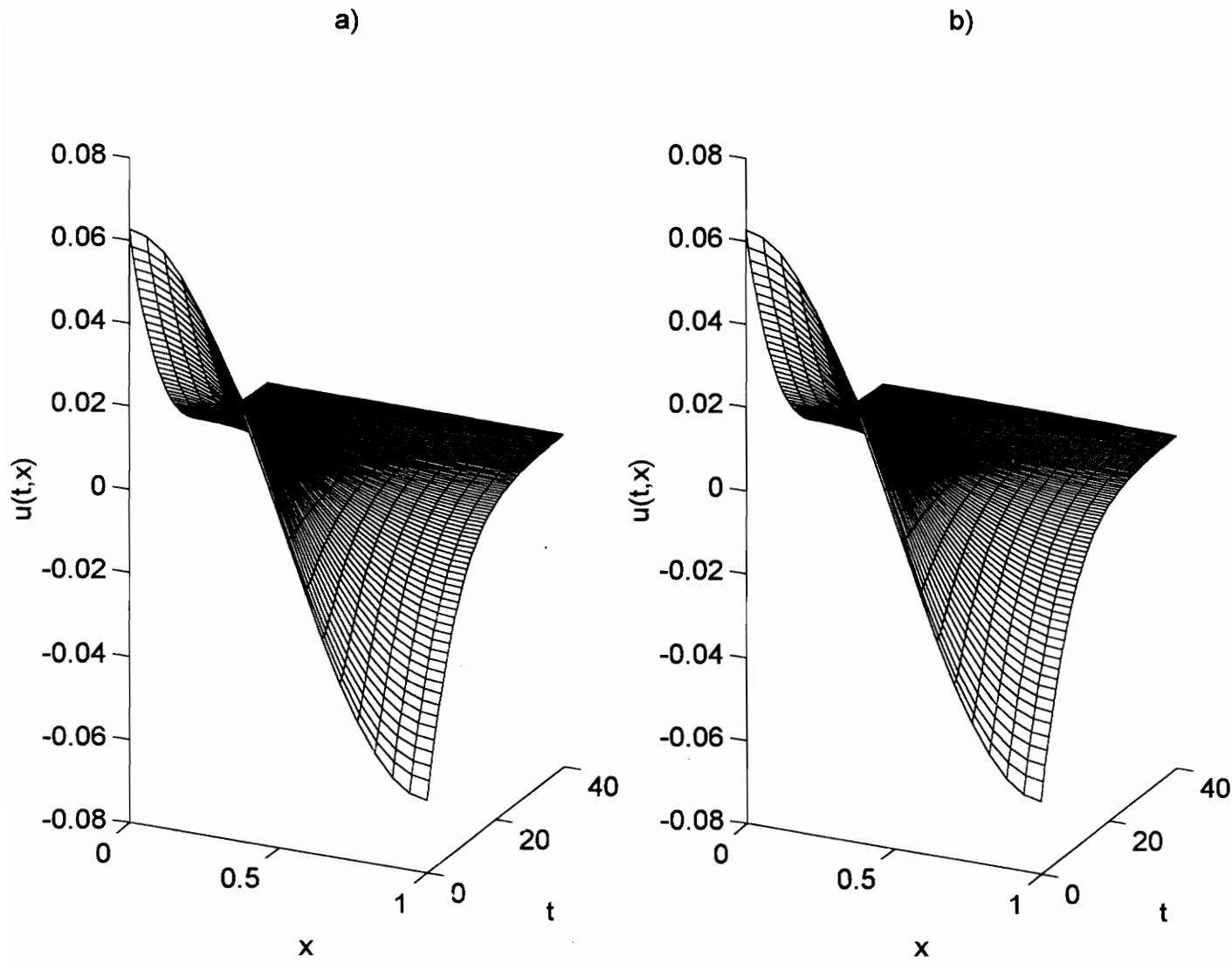


Figure 3.13: $\phi(x) = \frac{1}{16} \cos(\pi x)$ ($\text{Re}=60$): a) Galerkin method, b) G/C method

a)

b)

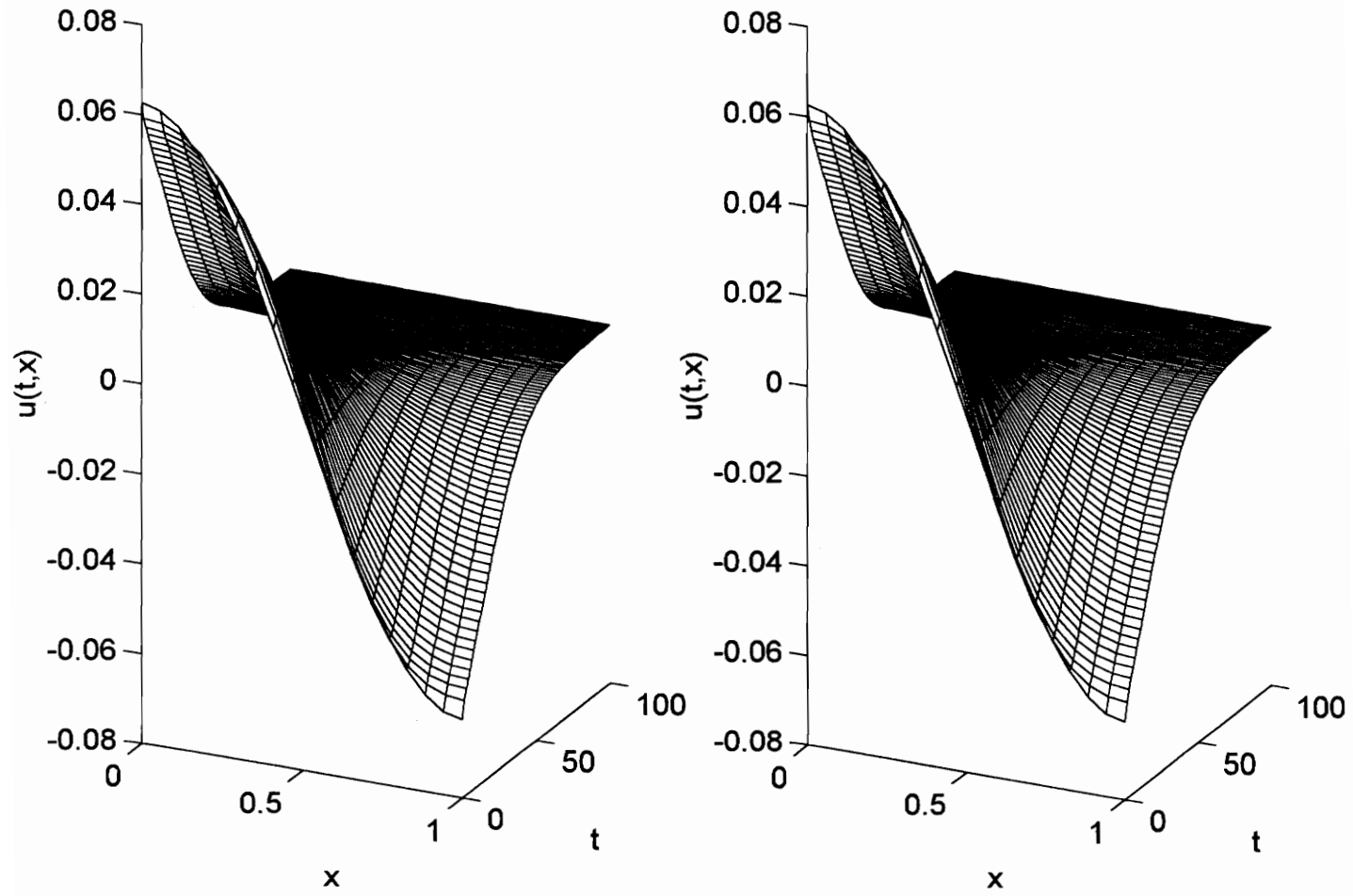


Figure 3.14: $\phi(x) = \frac{1}{16}\cos(\pi x)$ ($\text{Re}=120$): a) Galerkin method, b) G/C method

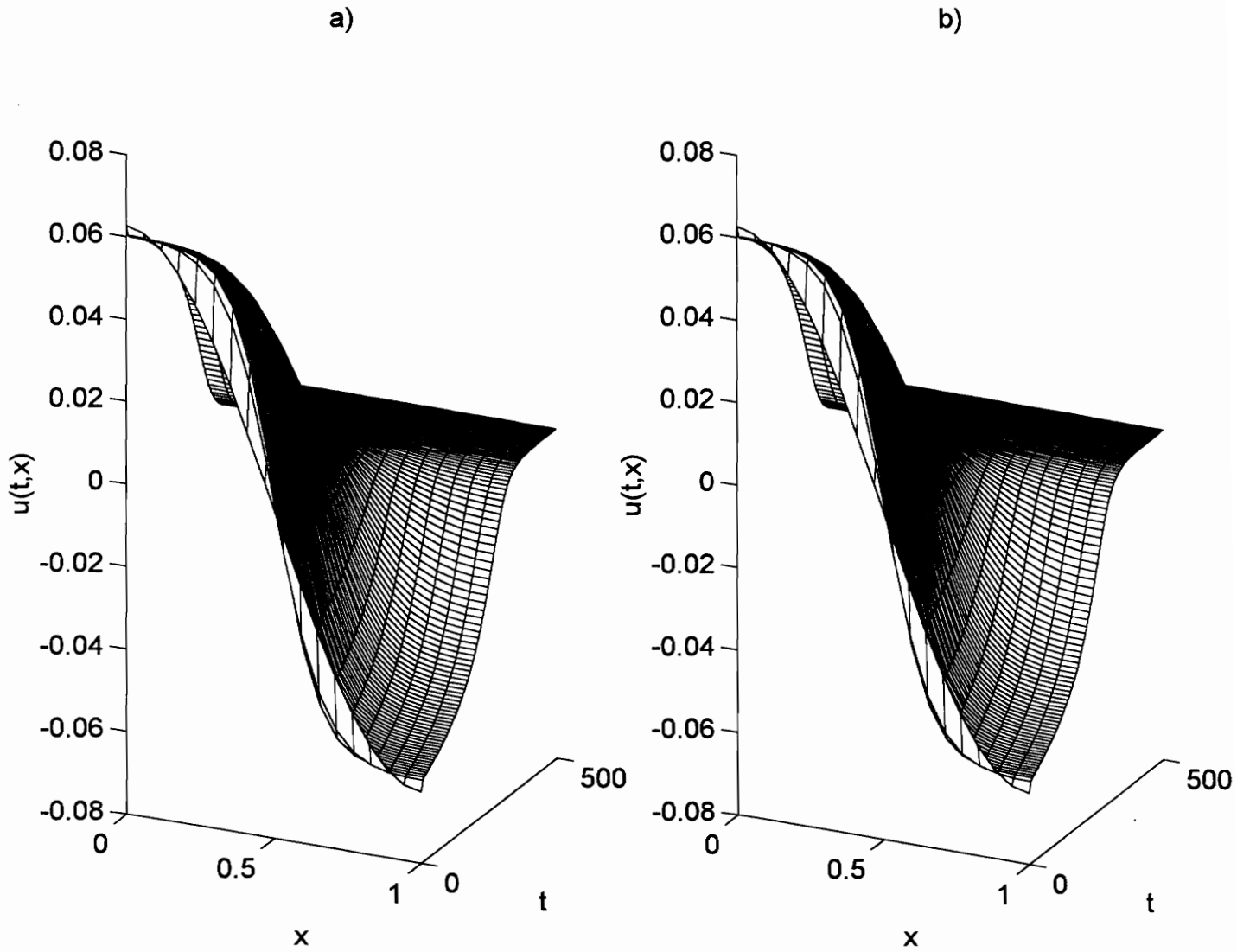


Figure 3.15: $\phi(x) = \frac{1}{16}\cos(\pi x)$ ($\text{Re}=240$): a) Galerkin method, b) G/C method

a)

b)

