Spectral Methods for Circuit Analysis

by

Ognen J. Nastov

S.B. E.E., Massachusetts Institute of Technology (1991)
S.M. E.E., Massachusetts Institute of Technology (1994)

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

February 1999

Signature of Author: . . . . . .
Department of Electrical Engineering and Computer Science
January 29, 1999

Certified by: . . . . . .
Jacob K. White
Professor of Electrical Engineering
Thesis Supervisor

Accepted by: . . . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Spectral Methods for Circuit Analysis

by

Ognen J. Nastov

Submitted to the Department of Electrical Engineering and Computer Science
on January 29, 1999 in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Abstract

Harmonic balance (HB) methods are frequency-domain algorithms used for high accuracy computation of the periodic steady-state of circuits. Matrix-implicit Krylov-subspace techniques have made it possible for these methods to simulate large circuits more efficiently. However, the harmonic balance methods are not so efficient in computing steady-state solutions of strongly nonlinear circuits with rapid transitions. While the time-domain shooting-Newton methods can handle these problems, the low-order integration methods typically used with shooting-Newton methods are inefficient when high solution accuracy is required.

We first examine possible enhancements to the standard state-of-the-art preconditioned matrix-implicit Krylov-subspace HB method. We formulate the BDF time-domain preconditioners and show that they can be quite effective for strongly nonlinear circuits, speeding up the HB runtimes by several times compared to using the frequency-domain block-diagonal preconditioner. Also, an approximate Galerkin HB formulation is derived, yielding a small improvement in accuracy over the standard pseudospectral HB formulation, and about a factor of 1.5 runtime speedup in runs reaching identical solution error.

Next, we introduce and develop the Time-Mapped Harmonic Balance method (TMHB) as a fast Krylov-subspace spectral method that overcomes the inefficiency of standard harmonic balance for circuits with rapid transitions. TMHB features a non-uniform grid and a time-map function to resolve the sharp features in the signals. At the core of the TMHB method is the notion of pseudo Fourier approximations. The rapid transitions in the solution waveforms are well approximated with pseudo Fourier interpolants, whose building blocks are complex exponential basis functions with smoothly varying frequencies. The TMHB features a matrix-implicit Krylov-subspace solution approach of same complexity as the standard harmonic balance method. As the TMHB solution is computed in a pseudo domain, we give a procedure for computing the real Fourier coefficients of the solution, and we also detail the construction of the time-map function. The convergence properties of TMHB are analyzed and demonstrated on analytic waveforms.

The success of TMHB is critically dependent on the selection of a non-uniform grid. Two grid selection strategies, direct and iterative, are introduced and studied. Both strategies are a priori schemes, and are designed to obey accuracy and stability requirements. Practical issues associated with their use are also addressed.

Results of applying the TMHB method on several circuit examples demonstrate that the TMHB method achieves up to five orders of magnitude improvement in accuracy compared to the standard harmonic balance method. The solution error in TMHB decays exponentially faster than the standard HB method when the size of the Fourier basis increases linearly. The TMHB method is also up to six times faster than the standard harmonic balance method in reaching identical solution accuracy, and uses up to five times less computer memory. The TMHB runtime speedup factor and storage savings favorably increase for stricter accuracy requirements, making TMHB well suited for high accuracy simulations of large strongly nonlinear circuits with rapid transitions.

Thesis Supervisor: Jacob K. White
Title: Professor of Electrical Engineering

To my mother Svetlana and sister Vesna

Acknowledgments

First and foremost, I would like to thank my thesis advisor Jacob White. This thesis would have never materialized without his continued support, skillful guidance, and superb expertise. In addition, I am most grateful for his encouragement and patience during the ups and downs of my Ph.D. research years.

I would also like to thank my Motorola manager Brian Mulvaney. I am extremely grateful for the opportunity to conduct the last year of my Ph.D. research as well as excited to start my "real-world" career in the first-rate environment of the Circuit Simulation Group at Motorola, Inc. in Austin, Texas.

My thanks to Jing Lee and Kiran Gullapalli, two of my colleagues at the Circuit Simulation Group at Motorola, for being friendly, patient, knowledgeable, and always available resources for the many questions and long discussions related to this research during the past year.

Thanks to my thesis readers, Bernard Lesieutre and Anthony Patera for their willingness to read and evaluate this work, and for their input, suggestions, and help towards finalizing this thesis document.

Thanks, also, to all of the other members of the Circuit Simulation Group at Motorola, as well as all the former and current members of the MIT VLSI-CAD group for their help and friendship.

On a more personal note, I would like to thank all my friends from across the United States and beyond for their friendship and all the happy times we shared during the years this research took place.

Finally, for all the unconditional love, patience, support, and encouragement throughout these years, I would like to thank my family: my mother Svetlana with her husband Tom, and my sister Vesna with her husband Scott.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The exploding demand for high performance wireless products has increased the need for more efficient, accurate, and robust simulation technologies for communication and other nonlinear analog circuits (e.g. RF amplifiers, mixers, power supplies, dc-dc converters). Designers of such circuits need to compute many different quantities of interest. Some of these quantities can be obtained from a small-signal analysis; however, many nonlinear effects, such as harmonic and intermodulation distortion or compression points can only be computed by obtaining circuit's steady-state.

The conventional time-domain transient circuit simulation technique [46, 47] involves forming the system of nonlinear ordinary differential equations (ODE) that describes the dynamics of the circuit, and solving them numerically as an initial-value problem. The system of ODEs are discretized using an integration method, and the resulting nonlinear algebraic system solved using the Newton's method. The sequence of linear problems at each Newton's iteration is solved by Gaussian elimination or perhaps an iterative linear solution algorithm.

When used in computing steady-state solutions of circuits, the described transient circuit simulation approach carries a high computational cost and is thus considered impractical. For many lightly damped circuits, it takes many periods of the excitation input signal in order for the simulated circuit to reach a periodic steady-state. It is also difficult to determine when the circuit has reached a steady state. In addition, most simulated circuits are described by stiff systems of ODEs. Since the simulation must follow the fast-varying signals for accuracy, many small time-steps are required, resulting in long simulation intervals and extraordinary run-times.

If the steady-state is periodic, the steady-state analysis can be formulated as a boundary value problem for the system of nonlinear ODEs that describes the circuit. The boundary conditions are given by the periodicity requirement. The simulation technologies for periodic steady-state analysis of nonlinear circuits belong to two broad classes: time-domain, and spectral (or frequency-

domain) methods. While these methods can be far more efficient than conventional transient analysis, they have many problems of their own.

Time-domain methods include the finite differences [4], and shooting-Newton methods [1, 4]. In the finite differences method, a finite-difference approximation is used to discretize the nonlinear system of ODEs on a finite set of time-points, transforming it into a system of nonlinear algebraic equations. These equations are then solved simultaneously with the boundary constraints via Newton's method. In the shooting-Newton method, the boundary-value problem is converted into a sequence of initial-value problems. The method begins with a guess of the solution at the beginning of the shooting interval. This guess is iteratively improved via the outer Newton loop to yield the special initial condition that results in a steady-state.

The non-equally spaced time-points in the time-domain methods are selected based on local error estimation and can easily follow the rapid transitions in the circuit's steady-state waveform, so these methods work well for highly nonlinear circuits. However, these methods achieve inferior polynomial convergence proportional to the order of the integration method used. The limited order of these methods limits their accuracy. In addition, finite-difference methods yield a large ill-conditioned linear system. While shooting-Newton methods yield a much smaller linear system, they require much storage, and are unable to handle distributed devices, quasi-periodic problems, and problems with large periods.

Spectral methods [38, 40, 42, 72] represent the periodic circuit response as weighted finite sums of global basis functions. The best known spectral method is the harmonic balance (HB) method [4, 5], which uses a truncated Fourier basis to approximate the solutions. The system of nonlinear ODEs is transformed into a nonlinear algebraic system in the frequency domain. This system is solved for the spectral coefficients of the solution via Newton's method. Note that the harmonic balance method is equivalent to a finite difference method on a uniform grid of order equal to the size of the Fourier basis.

The harmonic balance method achieves a superior accuracy compared to time-domain methods due to its exponential convergence nature. Its major drawback, however, is its inability to handle highly nonlinear problems and large circuits. These problems require many terms in the Fourier representation of the signals (i.e. many time-points in the uniform grid to capture the sharp features in the waveforms), and yield a large ill-conditioned linear system, both of which present complexity and storage problems for the harmonic balance method.

Both time-domain and spectral methods for steady-state circuit analysis have recently benefited from using preconditioned iterative linear solvers for solving the linear problem at each Newton step [7, 10, 11, 12, 15, 16]. In particular, the current state-of-the-art preconditioned matrix-implicit

Krylov-subspace realization of the harmonic balance method [7, 10, 11, 15] has made this method into a winning simulation strategy for large mildly-nonlinear circuits. However, strongly nonlinear circuits, and other circuits with solutions exhibiting rapid transitions, still present a serious bottleneck for spectral steady-state analysis techniques.

Given the clear theoretical advantage of spectral methods over time-domain methods with respect to achieved solution accuracy, a pertinent question is whether it is possible to enhance the existing harmonic balance method in some way, or develop new spectral methods in order to eliminate the aforementioned simulation bottleneck for strongly nonlinear and other circuits with rapid transitions.

In this dissertation we embark on a quest for better, more powerful spectral methods for circuit analysis. We start this journey with an overview of the current state-of-the-art preconditioned matrix-implicit Krylov-subspace harmonic balance method in Chapter 2. A study focusing on enhancements to this harmonic balance method is given in Chapter 3. In particular, we examine the impact of various preconditioners, and the choice of iterative linear solver. We also analyze the relative advantages of different formulations of the harmonic balance method.

Our quest continues with the introduction of the Time-Mapped Harmonic Balance method (TMHB) in Chapter 4. We build TMHB as a fast Krylov subspace spectral method utilizing a non-uniform grid to resolve the sharp features in the signals and therefore suited to efficiently obtain highly accurate steady-state solutions of strongly nonlinear and other circuits with rapid transitions. At the core of this new method is the grid selection strategy, and in Chapter 5 we present and study several such strategies. Results from applying the TMHB method and different grid selection strategies on several circuits are given in Chapter 6. Finally, in Chapter 7 we draw conclusions and suggest some avenues for future work.

Chapter 2

Harmonic Balance Method

Consider a circuit described with N nonlinear differential equations:

$$\dot{q}(v(t)) + i(v(t)) + u(t) = 0 \tag{2.1}$$

where $v(t) \in R^N$ is the vector of node voltages, $q(v(t)) \in R^N$ the vector of node charges (or fluxes), $i(v(t)) \in R^N$ the vector of resistive node currents, and $u(t) \in R^N$ the vector of input sources.

Let the circuit be driven by a single periodic excitation input source with period T. Finding the periodic steady-state solution of this circuit consists of computing the N steady-state waveforms $v(t)$ on the solution domain $t \in [0,T]$. The periodic steady-state solution of (2.1) satisfies the two-point constraint:

$$v(T) = v(0). \tag{2.2}$$

Time-domain methods [1, 4] (finite differences, shooting-Newton method) compute the steady-state by first discretizing the solution domain $[0,T]$. The time-derivatives are approximated with finite order integration formulas. For example, for the M-th order BDF formula, the time derivative of the charge function is

$$\dot{q}(v(t_i)) \approx \frac{1}{h} \sum_{j=0}^{M} \alpha_j q(v(t_{i-j})). \tag{2.3}$$

A Backward-Euler discretization of the circuit equations (2.1) therefore yields

$$\frac{1}{h_i}[q(v(t_i)) - q(v(t_{i-1}))] + i(v(t_i)) + u(t_i) = 0. \tag{2.4}$$

The finite difference methods then solve for the discrete solution samples $v(t_i)$, while the shooting-Newton method solves the problem

$$\varphi(v(0),0,T) - v(0) = 0 \tag{2.5}$$

where $\varphi()$ is the state transition function $\varphi(v(0),0,T) = v(T)$ computed from time integration of the circuit over one period.

Since the time-derivatives in the time-domain methods are local properties of a function (i.e. are computed using values of the function at a handful of nearby time points), these methods are exact for polynomials of low order M. In other words, the solution is represented by a sequence of low-order polynomials connected at the discrete time-points t_i. These methods can therefore at best achieve polynomial convergence with global errors $O(h^M)$.

Spectral methods [38, 40, 42, 72] approximate the solution waveforms as weighted finite sums of global, orthogonal, and smooth basis functions $\phi(t)$:

$$v(t) \approx \sum_{k=0}^{M-1} a_k \phi_k(t) \tag{2.6}$$

and solve for the spectral coefficients a_k. If the boundary conditions are periodic, a truncated Fourier basis is the obvious choice, as it automatically satisfies the boundary conditions:

$$v(t+T) \approx \sum_{k=-K}^{k=K} V_k e^{j2\pi k f(t+T)} = \sum_{k=-K}^{k=K} V_k e^{j2\pi k f t} \tag{2.7}$$

where $2K+1 = M$, $f = \frac{1}{T}$ is the periodic excitation input source frequency, and V_k are the Fourier coefficients. Note that the time-derivative of this approximation can be computed exactly:

$$\frac{d}{dt} v(t) \approx \sum_{k=-K}^{k=K} j2\pi k f V_k e^{j2\pi k f t}. \tag{2.8}$$

Due to the global nature of the spectral methods, as $k \mapsto \infty$ the k-th Fourier coefficient V_k decays faster than any negative power of k if the solution waveform $v(t)$ is infinitely smooth. This rapid decay of the coefficients means that the truncated Fourier approximation of the solution expanded by a few additional terms (by slightly increasing K) represents an exceedingly good approximation of the solution. As a result, spectral methods have errors that decay faster than any negative power of M, a property known as spectral accuracy, exponential convergence, or infinite-order accuracy.

The spectral methods will not achieve spectral accuracy if a discontinuity is present in the solution waveforms or one of their derivatives. In the context of circuit simulation, the smoothness of the waveforms will depend on the smoothness of the device models used for the circuit elements. If the device models are p-times continuously differentiable functions and periodic in all its derivatives, $V_k = O(k^{-p})$, and the spectral method's errors will be $O(M^{-p})$.

2.1 Formulation

In the standard harmonic balance (HB) method [4, 5], the solution waveforms are approximated with truncated Fourier series:

$$v(t) = \sum_{k=-K}^{k=K} V_k e^{j2\pi k f t} \tag{2.9}$$

with K the number of harmonics considered in the truncation. The method solves for the Fourier coefficients V_k. The approximation (2.9), in conjunction with the N circuit equations (2.1), results in the residual function:

$$f(V,t) \equiv \sum_{k=-K}^{K} j2\pi k f Q_k e^{j2\pi k f t} + i\left(\sum_{k=-K}^{K} V_k e^{j2\pi k f t}\right) + u(t) \tag{2.10}$$

where Q_k are the Fourier coefficients of $q(v(t))$ with $v(t)$ the truncated Fourier series approximations of the solution waveforms, i.e. $q\left(\sum_{k=-K}^{K} V_k e^{j2\pi k f t}\right) = \sum_{k=-K}^{K} Q_k e^{j2\pi k f t}$.

The residual function (2.10) is to be minimized on the solution domain $[0,T]$. This minimization is typically carried out by enforcing $f(V,t_m) = 0$ on a uniform grid of collocation (interpolation) points $t_m \in \{t_1, t_2, ..., t_M\}$ where $t_m = \frac{(m-1)T}{M}$ and $M = 2K+1$. This harmonic balance method is more accurately referred to as pseudospectral (or collocation) harmonic balance.

Equation (2.10) is now rewritten in the frequency domain yielding NM nonlinear algebraic equations

$$F(V) \equiv \Omega\Gamma q(\Gamma^{-1}V) + \Gamma i(\Gamma^{-1}V) + \Gamma u = 0 \tag{2.11}$$

where V is the node-voltage spectrum $V = [V_{-K,1}, ..., V_{-K,N}, V_{-K+1,1}, ..., ..., V_{K,N}]^T$ (a vector of Fourier coefficients) and Ω is the diagonal frequency-domain differentiation matrix

$$\Omega = \begin{bmatrix} j2\pi(-K)fI_N & & & \\ & j2\pi(-K+1)fI_N & & \\ & & \ddots & \ddots \\ & & & j2\pi K f I_N \end{bmatrix} \tag{2.12}$$

where I_N is the identity matrix of size N.

The matrices Γ and Γ^{-1} are DFT matrices that perform the conversions from time to frequency

and vice-versa

$$v = \Gamma^{-1} V = \begin{bmatrix} v(t_1) \\ v(t_2) \\ \vdots \\ v(t_M) \end{bmatrix} , \tag{2.13}$$

$$\Gamma^{-1} = \begin{bmatrix} e^{j2\pi(-K)ft_1} I_N & \dots & e^{j2\pi K ft_1} I_N \\ \vdots & & \vdots \\ e^{j2\pi(-K)ft_M} I_N & \dots & e^{j2\pi K ft_M} I_N \end{bmatrix}$$

where each $v(t) \in R^N$ is a vector of node voltages. Since the grid of time-points $t_1, \ldots, t_M$ is uniform, the DFT can be carried out in $O(NM \log M)$ operations using the FFT.

In harmonic balance, the nonlinear circuit devices are evaluated in the time-domain. As it can be seen from (2.11), the node-voltage spectrum V is transformed into the time-domain, the time-domain response of the nonlinear device functions $i(v)$ and $q(v)$ is calculated, and these waveforms are then converted back into the frequency domain.

Note that this time-domain evaluation of the nonlinear devices, and the subsequent conversion of the sampled nonlinear device response to the frequency domain can be a source of errors in harmonic balance due to aliasing effects.

2.1.1 Time-Domain Formulation

The harmonic balance method can also be written in the time-domain by converting the system of equations (2.11) from frequency to time:

$$f(v) \equiv \Gamma^{-1} \Omega \Gamma q(v) + i(v) + u = 0 \tag{2.14}$$

where v is the unknown vector of node voltage samples $v = [v(t_1), v(t_2), \ldots, v(t_M)]^T$ and $D = \Gamma^{-1} \Omega \Gamma$ is the spectral differential operator in the time-domain:

$$D = \begin{bmatrix} 0 & \alpha_1 I_N & \alpha_2 I_N & \dots & \alpha_{-1} I_N \\ \alpha_{-1} I_N & 0 & \alpha_1 I_N & \dots & \alpha_{-2} I_N \\ & \ddots & \ddots & \ddots & \\ \alpha_1 I_N & \alpha_2 I_N & \dots & \alpha_{-1} I_N & 0 \end{bmatrix} . \tag{2.15}$$

The time-domain differentiation matrix D is a block-dense, real, antisymmetric circulant matrix with a zero diagonal, compared to the complex diagonal differentiation matrix Ω. The coefficients

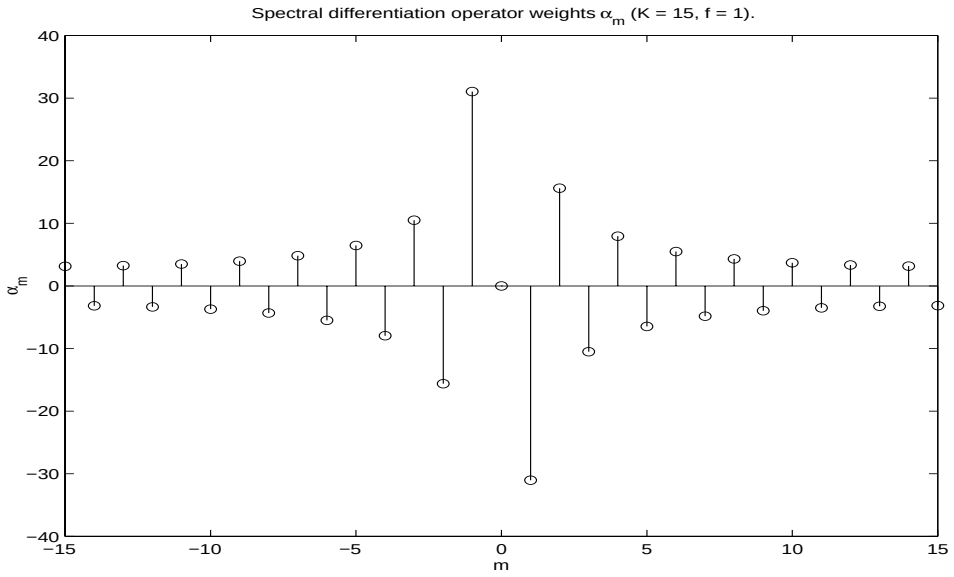

Figure 2.1: Coefficients α_m of the time-domain spectral differentiation operator D for $K = 15$ and $f = 1$.

α are the weights of the spectral time-domain difference operator, and are illustrated in Figure 2.1 for $K = 15$ and $f = 1$.

The time-domain formulation of the harmonic balance method is therefore

$$\begin{bmatrix} 0 & \alpha_1 I_N & \alpha_2 I_N & \dots & \alpha_{-1} I_N \\ \alpha_{-1} I_N & 0 & \alpha_1 I_N & \dots & \alpha_{-2} I_N \\ & \ddots & \ddots & \ddots & \\ \alpha_1 I_N & \alpha_2 I_N & \dots & \alpha_{-1} I_N & 0 \end{bmatrix} \begin{bmatrix} q(v(t_1)) \\ q(v(t_2)) \\ \dots \\ q(v(t_M)) \end{bmatrix} + \begin{bmatrix} i(v(t_1)) \\ i(v(t_2)) \\ \dots \\ i(v(t_M)) \end{bmatrix} + \begin{bmatrix} u(t_1) \\ u(t_2) \\ \dots \\ u(t_M) \end{bmatrix} = 0 \tag{2.16}$$

and can be compared to any time-domain finite difference method (for example, a finite difference method using Backward-Euler in non-uniform discretization of the domain $[0, T]$):

$$\begin{bmatrix} \frac{1}{h_1} I_N & & & -\frac{1}{h_1} I_N \\ -\frac{1}{h_2} I_N & \frac{1}{h_2} I_N & & \\ & \ddots & \ddots & \\ & & -\frac{1}{h_M} I_N & \frac{1}{h_M} I_N \end{bmatrix} \begin{bmatrix} q(v(t_1)) \\ q(v(t_2)) \\ \dots \\ q(v(t_M)) \end{bmatrix} + \begin{bmatrix} i(v(t_1)) \\ i(v(t_2)) \\ \dots \\ i(v(t_M)) \end{bmatrix} + \begin{bmatrix} u(t_1) \\ u(t_2) \\ \dots \\ u(t_M) \end{bmatrix} = 0 \tag{2.17}$$

where $t_1 = 0$, $h_1 = T - t_M$, and $h_m = t_m - t_{m-1}$ for all other m. The harmonic balance method can, therefore, be viewed as a finite-difference method of order M, the size of the Fourier basis. Note that the spectral time-domain difference operator D is much denser than the corresponding Backward-Euler difference operator (or any other operator coming from a low-order integration

scheme).

2.2 Solution Strategy

The system of NM nonlinear algebraic equations (2.11) are solved by applying the Newton's method. The application results in the iteration

$$J^{(l)}\left(V^{(l+1)}-V^{(l)}\right)=-F(V^{(l)}) \tag{2.18}$$

where l is the Newton iteration index. This iteration is a linear problem. The l-th Newton iteration Jacobian is

$$J^{(l)}=\Omega\Gamma C^{(l)}\Gamma^{-1}+\Gamma G^{(l)}\Gamma^{-1} \tag{2.19}$$

and is a $NM\times NM$ block-dense matrix.

The matrices C and G are

$$C\equiv\begin{bmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_M \end{bmatrix},\quad G\equiv\begin{bmatrix} G_1 & & & \\ & G_2 & & \\ & & \ddots & \\ & & & G_M \end{bmatrix} \tag{2.20}$$

where the blocks $C_m=[C_m(r,s)]$ and $G_m=[G_m(r,s)]$ consist of the elements

$$C_m(r,s) = \frac{dq(v_r(t_m))}{dv_s}, \tag{2.21}$$

$$G_m(r,s) = \frac{di(v_r(t_m))}{dv_s} \tag{2.22}$$

where r,s are the node indices, $r,s=1,2,\ldots,N$. The block-diagonal structure of G and C is due to the fact that the relations $q(v)$ and $i(v)$ are algebraic. The constituent sparse blocks C_m and G_m are simply the circuit capacitance and conductance matrices evaluated at the collocation time-points $t_m\in\{t_1,t_2,...,t_M\}$. Their sparsity depends on the topology of the circuit.

For linear circuits all C_m and G_m blocks are independent of the voltages and are the same, resulting in a iteration-invariant block-diagonal Jacobian. Newton's method then yields the exact solution in one iteration, and harmonic balance becomes equivalent to an AC (phasor) analysis. Note that if the time-domain formulation of harmonic balance is used, the Jacobian will be dense even for linear circuits because of the density of the spectral time-domain difference operator D that multiplies the block-diagonal C matrix. In other words, while in the frequency domain the block-diagonal structure of the frequency-domain HB Jacobian for linear circuits indicates that

`Guess` at a solution, x^0.
`Initialize` the search direction $p^0 = r^0 = b - Ax^0$.
`Set` $k = 1$.
`do` {
  `Compute` the new search direction, $p^k = Ap^{k-1}$.
  `Orthogonalize`, $p^k = p^k - \sum_{j=0}^{k-1} \beta_{k,j} p^j$.
  `Choose` α_k in
    $x^k = x^{k-1} + \alpha_k p^k$
    to minimize $\|r^k\| = \|b - Ax^k\|$.
  `If` $\frac{\|r^k\|}{\|r^0\|} < tol$, return v^k as the solution.
  `else` Set $k = k+1$.
}

Figure 2.2: GMRES algorithm for solving $Ax = b$.

there is no coupling between different frequencies, the time-domain HB Jacobian is dense because there is coupling (i.e. nonzero sensitivity) among the waveform samples $v(t_m)$ at the collocation grid times t_m.

For mildly-nonlinear circuits, the Jacobian is no longer block-diagonal. The off-diagonal blocks represent the inter-frequency coupling whose strength (measured with the ratio of the norms of the off-diagonal blocks to the diagonal blocks) will depend on the amount of nonlinearity in the circuit.

2.2.1 Matrix-Implicit Krylov-Subspace Approach

The explicit formation and direct factorization of the block-dense harmonic balance Jacobian J is computationally very expensive, $O(NM^3)$. A preconditioned iterative linear solution algorithm, such as the Generalized Minimum Residual algorithm (GMRES) [54] (summarized in Figure 2.2), can be used to reduce the factorization complexity to $O(NM^2)$. The tolerance of the linear solver can be loose, since the linear solver only computes the iterative solution updates for the outer Newton loop, and is not used for confirming convergence. Therefore, obtaining an approximate solution update by loosening the linear solver's tolerance only affects the outer Newton method's convergence rate, and not the accuracy of the final solution. In harmonic balance the GMRES tolerance default is typically 10^{-1}.

GMRES is a robust iterative algorithm for solving the linear problem. It guarantees convergence in maximum n steps (where n is the size of the linear system). It is much faster than relaxation, and only uses matrix-vector products. Furthermore, GMRES allows implicit matrices (while direct factorization requires explicit matrices). The implicit nature of GMRES also allows usage of operators such as the FFT. However, good preconditioners are essential for GMRES as GMRES is slower than direct factorization if it takes all n steps to solve the linear system.

The diagonal blocks of the Jacobian work well as a standard preconditioner in many circuit examples, particularly for mildly nonlinear circuits with weak off-diagonal blocks in the Jacobian [11, 15]. This frequency-domain block-diagonal preconditioner is

$$P_f = \begin{bmatrix} j2\pi(-K)f\bar{C}_0 + \bar{G}_0 & & & \\ & j2\pi(-K+1)f\bar{C}_0 + \bar{G}_0 & & \\ & & \ddots & \\ & & & j2\pi K f\bar{C}_0 + \bar{G}_0 \end{bmatrix} \tag{2.23}$$

where the blocks $\bar{C}_0 = [\bar{C}_0(r,s)]$ and $\bar{G}_0 = [\bar{G}_0(r,s)]$ contain the elements

$$\bar{C}_0(r,s) = \frac{1}{M}\sum_{m=1}^{M} C_m(r,s), \tag{2.24}$$

$$\bar{G}_0(r,s) = \frac{1}{M}\sum_{m=1}^{M} G_m(r,s) \tag{2.25}$$

with $r,s = 1,2,\ldots,N$ the node indices. $\bar{C}_0$ and $\bar{G}_0$ simply represent matrices made up from the DC (k = 0) Fourier coefficients of the circuit's capacitance and conductance matrices, with the averaging in (2.24) and (2.25) done on node by node basis.

To demonstrate the effectiveness of this block diagonal preconditioning consider the operational transconductance (OTA) amplifier circuit shown in Figure 2.3. Table 2.1 illustrates that the number of GMRES iterations is greatly reduced by the block-diagonal preconditioning. In addition, Table 2.1 indicates that the number of GMRES iterations increases with number of harmonics without preconditioning.

The GMRES linear solver requires forming the matrix-vector product

$$Jp^k = (\Omega\Gamma C + \Gamma G)\,\Gamma^{-1}p^k \tag{2.26}$$

where p^k is the search direction in the k-th GMRES iteration. This matrix-vector product can be formed implicitly by a sequential evaluation using 3 FFTs, reducing the complexity of the harmonic balance method to the complexity of the time-frequency conversions, which is $O(NM\log M)$ for the FFTs. To illustrate this again consider the operational transconductance (OTA) amplifier

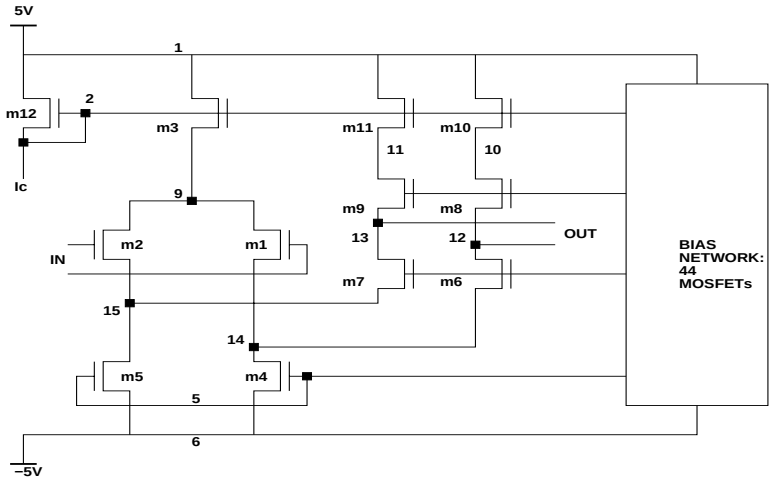

Figure 2.3: OTA Amplifier (56 MOSFETs).

K	I_L (GMRES)	I_L (PGMRES)
10	139	41
14	155	48
18	159	49
20	173	47
32	200	41

Table 2.1: Effectiveness of block-diagonal preconditioning in harmonic balance analysis of the OTA amplifier. K is the number of harmonics, and I_L the number of linear iterations in a typical Newton iteration. PGMRES refers to a GMRES method with block-diagonal preconditioning.

circuit shown in Figure 2.3. The advantages of the matrix-implicit approaches are demonstrated in Figure 2.4, which shows the computational cost vs. the number of harmonics for the different factorization approaches in the harmonic balance method.

The extrapolated slopes (straight dashed lines) in Figure 2.4 correspond to the exponent β in the $O(M^\beta)$ observed computational complexity versus the number of harmonics K, where $M = 2K+1$. For the Gaussian factorization approach $\beta = 3.1$, for the preconditioned GMRES scheme $\beta = 1.8$, and for the matrix-implicit preconditioned GMRES approach $\beta = 1.1$, i.e. slightly faster than linear as expected from an $M \log M$ dependence.

The FFT-based, matrix-implicit, GMRES preconditioned pseudospectral harmonic balance method [7, 10, 11, 15] is the current state-of-the art steady-state circuit analysis technique. It is commonly used to analyze circuits with hundreds of devices. A number of other less effective

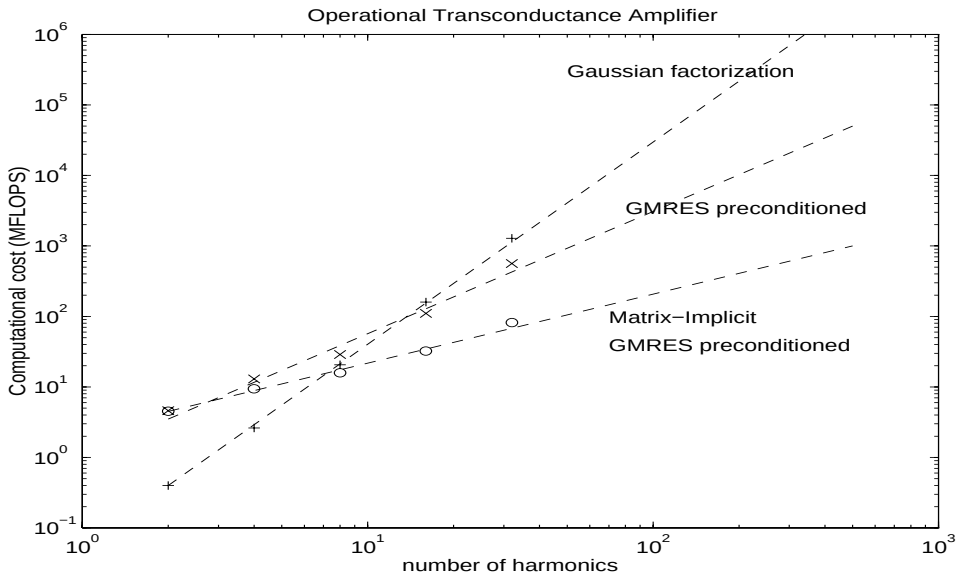

Figure 2.4: The superiority of the matrix-implicit preconditioned approach in harmonic balance analysis of the OTA amplifier.

modifications to improve the accuracy [3], the convergence [8, 17] and the computational efficiency [9] of the harmonic balance method have also been investigated. This method, however, is still not universally successful when applied to different circuits [11, 15]. In particular, highly nonlinear circuits, as well as other circuits with rapid transitions still cause problems. These circuits exhibit waveforms with sharp features which require many harmonics, and many linear (GMRES) and nonlinear (Newton) iterations in order for the method to converge as well as to obtain a reasonable accuracy. In the next Chapter, we study possible ways to enhance the harmonic balance method in order to overcome these limitations.

Chapter 3

Enhancements to Harmonic Balance

In this Chapter we study possible enhancements to the standard harmonic balance (HB) method reviewed in Chapter 2. We will first examine the impact of the choice of different preconditioners for the GMRES iterative linear solver in the matrix-implicit Krylov-subspace HB method. We will also derive the approximate Galerkin formulation of the harmonic balance method and compare its performance to the standard pseudospectral formulation. At the end, we add a few remarks concerning the choice of an iterative linear solver in HB.

3.1 Time-Domain Preconditioners

If a linear system is solved by an iterative method such as GMRES, the convergence rate of the solver depends on the spectral properties of the coefficient matrix. The idea of preconditioning is to transform the original linear system into one that is equivalent (i.e. has the same solution), but which has more favorable spectral properties. The preconditioner is a matrix that performs this transformation. One way to design a preconditioner is to construct a matrix that somehow approximates the coefficient matrix, and is easy to invert.

For example, in the frequency-domain harmonic balance formulation, the l-th Newton iteration linear problem

$$J^{(l)}\Delta V^{(l)} = -F(V^{(l)}) \tag{3.1}$$

where $\Delta V^{(l)} = V^{(l+1)} - V^{(l)}$, is transformed into

$$JP^{-1}P\Delta V = -F(V) \tag{3.2}$$

using a right-preconditioning approach. The Newton iteration indices were dropped for the sake

of clarity. Setting $P\Delta V = U$, we then first solve the system

$$JP^{-1}U = -F(V) \tag{3.3}$$

with GMRES. Since the preconditioner P was chosen to be easily invertible, the GMRES matrix-vector products $JP^{-1}p^k$ can be computed efficiently. To further speed up the matrix-vector product, we can invert the preconditioner once, and store it in its LU decomposed form. If the preconditioner is well designed, the linear system (3.3) will have more favorable spectral properties than the original linear system (3.1), and GMRES will be able to converge (or reach the desired solution tolerance) in significantly fewer iterations. At the end, we obtain the actual solution ΔV from:

$$\Delta V = P^{-1}U. \tag{3.4}$$

The advantages of the right style of preconditioning (versus left) is that right preconditioning does not change the right hand side $-F(V)$ of the system, and therefore the RHS of the original linear problem, transformed linear problem, and the outer Newton loop stay the same. In contrast, a left preconditioning GMRES will in essence solve the original linear problem to various degrees of accuracy in each Newton iteration due to the scaling of the RHS of the original linear system by P^{-1} in the transformed linear problem. In practice, left preconditioning has shown to cause costly increases in both the numbers of GMRES and Newton iterations.

In the frequency-domain formulation of harmonic balance, the diagonal blocks of the frequency-domain Jacobian

$$P_f = \begin{bmatrix} j2\pi(-K)f\bar{C_0} + \bar{G_0} & & & \\ & j2\pi(-K+1)f\bar{C_0} + \bar{G_0} & & \\ & & \ddots & \\ & & & j2\pi K f\bar{C_0} + \bar{G_0} \end{bmatrix} \tag{3.5}$$

are commonly used as a right preconditioner. This preconditioner can be easily and efficiently formed and inverted due to its block-diagonal structure.

The block-diagonal preconditioner (3.5) works well for most circuits, but not as well for highly nonlinear circuits [11, 15]. In Chapter 2 we mentioned that the off-diagonal blocks in the frequency domain HB Jacobian represent the inter-frequency coupling; their strength depends on the degree of nonlinearity of the circuit. The block-diagonal preconditioner does not include these blocks, which is why this preconditioner is not as efficient for highly nonlinear circuits.

Time-domain steady-state methods, such as finite-differences, are well-suited for nonlinear circuits. Since the harmonic balance method can be viewed as a time-domain method, we will use

this point of view to design preconditioners that we hope to be more efficient in simulating highly nonlinear circuits. Note that centered-difference preconditioning in spectral methods is considered in [42], upwind finite-difference preconditioning in [80], and finite element preconditioning in [69].

The time-domain formulation of the harmonic balance method given in Section 2.1 showed that the harmonic balance method is essentially a time-domain method of order M, which is the size of the Fourier basis used in the truncated series approximation of the solution waveforms. If the time-domain harmonic balance formulation

$$f(v) \equiv \Gamma^{-1}\Omega\Gamma q(v) + i(v) + u = 0 \tag{3.6}$$

is solved by the Newton's method, the linear problem solved at the l-th Newton iteration is

$$J_t^{(l)} \Delta v^{(l)} = -f(v^{(l)}) \tag{3.7}$$

where $\Delta v^{(l)} = v^{(l+1)} - v^{(l)}$.

The time-domain Jacobian at the l-th Newton iteration is

$$J_t^{(l)} = (\Gamma^{-1}\Omega\Gamma)C^{(l)} + G^{(l)} = DC^{(l)} + G^{(l)} \tag{3.8}$$

and it is a real block-dense matrix whose structure is

$$J_t = \begin{bmatrix} G_1 & \alpha_1 C_2 & \alpha_2 C_3 & \dots & \alpha_{-1} C_M \\ \alpha_{-1} C_1 & G_2 & \alpha_1 C_3 & \dots & \alpha_{-2} C_M \\ & \ddots & \ddots & \ddots & \\ \alpha_1 C_1 & \alpha_2 C_2 & \dots & \alpha_{-1} C_{M-1} & G_M \end{bmatrix} \tag{3.9}$$

where the coefficients α are the weights of the spectral differentiation operator in the time-domain.