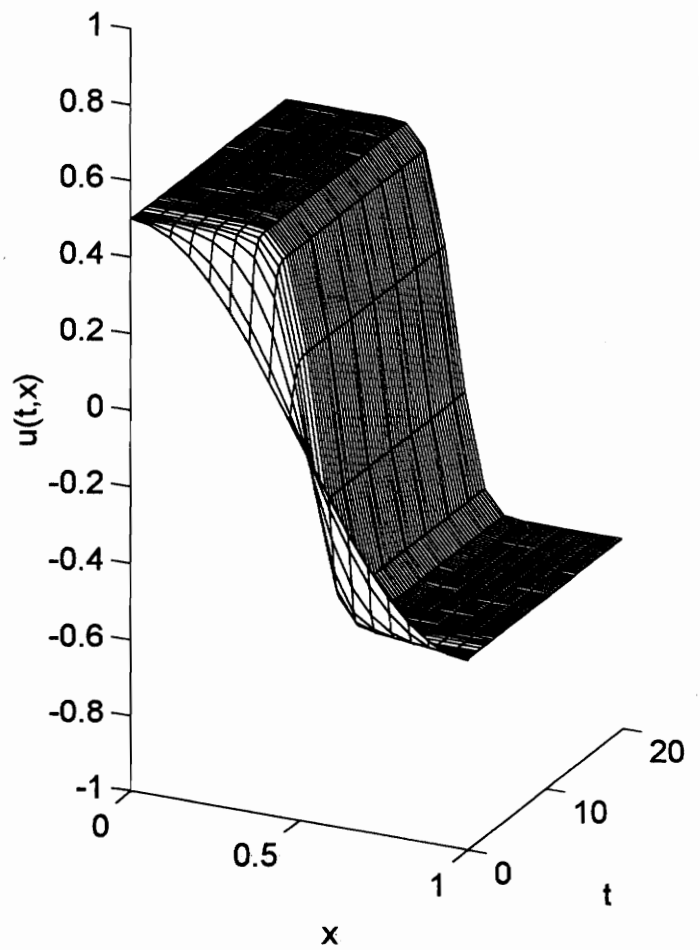
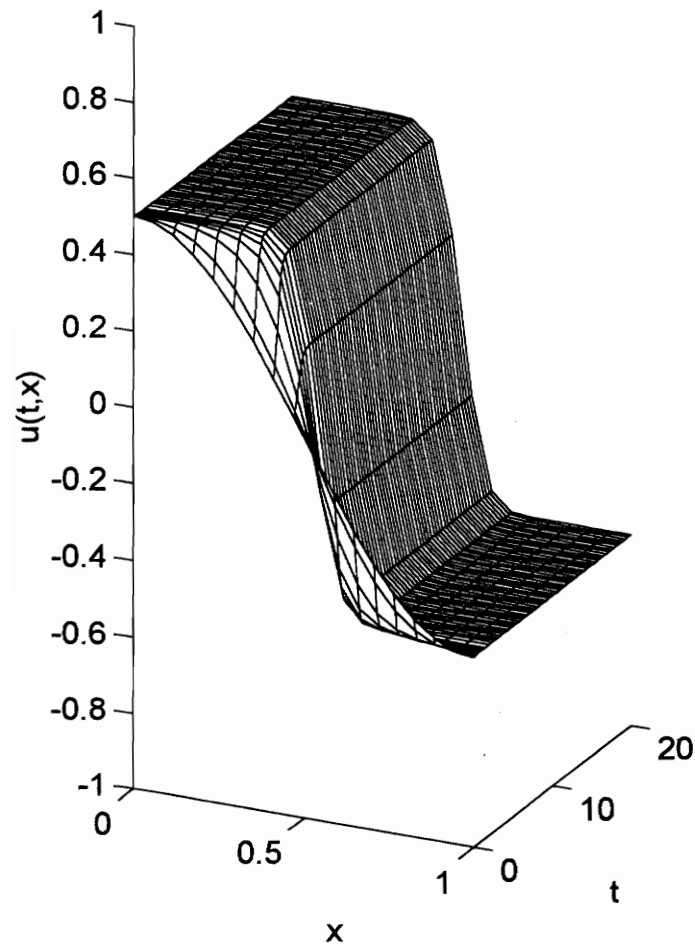


Figure 3.16: $\phi(x) = \frac{1}{2}\cos(\pi x)$ ($\text{Re}=60$): a) Galerkin method, b) G/C method

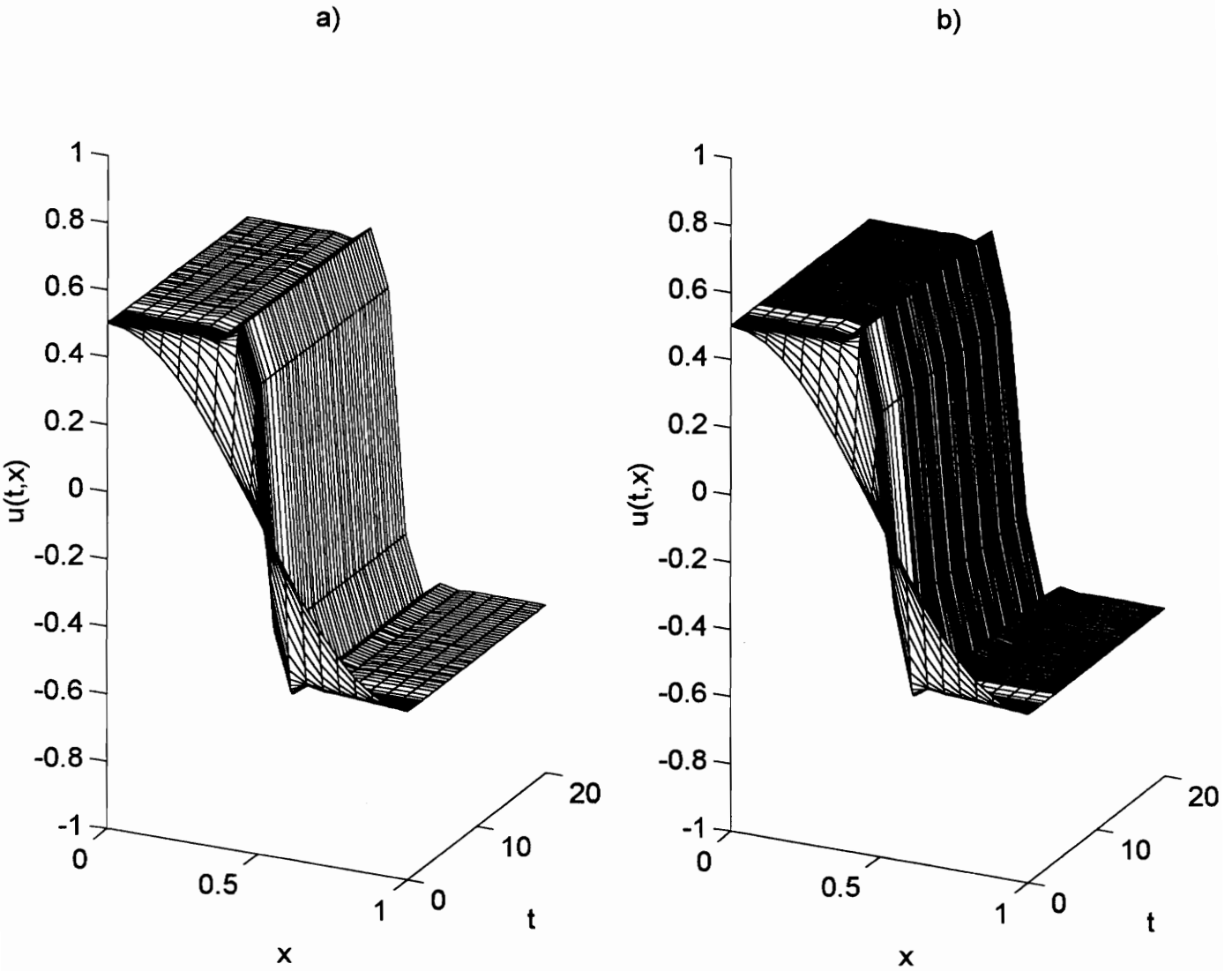


Figure 3.17: $\phi(x) = \frac{1}{2}\cos(\pi x)$ ($\text{Re}=120$): a) Galerkin method, b) G/C method

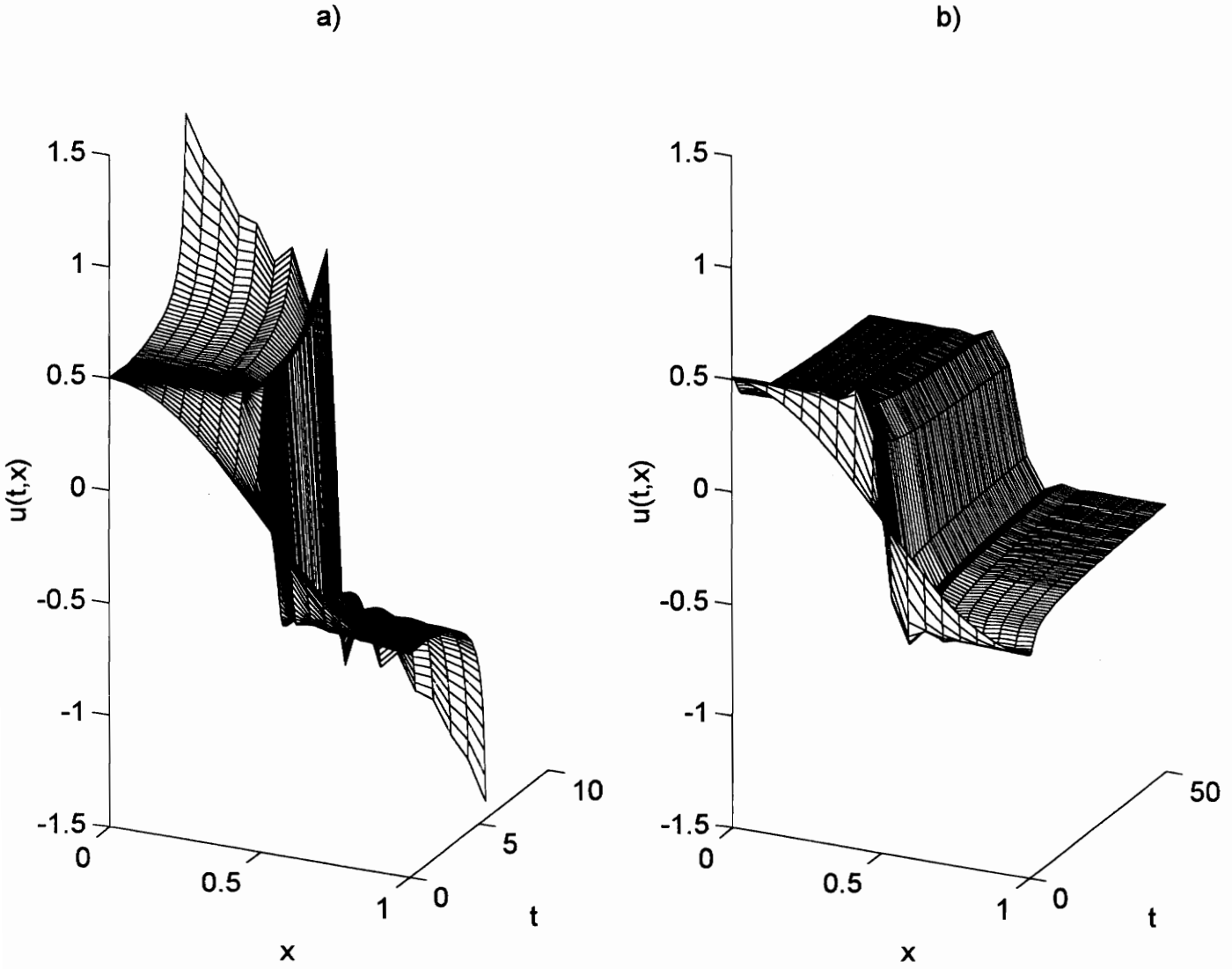


Figure 3.18: $\phi(x) = \frac{1}{2}\cos(\pi x)$ ($\text{Re}=240$): a) Galerkin method, b) G/C method

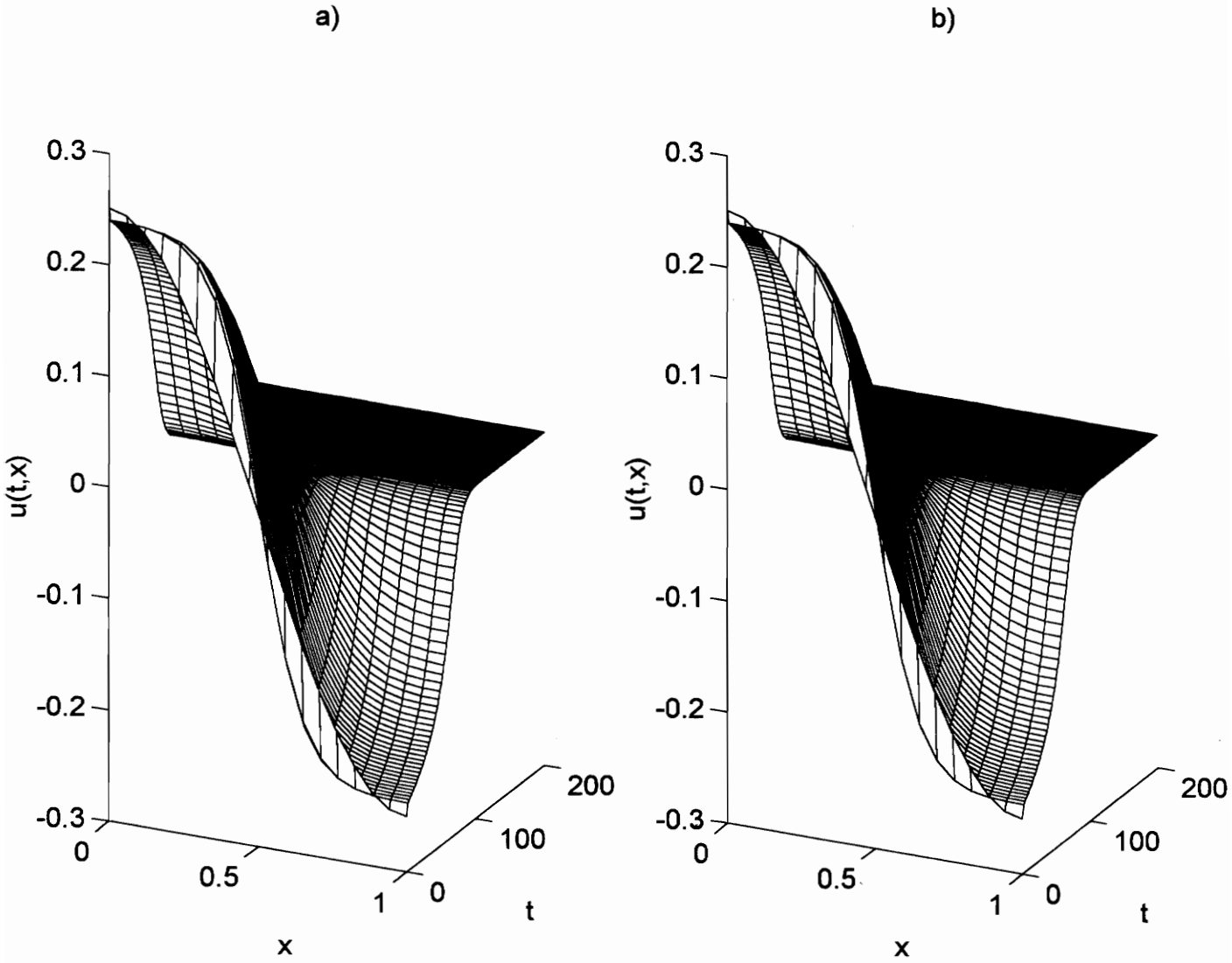


Figure 3.19: $\phi(x) = \frac{1}{4}\cos(\pi x)$ ($Re=60$): a) Galerkin method, b) G/C method