We would like to design a preconditioner that will in some way approximate the time-domain Jacobian J_t, but will have an easily invertible block structure. Since the time-domain formulation of the harmonic balance method is a finite-difference method in disguise, one idea is use the Jacobian matrix from a lower-order finite difference scheme on a uniform grid as a preconditioner. For example, the Backward-Euler (or BDF-1) finite difference Jacobian on a uniform grid is

$$J_{BE} = \begin{bmatrix} \frac{C_1}{h} + G_1 & & & & -\frac{C_M}{h} \\ -\frac{C_1}{h} & \frac{C_2}{h} + G_2 & & & \\ & \ddots & \ddots & & \\ & & & -\frac{C_{M-1}}{h} & \frac{C_M}{h} + G_M \end{bmatrix}. \tag{3.10}$$

Similarly, the uniform grid BDF-2 finite-difference Jacobian is

$$J_{BDF2} = \begin{bmatrix} \frac{3C_1}{2h}+G_1 & & & \frac{C_{M-1}}{2h} & -\frac{C_M}{h} \\ -\frac{2C_1}{h} & \frac{3C_2}{2h}+G_2 & & & \frac{C_M}{2h} \\ \frac{C_1}{2h} & -\frac{2C_2}{h} & \frac{3C_3}{2h}+G_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ & & \frac{C_{M-2}}{2h} & -\frac{2C_{M-1}}{h} & \frac{3C_M}{2h}+G_M \end{bmatrix} \tag{3.11}$$

and so on for the higher-order BDF integration schemes. In general, the BDF-n time-domain preconditioner will have $n+1$ bands of blocks. For a more efficient factorization, the cluster of blocks in the upper-right corner of the finite-difference Jacobians is ignored. For example, the BDF-2 preconditioner $P_{t,2}$ is:

$$P_{t,2} = \begin{bmatrix} \frac{3C_1}{2h}+G_1 & & & & \\ -\frac{2C_1}{h} & \frac{3C_2}{2h}+G_2 & & & \\ \frac{C_1}{2h} & -\frac{2C_2}{h} & \frac{3C_3}{2h}+G_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ & & \frac{C_{M-2}}{2h} & -\frac{2C_{M-1}}{h} & \frac{3C_M}{2h}+G_M \end{bmatrix}. \tag{3.12}$$

3.1.1 Complexity Analysis of Preconditioning Approaches

The complexity of the preconditioned matrix-implicit Krylov-subspace harmonic balance is limited by the FFT to $O(NM\log M)$. It is therefore crucial to investigate the number of required FFTs in each GMRES iteration for different combinations of preconditioning approaches and harmonic balance domain formulations.

For the time-domain HB formulation, a time-domain preconditioner P_t transforms the matrix-vector product into

$$J_t P_t^{-1} p^k = (\Gamma^{-1}\Omega\Gamma C + G) P_t^{-1} p^k \tag{3.13}$$

where p^k is the time-domain search direction in the k-th GMRES iteration. The above matrix-vector product requires only 2 FFTs. On the other hand, the matrix-vector product in the frequency-domain formulation using the frequency-domain block-diagonal preconditioner P_f is

$$J P_f^{-1} p^k = (\Omega\Gamma C + \Gamma G)\,\Gamma^{-1} P_f^{-1} p^k \tag{3.14}$$

and requires 3 FFTs.

If we now reverse the situation, namely, use the frequency-domain preconditioner in the time-domain HB formulation, the matrix-vector product becomes

$$J_t \Gamma^{-1} P_f^{-1} \Gamma p^k = (\Gamma^{-1} \Omega \Gamma C + G) \Gamma^{-1} P_f^{-1} \Gamma p^k \tag{3.15}$$

and now requires 4 FFTs. In comparison, using a time-domain preconditioner in the frequency domain HB formulation, results in the matrix-vector product

$$J \Gamma P_t^{-1} \Gamma^{-1} p^k = (\Omega \Gamma C + \Gamma G) \Gamma^{-1} \Gamma P_t^{-1} \Gamma^{-1} p^k = (\Omega \Gamma C + \Gamma G) P_t^{-1} \Gamma^{-1} p^k \tag{3.16}$$

which again requires 3 FFTs. While we can conclude that it is best to have both time and frequency HB formulations implemented, if we have to choose one formulation, it would be the frequency-domain one, particularly since the frequency-domain preconditioner works well for most circuits and should therefore be the default choice of a preconditioner.

3.1.2 Time-Domain Preconditioning Results

In order to investigate the potential merits of the BDF preconditioners, we used the standard, right preconditioned GMRES matrix-implicit HB method to simulate two highly nonlinear circuits: a diode rectifier, and a DC-DC converter. Statistics for these two circuits are given in Table 6.1 (Chapter 6). The block-diagonal frequency domain preconditioner is not effective for these two circuits.

The standard HB method was implemented in Mica, Motorola's SPICE-like circuit simulator, in the computer language C [55]. All computer runs were done on Sun Ultra-2 350MHz UNIX workstations.

Three different preconditioners were used in the HB GMRES solver: the block-diagonal frequency domain preconditioner (FREQ), and the BDF-1 and BDF-2 time-domain preconditioners. The HB method was formulated in the frequency domain. The DC (equilibrium, $\dot{q} = 0$) solution of the circuits was used as an initial Newton guess.

Figure 3.1 summarizes the influence of the three preconditioners on the Newton and GMRES solvers in HB simulation of the diode rectifier circuit. On the left, plot (A) shows the total number of Newton iterations I_N for each HB run at increasing number of harmonics K. While FREQ induces increases in the number of Newton iterations, the BDF preconditioners keep this number independent of K. In plot (B), the average number of GMRES iterations per Newton step I_L/I_N is plotted against the number of harmonics K (where I_L is the total number of GMRES iterations). The FREQ preconditioner is again visibly ineffective and causes an increase in the average number

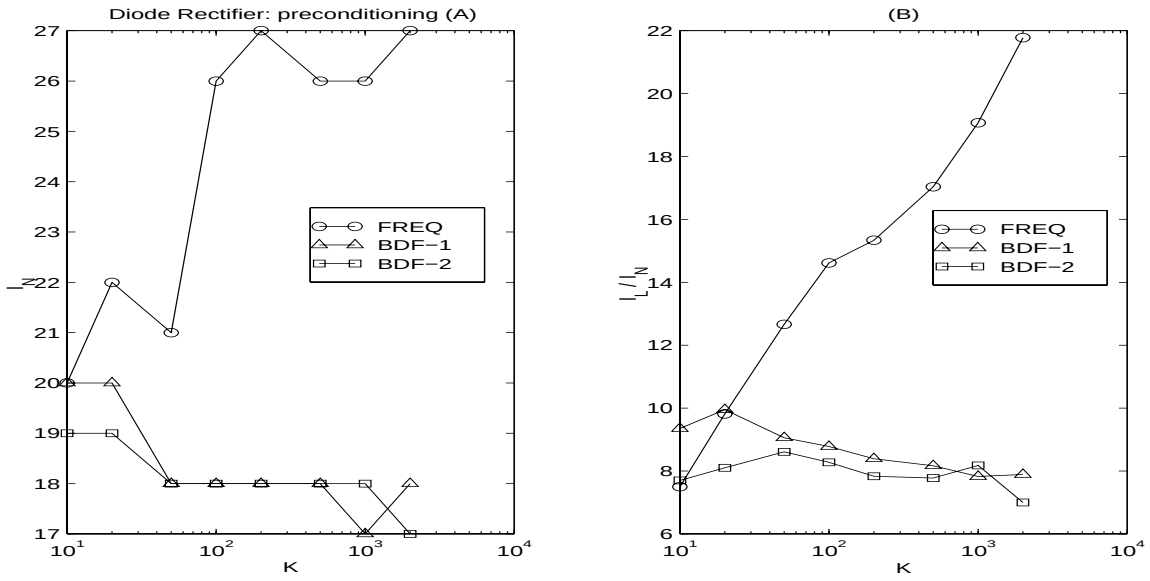

Figure 3.1: Diode Rectifier: (A) Newton iterations; (B) average GMRES iterations for different preconditioners.

of linear iterations, while the time-domain BDF preconditioners are quite effective and keep the both the average and the total number of linear iterations independent of K.

Figure 3.2 shows the total CPU time performance for the preconditioned HB approaches in simulating the diode rectifier. The performance of BDF-1 and BDF-2 is almost identical, and both perform increasingly better than FREQ when a growing number of harmonics are considered. For example, at $K = 1000$, HB runs using BDF-1 and BDF-2 are twice faster than the FREQ preconditioner.

The HB simulations of the second circuit, the DC-DC converter, are summarized in Figures 3.3 and 3.4. While the number of Newton iterations are similar, and increase for all preconditioners, the BDF preconditioners once again maintain the average number of GMRES iterations per Newton step independent of the number of harmonics K. This results in considerable total HB CPU time speedups for large K: for example, at $K = 1000$, the HB run using the BDF-2 preconditioner is 3.7 times faster than the HB run using the FREQ preconditioner.

We can conclude that the BDF preconditioners can be quite effective for highly nonlinear circuits. They keep the average number of GMRES iterations bounded, and can have a similar effect on the number of Newton iterations as well. Factoring the BDF preconditioners is more expensive than factoring the block-diagonal FREQ preconditioner since they contain off-diagonal blocks. However, due to the independence of the average number of GMRES iterations from the number of harmonics K in the case of BDF preconditioners, a performance speedup is present for larger

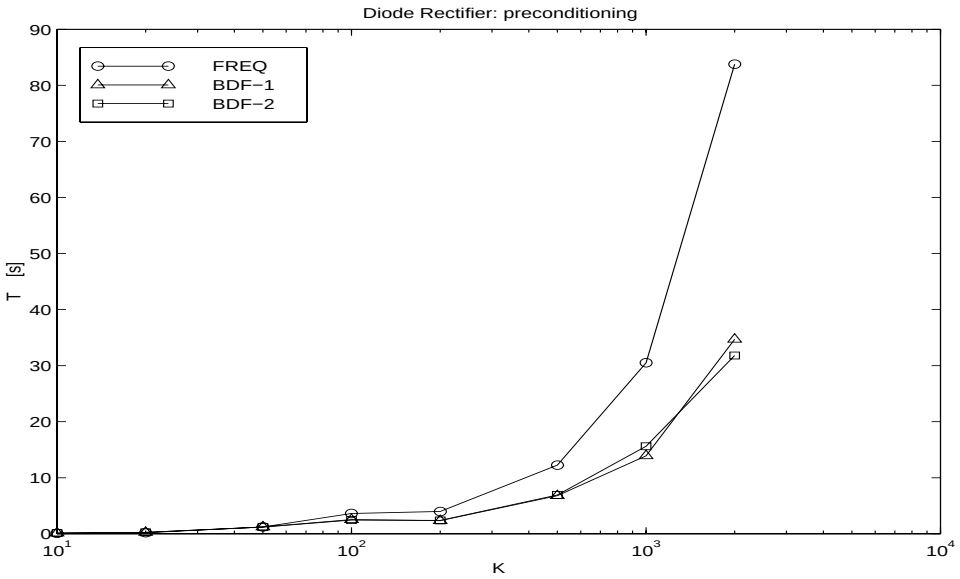

Figure 3.2: Diode Rectifier: CPU time performance of different preconditioners.

K. The HB method using BDF preconditioners is shown to be several times faster than the HB method using the FREQ preconditioner, particularly in the runs with large number of harmonics K; in fact, the speedup factors increase with increasing number of harmonics K.

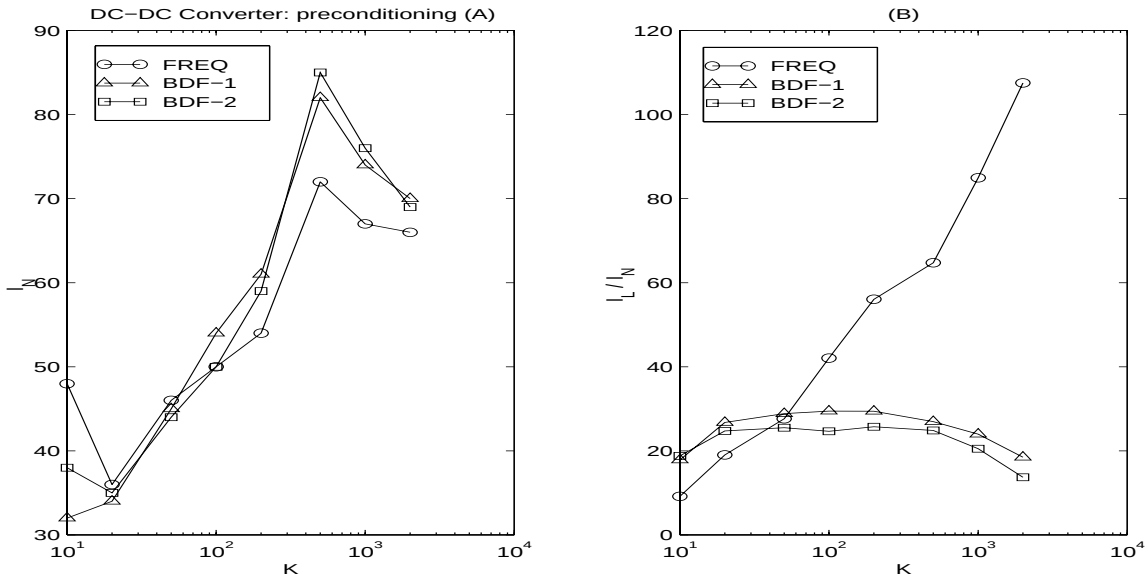

Figure 3.3: DC-DC Converter: (A) Newton iterations; (B) average GMRES iterations for different preconditioners.

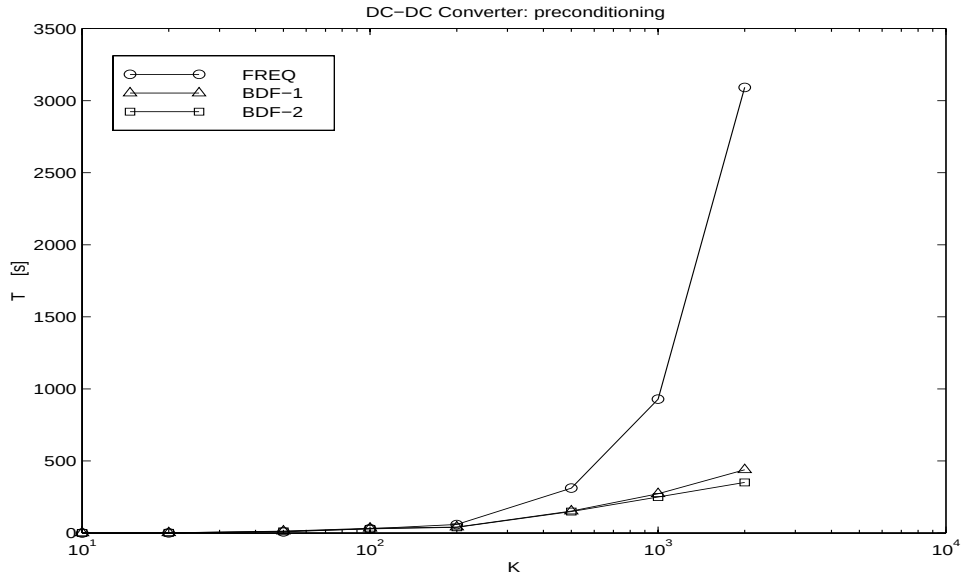

Figure 3.4: DC-DC Converter: CPU time performance of different preconditioners.

3.2 Approximate Galerkin Harmonic Balance

In this section we introduce a different formulation of the harmonic balance method. Recall from Chapter 2 that in the harmonic balance the solution waveforms are approximated with truncated Fourier series

$$v(t) = \sum_{k=-K}^{k=K} V_k e^{j2\pi kft}. \tag{3.17}$$

This approximation, in conjunction with the N circuit equation

$$\dot{q}(v(t)) + i(v(t)) + u(t) = 0 \tag{3.18}$$

results in the residual function

$$f(V,t) \equiv \sum_{k=-K}^{K} j2\pi kfQ_k e^{j2\pi kft} + i\left(\sum_{k=-K}^{K} V_k e^{j2\pi kft}\right) + u(t) \tag{3.19}$$

where Q_k are the Fourier coefficients of $q(v(t))$ with $v(t)$ the truncated Fourier series approximations of the solution waveforms, i.e. $q\left(\sum_{k=-K}^{K} V_k e^{j2\pi kft}\right) = \sum_{k=-K}^{K} Q_k e^{j2\pi kft}$.

The residual function (3.19) is to be minimized on the solution domain $[0,T]$. In the collocation or pseudospectral harmonic balance method, this minimization is carried out by enforcing $f(V,t_m) = 0$ on a uniform grid of collocation (interpolation) points $t_m \in \{t_1, t_2, ..., t_M\}$ where $t_m = \frac{(m-1)T}{M}$ and $M = 2K+1$.

Instead of using a collocation method, we can minimize the norm of the residual $f(V,t)$ using a mean weighted residual method

$$\frac{1}{T}\int_0^T f(V,t)\omega_l(t)dt = 0 \tag{3.20}$$

using a set of M suitable test functions $\omega_l(t)$. For example, if the set of test functions is $\omega_l(t) = t^l$, the moment method is generated. If $\omega_l(t) = \delta(t - t_l)$ with t_l the same time points forming the collocation grid, the collocation (pseudospectral) method is generated.

When the sets of basis and test functions coincide, the mean weighted residual method becomes the Galerkin (spectral) method. The Galerkin method is theoretically better than the collocation (pseudospectral) method. Galerkin's method reduces the global error across the continuum of the solution domain $[0,T]$ by using an integral norm of the residual. In comparison, the collocation method simply ensures that the residual vanishes at a discrete collocating set of points distributed uniformly across the solution domain.

Both the Galerkin and the collocation method use a finite set of trigonometric polynomials (i.e. the truncated Fourier basis) to approximate the solution waveforms. As it will be shown in the

next section, the set of Fourier coefficients for the Galerkin method are the truncated set of the exact Fourier coefficients of the infinite Fourier series of the solution. The computed Galerkin approximations of the solutions are thus also known as Fourier truncations.

In comparison, the collocation method uses a different set of Fourier coefficients. The collocation Fourier coefficients are such that the approximate collocation solution interpolates (i.e. exactly passes through) the set of collocation points. The computed collocation approximations of the solutions are thus known as Fourier interpolations.

The computation of the exact Fourier coefficients in the Galerkin method requires an exact evaluation of integrals, while the DFT (or FFT) can be used to compute the Fourier coefficients in the collocation method. This explains the popularity of the collocation method for complicated nonlinear problems.

3.2.1 Derivation

Given the theoretical advantages of the Galerkin method over collocation, we now proceed to derive the Galerkin harmonic balance method. We choose the set of test functions to be the same set of $M = 2K+1$ Fourier basis functions used in approximating the solution waveforms, i.e. $\omega_l(t) = e^{j2\pi l ft}$ with $l \in [-K, -K+1, ..., K]$. The Fourier basis functions are periodic and mutually orthogonal, which can results in simplified evaluation of the integral in (3.20).

Applying the Galerkin method to minimize the harmonic balance residual function (3.19) yields a system of NM nonlinear equations

$$F_l(V) \equiv \frac{1}{T}\int_{t=0}^{T} f(V,t)e^{-j2\pi l ft}dt = 0 \tag{3.21}$$

where $l \in [-K, -K+1, ..., K]$. Expanding for $f(V,t)$, the system becomes

$$F_l(V) \equiv \frac{1}{T}\int_{t=0}^{T} \dot{q}(v(t))e^{-j2\pi l ft}dt + \frac{1}{T}\int_{t=0}^{T} i(v(t))e^{-j2\pi l ft}dt + \frac{1}{T}\int_{t=0}^{T} u(t)e^{-j2\pi l ft}dt = 0 \tag{3.22}$$

where $v(t)$ is the truncated Fourier series approximation in (3.17).