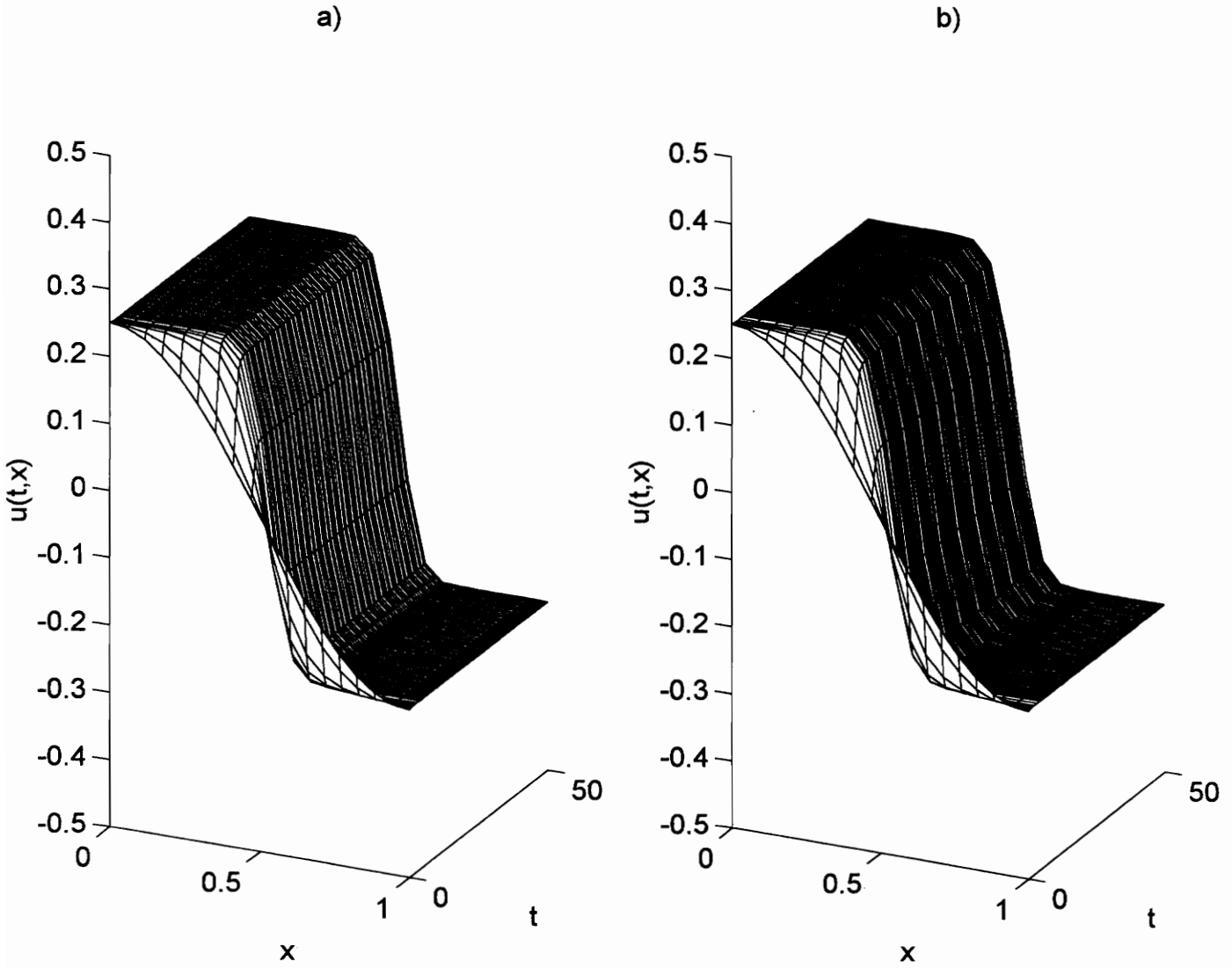


Figure 3.20: $\phi(x) = \frac{1}{4}\cos(\pi x)$ ($\text{Re}=120$): a) Galerkin method, b) G/C method

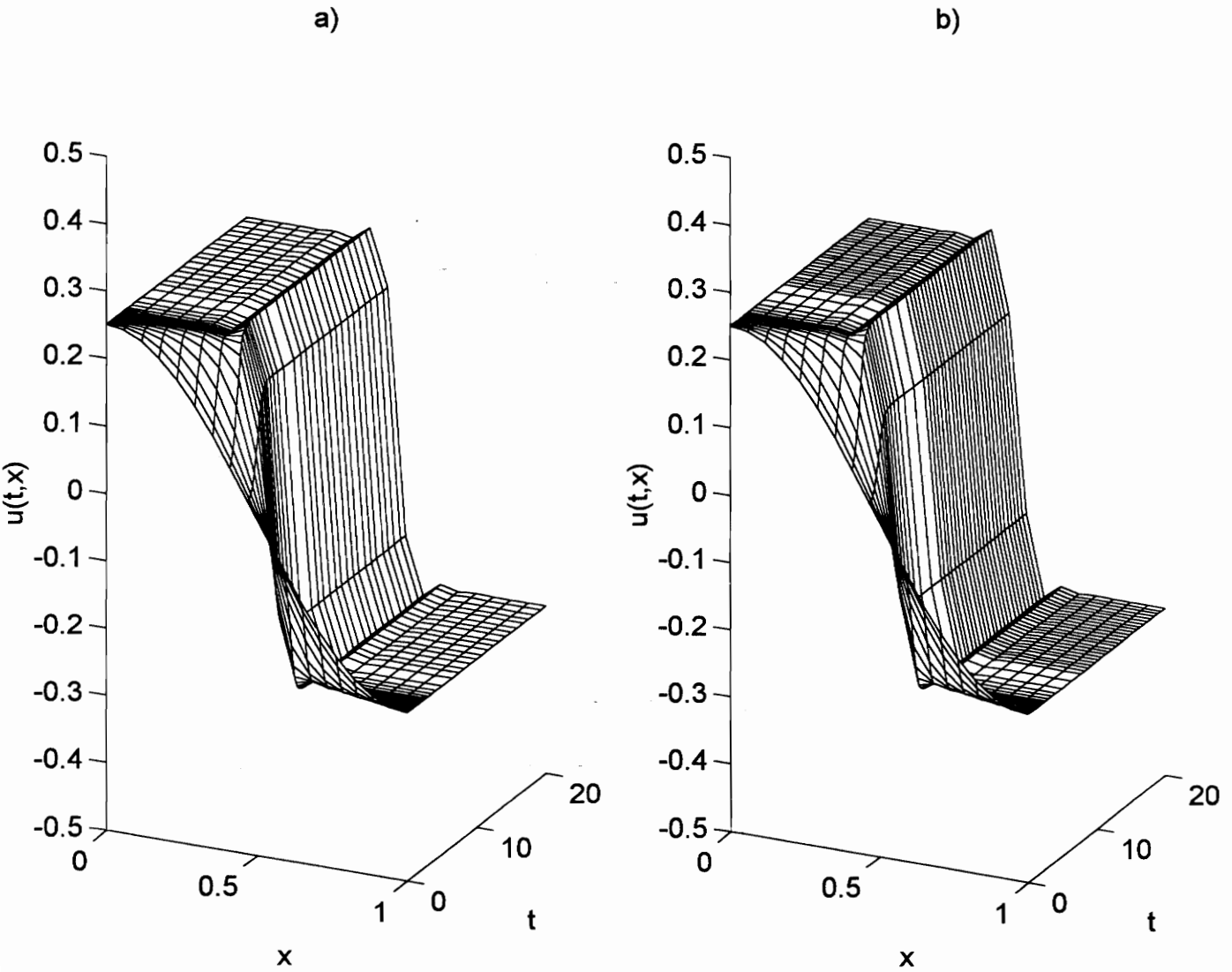


Figure 3.21: $\phi(x) = \frac{1}{4}\cos(\pi x)$ ($\text{Re}=240$): a) Galerkin method, b) G/C method

CHAPTER 3. NUMERICAL EXPERIMENTS

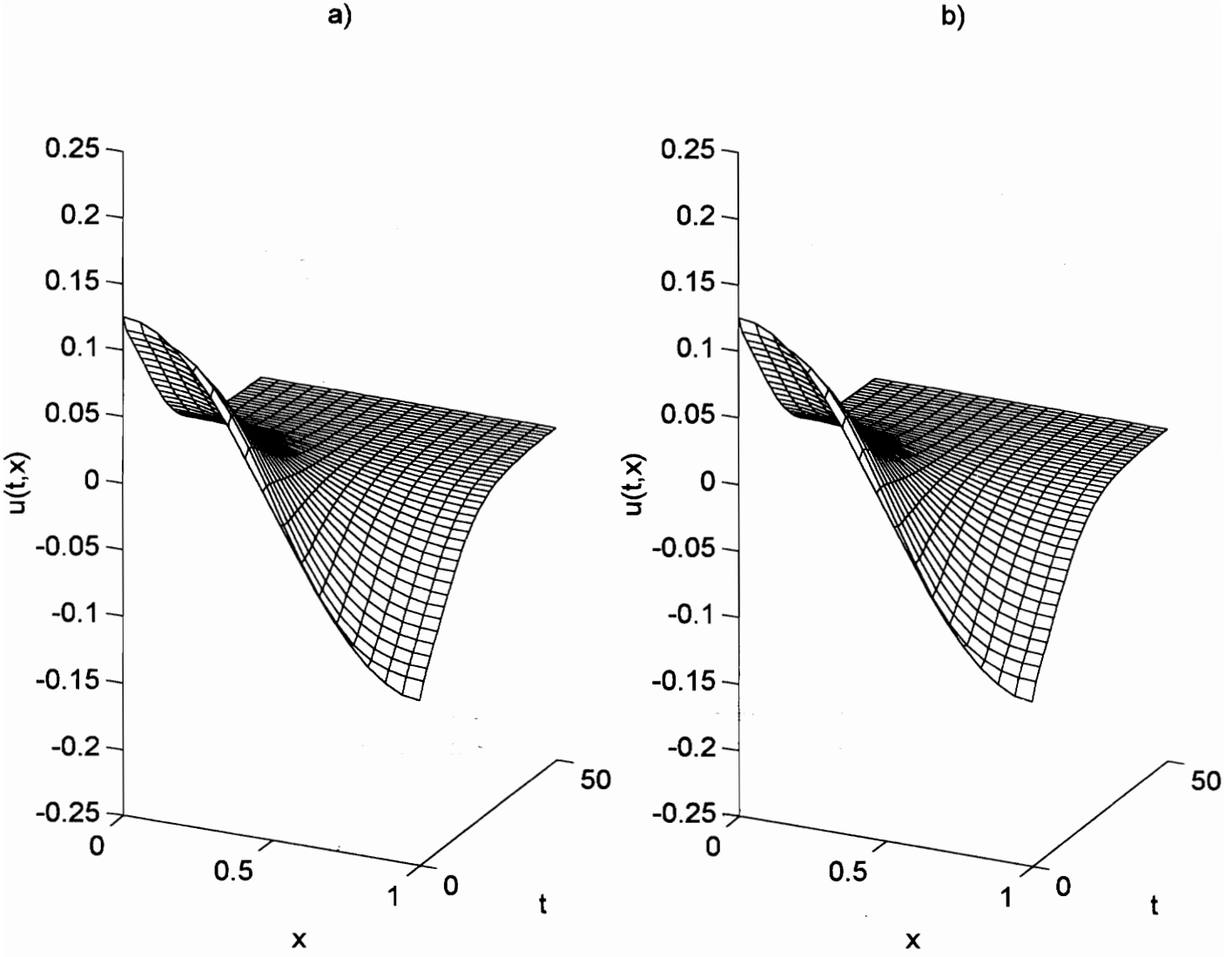


Figure 3.22: $\phi(x) = \frac{1}{8}\cos(\pi x)$ ($\text{Re}=60$): a) Galerkin method, b) G/C method

a)

b)

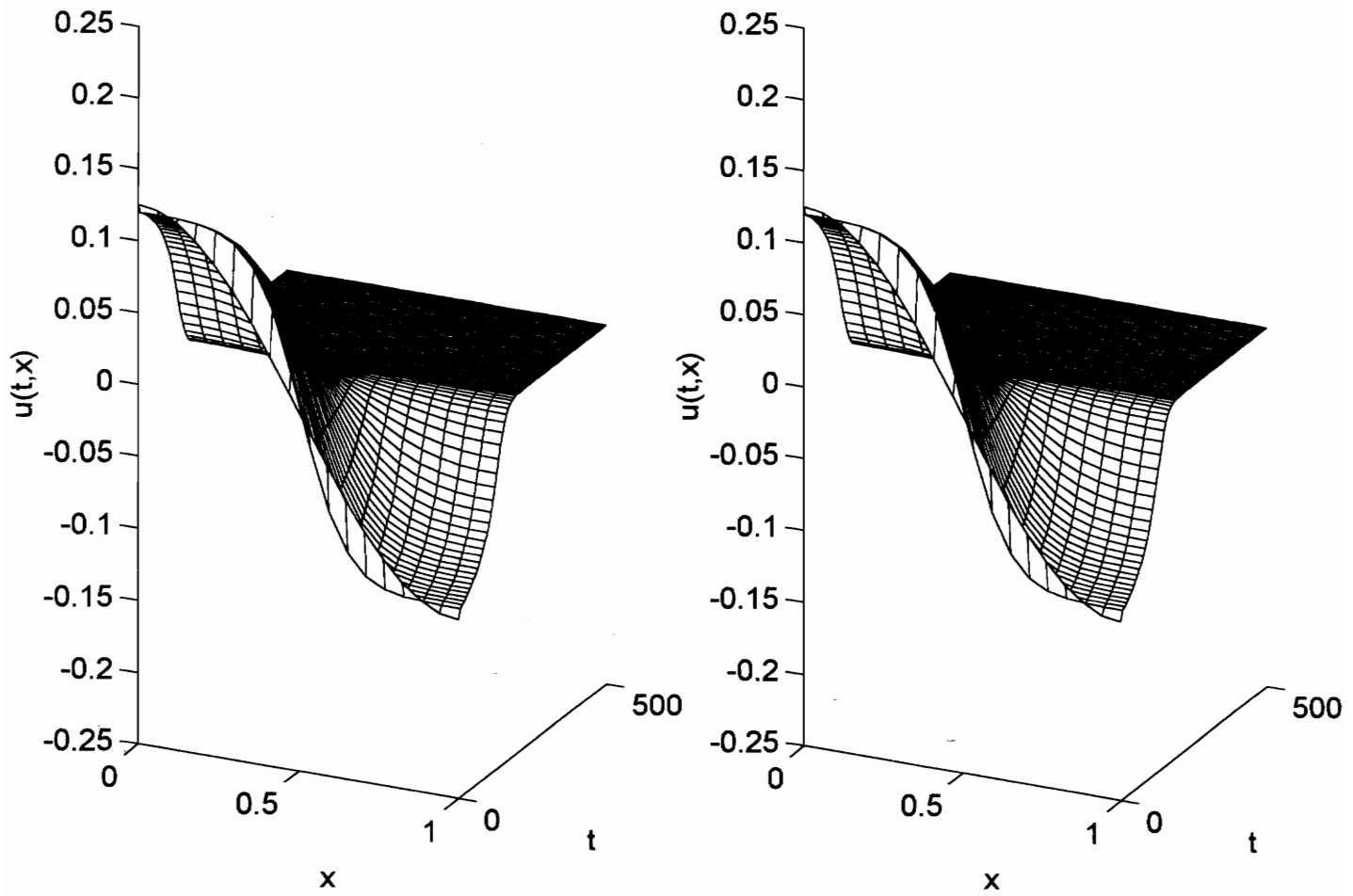


Figure 3.23: $\phi(x) = \frac{1}{8} \cos(\pi x)$ ($\text{Re}=120$): a) Galerkin method, b) G/C method

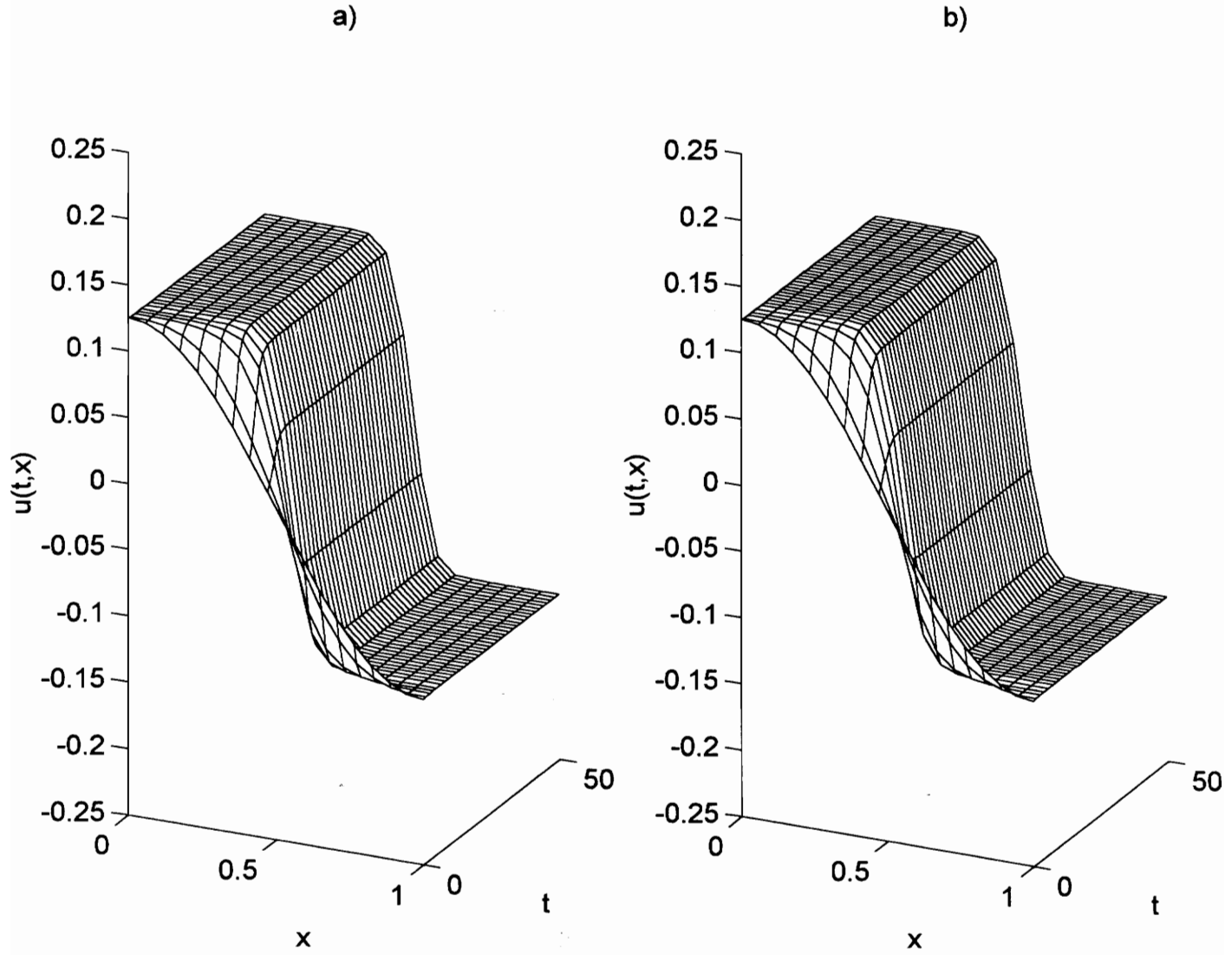


Figure 3.24: $\phi(x) = \frac{1}{8} \cos(\pi x)$ ($Re=240$): a) Galerkin method, b) G/C method

CHAPTER 3. NUMERICAL EXPERIMENTS

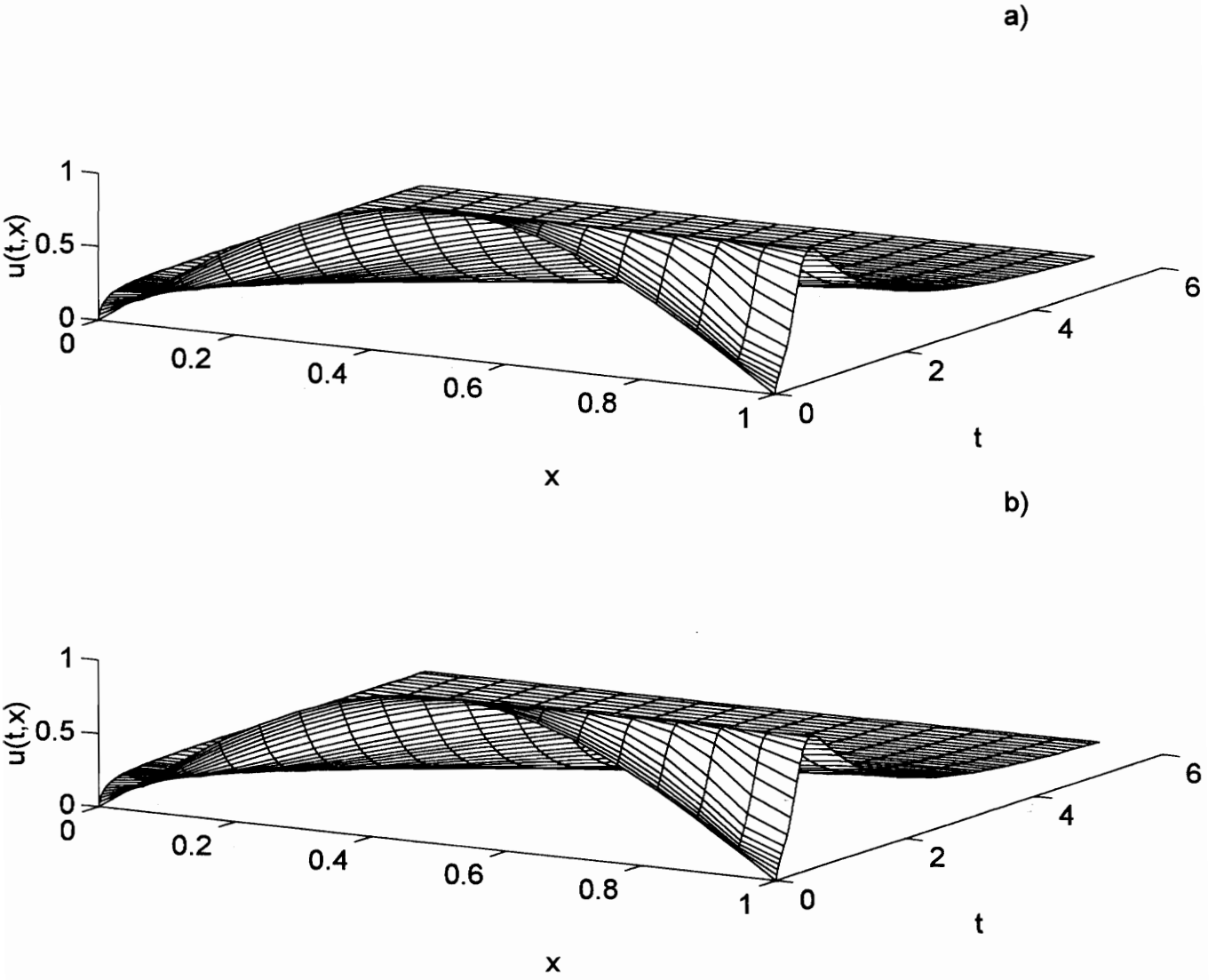


Figure 3.25: $\phi(x) = \sin(\pi x)$ ($Re=60$): a) Galerkin method, b) G/C method