The Galerkin system of equations (3.22) is solved by applying the Newton's method. The Jacobian for the Newton iteration is

$$J = J_q + J_i = \frac{\partial F_q(V)}{\partial V} + \frac{\partial F_i(V)}{\partial V} \tag{3.23}$$

where $F_q = [F_{q,-K}, F_{q,-K+1}, ..., F_{q,K}]^T$ and $F_i = [F_{i,-K}, F_{i,-K+1}, ..., F_{i,K}]^T$ are the terms in the Galerkin equations (3.22) corresponding to the integrals of the node charges time derivative $\dot{q}(v)$ and currents $i(v)$ respectively, i.e.

$$F_{q,l}(V) = \frac{1}{T}\int_{t=0}^{T} \dot{q}(v(t))e^{-j2\pi l ft}dt, \tag{3.24}$$

$$F_{i,l}(V) = \frac{1}{T}\int_{t=0}^{T} i(v(t))e^{-j2\pi l\, ft}dt. \tag{3.25}$$

For the sake of clarity, we will focus on the resistive part of the Jacobian J_i; the derivation of the capacitive part J_q is similar.

Using the chain rule, the resistive part of the Jacobian J_i is

$$J_i = \frac{\partial F_i(V)}{\partial V} = \frac{\partial F_i(V)}{\partial v}\frac{\partial v}{\partial V}. \tag{3.26}$$

Substituting (3.25) into (3.26) we obtain

$$J_i = \frac{1}{T}\int_{t=0}^{T} dt \frac{\partial i(v(t))}{\partial v} \begin{bmatrix} e^{-j2\pi(-K)\, ft} \\ e^{-j2\pi(-K+1)\, ft} \\ \vdots \\ e^{-j2\pi(K)\, ft} \end{bmatrix} \begin{bmatrix} \frac{\partial v}{\partial V_{-K}} & \frac{\partial v}{\partial V_{-K+1}} & \cdots & \frac{\partial v}{\partial V_K} \end{bmatrix}. \tag{3.27}$$

Since $v(t)$ is approximated by the truncated Fourier series (3.17), the terms $\frac{\partial v}{\partial V_k} = e^{j2\pi(k)\, ft}$, and we arrive at

$$J_i = \frac{1}{T}\int_{t=0}^{T} dt \frac{\partial i(v(t))}{\partial v} \begin{bmatrix} e^{-j2\pi(0)\, ft} & e^{-j2\pi(1)\, ft} & \cdots & e^{-j2\pi(2K)\, ft} \\ e^{-j2\pi(-1)\, ft} & e^{-j2\pi(0)\, ft} & \cdots & e^{-j2\pi(2K-1)\, ft} \\ \vdots & \vdots & \vdots & \vdots \\ e^{-j2\pi(-2K)\, ft} & e^{-j2\pi(-2K+1)\, ft} & \cdots & e^{-j2\pi(0)\, ft} \end{bmatrix}. \tag{3.28}$$

The resistive part of the Jacobian J_i is therefore a matrix whose entries are Fourier transform integrals. Each of these integrals picks out the exact Fourier coefficients $\tilde{G}_k^*$ of $\frac{\partial i(v(t))}{\partial v}$ i.e.

$$J_i = \begin{bmatrix} \tilde{G}_0^* & \tilde{G}_{-1}^* & \cdots & \tilde{G}_{-2K}^* \\ \tilde{G}_1^* & \tilde{G}_0^* & \cdots & \tilde{G}_{-2K+1}^* \\ \vdots & \vdots & \cdots & \vdots \\ \tilde{G}_{2K}^* & \tilde{G}_{2K-1}^* & \cdots & \tilde{G}_0^* \end{bmatrix}. \tag{3.29}$$

In practice, the Fourier transform integrals will be evaluated by a numerical integration rule, and the $\tilde{G}_k^*$ blocks will only be approximations to the exact Fourier coefficients. It is particularly interesting to see what happens if the trapezoidal rule is used for this evaluation. Since the integrands are periodic, the trapezoidal rule is equivalent to Gaussian quadrature and is exponentially accurate. If the grid used in the trapezoidal rule evaluation is the set of M uniformly spaced time points $t_m \in t_1, t_2, ..., t_M$ with $t_m = \frac{(m-1)T}{M}$, the trapezoidal rule is equivalent to an M-point DFT. The resistive

part of the Jacobian then becomes a circulant matrix

$$J_i = \begin{bmatrix} \tilde{G}_0 & \tilde{G}_{-1} & \cdots & \tilde{G}_1 \\ \tilde{G}_1 & \tilde{G}_0 & \cdots & \tilde{G}_2 \\ \vdots & \vdots & \cdots & \vdots \\ \tilde{G}_{-1} & \tilde{G}_{-2} & \cdots & \tilde{G}_0 \end{bmatrix} = \Gamma G \Gamma^{-1} \tag{3.30}$$

and is identical to the resistive part of the Jacobian J_i in the pseudospectral harmonic balance formulation. A similar result can be obtained for the capacitive part J_q of the Jacobian.

The Galerkin harmonic balance method is, therefore, equivalent to the pseudospectral harmonic balance when the Galerkin integrals are evaluated using a trapezoidal rule on a uniform grid of same size M as the size of the Fourier basis.

In order to achieve a better approximation to the resistive part of the actual Galerkin harmonic balance Jacobian (3.29) (which contains exact Fourier coefficients), as well as to the capacitive part, the number of uniformly spaced points in the trapezoidal rule can be increased to $M_\sigma = \sigma 2K + 1$ (where $\sigma > 1$). This approximate Galerkin Jacobian can be computed from the pseudospectral harmonic balance Jacobian on the uniform collocation grid of M_σ points : it is the central $NM \times NM$ part of this pseudospectral HB Jacobian. This approximate Galerkin Jacobian will converge to the actual Galerkin Jacobian in the limit $\sigma \mapsto \infty$.

When the pseudospectral harmonic balance uses the larger M_σ point uniform collocation grid and the oversampled DFT transform matrices Γ and Γ^{-1} of size NM_σ, the method is an oversampled pseudospectral HB (i.e. a pseudospectral HB with oversampling transforms).

In summary, the middle $NM \times NM$ part of the oversampled pseudospectral Jacobian approximates the actual Galerkin Jacobian increasingly better for increasing transform lengths. In other words, the oversampled pseudospectral HB represents an approximate Galerkin HB method (AGHB).

As an illustration, consider the case $K = 1$ (i.e. $M = 3$). When $\sigma = 1$ (no oversampling), the resistive part of the pseudospectral Jacobian is

$$J_i^{(C)} = \begin{bmatrix} \tilde{G}_0 & \tilde{G}_{-1} & \tilde{G}_1 \\ \tilde{G}_1 & \tilde{G}_0 & \tilde{G}_{-1} \\ \tilde{G}_{-1} & \tilde{G}_1 & \tilde{G}_0 \end{bmatrix}. \tag{3.31}$$

We now introduce oversampling with $\sigma = 2$. The collocation grid is now $M_\sigma = 5$, and the resistive

part of the oversampled pseudospectral Jacobian is

$$J_i = \begin{bmatrix} \tilde{G}_0 & \tilde{G}_{-1} & \tilde{G}_{-2} & \tilde{G}_2 & \tilde{G}_1 \\ \tilde{G}_1 & \tilde{G}_0 & \tilde{G}_{-1} & \tilde{G}_{-2} & \tilde{G}_2 \\ \tilde{G}_2 & \tilde{G}_1 & \tilde{G}_0 & \tilde{G}_{-1} & \tilde{G}_{-2} \\ \tilde{G}_{-2} & \tilde{G}_2 & \tilde{G}_1 & \tilde{G}_0 & \tilde{G}_{-1} \\ \tilde{G}_{-1} & \tilde{G}_{-2} & \tilde{G}_2 & \tilde{G}_1 & \tilde{G}_0 \end{bmatrix}. \tag{3.32}$$

Its middle 3×3 part represents the resistive part of the approximate Galerkin Jacobian:

$$J_i^{(G)} = \begin{bmatrix} \tilde{G}_0 & \tilde{G}_{-1} & \tilde{G}_{-2} \\ \tilde{G}_1 & \tilde{G}_0 & \tilde{G}_{-1} \\ \tilde{G}_2 & \tilde{G}_1 & \tilde{G}_0 \end{bmatrix}. \tag{3.33}$$

Note the differences between the lower-left $(1,3)$ and upper-right $(3,1)$ corner terms in (3.33) and (3.31).

Given the theoretical advantages of the Galerkin method over collocation, we may suspect some sort of accuracy enhancements by using the oversampling transforms in the pseudospectral HB method. Consider the computation of $\tilde{G}_l^*$ with the Galerkin (spectral) method

$$\tilde{G}_l^* = \frac{1}{T}\int_0^T \frac{di(v)}{dv} e^{-j2\pi l f t} dt \tag{3.34}$$

with the uniform M_σ-point trapezoidal rule (or equivalently, with an oversampled pseudospectral method with oversampling factor σ). Define

$$g^*(t) = \frac{di(v(t))}{dv} \tag{3.35}$$

and let its exact infinite Fourier representation be given by

$$g^*(t) = \sum_{k=-\infty}^{\infty} G_k^* e^{j2\pi k f t}. \tag{3.36}$$

Let the approximate pseudospectral representation with truncated Fourier series be given by

$$g(t) = \sum_{k=-K}^{K} \tilde{G}_k e^{j2\pi k f t}. \tag{3.37}$$

Using the trigonometric interpolation theorem [43], it can be shown that $\tilde{G}_k$ differs from $\tilde{G}_k^*$ by a perturbing aliasing sum:

$$\tilde{G}_k = \tilde{G}_k^* + \sum_{j=-\infty,\ j\neq 0}^{\infty} \tilde{G}_{k+jM_\sigma}^* \tag{3.38}$$

where $M_\sigma = \sigma 2K + 1$.

The presence of the aliasing term in the collocation Fourier coefficients indicates that the error for the collocation method will always be greater that the error for the Galerkin method. On the other hand, it can be shown that the aliasing term is asymptotically of the same order as the truncation error [67, 39], which implies that the errors for the Galerkin and collocation methods will decay at the same rate in the asymptotic limit [42].

To illustrate the perturbing aliasing terms in the approximate Fourier coefficients $\tilde{G}_k$, look at the case $K = 1$, and no oversampling (i.e. $\sigma = 1$). The approximate Fourier coefficient $\tilde{G}_2$ is

$$\tilde{G}_2 = \tilde{G}_2^* + (\tilde{G}_{-1}^* + \tilde{G}_5^* + \tilde{G}_{-4}^* + \tilde{G}_8^*). \tag{3.39}$$

Note the presence of the $\tilde{G}_{-1}^*$ coefficient in the aliasing sum which is particularly worrisome, since typically $|\tilde{G}_{-1}^*| > |\tilde{G}_2^*|$.

If the oversampling is set to $\sigma = 2$, the same approximate Fourier coefficient is

$$\tilde{G}_2 = \tilde{G}_2^* + (\tilde{G}_{-3}^* + \tilde{G}_7^* + \tilde{G}_{-8}^* + \tilde{G}_{12}^*). \tag{3.40}$$

Note how all perturbing coefficients here now have indices with absolute values greater than 2. In general the magnitudes of the perturbing coefficients will be smaller than $|\tilde{G}_2^*|$.

3.2.2 Approximate Galerkin Harmonic Balance Results

In this section we compare the performance of the oversampled pseudospectral HB (i.e. approximate Galerkin harmonic balance, or AGHB) with the pseudospectral HB formulation (i.e. PSHB) on the two nonlinear circuits (the diode rectifier and the DC-DC converter) used in the time-domain preconditioning experiments. A right preconditioned matrix-implicit GMRES linear solver was used in both formulations, with BDF-2 as a preconditioner. The initial Newton guess was a shooting-Newton solution computed with loose tolerance.

The HB methods were implemented in Mica, Motorola's SPICE-like circuit simulator, in the computer language C [55]. All computer runs were done on Sun Ultra-2 350MHz UNIX workstations.

Figures 3.5 and 3.6 show the L_∞ norm of the frequency-domain pointwise error ε_f in a solution waveform for the two circuits simulated with PSHB and AGHB with increasing oversampling σ. The frequency-domain pointwise error ε_f is computed as:

$$\varepsilon_f(kf) = |V_k^* - V_k| \tag{3.41}$$

where V_k^* is the k-th Fourier coefficient of the exact solution, and V_k is the k-th Fourier coefficient of the solution computed by PSHB or AGHB. Since no exact solution for these circuits exists, a

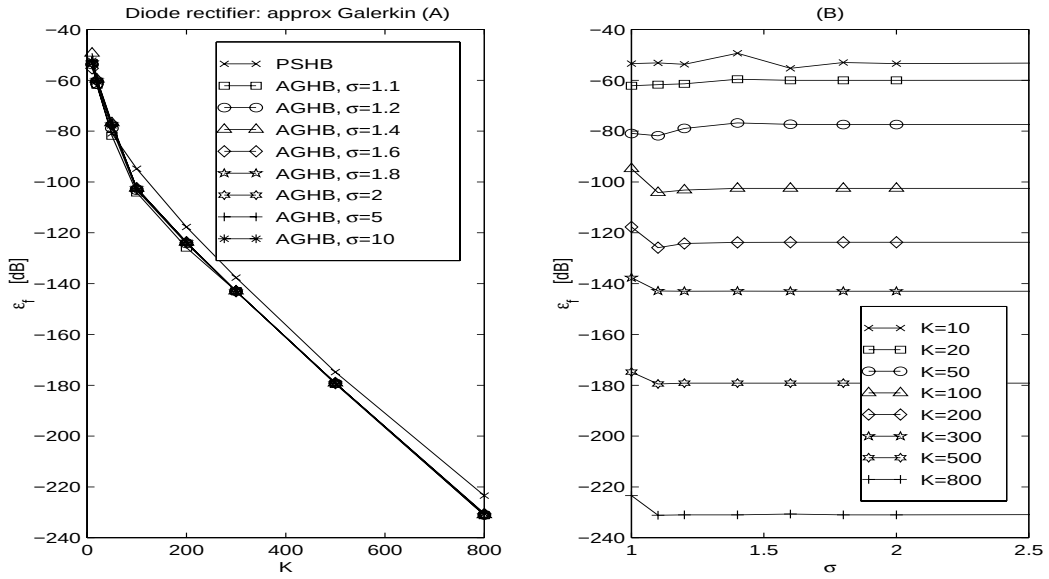

Figure 3.5: Diode Rectifier, AGHB L_∞ norm of the frequency-domain pointwise error in i_{VIN} versus: (A) number of harmonics K; (B) oversampling factor σ.

PSHB solution computed with a very large number of harmonics ($K = 5000$) was used for this purpose. Note that the points at $\sigma = 1$ in the plots correspond to the PSHB solutions.

While the Galerkin formulation is theoretically better than the collocation (pseudospectral) formulation, the plots in Figures 3.5 and 3.6 show only a small improvement in accuracy when AGHB is used: up to 7.5dB for the i_{VIN} waveform in the diode rectifier, and up to 12dB for the DC-DC converter's v_{COIL}. The reduction in the error initially increases with an increasing number of harmonics K, but saturates after reaching its peak value. The plots show that only minor 10% to 20% oversampling is needed to reach the peak error reduction at each K. This is a good outcome since the computational cost of AGHB is limited by the length of the FFTs, i.e. its complexity is $O(NM_\sigma \log M_\sigma)$. If a larger value of σ was required to get the error reduction, the benefits of AGHB would not have been worth the extra cost.

Note that if a direct factorization is used for the linear problems in both AGHB and PSHB, the oversampling in AGHB does not introduce an additional computational cost as the complexity is entirely dominated by the expense of the factorization.

Since AGHB is more accurate than the PSHB, to ensure a fair comparison of the two formulations we compare total CPU times needed for the two formulations to reach identical solution accuracy. Figure 3.7 shows these times for the DC-DC Converter, with AGHB using 20% oversampling ($\sigma = 1.2$). The plot indicates that AGHB is on average 1.5 times faster than the PSHB in computing the solution to same accuracy. The efficiency savings come primarily from the fact

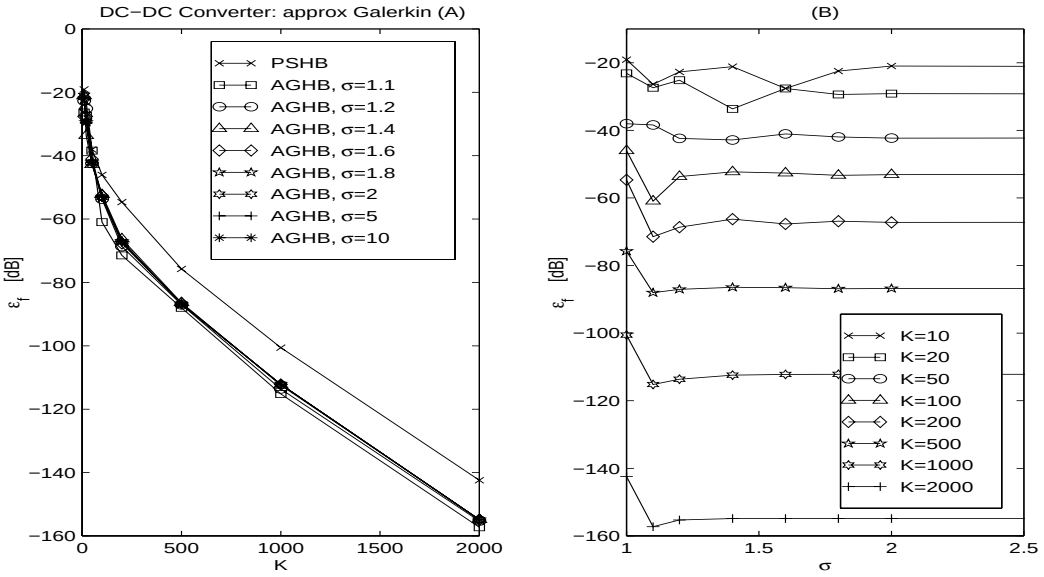

Figure 3.6: DC-DC Converter, AGHB L_∞ norm of the frequency-domain pointwise error in v_{COIL} versus: (A) number of harmonics K; (B) oversampling factor σ.

that AGHB needs to use fewer harmonics than PSHB to reduce the error to the same amount. This reduction in the size of the problem easily offsets the cost increase due to the use of oversampling transforms in AGHB.

In order to examine the possible effects of the AGHB to the GMRES linear solver and the Newton nonlinear solver, Figures 3.8 and 3.9 plot the average GMRES iterations per Newton step (computed as the ratio of the total GMRES iterations I_L to the number of Newton iterations), as well as the Newton iterations as functions of the oversampling factor σ. The points at $\sigma = 1$ correspond to PSHB. While the AGHB formulation does not seem to have any effect on the Newton solver, as σ is increased, the average number of GMRES iterations drops by a third. The Galerkin formulation, therefore, generates a better conditioned linear system. Unfortunately, we are unable to take advantage of this property of AGHB in practice since the increased cost of AGHB at larger σ completely overwhelms the savings generated by the reduction in the GMRES iterations.