CHAPTER 3. NUMERICAL EXPERIMENTS

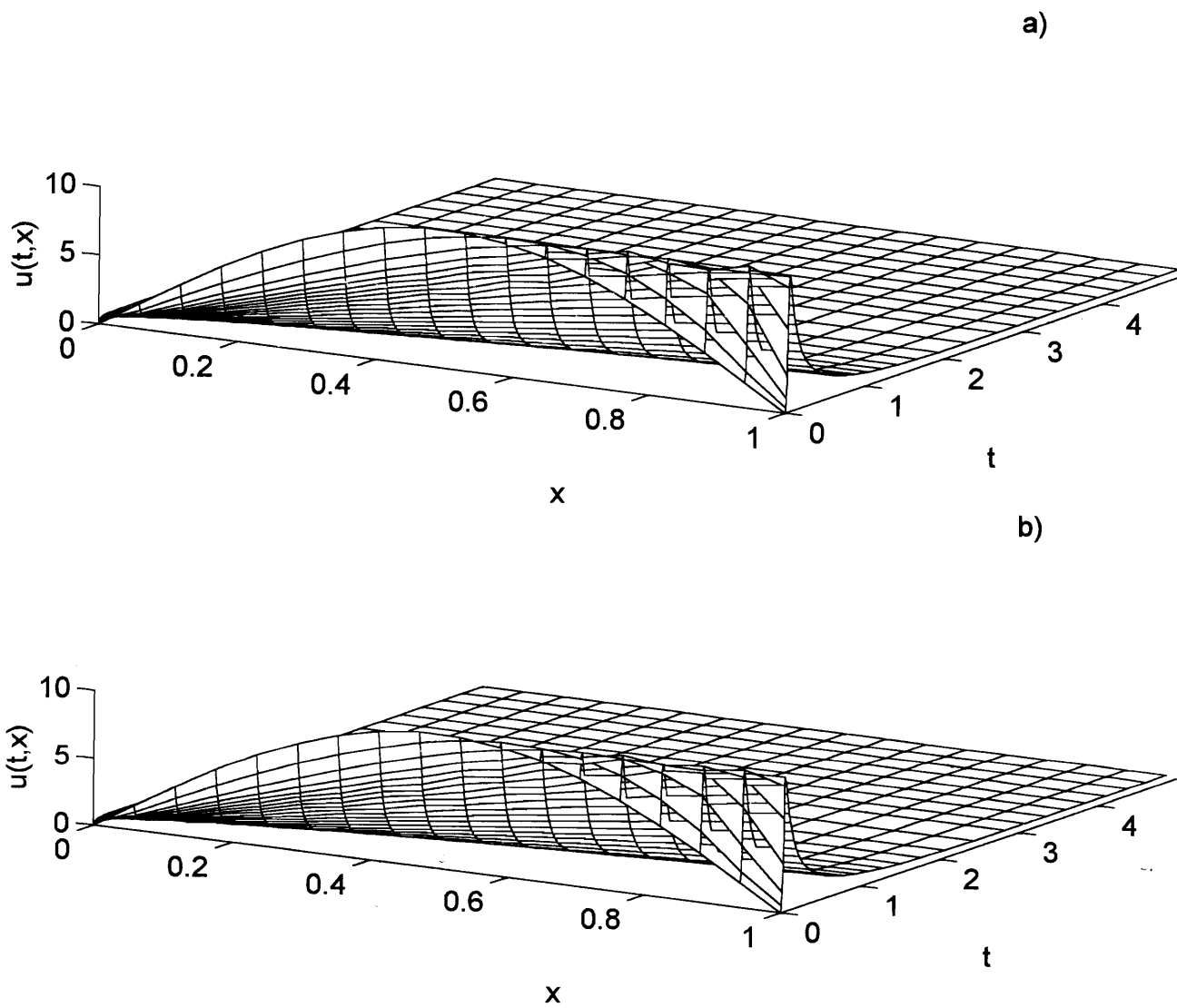


Figure 3.26: $\phi(x) = 10\sin(\pi x)$ ($Re=60$): a) Galerkin method, b) G/C method

CHAPTER 3. NUMERICAL EXPERIMENTS

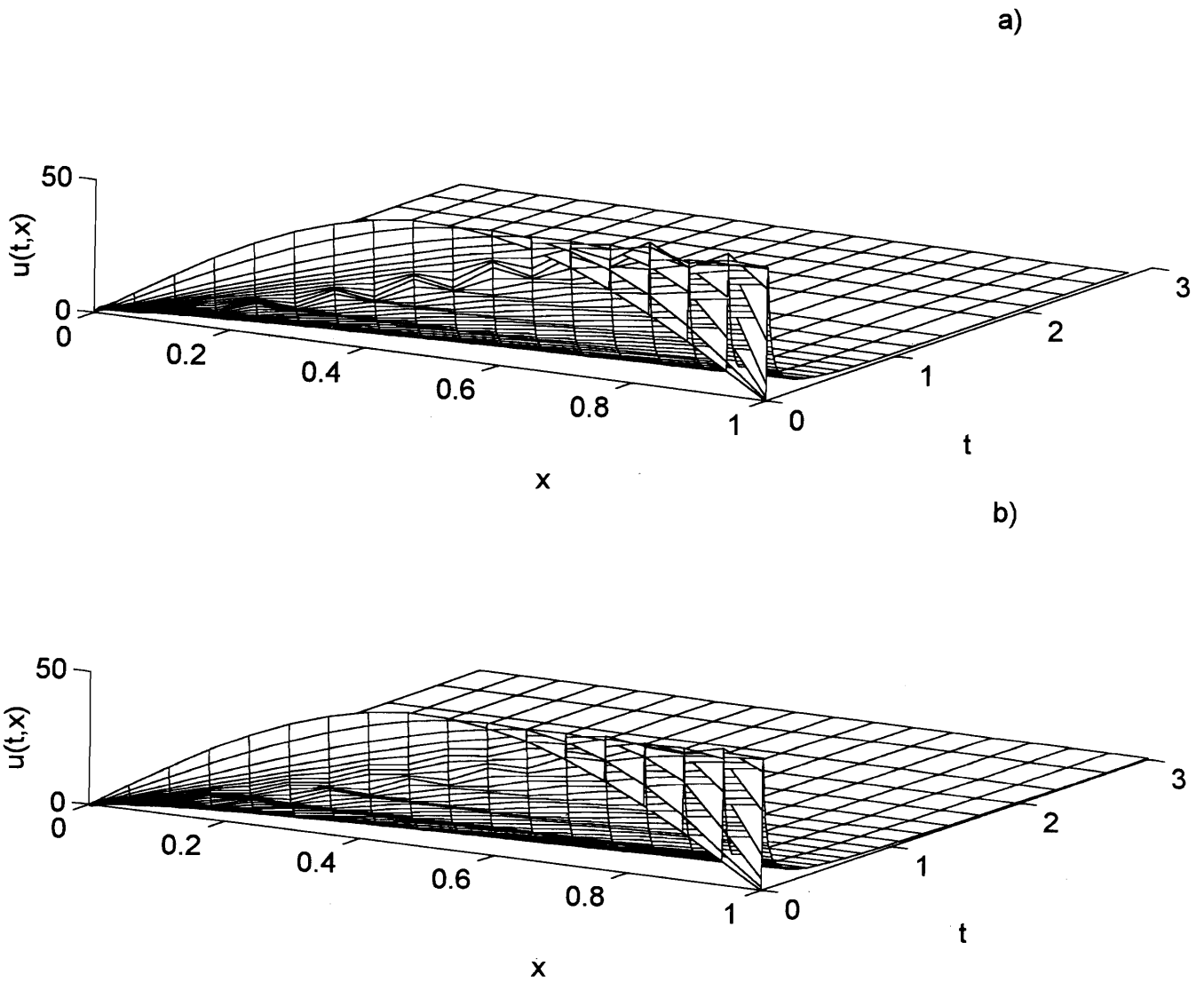
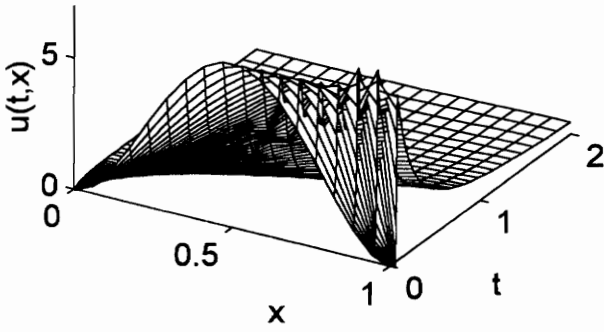


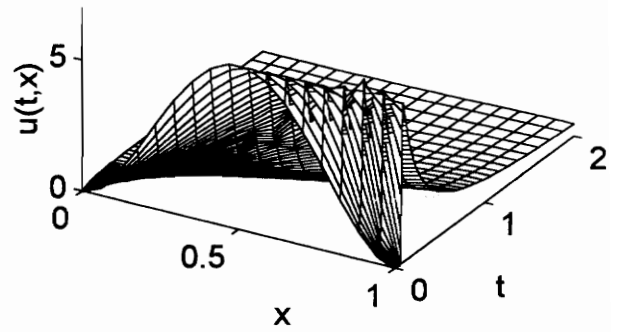
Figure 3.27: $\phi(x) = 50\sin(\pi x)$ ($\text{Re}=60$): a) Galerkin method, b) G/C method

CHAPTER 3. NUMERICAL EXPERIMENTS

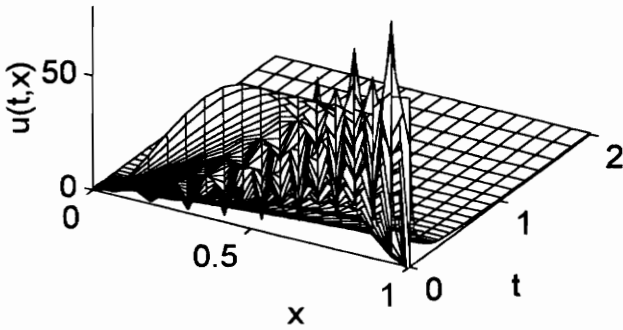
a)



b)



c)



d)

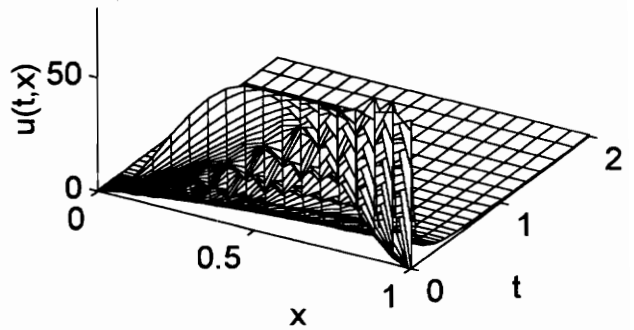


Figure 3.28: a) $\phi(x) = 100x^2(1-x)^2$, Galerkin method, b) $\phi(x) = 100x^2(1-x)^2$, G/C method, c) $\phi(x) = 1000x^2(1-x)^2$, Galerkin method, d) $\phi(x) = 1000x^2(1-x)^2$, G/C method

CHAPTER 3. NUMERICAL EXPERIMENTS

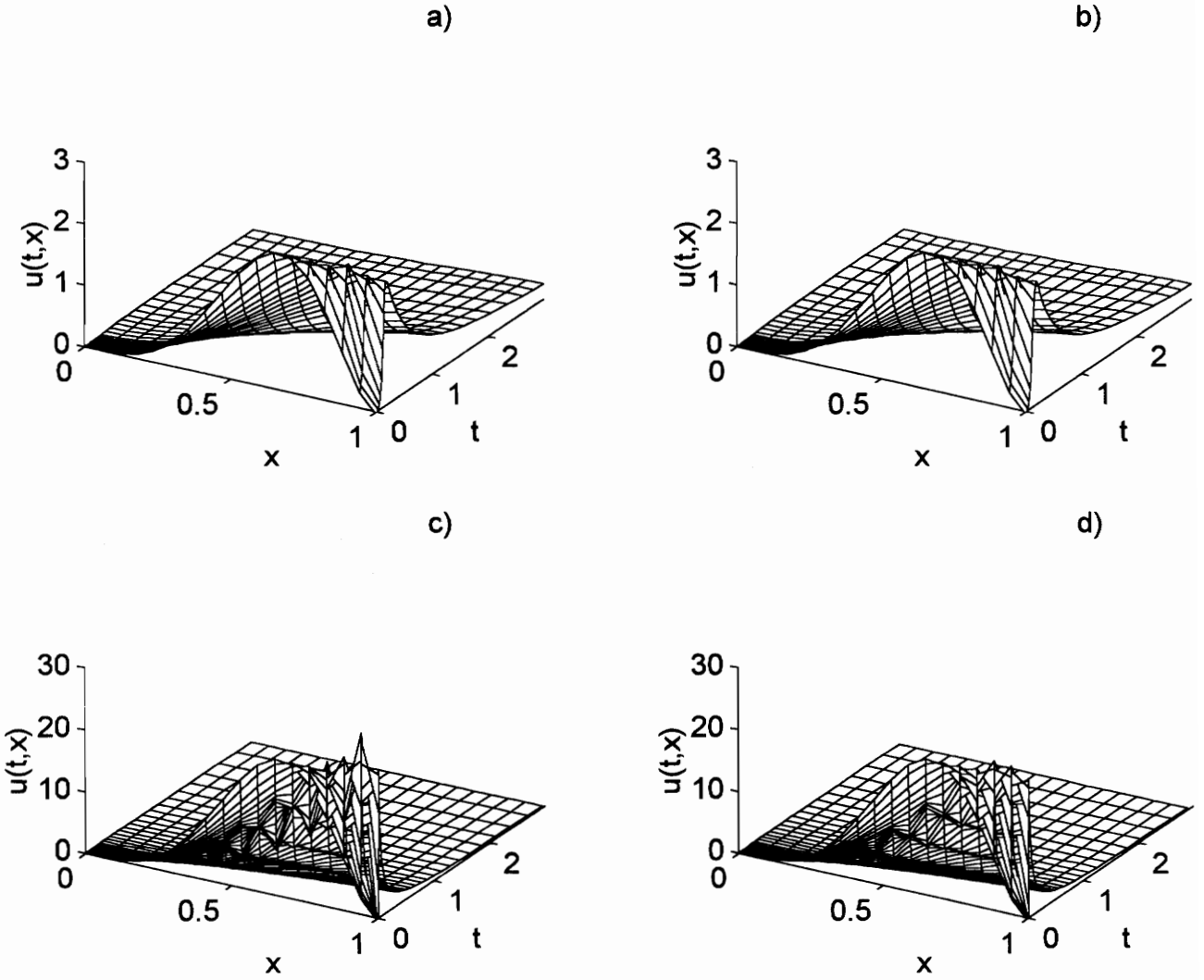


Figure 3.29: a) $\phi(x) = 100x^4(1-x)^2$, Galerkin method, b) $\phi(x) = 100x^4(1-x)^2$, G/C method, c) $\phi(x) = 1000x^4(1-x)^2$, Galerkin method, d) $\phi(x) = 1000x^4(1-x)^2$, G/C method