We can conclude the approximate Galerkin formulation of the HB method does indeed bring some improvements to the standard pseudospectral HB method. The achieved increase in accuracy is small, and does not improve with increasing number of harmonics K (as predicted by the theory, the Galerkin and collocation spectral methods have errors that will decay at the same rate in the asymptotic limit [42]). Nevertheless, given that only a small 20% oversampling in AGHB yieds the peak increases in accuracy, the approach is cost effective for matrix-implicit HB implementations. The simulated examples show that the maximal observed 12dB improvement in accuracy

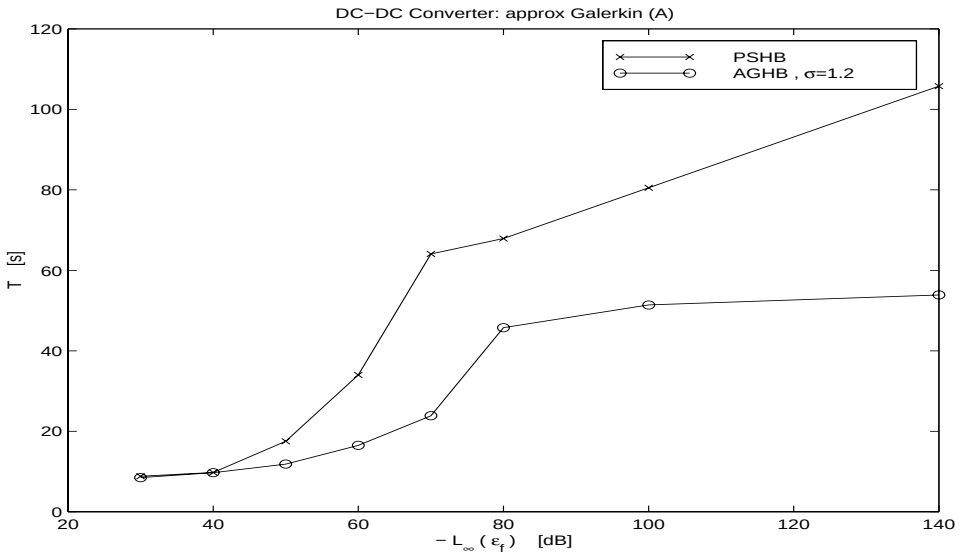

Figure 3.7: DC-DC Converter, total CPU times for PSHB and AGHB to reach a specific solution accuracy in i_{VIN}.

is equivalent to a simulation run time speedup of 1.5 comparing AGHB and PSHB runs reaching identical solution error.

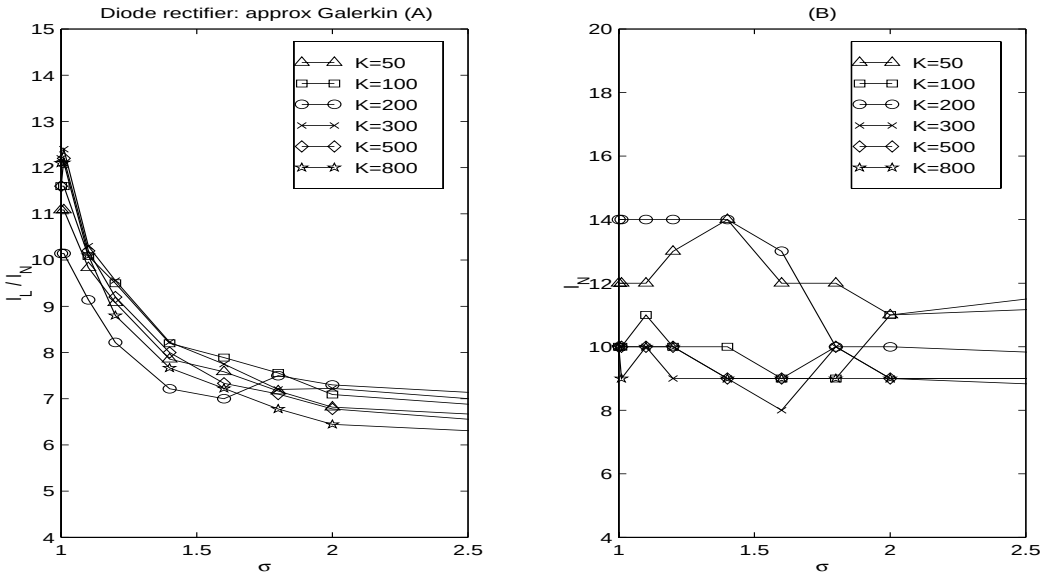

Figure 3.8: Diode Rectifier, AGHB: (A) average GMRES iterations; (B) Newton iterations, both versus σ.

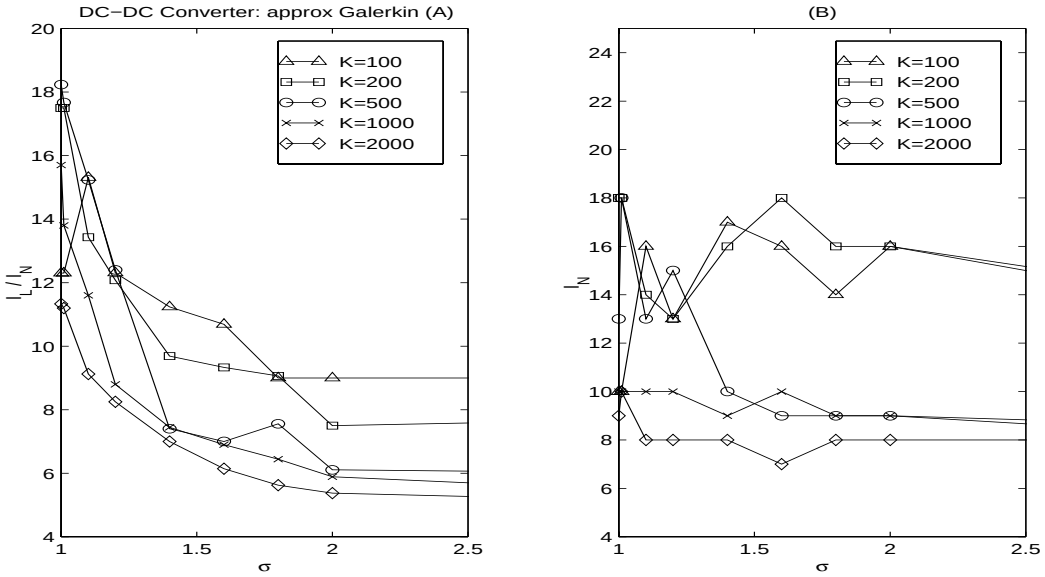

Figure 3.9: DC-DC Converter, AGHB: (A) average GMRES iterations; (B) Newton iterations, both versus σ.

3.3 Selection of Iterative Linear Solver

There are many other iterative linear solver methods besides GMRES, and many more are being developed. It is therefore worthwhile to examine what are the relative advantages and disadvantages of GMRES versus some other iterative methods that can be used in implementations of harmonic balance. One such alternate method is the Quasi Minimum Residual (QMR) algorithm [54].

The GMRES method holds the theoretical advantage over QMR in achieving the largest residual reduction in each iteration. In addition, GMRES requires only one matrix-vector product computation, versus two for the QMR. The Krylov subspace formed by the GMRES method is orthogonal, while QMR uses a bi-orthogonal Krylov subspace. GMRES is easier to code and is more robust in practice.

Unfortunately, GMRES requires storage of the entire Krylov subspace. This can become prohibitive for large problems for which the preconditioning is not effective and GMRES takes many iterations to reach the desired accuracy in solving the linear system. In addition, GMRES performs full back-orthogonalization, and therefore the number of inner products grows with the iteration number.

In comparison, QMR only requires storage of the last two vectors in the Krylov subspace since it only back-orthogonalizes for two steps. Note however that if the matrix-vector product is computationally expensive (as it is in the harmonic balance case due to the FFTs and the preconditioner), the lower back-orthogonalization cost of the QMR is irrelevant.

A transpose-free variant of QMR, TFQMR [54] requires only one matrix-vector multiplication, and is therefore more efficient than QMR. Similarly, there are variants of the GMRES algorithm that address the memory problems with this linear solver. The restarted GMRES algorithm restarts the GMRES algorithm after some maximum number of iterations. The quasi GMRES variants (QGMRES and DQGMRES) [54] use a truncated back-orthogonalization.

Nevertheless, given its proven reliability record in practice, we recommend GMRES as a default linear solver for harmonic balance. QMR, TFQMR, and the quasi variants of GMRES are not as robust, and are therefore only prefered in situations when ineffective preconditioning in GMRES in a large problem results in large number of iterations and memory storage issues. It is therefore recommended to have at least one of the QMR methods (preferably TFQMR) as a second choice in the implementation of harmonic balance.

Chapter 4

Time-Mapped Harmonic Balance

One of the main advantages of the shooting-Newton method for steady-state analysis of circuits is that it is a time-domain method which can select time-points based on local error estimation. Shooting-Newton methods can, therefore, easily handle circuits whose solution waveforms undergo rapid transitions, such as many highly nonlinear circuits, by increasing the time-point grid resolution to resolve the sharp waveform features.

The main advantage of the harmonic balance (HB) method is its spectral accuracy. The solution waveforms converge exponentially fast with increasing harmonics in the harmonic balance method, in contrast to the limited order polynomial convergence for the shooting-Newton and other time-domain methods. Unfortunately, the effective time-points used by the harmonic balance method are uniformly spaced, and this forces the method to use a large number of harmonics when the circuit solution contains very rapid transitions.

In this Chapter we introduce the Time-Mapped Harmonic Balance (TMHB) method. The main idea behind this new method is to utilize a non-uniform time-point grid with increased resolution in the fast varying regions of the solution waveforms, while retaining all of the advantages of the state-of-the-art preconditioned matrix-implicit Krylov-subspace harmonic balance method (referred to in the remaining text as the standard HB method). Since the non-uniform time-point grid is better adapted to fast-varying solution waveforms than a uniform grid with same number of time-points, one would hope and expect better accuracy in the computed solutions from the TMHB method compared to the standard HB method.

A non-uniform collocation grid in HB is certainly possible if the time-frequency conversions are done by the Almost Periodic Fourier Transform (APFT) [4]. Recall, however, that the computational complexity of the standard HB method is limited by the complexity of the time-frequency conversion. By replacing the FFT with the APFT the run-times would increase from $O(NM\log M)$

to $O(NM^2)$, which would definitely be an unacceptable outcome.

In the TMHB method, we keep the FFT and yet are able to use a non-uniform grid in the TMHB by mapping (transforming) the circuit problem and solving it in a new, pseudo-time domain. In this new domain, the sharp features in circuit's waveforms flatten out to certain extent, and the rapid transitions are less rapid. We are thus able to compute the solutions to a higher accuracy using a uniform time-point grid in this pseudo-time domain. As we know how this uniform grid is related to the non-uniform grid in real time, the solution is then transformed back into the real time domain.

At the core of the TMHB method is the non-uniform grid selection strategy. The success of the TMHB transformation of the circuit problem is critically dependent on how well these non-uniform time-points are placed. The grid selection strategy is a topic of Chapter 5.

In this Chapter we derive the Time-Mapped Harmonic Balance algorithm, give a Krylov-subspace based solution technique and describe the post-processing procedure used to obtain the actual Fourier coefficients from the TMHB solution. We also describe the procedure used to construct the time-map function which relates the non-uniform grid to a uniform pseudo-time grid. We study the error convergence properties of the TMHB method, and demonstrate these properties on analytic waveforms.

In [68] the authors use coefficient smoothing techniques for fast varying solutions in fluid dynamics. Spectral methods using mapping (also known as transformation, change-of-coordinate, or pseudo) techniques have been applied to solving explicit scalar (or at best small systems of) first or second order ODEs and PDEs in a number of references, mostly dealing with problems in fluid dynamics [43, 70, 71, 76, 77, 81, 82, 83].

4.1 Formulation

As we did in formulating the standard HB method in Chapter 2, we start by considering a circuit whose dynamics is described with N nonlinear differential equations:

$$\dot{q}(v(t)) + i(v(t)) + u(t) = 0 \tag{4.1}$$

where $v(t) \in R^N$ is the vector of node voltages, $q(v(t)) \in R^N$ the vector of node charges (or fluxes), $i(v(t)) \in R^N$ the vector of resistive node currents, and $u(t) \in R^N$ the vector of input sources.

Let the circuit be driven by a single periodic excitation input source with period T. Finding the periodic steady-state solution of this circuit consists of computing the N steady-state waveforms $v(t)$ on the solution domain $t \in [0, T]$. The periodic steady-state solution of (4.1) satisfies the

two-point constraint:

$$v(T) = v(0). \tag{4.2}$$

In the standard HB method, the solution waveforms are approximated with truncated Fourier series:

$$v(t) = \sum_{k=-K}^{k=K} V_k e^{j2\pi kft} \tag{4.3}$$

with $f = \frac{1}{T}$ the periodic excitation input source frequency, and K the number of harmonics considered in the truncation. The method solves for the Fourier coefficients V_k. The approximation (4.3), in conjunction with the N circuit equations (4.1), results in the residual function:

$$f(V,t) \equiv \sum_{k=-K}^{K} j2\pi kfQ_k e^{j2\pi kft} + i\left(\sum_{k=-K}^{K} V_k e^{j2\pi kft}\right) + u(t) \tag{4.4}$$

where Q_k are the Fourier coefficients of $q(v(t))$ with $v(t)$ the truncated Fourier series approximations of the solution waveforms, i.e. $q\left(\sum_{k=-K}^{K} V_k e^{j2\pi kft}\right) = \sum_{k=-K}^{K} Q_k e^{j2\pi kft}$.

The residual function (4.4) is to be minimized on $[0,T]$.

In order to enhance the clarity of the presentation, we now introduce the following notation for grids of time-points.

Definition 1 The set of M time-points $\rho^M = \{t_1, t_2, ..., t_M\}$ is a grid if (1) $\forall t_m \in [0,T)$; and (2) $0 = t_1 < t_2 < ... < t_m < T$ (strict monotonicity property). The grid spacing h_m is $h_m = t_{m+1} - t_m$ with $t_{M+1} = T$, and the grid size is M.

Definition 2 The grid ρ^M is uniform if all its spacings satisfy $h_m = const.$, i.e. $t_m = \frac{(m-1)T}{M}$. Uniform grids are denoted ρ_u.

In the standard (pseudospectral) HB method, the minimization of (4.4) is carried out by enforcing $f(V,t_m) = 0$ on the uniform grid $t_m \in \rho_u^M$ of collocation (interpolation) points where $M = 2K+1$.

In contrast, the Time-Mapped Harmonic Balance (TMHB) utilizes a non-uniform grid of time-points ρ^M. The non-uniform grid has increased resolution in the regions of the solution waveforms with rapid transitions. The non-uniform grid resolves the sharp waveform features in order to increase the solution accuracy of the TMHB method.

We now introduce the notion of pseudo time $\hat{t}$ and denote the grids of pseudo time-points with $\hat{\rho}$. The pseudo time $\hat{t}$ coordinate is related to real time via the time-map function λ such that

$$t = \lambda(\hat{t}) \tag{4.5}$$

The time-map function maps the uniform grid of pseudo time-points $\hat{\rho}_u^M$ into the non-uniform grid of real time-points ρ^M such that $\lambda : \hat{t}_m \mapsto t_m$ where $\hat{t}_m \in \hat{\rho}_u^M$, $t_m \in \rho^M$, $\lambda(0) = 0$, and $\lambda(T) = T$.

The time-map function $\lambda(\cdot)$ is constructed from the non-uniform grid of time-points ρ^S which is generated by the grid selection algorithm. This non-uniform grid is spectrally interpolated to yield $\lambda(\cdot)$. The details of this construction are given in Section 4.4.

To derive the Time-Mapped Harmonic Balance (TMHB) method, consider that

$$\frac{d}{dt} = \frac{1}{\lambda'(\hat{t})}\frac{d}{d\hat{t}} \quad . \tag{4.6}$$

Replacing the time-derivative in (4.1) with (4.6) yields

$$\frac{1}{\lambda'(\hat{t})}\frac{d}{d\hat{t}}q(v(\lambda(\hat{t})) + i(v(\lambda(\hat{t}))) + u(\lambda(\hat{t})) = 0 \tag{4.7}$$

and the two-point constraint (4.2) becomes

$$v(\lambda(T)) = v(\lambda(0)). \tag{4.8}$$

The solution waveforms in TMHB are approximated with truncated pseudo Fourier series:

$$v(t) = v(\lambda(\hat{t})) = \sum_{k=-K}^{k=K} \hat{V}_k e^{j2\pi k f \hat{t}} \tag{4.9}$$

where $\hat{V}_k$ are the pseudo Fourier coefficients of the solution waveforms. Equations (4.7) and (4.9) yield the residual function

$$\hat{f}(\hat{V},\hat{t}) \equiv \frac{1}{\lambda'(\hat{t})}\sum_{k=-K}^{K} j2\pi k f \hat{Q}_k e^{j2\pi k f \hat{t}} + i\left(\sum_{k=-K}^{K} \hat{V}_k e^{j2\pi k f \hat{t}}\right) + u(\hat{t}) \tag{4.10}$$

where $\hat{Q}_k$ are the pseudo Fourier coefficients of $q(v(\lambda(\hat{t})))$ with $v(\lambda(\hat{t}))$ the truncated pseudo Fourier approximations of the solution waveforms, i.e. $q\left(\sum_{k=-K}^{K} \hat{V}_k e^{j2\pi k f \hat{t}}\right) = \sum_{k=-K}^{K} \hat{Q}_k e^{j2\pi k f \hat{t}}$.

The residual function (4.10) is to be minimized on $[0,T]$. The minimization is carried out by a collocation method, enforcing $\hat{f}(\hat{V},\hat{t}) = 0$ on the uniform pseudo grid of collocation points $\hat{\rho}_u^M$.

The non-uniform grid in real time in effect "stretches" out those regions of the solution waveforms with sharp features. As a result, the TMHB solution $v(t)$ in real time is the smoother waveform $v(\lambda(\hat{t}))$ when viewed in pseudo time, as illustrated in Figure 4.1. Since the waveform is smoother in pseudo time, its features are more easily resolved with an M-point uniform pseudo grid, compared to resolving the original fast varying waveform in real time with an M-point uniform real time grid in the HB method. Thus one expects better accuracy from the solution of the TMHB method.

Another way of seeing why TMHB should be more accurate than standard HB is to think of how the set of pseudo Fourier series basis functions interpolates the solution waveforms. The

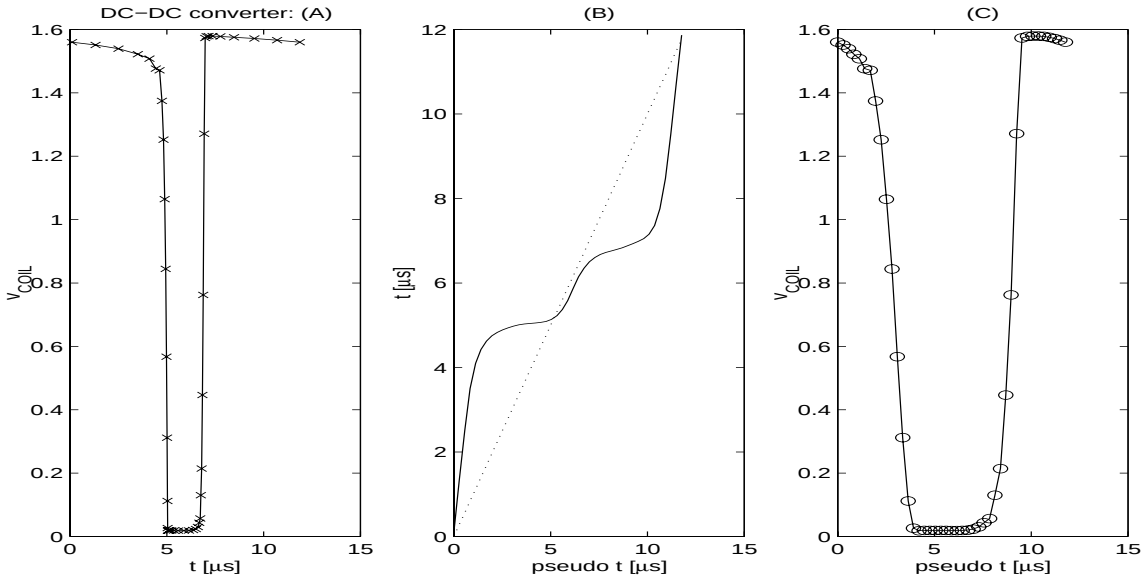

Figure 4.1: The smoothing effect of the non-uniform grid of TMHB: (A) v_{COIL} in real time (non-uniform ρ^M grid); (B) time-map function $t = \lambda(\hat{t})$; (C) v_{COIL} in pseudo time (uniform $\hat{\rho}_u^M$ grid).

pseudo Fourier series basis in real time is the set of $e^{j2\pi f k \lambda^{-1}(t)}$ oscillatory functions. These basis functions smoothly increase their frequency in the regions where the non-uniform grid is dense. The more rapidly changing pseudo basis function is thus able to capture the rapid transitions in the solution waveform more easily. Two basis functions, one from standard, and one from a pseudo Fourier series waveform approximation are given in Figure 4.2. The non-uniform grid was such that it had a peak increase in density in the middle of the domain at $t = 10ms$ (the time map function (4.43) was used) to accommodate rapid transitions occurring in the middle of the interval.