CHAPTER 3. NUMERICAL EXPERIMENTS

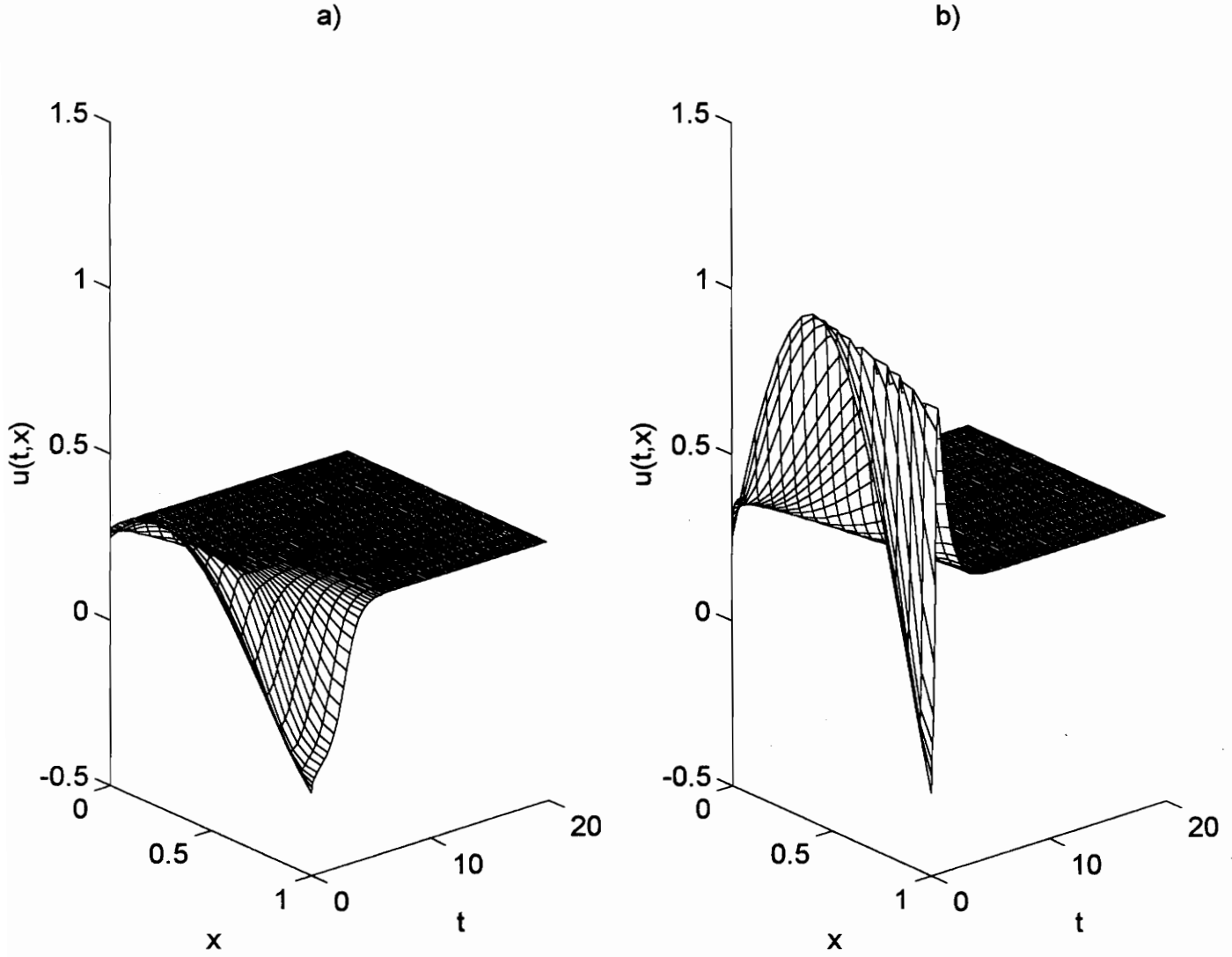


Figure 3.30: G/C approximations ($\text{Re}=60$): a) $\phi(x) = \frac{1}{4}\cos(\pi x) + \frac{1}{4}\sin(\pi x)$, b) $\phi(x) = \frac{1}{4}\cos(\pi x) + \sin(\pi x)$

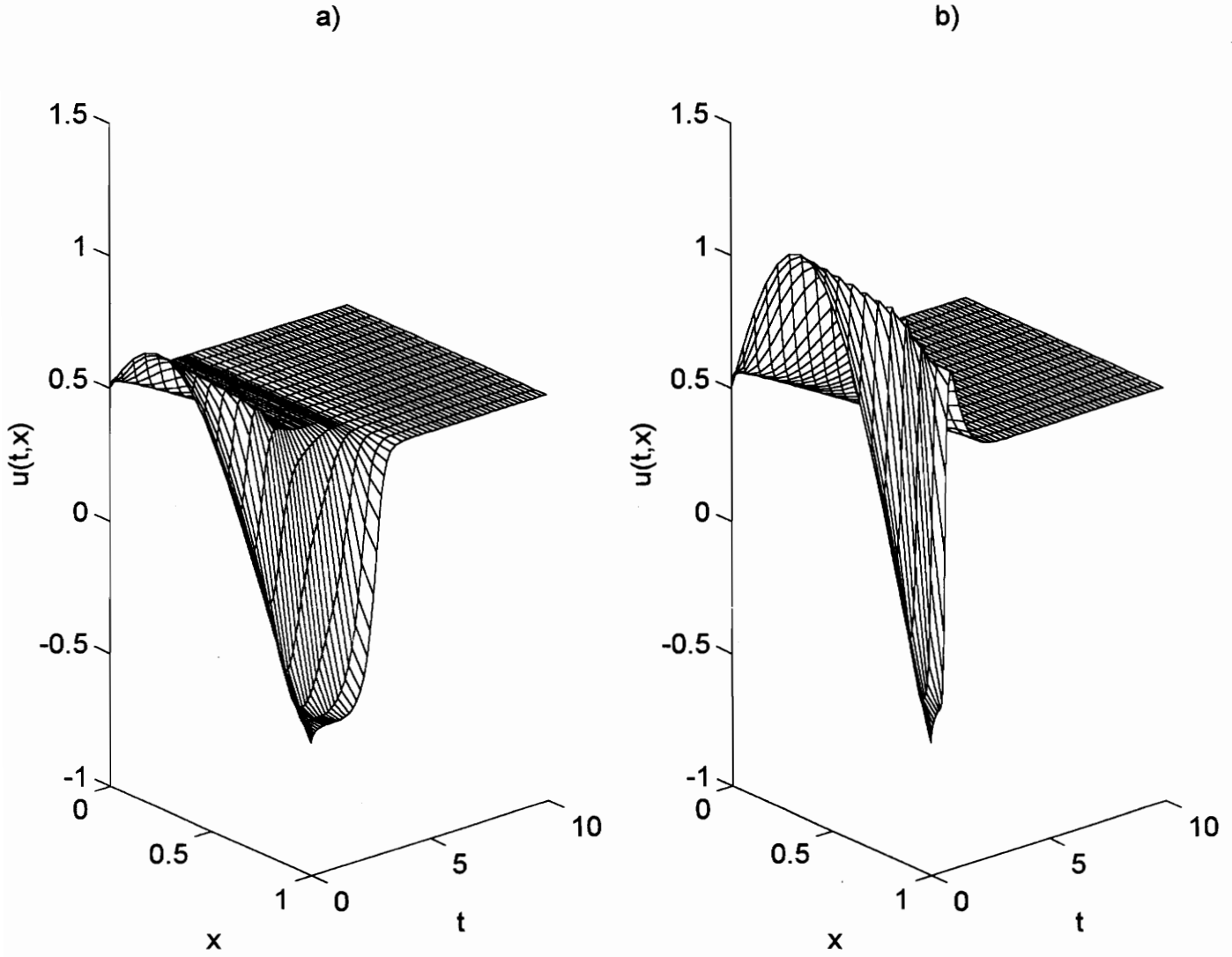


Figure 3.31: G/C approximations ($Re=60$): a) $\phi(x) = \frac{1}{2}\cos(\pi x) + \frac{1}{2}\sin(\pi x)$, b) $\phi(x) = \frac{1}{2}\cos(\pi x) + \sin(\pi x)$

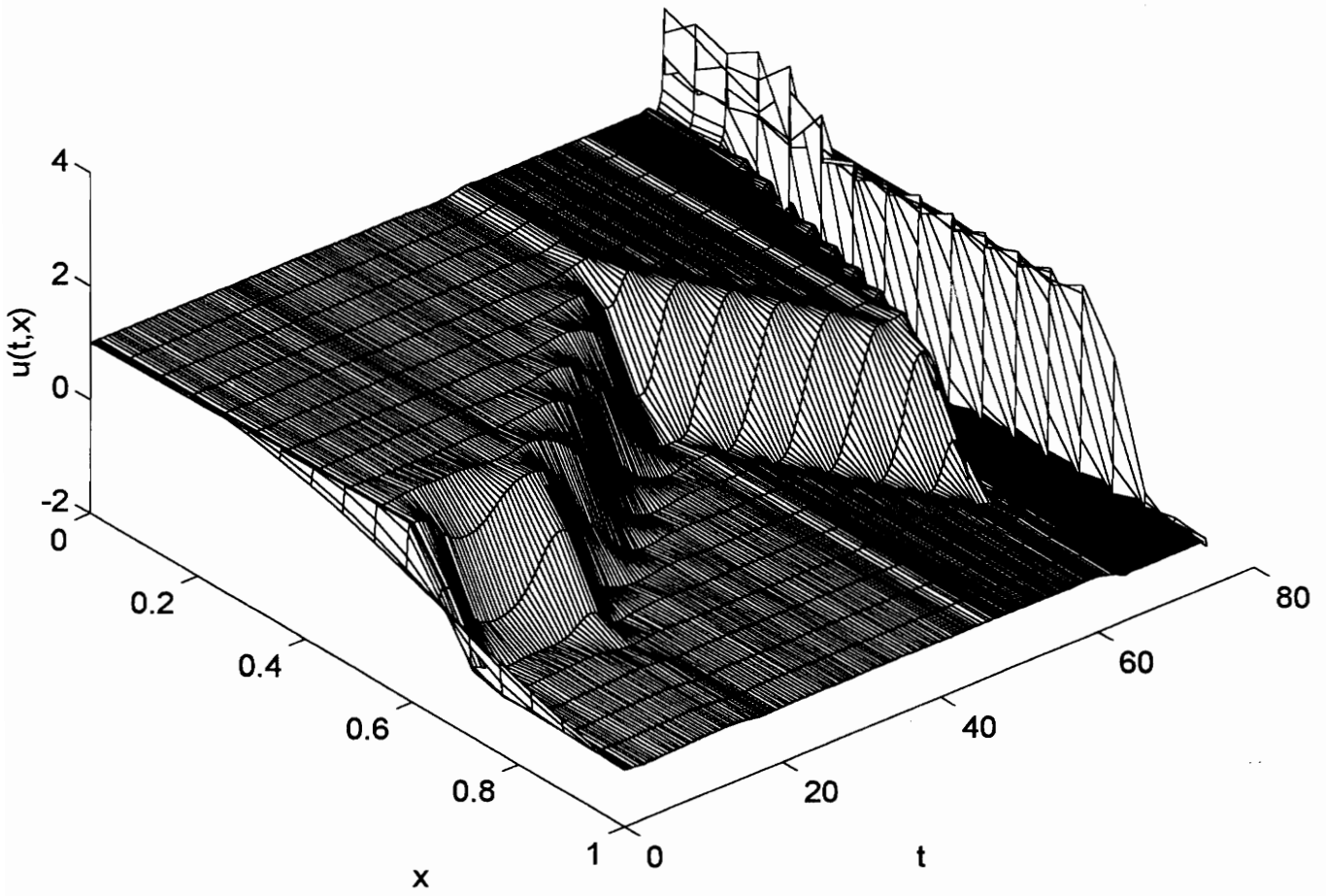


Figure 3.32: G/C approximations ($Re=60$): $\phi(x) = \cos(\pi x) + \frac{1}{2}\sin(\pi x)$