In addition, the magnitude of the pseudo series coefficient for the highest frequency (largest k) pseudo basis functions need not be as large as the corresponding magnitudes in the standard series basis Functions. The high frequency basis coefficient magnitudes in the standard series need to be larger in order to increase the slope of the basis function and capture the rapid transitions. This causes a pronounced Gibbs effect and a not-so-well approximating solution interpolant. In contrast, due to smaller high frequency basis magnitudes, the pseudo basis functions cause much smaller Gibbs oscillations and are better building blocks in interpolating the fast varying solution waveform.

Equation (4.10) is now rewritten in the frequency domain yielding NM nonlinear algebraic equations

$$\hat{F}(\hat{V}) \equiv \Gamma\Lambda\Gamma^{-1}\Omega\Gamma q(\Gamma^{-1}\hat{V}) + \Gamma i(\Gamma^{-1}\hat{V}) + \Gamma u = 0 \tag{4.11}$$

where $\hat{V}$ is the pseudo node-voltage spectrum $\hat{V} = [\hat{V}_{-K,1}, ..., \hat{V}_{-K,N}, \hat{V}_{-K+1,1}, ..., ..., \hat{V}_{K,N}]^T$ (a vec-

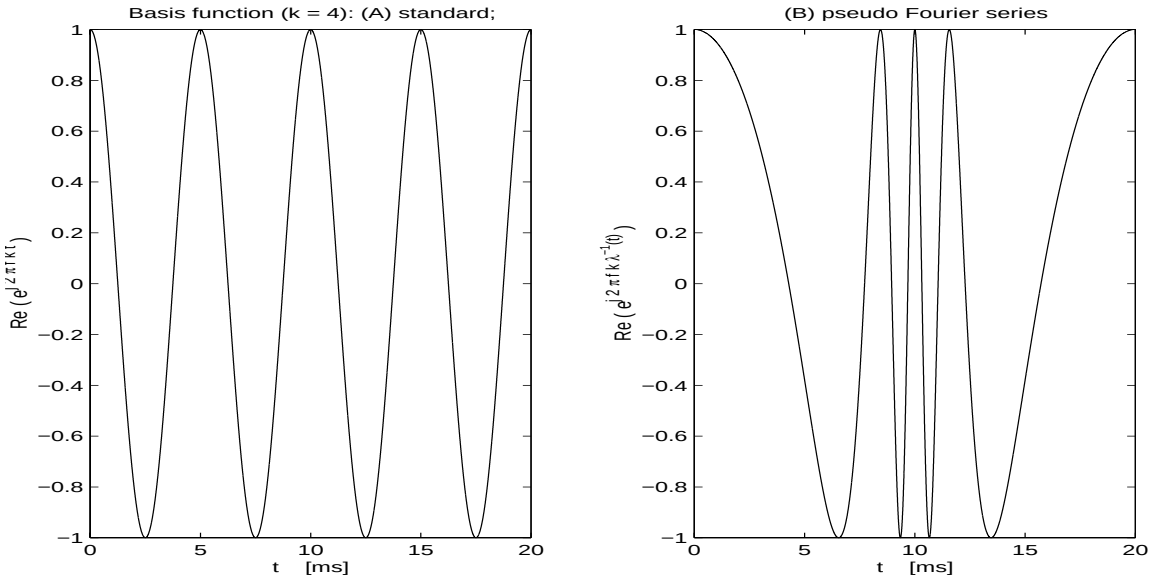

Figure 4.2: Basis functions, ($k = 4$), for the (A) standard, and (B) a pseudo Fourier series.

tor of pseudo Fourier coefficients), and Ω is the diagonal frequency-domain differentiation matrix

$$\Omega = \begin{bmatrix} j2\pi K f I_N & & & \\ & j2\pi(K-1) f I_N & & \\ & & \ddots & \ddots \\ & & & j2\pi(-K) f I_N \end{bmatrix}, \tag{4.12}$$

Λ is the diagonal matrix

$$\Lambda \equiv \begin{bmatrix} \frac{1}{\lambda'(\hat{t}_1)} I_N & & & \\ & \frac{1}{\lambda'(\hat{t}_2)} I_N & & \\ & & \ddots & \\ & & & \frac{1}{\lambda'(\hat{t}_M)} I_N \end{bmatrix} \tag{4.13}$$

and I_N is the identity matrix of size N.

The matrices Γ and Γ^{-1} are DFT matrices that perform the conversions from pseudo time to

frequency and vice-versa

$$v = \Gamma^{-1}\hat{V} = \begin{bmatrix} v(\lambda(\hat{t}_1)) \\ v(\lambda(\hat{t}_2)) \\ \vdots \\ v(\lambda(\hat{t}_M)) \end{bmatrix}, \tag{4.14}$$

$$\Gamma^{-1} = \begin{bmatrix} e^{j2\pi(-K)f\hat{t}_1}I_N & \dots & e^{j2\pi Kf\hat{t}_1}I_N \\ \vdots & & \vdots \\ e^{j2\pi(-K)f\hat{t}_M}I_N & \dots & e^{j2\pi Kf\hat{t}_M}I_N \end{bmatrix}$$

where each $v(\lambda(\hat{t})) \in R^N$ is a vector of node voltages. Since the pseudo time grid $\hat{\rho}_u^M$ is uniform, the DFT can be carried out in $O(NM\log M)$ operations using the FFT just as in the standard HB.

4.2 Matrix-Implicit Krylov-Subspace Solution Technique

The system of NM nonlinear algebraic equations (4.11) are solved for the pseudo Fourier coefficients $\hat{V}$ by applying Newton's method. The application results in the iteration

$$J^{(l)}\left(\hat{V}^{(l+1)} - \hat{V}^{(l)}\right) = -\hat{F}(\hat{V}^{(l)}) \tag{4.15}$$

where l is the Newton iteration index. The l-th Newton iteration Jacobian is

$$J^{(l)} = \Gamma\Lambda\Gamma^{-1}\Omega\Gamma C^{(l)}\Gamma^{-1} + \Gamma G^{(l)}\Gamma^{-1} \tag{4.16}$$

and is a $NM \times NM$ block-dense matrix.

The block-diagonal matrices C and G are

$$C \equiv \begin{bmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_M \end{bmatrix}, \quad G \equiv \begin{bmatrix} G_1 & & & \\ & G_2 & & \\ & & \ddots & \\ & & & G_M \end{bmatrix} \tag{4.17}$$

where the blocks $C_m = [C_m(r,s)]$ and $G_m = [G_m(r,s)]$ consist of the elements

$$C_m(r,s) = \frac{dq(v_r(\lambda(\hat{t}_m)))}{dv_s} = \frac{dq_r(v(t_m))}{dv_s}, \tag{4.18}$$

$$G_m(r,s) = \frac{di(v_r(\lambda(\hat{t}_m)))}{dv_s} = \frac{di_r(v(t_m))}{dv_s} \tag{4.19}$$

where r,s are the node indices, $r,s = 1,2,\ldots,N$. The matrices G and C remain block-diagonal as in the standard HB as $q(v)$ and $i(v)$ remain algebraic. The constituent sparse blocks G_m and C_m are the circuit conductance and capacitance matrices evaluated in real time on the non-uniform grid of real collocation time-points $t_m \in \rho^M$. The sparsity of these blocks depends on the circuit topology.

The grid selection algorithm will in general generate mildly non-uniform grids even for linear circuits. This is caused by the grid selection scheme which distributes the time-points following the rate of change in the solution waveforms, which is not constant for the pure sinusoidal signals in a linear circuit. The multiplication by $\Gamma\Lambda\Gamma^{-1}$ in the formation of the Jacobian prevents the decoupling of the Jacobian and the equations as in standard HB for linear circuits. The decoupling is possible only if Λ happens to be an identity matrix, meaning that $t = \lambda(\hat{t}) \equiv \hat{t}$, i.e. that the pseudo time non-uniform grid in TMHB is in fact uniform.

This should not be a cause for an alarm since TMHB is a method meant to be used on highly nonlinear circuits whose solutions undergo rapid transitions. For linear and even mildly nonlinear circuits driven by pure sinusoids, the uniform grid in the standard HB is effective enough, so that introducing a non-uniform grid with TMHB would, at best, yield only minor improvements in solution accuracy.

The Newton iteration (4.15) is a linear problem, and explicitly forming and factoring the block-dense TMHB Jacobian J would be very expensive, $O(NM^3)$. As in standard HB, a preconditioned iterative linear solver (GMRES) is used to reduce the complexity to $O(NM^2)$, with further reductions in complexity obtained by implicitly forming the GMRES matrix-vector product by sequential evaluation using FFTs, to $O(NM\log M)$. Therefore the complexity of TMHB is the same as the complexity of the state-of-the-art matrix-implicit Krylov-subspace standard HB.

The memory requirements for the TMHB method are also of the same complexity as the standard HB, growing linearly with $O(NM)$ due to the storage of the Krylov subspace vectors in the GMRES algorithm.

4.2.1 Speeding Up the Matrix-Vector Product

The matrix-vector product

$$Jp^k = \left(\Gamma\Lambda\Gamma^{-1}\Omega\Gamma C + \Gamma G\right)\Gamma^{-1}p^k \tag{4.20}$$

where p^k is the search direction in the k-th GMRES iteration, requires 5 FFTs. It is possible to reduce this number of FFTs to 3 by a simple algebraic reformulation of the system of algebraic equations (4.11). Multiplying (4.11) by $\Gamma\Lambda^{-1}\Gamma^{-1}$ results in the system:

$$\Omega\Gamma q(\Gamma^{-1}\hat{V}) + \Gamma\Lambda^{-1} i(\Gamma^{-1}\hat{V}) + \Gamma\Lambda^{-1}u = 0. \tag{4.21}$$

When Newton's method is applied to (4.21), the Jacobian at the l-th Newton iteration is:

$$J^{(l)} = \Omega\Gamma C^{(l)}\Gamma^{-1} + \Gamma\Lambda^{-1}G^{(l)}\Gamma^{-1} \tag{4.22}$$

where Λ^{-1} is computed trivially as Λ is a diagonal matrix. The k-th GMRES iteration matrix-vector product used to solve the linear subproblem at each Newton iteration is now

$$Jp^k = \left(\Omega\Gamma C + \Gamma\Lambda^{-1}G\right)\Gamma^{-1}p^k \tag{4.23}$$

and requires 3 FFTs. As the FFT limits the complexity of the TMHB, and the GMRES computation time dominates the total TMHB runtime, a substantial speedup is achieved by this 40% reduction in the number of FFTs in the matrix-vector product computation. Note that the matrix-vector product in the standard HB also requires 3 FFTs. The only (minor) overhead in TMHB matrix-vector product comes from the multiplication by the diagonal matrix Λ^{-1}.

Note that we could choose to reformulate only the GMRES linear problem, and not the system of equations (4.11), arriving at the same formulation for the matrix-vector product with 3 FFTs. In this case the right hand sides (RHS) for the outer Newton loop and the inner GMRES iterations will be different. This approach has shown to cause a drastic increase in both the number of Newton and GMRES iterations in practice, and is not recommended.

4.2.2 Linear Device Treatment

Linear elements (such as linear resistors, capacitors, and controlled voltage sources) are characterized by linear device functions $i(v)$ and $q(v)$. Therefore, in most circuit simulators these devices can be evaluated only once, rather than at each collocation point. Consider such an implementation of HB where the linear devices are treated separately. In effect, the steady-state circuit problem (4.1) can now be expanded with an additional term and written as:

$$\dot{q}(v(t)) + i(v(t)) + \int_{-\infty}^{t} y(t-\tau)v(\tau)d\tau + u(t) = 0 \tag{4.24}$$

where $q(v(t)) \in R^N$ and $i(v(t)) \in R^N$ are now, respectively, the vectors of node charges (or fluxes) and resistive currents from the nonlinear circuit devices, $y(t) \in R^N$ is the matrix-valued impulse response of the circuit with all the nonlinear devices removed, and $u(t) \in R^N$ the vector of input sources. If $y(t)$ is causal and has finite energy, then

$$\Gamma\int_{-\infty}^{t} y(t-\tau)v(\tau)d\tau = YV. \tag{4.25}$$

The TMHB system of NM nonlinear equations (4.11) is now expanded with this additional term

$$\hat{F}(\hat{V}) \equiv \Gamma\Lambda\Gamma^{-1}\Omega\Gamma q(\Gamma^{-1}\hat{V}) + \Gamma i(\Gamma^{-1}\hat{V}) + Y\hat{V} + \Gamma u = 0 \tag{4.26}$$

where $Y = Y_G + \Omega Y_C$ is a block-diagonal matrix, with Y_G and Y_C being its resistive and capacitive parts which are also block diagonal. Note that due to the linearity of the devices described by Y, all of the constituent sparse blocks Y_{Gm} and Y_{Cm} of Y_G and Y_C respectively, are the same, while Y's diagonal blocks differ as a result of a multiplication by Ω.

The matrix-vector product (4.20) is now

$$Jp^k = \left(\Gamma\Lambda\Gamma^{-1}\Omega\Gamma C + \Gamma G\right)\Gamma^{-1}p^k + Yp^k \tag{4.27}$$

and still requires 5 FFTs. The matrix-vector product (4.23) corresponding to the reformulated problem (4.21) is now

$$Jp^k = \left(\Omega\Gamma C + \Gamma\Lambda^{-1}G\right)\Gamma^{-1}p^k + \Gamma\Lambda^{-1}\Gamma^{-1}Yp^k \tag{4.28}$$

and requires 2 additional FFTs bringing the total to 5 FFTs.

There are several ways of getting around this problem. If the circuit contains very few linear elements, we may skip the sequence of operations $\Gamma\Lambda^{-1}\Gamma^{-1}$ in front of Y, removing the 2 costly additional FFTs. In practice this has shown to work well with only minor increases in the number of GMRES iterations if the only linear elements in the circuit are a few controlled sources.

A better approach is to treat all linear elements as nonlinear elements and retain the original reformulation of the matrix-vector product (4.23) with 3 FFTs. The overhead of this approach is the repeated evaluation of linear elements such as resistors and capacitors at each collocation point. This overhead is minor, however, when one considers that: (1) the cumulative device evaluation time in HB or TMHB is only a small fraction of the cumulative time spent in solving the linear problem with the GMRES iterations; (2) the evaluation of a resistor, capacitor, or a controlled source takes a fraction of a time needed to evaluate a modern model for a nonlinear device such as a bipolar or MOS transistor; and (3) modern communication circuits consist of large number of transistors comparable to the number of linear devices.

One final point is that the linear devices found on a chip such as resistors or capacitors, are in fact nonlinear as they are nonlinear functions of temperature, which in turn is a nonlinear function of the on-chip voltages and currents. It is therefore important to treat these devices as nonlinear via an implementation of an appropriate device model for accurate circuit simulation.

4.2.3 Preconditioning

For the reformulated TMHB, the diagonal blocks of the TMHB Jacobian (4.22)

$$P_f = \begin{bmatrix} j2\pi(-K)f\bar{C}_0 + \bar{G}_{0,\lambda} & & & \\ & j2\pi(-K+1)f\bar{C}_0 + \bar{G}_{0,\lambda} & & \\ & & \ddots & \\ & & & j2\pi K f\bar{C}_0 + \bar{G}_{0,\lambda} \end{bmatrix} \tag{4.29}$$

where the blocks $\bar{C}_0 = [\bar{C}_0(r,s)]$ and $\bar{G}_0 = [\bar{G}_0(r,s)]$ contain the elements

$$\bar{C}_0(r,s) = \frac{1}{M}\sum_{m=1}^{M} C_m(r,s), \tag{4.30}$$

$$\bar{G}_{0,\lambda}(r,s) = \frac{1}{M}\sum_{m=1}^{M} \lambda'(\hat{t}_m) G_m(r,s) \tag{4.31}$$

with $r,s = 1,2,\ldots,N$ the node indices, are the standard preconditioner for the GMRES solver in the TMHB method, just as the diagonal blocks of the HB Jacobian are the default preconditioner in the standard HB method [11, 15]. The diagonal blocks of the TMHB Jacobian consist of the DC Fourier coefficients of the matrices C and $\Lambda^{-1}G$. These DC coefficients can be quickly computed by averaging the entries of C and $\Lambda^{-1}G$ corresponding to each circuit node. The diagonal block structure of the preconditioner allow application of fast direct block factorization algorithms.

This standard frequency-domain preconditioner works well in most circuit examples. The grid selection strategy in the TMHB requires a computation of a solution guess, which can be also used as a good starting initial guess for the Newton iterations of the TMHB. The usage of this initial guess drastically reduces the total number of Newton iterations, and also has a reduction effect on the number of GMRES iterations as well. As a consequence, the standard preconditioner is quite adequate in most circuit examples. For some strongly nonlinear circuits, the time-domain BDF preconditioners from Chapter 3 are more effective than the standard preconditioner in TMHB as they are in the HB method.

Right preconditioning is used in the TMHB method. With right preconditioning the linear problem $J\Delta\hat{V} = -F$ becomes $JP^{-1}P\Delta\hat{V} = -F$ where P is the preconditioner. By letting $P\Delta\hat{V} = \hat{U}$, the GMRES algorithm first solves the linear problem $JP^{-1}\hat{U} = -F$, and then computes $\Delta\hat{V} = P^{-1}\hat{U}$.

The right style of preconditioning does not change the RHS of the linear problem in GMRES, and this RHS is the same as the RHS used in the outer Newton loop. As it was noted earlier in this section, when these two RHS sides are different, costly increases in both the number of Newton and GMRES iterations are noted in practice.

4.3 Computing the Real Time Fourier Coefficients

The TMHB method solves for the pseudo Fourier coefficients $\hat{V}$ of the solution waveforms. These pseudo Fourier coefficients can be related to the real Fourier coefficients V. First note that if time-domain waveforms are desired, due to (4.9), an inverse FFT readily yields the voltage waveforms at the non-uniform grid ρ^M of real time-points:

$$v(t) \equiv v(\lambda(\hat{t})) = \Gamma^{-1}\hat{V}. \tag{4.32}$$

To compute the actual Fourier coefficients V, we use the following "unmap" procedure. We first introduce a non-uniform oversampled grid in pseudo time $\hat{\rho}^{M_\sigma}$. This grid is chosen such that the time-map function $\lambda(\cdot)$ maps it into a uniform oversampled grid in real time $\rho_u^{M_\sigma}$, i.e. $\lambda : \hat{t}_m \mapsto t_m$ where $\hat{t}_m \in \hat{\rho}^{M_\sigma}$ and $t_m \in \rho_u^{M_\sigma}$. The number of collocation points in the oversampled grids are $M_\sigma = \sigma M$ where $\sigma > 1$ is the oversampling factor.

Since from (4.5) $\hat{t} = \lambda^{-1}(t)$, (4.9) can be rewritten as

$$v(t) = \sum_{k=-K}^{k=K} \hat{V}_k e^{j2\pi k f \lambda^{-1}(t)}. \tag{4.33}$$

This summation is evaluated at real times $t_m \in \rho_u^{M_\sigma}$ to give the solution waveforms $v(t_m)$ at the oversampled uniform grid in real time. Note that (4.33) cannot be carried out by an inverse FFT since the pseudo time-points $\lambda^{-1}(t_m) \in \hat{\rho}^{M_\sigma}$ form a non-uniform grid. Due to this non-uniform grid, the complex exponentials in the summation are not equally spaced along the unit circle in the complex plane as required by the FFT.

Finally, since the $v(t)$'s values are now known on a (oversampled) uniform grid in real time, we can use the FFT to compute the real Fourier coefficients V

$$V = \Gamma v(t) \tag{4.34}$$

where $t \in \rho_u^{M_\sigma}$.