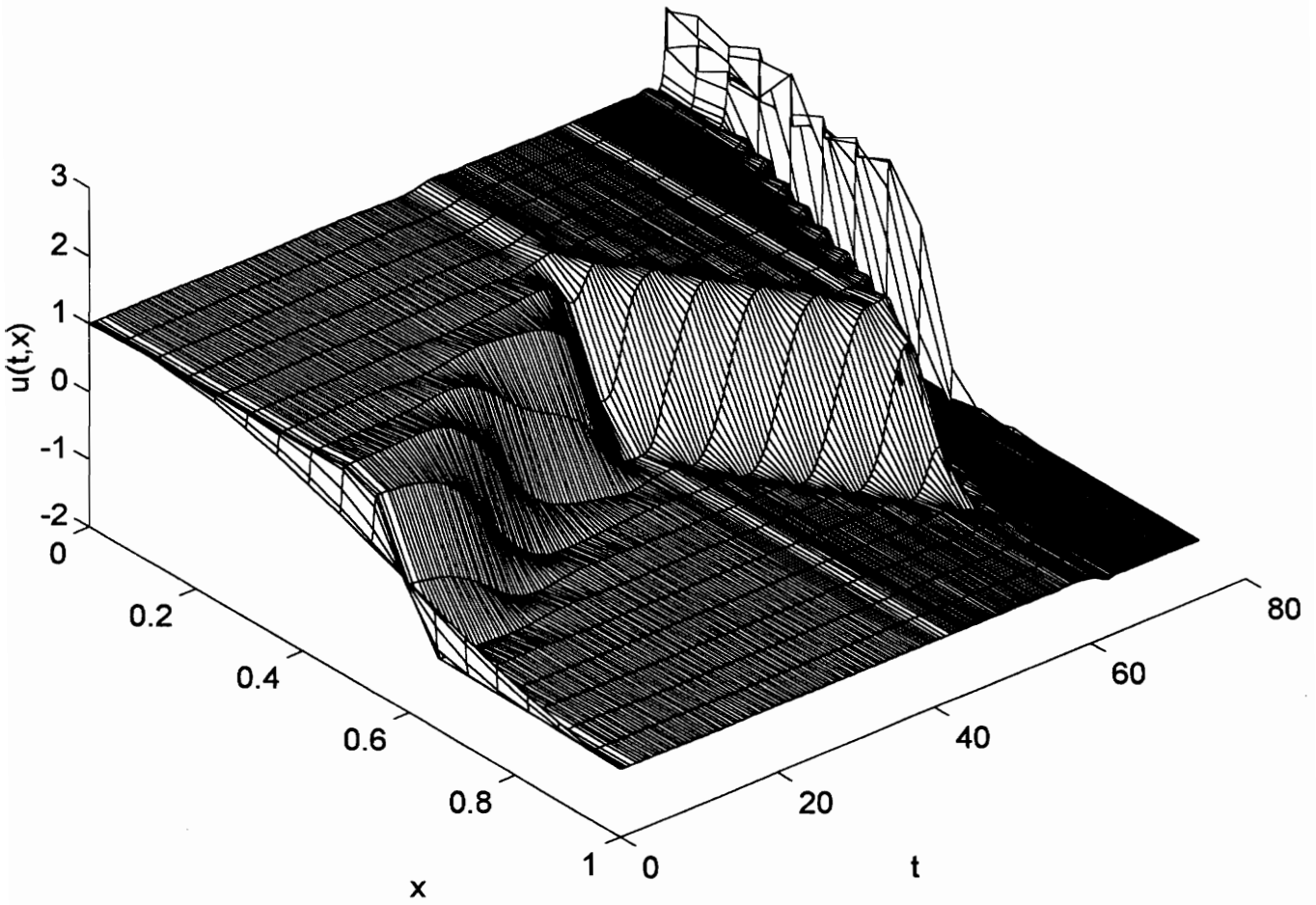


Figure 3.33: G/C approximations ($\text{Re}=60$): $\phi(x) = \cos(\pi x) + \frac{1}{4}\sin(\pi x)$

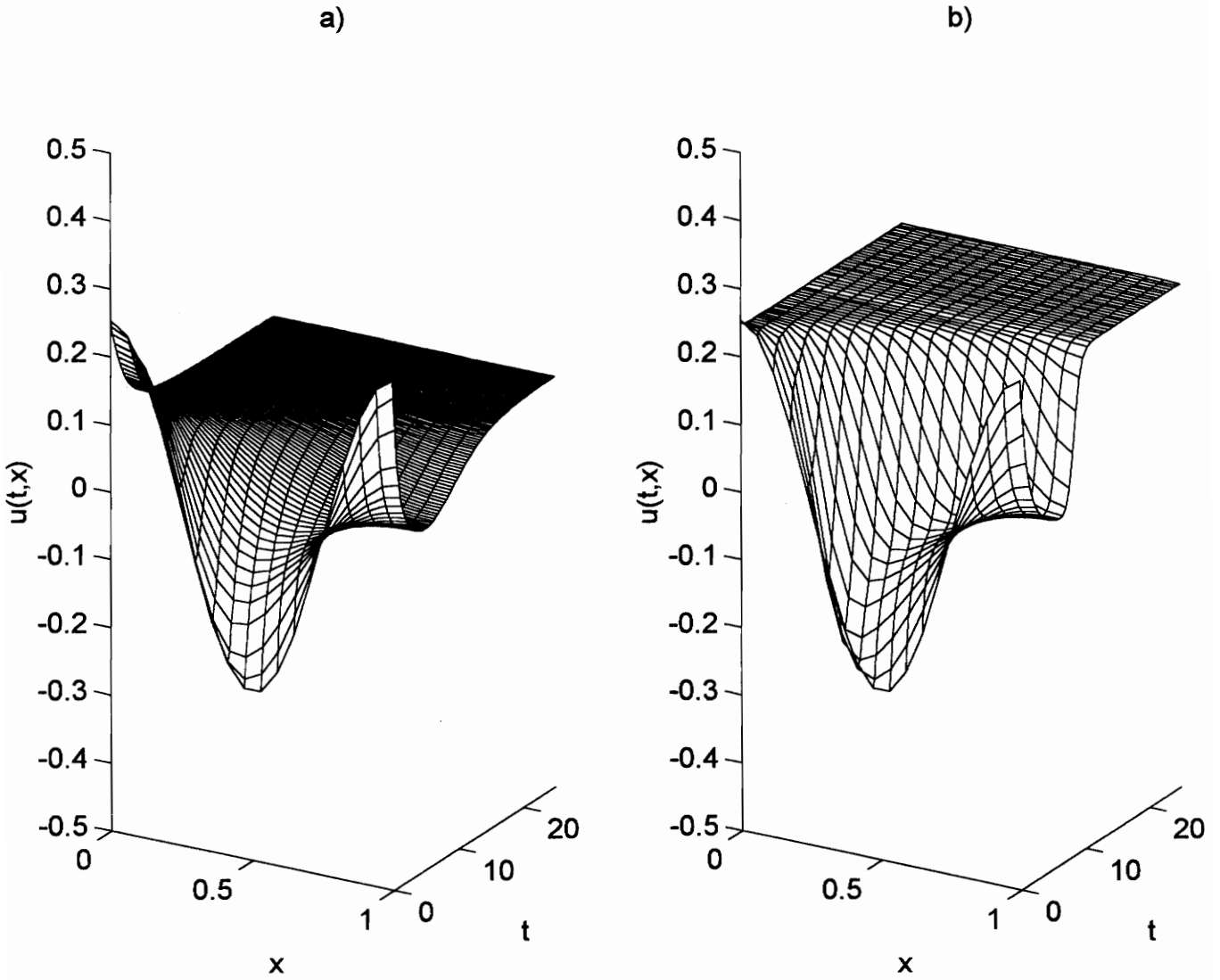


Figure 3.34: G/C approximations, $\phi(x) = \frac{1}{4}\cos(2\pi x)$: a) $\text{Re}=60$, b) $\text{Re}=120$

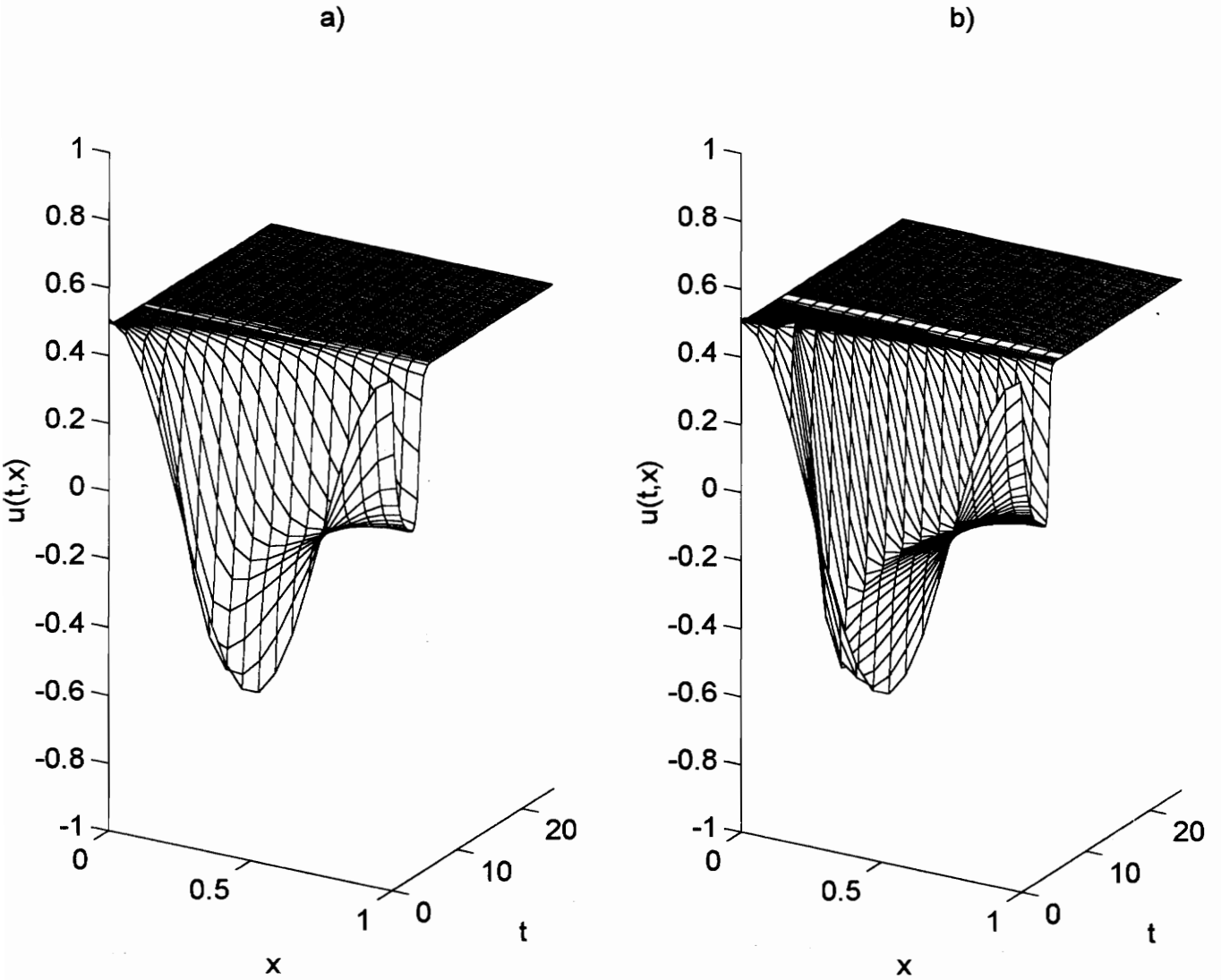


Figure 3.35: G/C approximations, $\phi(x) = \frac{1}{2}\cos(2\pi x)$: a) $\text{Re}=60$, b) $\text{Re}=120$

CHAPTER 3. NUMERICAL EXPERIMENTS

a)

b)

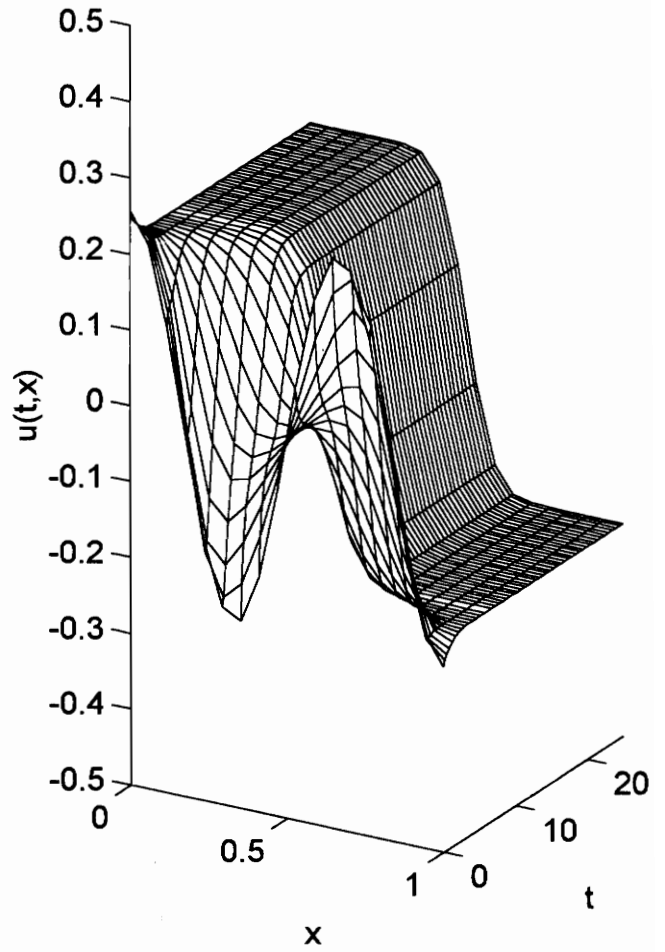
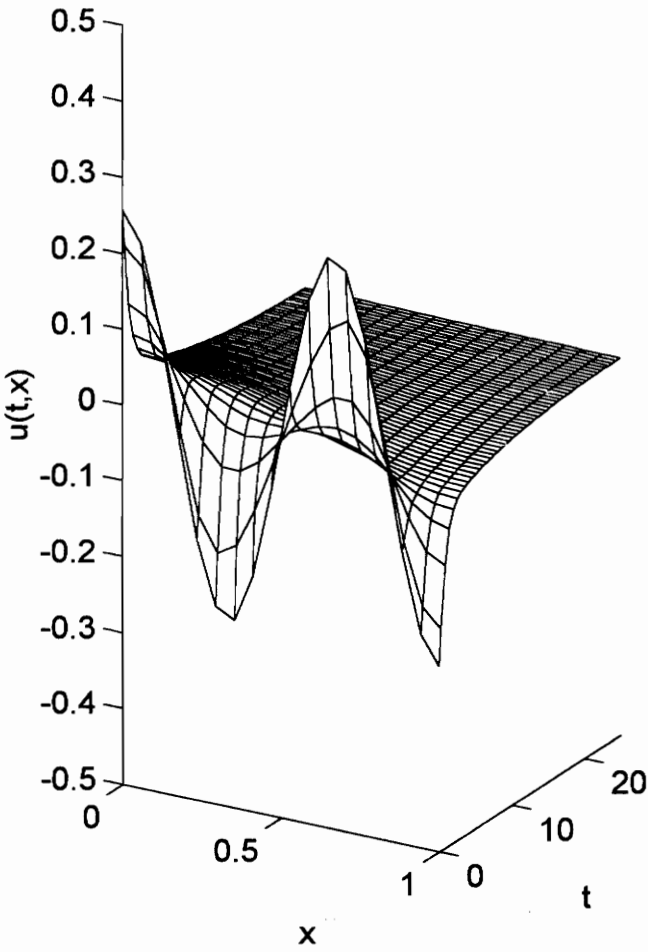