Note that this procedure actually yields $M_\sigma = 2K_\sigma + 1$ Fourier coefficients, which is more than the M Fourier coefficients expected. The additional Fourier coefficients represent the higher frequencies $2\pi k f$ where $K < k \leq K_\sigma$ captured by the non-uniform grid in TMHB. These coefficients are shown to match the Fourier coefficients of the "exact" solution quite well (see Figure 4.3). Without oversampling, these coefficients would be zero and the additional accuracy obtained by the TMHB method would be lost.

In effect the M pseudo Fourier coefficients $\hat{V}$ "pack" high frequency information content. In order to "unpack" this content, we must carry the described "unmap" procedure utilizing the oversampling frequencies between K and K_σ.

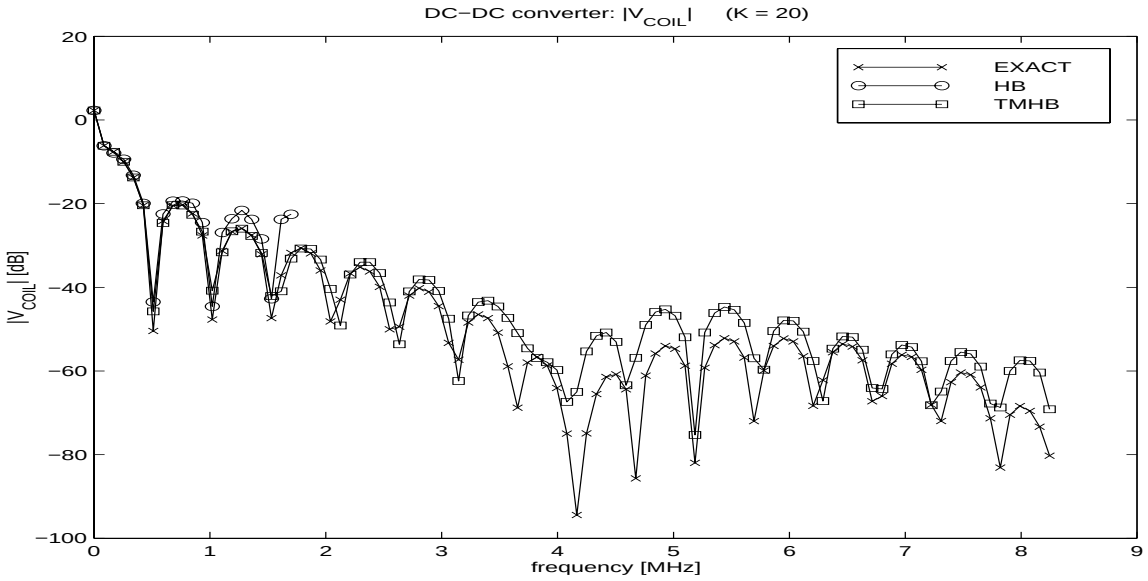

Figure 4.3: TMHB matching of high-frequency coefficients.

The rate of oversampling is determined by the Nyquist frequency $f_\sigma = 1/(2h_{min})$ corresponding to the smallest spacing $h_{min} = \min(h_m)$ for $\forall m$ in the non-uniform grid in real time ρ^M, where from

$$K_\sigma = \left\lceil \frac{f_\sigma}{f} \right\rceil = \left\lceil \frac{1}{2h_{min} f} \right\rceil \tag{4.35}$$

and $\sigma = K_\sigma / K$.

The unmap procedure described above is in essence an oversampled pseudo Fourier interpolation of the solution waveforms $v(t)$. This interpolation uses the discrete waveform values of $v(t)$ at the non-uniform grid in real time ρ^M to generate the discrete values $v(t_m)$ at the oversampled uniform grid in real time $\rho_u^{M_\sigma} \ni t_m$. It is crucial to use a spectrally accurate oversampled interpolation in order to preserve the accuracy of the solution. Local interpolation schemes (linear or quadratic) are not suitable for this task as they introduce errors that are larger than the errors from the Fourier approximation of the solution.

4.3.1 Complexity of the Unmap Procedure

The complexity of the post-processing unmap procedure is $O(\sigma M^2)$ per waveform since the summation (4.33) is a multiplication of $\hat{V}$ by a dense matrix of size $\sigma M \times M$. The unmap procedure is typically applied on only those few waveforms whose Fourier coefficients are wanted. In comparison, the complexity of the TMHB method is $O(NM \log M)$, i.e. limited by the FFT. This means that the worst cases when the computation time for the unmap procedure can be a significant portion of

the computation time for the TMHB are if: (1) the circuit is extremely non-linear (requiring vast variations in the grid resolution translating into large σ and large M); and either if (2a) the circuit is small (N small) or (2b) the unmap procedure is carried on all N waveforms for a circuit of arbitrary size.

In practice, however, the situation is not bad at all. Recall that the actual computation times are roughly equal to some computation complexity constant times the order of complexity. The constant for the TMHB method applied to highly nonlinear circuits is much larger than the constant for the unmap procedure since it is proportional to the total number of GMRES iterations performed. This number in turn is larger for very nonlinear circuits since for these circuits the standard preconditioner made up from diagonal blocks of the Jacobian J is not terribly effective.

Furthermore, stability considerations in the grid selection algorithm limit the rate of change of the TMHB grid resolution (i.e. the ratios of neighboring grid spacings should be kept bounded). This in turn limits the oversampling factor σ. Finally, for large circuits, computer memory limits set a bound on M.

4.4 Construction of the Time-Map Function

The time-map function λ maps the uniform grid in pseudo time $\hat{\rho}_u^M$ to the non-uniform grid ρ^M in real time. In addition, in the unmap procedure used to compute the real Fourier coefficients, the time-map function λ maps the non-uniform oversampled grid in pseudo time $\hat{\rho}^{M_\sigma}$ into the uniform oversampled grid $\rho_u^{M_\sigma}$ in real time.

The first step in constructing the time-map function λ is to determine a set of S non-uniform real time time-points. The success of the TMHB method is crucially dependent on this time-point selection, and several different selection strategies are presented in Chapter 5. The strategies requires an initial guess for the solution waveforms. In particular, an approximate solution is computed using a shooting-Newton method [4] with a low-order time integration scheme. The S non-uniform time-points for the TMHB method are then selected based on balancing two criteria: using small time-steps in the fast-varying regions of the approximate solution waveforms, and insuring that the time-steps do not change too rapidly. Although using a shooting-Newton method to compute the approximate solution is expensive, the cost is kept low by loosening the convergence tolerance. In addition, this shooting-Newton solution is useful as an initial guess for the TMHB.

Given the S non-uniform real time time-points, we now describe the construction of the time-map function $t = \lambda(\hat{t})$. In order to preserve the spectral accuracy of the TMHB method, the time-map function must be smooth (or more precisely, at least as many times continuously differentiable

as the functions describing the circuit element's device models). Also, we must be able to compute its first derivative with spectral accuracy or better as it is used in the TMHB formulation (4.10). Furthermore, to ensure the strict monotonicity of the non-uniform grid of real time-points, the time-map function must be strictly monotonic, i.e. $\lambda'(\hat{t}) > 0$ for all $\hat{t} \in [0, T]$. Finally, for unmap purposes, we also need to be able to compute $\lambda^{-1}(t)$.

We first represent $\lambda(\hat{t})$ as a sum of a linear part and a T-periodic part $\lambda_\phi(\hat{t})$

$$t = \lambda(\hat{t}) = \hat{t} + \lambda_\phi(\hat{t}). \tag{4.36}$$

This split of the time-map function is illustrated in Figure 4.4.

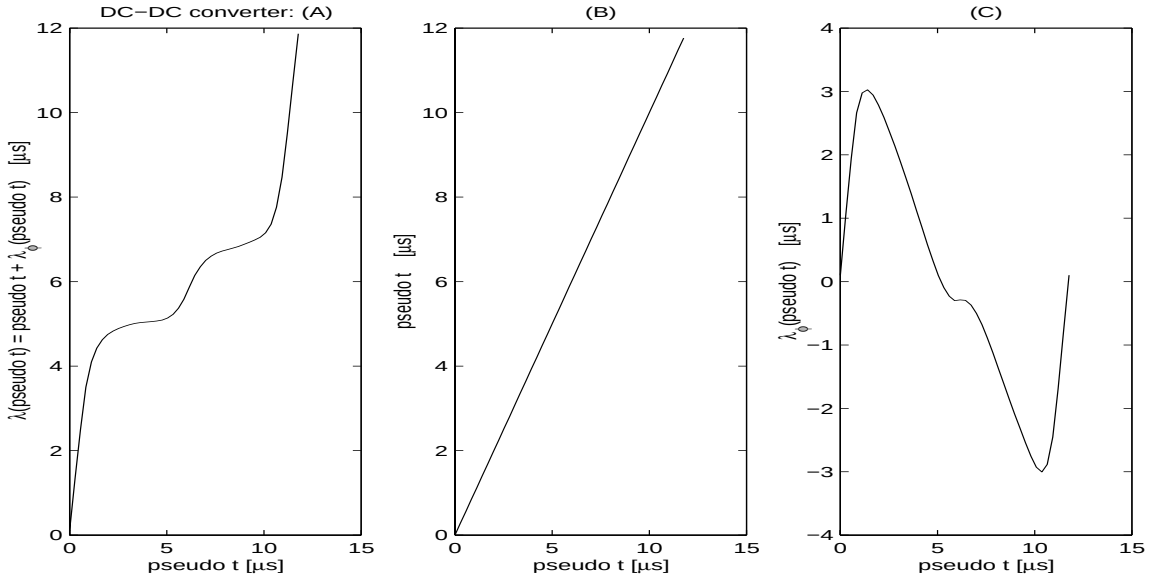

Figure 4.4: Split of the time-map function: (A) time-map function $\lambda(\hat{t})$; (B) linear part $\hat{t}$; (C) T-periodic part $\lambda_\phi(\hat{t})$.

The periodic part $\lambda_\phi(\hat{t})$ is chosen to be a Fourier polynomial interpolant $\phi(\hat{t})$ of order S such that the interpolatory condition

$$t_s = \hat{t}_s + \phi(\hat{t}_s) \tag{4.37}$$

is exactly satisfied at the points $(\hat{t}_s, t_s)$ where $t_s \in \rho^S$ are the S non-uniform real-time time-points, and $\hat{t}_s \in \hat{\rho}_u^S$ are S uniform pseudo time-points. The interpolant $\phi(\hat{t})$ is the truncated Fourier series

$$\phi(\hat{t}) = \sum_{k=-J}^{J} \Phi_k e^{j2\pi k f \hat{t}} \tag{4.38}$$

where $2J+1=S$. The coefficients Φ_k can be computed with an inverse FFT of size S

$$\begin{bmatrix} \Phi_{-J} \\ \dots \\ \Phi_J \end{bmatrix} = \Gamma^{-1} \begin{bmatrix} t_1 - \hat{t}_1 \\ \dots \\ t_S - \hat{t}_S \end{bmatrix} . \tag{4.39}$$

Thus the time-map function is constructed as:

$$\lambda(\hat{t}) = \hat{t} + \sum_{k=-J}^{J} \Phi_k e^{j2\pi k f \hat{t}} \tag{4.40}$$

and this approximation exactly passes through the points $(\hat{t}_s, t_s)$.

The first derivative of the time-map function is

$$\lambda'(\hat{t}) = 1 + \sum_{k=-J}^{J} j2\pi k f \Phi_k e^{j2\pi k f \hat{t}} \tag{4.41}$$

and is exact.

The $\lambda(\cdot)$ function (4.40) and its first derivative (4.41) are now evaluated on the grid of M uniform pseudo time-points $\hat{\rho}_u^M$ to yield the M-point non-uniform grid in real time ρ^M and the matrix of time-map derivatives Λ.

Due to the Fourier nature of the representation (4.40), $\lambda(\hat{t})$ may exhibit high frequency oscillations and violate the monotonicity requirement. In practice, for the grids selected by the grid selection algorithm, if S is sufficiently large, this violation rarely happens. If a violation does happen, it can be resolved by damping the time-map function with an exponential filter μ_k, yielding a filtered construction

$$\lambda_\mu(\hat{t}) = \hat{t} + \sum_{k=-J}^{J} \mu_k \Phi_k e^{j2\pi k f \hat{t}} \tag{4.42}$$

where $\mu_k = e^{-\delta\left(\frac{k}{S}\right)^\gamma}$ and δ and γ are filter parameters [83]. Note that the filtered approximation no longer passes through the points $(t_s, \hat{t}_s)$, as illustrated in Figure 4.5.

The filtered approximation can introduce an offset τ such that $\lambda(0)=\tau$ and $\lambda(T)=T+\tau$. This offset causes no problems to the TMHB method. Strong filtering, however, can deteriorate the quality of the constructed time-map function by making its flat regions less flat. As the flat regions in the time-map function correspond to the highest grid resolution density, the filtering can cause a substantial resolution reduction in these regions, and this, in turn, can reduce the effectiveness of the TMHB method.

The values of $\lambda^{-1}(t)$ at the oversampled uniform times $t_m \in \rho_u^{M\sigma}$ are required in the unmap procedure used to compute the real Fourier coefficients. This computation is accomplished by applying Newton's method to the nonlinear equation $\lambda(\hat{t}_m) - t_m = 0$ and solving for $\hat{t}_m$ at each time point t_m.

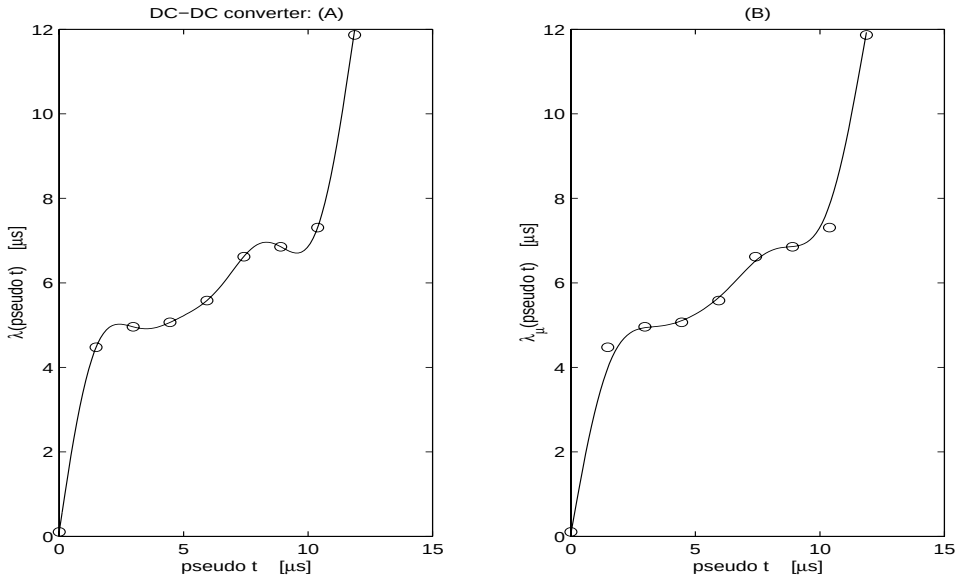

Figure 4.5: Filtering of the time-map function: (A) a non-monotonic time-map function $\lambda(\hat{t})$; (B) the filtered monotonic time-map function $\lambda_\mu(\hat{t})$ ($S = 8$, $\delta = 3.5$, and $\gamma = 2$).

4.4.1 Explicit Time-Map Functions

Almost all of the mapping references given in the introduction of this Chapter use parameterized explicit mapping functions since the problems being solved are given by explicit equations, and the solution behavior is well known. Similarly, an early implementation of the TMHB method used an explicit functional form of the time-map function. In particular, for a diode rectifier circuit whose waveforms had rapid transitions in the middle of the solution interval $[0,T]$, the time-map function [43, 77]

$$\lambda_E(\hat{t}) = \frac{T}{\pi}\left(\tan^{-1}\left\{L\tan\left[\pi\left(\frac{\hat{t}}{T}-\frac{1}{2}\right)\right]\right\}+\frac{\pi}{2}\right) \tag{4.43}$$

was used, with the parameter L controlling the increase of the grid resolution. A plot of this explicit time-map function is shown in Figure 4.6.

This explicit time-map function is well-suited for this particular circuit problem. The function is infinitely smooth, obeys the strict monotonicity requirement, and its exact first derivative and inverse can be written down explicitly as well. However, automating the choice of an appropriate explicit time-map function is at best very complicated when presented with an arbitrary circuit problem whose many solution waveforms may undergo an unknown number of rapid transitions throughout the solution domain $[0,T]$.

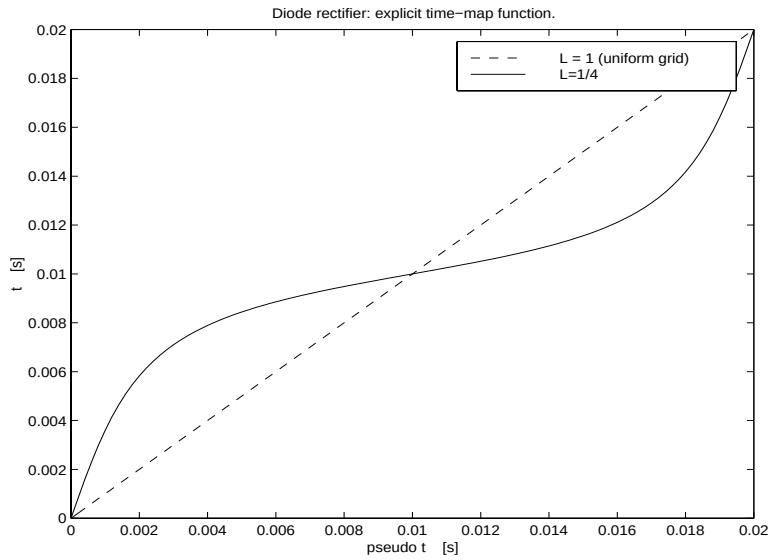

Figure 4.6: The explicit time-map function $\lambda_E(\hat{t})$

.

4.5 Error Convergence Analysis of TMHB

In this section we will study the convergence properties of TMHB, and attempt to understand more precisely why the TMHB method can achieve significant improvement in solution accuracy when compared to the HB method. The basics of the analysis presented here are similar to the convergence analysis for pseudo Chebyshev approximations for stiff problems in [71] and the analyses in [73, 76].

Let the exact periodic steady-state solution of the system of circuit equations (4.1) have the infinite Fourier series representation

$$v^*(t) = \sum_{k=-\infty}^{\infty} V_k^* e^{j2\pi k f t} \tag{4.44}$$

in the real time coordinate t, and the infinite pseudo Fourier series representation

$$v^*(t) = v^*(\lambda(\hat{t})) = \hat{v}^*(\hat{t}) = \sum_{k=-\infty}^{\infty} \hat{V}_k^* e^{j2\pi k f \hat{t}} \tag{4.45}$$

in the pseudo time coordinate $\hat{t}$. The Galerkin formulation of the HB method represents its approximate solution $v(t)$ with the truncated Fourier series

$$v(t) = \sum_{k=-K}^{k=K} V_k^* e^{j2\pi k f t} \tag{4.46}$$

while the Galerkin TMHB method uses a truncated pseudo Fourier series to represent its approximate solution $\hat{v}(\hat{t})$

$$\hat{v}(\hat{t}) = \sum_{k=-K}^{k=K} \hat{V}_k^* e^{j2\pi k f \hat{t}} \tag{4.47}$$

The Galerkin formulations of both HB and TMHB, in other words, use a truncation of the standard and pseudo infinite Fourier series respectively in approximating the solution.

We now proceed to bound the global truncation errors for the HB and the TMHB methods. We assume that the solution waveforms are infinitely smooth for clarity.