Figure 3.36: G/C approximations, $\phi(x) = \frac{1}{4}\cos(3\pi x)$: a) $Re=60$, b) $Re=120$

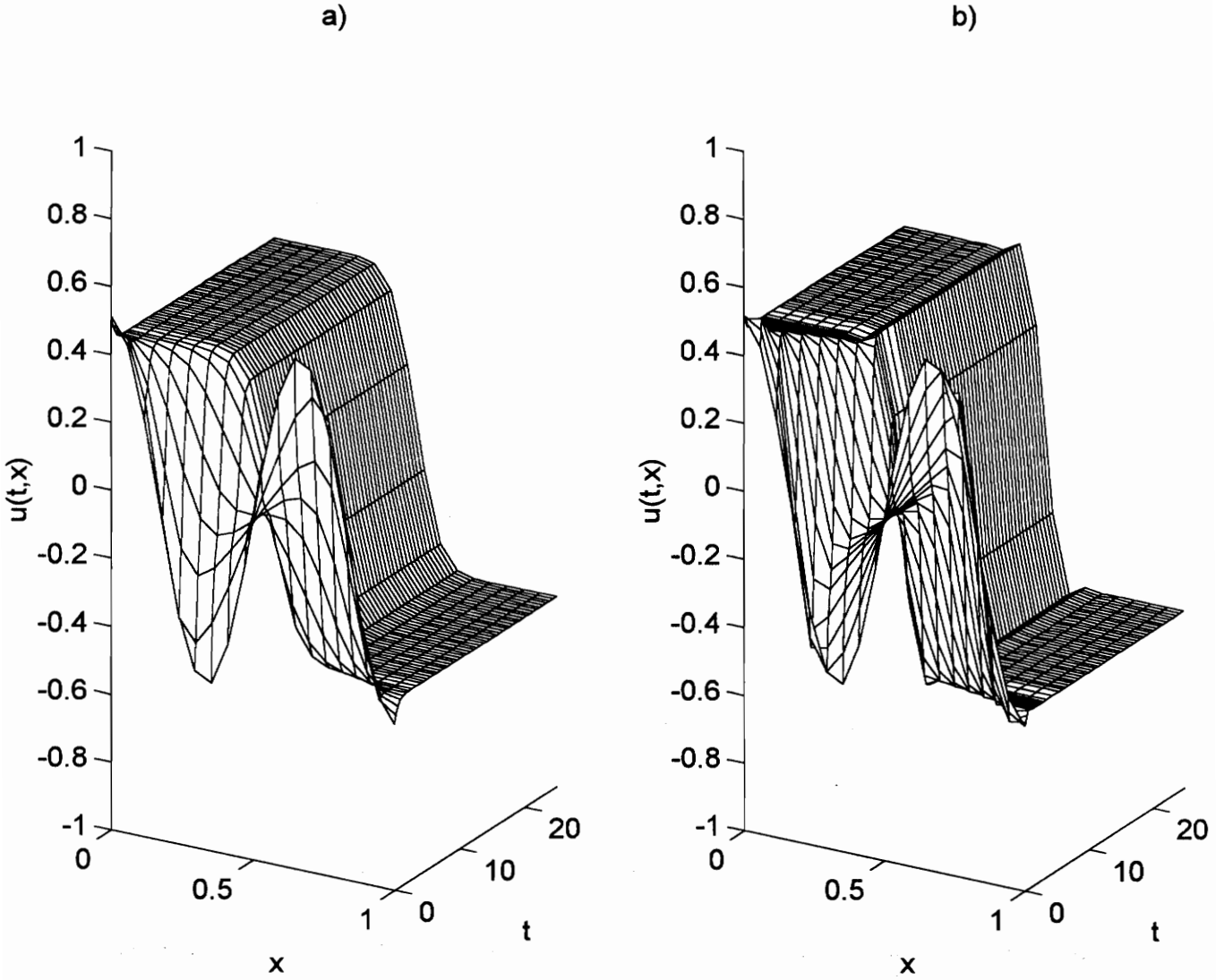


Figure 3.37: G/C approximations, $\phi(x) = \frac{1}{2}\cos(3\pi x)$: a) $\text{Re}=60$, b) $\text{Re}=120$

Chapter 4

CONCLUSIONS

4.1 Overview of Results

In this paper we used the Galerkin method and the Galerkin/Conservation method (outlined in Chapter 2) to approximate solutions to the homogeneous initial/boundary value problem

$$u_t(t, x) + u(t, x)u_x(t, x) = \epsilon u_{xx}(t, x) \quad (4.1.1)$$

$$u(0, x) = \phi(x) \quad (4.1.2)$$

$$u_x(t, 0) = u_x(t, 1) = 0 \quad (4.1.3)$$

where $\epsilon = \frac{1}{Re}$ is the reciprocal of the Reynolds number. A solution to this IVBP, starting initially at $u(0, x) = \phi(x)$, describes the evolution of the function $u(t, x)$ over time subject to Neumann boundary conditions (4.1.3) on the closed x -interval $[0, 1]$. Using for the most part $N=16$ elements, we compared our results obtained from the two approximation methods on the bases of accuracy, speed (computational execution time), and behavior. We found that for many combinations of initial conditions and Reynolds numbers the two methods produced virtually identical results. In particular, the results were identical in every case where both approximate solutions converged. In a few examples however, the Galerkin solution grew exponentially in time despite the convergent behavior of the Galerkin/Conservation solution. Together with our findings that the Galerkin/Conservation method is generally 'faster' (with respect to execution time) and more accurate, these observations indicate that the Galerkin/Conservation method is 'better' in the sense that it is computationally more efficient and intuitively more believable.

CHAPTER 4. CONCLUSIONS

The key problem is determining the form of the steady state solutions, i.e. the solutions $u_{ss}(x)$ such that $u(t, x) \rightarrow u_{ss}(x)$ as $t \rightarrow \infty$ where $u(t, x)$ solves the initial/boundary value problem. Byrnes, Gilliam, and Shubov [6] proved that for $\epsilon > 0$ there is a constant k such that $u_{ss}(x) \equiv C$ for some constant C as long as at $\|\phi(x)\| \leq k\epsilon$. That is, for initial data 'small' enough, the steady state solution will be a constant. We have offered data in this paper which certainly supports this theory. Moreover, some of our numerical results (see Examples 3.2.4, 3.3.1, 3.4.2, 3.4.7) illustrate a phenomena where initial conditions which are not 'small' evolve to steady state solutions which are not everywhere constant. We have even seen cases (see Examples 3.4.2, 3.4.7) where numerical solutions (corresponding to a given initial condition) converge to a constant for a certain value of ϵ but converge to a nonconstant steady state for some $\hat{\epsilon} < \epsilon$.

4.2 Conclusions

The numerical results contained in this paper seem to imply the existence of nonconstant steady state solutions $u_{ss}(x)$ for certain initial conditions and Reynolds numbers. The convergence of numerical solutions to states which are not everywhere constant can be the result of several things; unfortunately, it is not easy to determine the exact cause(s). First, the condition that a solution $u(t, x)$ converges to $u_{ss}(x)$ in time may not be sufficient for $u_{ss}(x)$ to satisfy the steady state equation:

$$u_x(x)u(x) - \epsilon u_{xx}(x) = 0 \tag{4.2.1}$$

$$u_x(0) = u_x(1) = 0. \tag{4.2.2}$$

That is, the steady state that the solutions reach may not be an equilibrium. Second, the numerical solution $u^N(t, x)$ may not accurately reflect the steady state solution $u_{ss}(x)$ as $t \rightarrow \infty$. We have seen inaccuracies in some of our examples which were substantial enough to cause our Galerkin and Galerkin/Conservation solutions to differ drastically. Clearly, based on uniqueness theorems of partial differential equation theory, at least one of the

CHAPTER 4. CONCLUSIONS

numerical solutions has to be false since there exists one and only one exact solution for a given initial condition. At the same time, one might argue that nearly perfect similarities in the approximate solutions obtained by the two different numerical methods (which we have seen in many examples) is evidence to support that both approximations are representative of the exact solution. A final possible reason for the nonconstant steady state convergence could be that there really exist L_2 functions $u_{ss}(x)$ that satisfy a weak form of the steady state equation. The latter possibility is perhaps interesting enough to trigger more advanced research on this topic.

4.3 Open Problems

Although we have established no proven results from our observations, our numerical results give rise to several interesting questions. Hopefully, the data we have collected in this paper will serve as a motivator for research into the following unknowns:

- Can one prove that, for 'large' initial data, there exist non-constant weak solutions to the steady state equation?
- If all steady state solutions are constant despite the size of the initial data, is it possible for $u(t, x)$ to converge (in time) to $\hat{u}(x)$ where $\hat{u}(x)$ is not a steady state solution?
- Can one prove (for either of our approximate solutions) that $u^N(t, x)$ converges to $u^N(x)$ in some topology?

REFERENCES

- [1] Burgers, J.M., "Mathematical examples illustrating relations occurring in the theory of turbulent fluid motion", *Trans. Roy. Neth. Acad. Sci.* **17**, Amsterdam, (1939), 1-53.
- [2] Burgers, J.M., "A mathematical model illustrating the theory of turbulence", *Adv. in Appl. Mech.* **1**, (1948), 171-199.
- [3] Burgers, J.M., "Statistical problems connected with asymptotic solution of one-dimensional nonlinear diffusion equation", in M. Rosenblatt and C. van Atta (eds.), *Statistical Models and Turbulence*, Springer, Berlin, (1972), 41-60.
- [4] Burns, J.A. and Kang, S., *A control problem for Burgers' equation with bounded input/output*, Institute for Computer Applications for Science and Engineering, ICASE Report 90-45, (1990), NASA Langley Research Center, Hampton, VA.
- [5] Burns, J.A. and Kang, S., *A stabilization problem for Burgers' equation with unbounded control and observation*, International Series of Numerical Mathematics, Vol. 100, Birkhauser Verlag Basel, (1991).
- [6] Byrnes, C.I., Gilliam, D.S., and Shubov, V.I., "Boundary Control, Feedback Stabilization, and the Existence of Attractors for a Viscous Burgers' Equation", (1990).
- [7] Cole, J.D., "On a quasi-linear parabolic equation occurring in aerodynamics", *Quart. Appl. Math.*, **IX**, (1951), 225-236.
- [8] Fletcher, C.A.J., "Burgers' Equation: A Model for all Reasons", *Numerical Solutions of Partial Differential Equations*, J. Noye (Editor), North-Holland Publ. Co. (1982).
- [9] Fletcher, C.A.J., *Computational Galerkin Methods*, Springer-Verlag, New York, New York, (1984), 114-119.
- [10] Hopf, E., "The partial differential equation $u_t + uu_x = \mu u_{xx}$ ", *Comm. Pure Appl. Math.*, **3**, (1950), 201-230.
- [11] Lighthill, M.J., "Viscosity effects in sound waves of finite amplitude", in G.K. Batchelor and R.M. Davies (eds.), *Surveys in Mechanics*, Cambridge University Press, Cambridge, (1956), 250-351.
- [12] Marrekchi, H., "Dynamic Compensators for A Nonlinear Conservation Law", Virginia Polytechnic Institute and State University, (1993).

REFERENCES

- [13] Stoer, J. and Bulirsch R., *Introduction to Numerical Analysis*, Springer-Verlag, New York, (1980).

VITA

Steven Morrison Pugh was born in Roanoke, Virginia on November 22, 1969. He graduated from William Byrd High School in Vinton, Virginia in 1987 and later attended Virginia Western Community College in Roanoke, Virginia and Virginia Commonwealth University in Richmond, Virginia. Steve transferred to Virginia Polytechnic Institute and State University in January, 1991 and graduated with Bachelor of Science and Master of Science degrees in Mathematics in 1993 and 1995, respectively.

He is a member of Pi Mu Epsilon National Mathematics Honorary Society, the American Mathematical Society, and the Society for Industrial and Applied Mathematics.

In July, 1995, Steve began work as a programmer at Circuit City Stores, Inc. in Richmond Virginia.

A handwritten signature in cursive script that reads "Steven M. Pugh". The signature is written in black ink and is centered on the page.