Recall that the Sobolev norm of integer order q of a T-periodic waveform $u(t)$ is given by

$$\|u\|_q^2 = \frac{1}{T} \sum_{j=0}^{q} \int_{t=0}^{T} |u^{(j)}|^2 dt = \sum_{j=0}^{q} |u|_j^2 \tag{4.48}$$

where $|u|_j$ is the L_2 norm of the j-th derivative of $u(t)$, i.e.

$$|u|_j^2 = \frac{1}{T} \int_{t=0}^{T} |u^{(j)}|^2 dt = (2\pi f)^{2j} \sum_{k=-\infty}^{\infty} k^{2j} |U_k|^2 \tag{4.49}$$

with $u(t) = \sum_{k=-\infty}^{\infty} U_k e^{j2\pi k f t}$ the infinite Fourier series representation of $u(t)$.

The principal approximation theory result for the Galerkin spectral methods [40, 42] states that for any $0 \leq q \leq p$ there exists a constant C independent of K such that

$$\|v^* - v\|_q \leq C K^{q-p} |v^*|_p. \tag{4.50}$$

The quantity $\|v^* - v\|_q$ is the truncation error measured in the q-th Sobolev norm. The inequality (4.50) states that the truncation error is bounded by the norm of the derivatives of the approximated waveform.

The inequality (4.50) defines a family of error bounds. For a fixed q, the truncation error is bounded by the smallest of the right-hand sides of (4.50). If the right-hand sides of (4.47) decrease as p is increased, and K is sufficiently large, the error bound tightens, and the method achieves spectral accuracy.

If the waveform has rapid transitions, as p is increased, the norms of the derivatives $|v^*|_p$ may generate a sequence with a more rapid increase rate than the decay rate of the CK^{q-p} sequence when $K \leq K_0$ where K_0 is fairly large. This in turn means that the bound of the truncation error, i.e. the right-hand side of (4.50) will grow for increasing p. Instead of achieving spectral accuracy, the bound of the truncation error will be $O(K^{-p_m})$ (i.e. the error will have polynomial convergence) where p_m is the value of p at which the error bound attains a minimum.

Spectral accuracy will eventually be retrieved for large K, i.e. when $K > K_0$. In other words, when the number of time-points $M = 2K + 1$ is larger than $M_0 = 2K_0 + 1$, the sharp features in

the waveform will be sufficiently well resolved, and the error of the spectral method will start its spectral (infinite order) decay. However, the number of harmonics K_0 which defines the threshold of the spectral convergence of the error can be very large for very fast varying waveforms.

The main mechanism through which the TMHB attains a faster rate of error convergence is by working in the pseudo time domain, in which the waveform $\hat{v}^*$ is smoother. When $K < K_0$ i.e. the number of harmonics is smaller than the spectral accuracy threshold, the sequence $|\hat{v}^*|_p$ does not increase as rapidly as the sequence $|v^*|_p$ for increasing p, and the TMHB error is thus more tightly bound than the HB error. When the number of harmonics K approaches the spectral accuracy threshold K_0, the HB error bound will quickly catch up with the TMHB error bound and both methods will be converging at the same rate.

We will now proceed to derive the HB and TMHB truncation error bounds, and investigate their values and decay rates using an explicit waveform and explicit time-map function. Setting $q = 0$ in (4.50), the truncation error for the HB method should satisfy

$$\|v^* - v\|_0 \leq CK^{-p}|v^*|_p. \tag{4.51}$$

More specifically,

$$\begin{aligned}
\|v^* - v\|_0^2 &= |v^* - v|^2 \\
&= \sum_{|k|=K+1}^{\infty} |V_k^*|^2 \\
&= (2\pi f)^{-2p} \sum_{|k|=K+1}^{\infty} k^{-2p}(2\pi f)^{2p} k^{2p} |V_k^*|^2 \\
&\leq (2\pi f)^{-2p} K^{-2p} \sum_{|k|=K+1}^{\infty} k^{2p}(2\pi f)^{2p} |V_k^*|^2 \\
&\leq (2\pi f)^{-2p} K^{-2p} |v^*|_p^2.
\end{aligned} \tag{4.52}$$

The truncation error for the HB method is therefore bounded by

$$|v^* - v| \leq (2\pi f)^{-p} K^{-p} |v^*|_p. \tag{4.53}$$

Similarly, the truncation error for the TMHB method satisfies

$$|\hat{v}^* - \hat{v}| \leq (2\pi f)^{-p} K^{-p} |\hat{v}^*|_p. \tag{4.54}$$

The right-hand sides of (4.53) and (4.54) define families of error bounds as p takes different integer values i.e. $1 \leq p \leq \infty$. The errors will be bounded by the minimums of these bound families.

Let $B(K,p)$ be the truncation error bound family for the HB method, i.e.

$$B(K,p) = (2\pi f)^{-p} K^{-p} |v^*|_p \tag{4.55}$$

and $\hat{B}(K,p)$ the truncation error bound family for the TMHB method:

$$\hat{B}(K,p) = (2\pi f)^{-p} K^{-p} |\hat{v}^*|_p. \tag{4.56}$$

The standard truncation error bound $B_m(K)$ for the HB method is

$$B_m(K) = \min_p B(K,p) \tag{4.57}$$

and the pseudo truncation error bound for the TMHB method is

$$\hat{B}_m(K) = \min_p \hat{B}(K,p). \tag{4.58}$$

When $K < K_0$, as argued above, we expect the TMHB error bound to be tighter than the HB error bound, i.e. $\hat{B}_m < B_m$ since the waveform in pseudo time $\hat{v}^*$ is smoother than v^* and thus generates a more slowly increasing sequence of derivative norms $|\hat{v}^*|_p$ than the sequence $|v^*|_p$.

We will now illustrate the error convergence mechanism of the TMHB method by using an explicit analytic waveform $v^*(t)$ and an explicit time-map function $t = \lambda(\hat{t})$. The analytic waveform is

$$v^*(t) = (1 + \tanh(S(t-0.5))) \tag{4.59}$$

on $t \in [0,1]$ with the parameter $S = 1000$ determining the steepness of the rapid transition at $t = 0.5$. The time-map function is

$$t = \lambda(\hat{t}) = \frac{1}{\pi}\left(\arctan(L\tan(\pi(\hat{t}-0.5))) + \frac{\pi}{2}\right) \tag{4.60}$$

with $L = 0.01$. The time-map function is in essence a change of coordinate, transforming the waveform $v^*(t)$ into the smoother function $\hat{v}(\hat{t}) = v(\lambda(\hat{t}))$ in the pseudo time coordinate. Figure 4.7 illustrates the analytic waveform in real time, the time-map function, and the smoothed waveform in pseudo time.

While the waveform $v^*(t)$ is not periodic, it can be periodized with period $T = 2$ by adding to it its mirror (flip) image $v^*(2-t)$ for $t \in [1,2]$. To compute the standard error bound families $B(K,p)$ corresponding to $v^*(t)$ and pseudo error bound families $\hat{B}(K,p)$ corresponding to $\hat{v}^*(\hat{t})$, we compute the waveform derivative norms $|\cdot|_p$ for the integer range $1 \le p \le 50$ by numerically evaluating the norm integrals with the trapezoidal rule. The derivative discontinuities at $t = 1$ are numerically ignored by integrating over only the first half of the period, i.e. over $t \in [0,1]$.

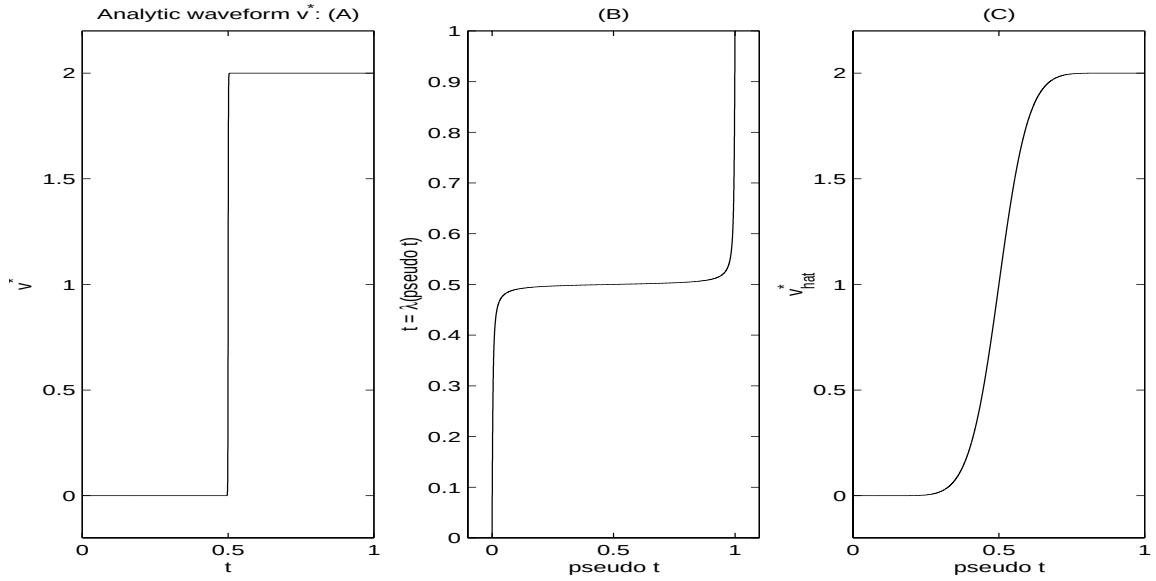

Figure 4.7: The analytic waveform $v^*(t)$: (A) plot of v^* in real time; (B) time-map function; (C) plot of $\hat{v}^*$ in pseudo time.

As our primary interest is to see how the v^* and $\hat{v}^*$ error bounds behave for increasing K, we compute the minimums $B_m(K)$ and $\hat{B}_m(K)$ of the error bound families at each K. Figure 4.8 shows a plot of these error bounds in dB versus K. The left plot is linear in K and demonstrates that the error bound $\hat{B}_m(K)$ of the pseudo approximation is much tighter than the error bound $B_m(K)$ of the standard approximation for small and moderate values of K. For example, at $K = 300$ the pseudo error bound is 130dB tighter than the standard error bound. As a result, the truncation error of the pseudo approximation will also be several orders of magnitude smaller than the truncation error of the standard approximation for a range of values for K.

The right plot in Figure 4.8 plots the error bounds versus a log scale for the number of harmonics K. The operating range of the TMHB method is the range of K where the pseudo error bound is smaller than the standard error bound. This range is clearly visible, and it can be seen that it extends up to $K = K_0 = 6300$. For small K, the standard error bound decreases at a rate of about 20 dB per decade, which indicates a polynomial order convergence of $O(K^{-1})$, while the pseudo error bound decreases at rate of 80 dB per decade i.e. with $O(K^{-4})$. As K approaches the spectral accuracy threshold $K_0 = 6300$, the decay rate of the standard error bound increases, and the value of the standard error bound catches up with the value of the pseudo error bound at the threshold. For K above the threshold, both error bounds have spectral decay, although a limiting rate of 1000 dB per decade is observed which corresponds to $O(K^{-50})$ (explained below).

Note that when the standard error bound finally catches up with the pseudo error bound at

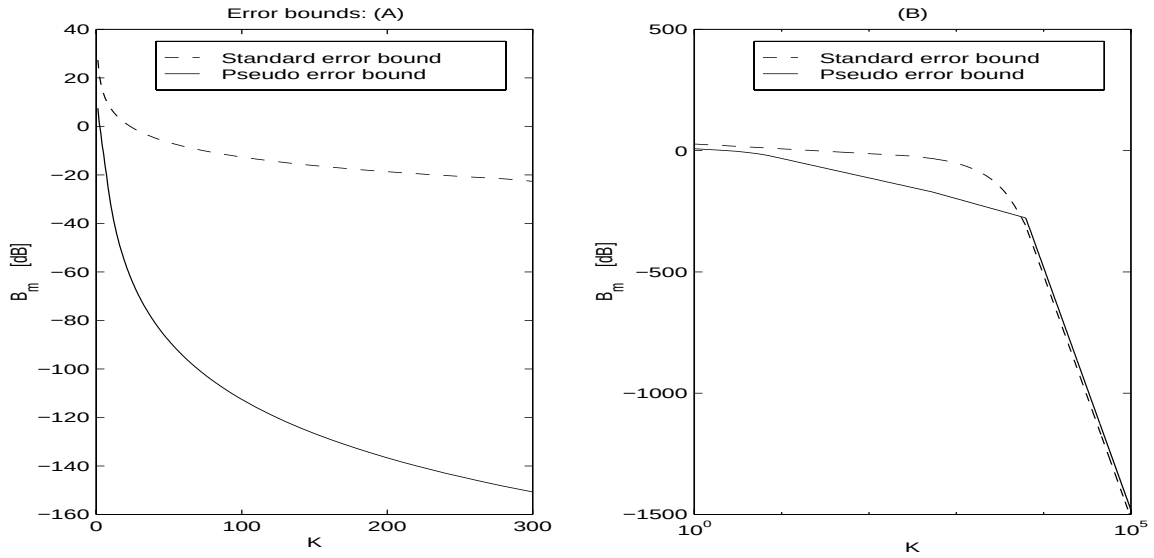

Figure 4.8: Convergence of the standard $B(K)$ and pseudo $\hat{B}(K)$ error bounds in dB versus K: (A) linear plot; (B) log plot.

$K = K_0$, both bounds have reached very small values of around -270 dB which is close to the double precision accuracy (-300 dB) of most numerical computations.

The convergence exponents of the error bounds in Figure 4.8 can be explained by plotting the family (set) of standard and pseudo error bounds for fixed K versus the norm order p (see Figure 4.9). The values of p at which the families of error bounds attain the minimum are the convergence exponents of the error bounds (taken with a negative sign). For $K = 100$ (plot A) the minimum of the standard error bound family is at $p = 1$, while the pseudo error bound family has a much smaller minimum at $p = 4$. At around $K = K_0 = 6300$ (plot B) both error bound families reverse the increasing trend for large p. For $K = 10000$ (plot C) both error bound families decay monotonically at the fastest possible rate over the entire p range, so their minimums are at the largest considered p in our computation. The limiting convergence rate exponents of the error bounds in Figure 4.8 are 50 because the bound families were numerically computed up to the value of $p = 50$.

As it was explained earlier, the reason for the smaller pseudo error bound in the operating range of the TMHB method $K < K_0$ is due to the more slowly increasing sequence of derivative norms $|\hat{v}^*(t)|_p$ of the waveform in pseudo time coordinates. Figure 4.10 plots the sequences of standard derivative norms $|v^*(t)|_p$ and pseudo derivative norms $|\hat{v}^*(t)|_p$ in dB versus p. As it can be seen, the transformation into pseudo time coordinates via the time-map function yields a slower rate of increase of the pseudo derivative norms when p is small. The quality of the time-map function will

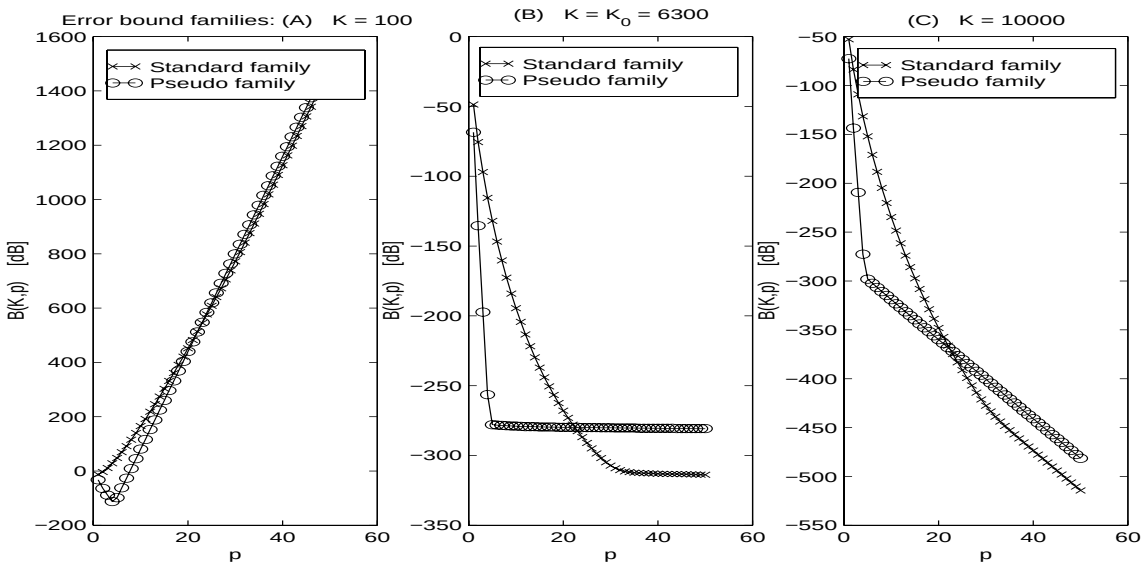

Figure 4.9: The standard $B(K,p)$ and pseudo $\hat{B}(K,p)$ error bound families in dB versus p: (A) $K = 100$; (B) $K = K_0 = 6300$; (C) $K = 10000$.

determine how much slower will the pseudo derivative norms increase compared to the standard derivative norms; it will also determine for how many values of p will this rate slowing be in effect.

While our error convergence analysis used Galerkin (spectral) Fourier approximations, it can be readily extended for the collocation (pseudospectral) Fourier approximations since the aliasing term present in the collocation Fourier coefficients is of the same order as the truncation error of the method [67, 39] (Chapter 3).

The analysis also holds if the solution waveforms are not infinitely differentiable. In that case the family of error bounds on the right-hand side of (4.50) has a finite number of members.

4.6 Demonstration of the Error Convergence of Pseudo Fourier Series

The greater accuracy of the TMHB method comes from the smaller global truncation error of the pseudo Fourier series for the (smoother) solution waveform in pseudo time, compared to the global truncation error of the standard HB Fourier series approximation of the solution waveform in real time.

Furthermore, from the results of the analysis in Section 4.5, when K is small or moderate, we would expect the error in the TMHB method to decrease $O(K^{-\hat{p}})$ compared to the error convergence rate $O(K^{-p})$ for the HB method, where the convergence rate exponents satisfy $\hat{p} > p$. In

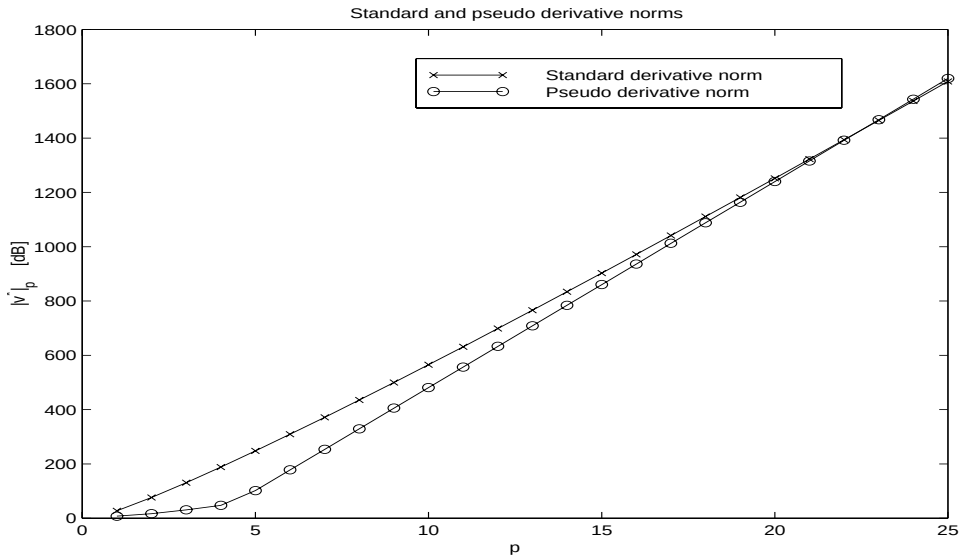

Figure 4.10: The standard ($|v^*|_p$) and pseudo ($|\hat{v}^*|_p$) derivative norms in dB versus p of the analytic waveform v^*.

other words, the error in the TMHB method should decrease exponentially faster than the standard HB error with increasing number of harmonics K (or equivalently, with increasing number of collocation time-points $M = 2K+1$).

To demonstrate the superior error convergence of the pseudo Fourier series, consider approximating an analytic waveform with both standard (uniform grid) Fourier series and (non-uniform grid) pseudo Fourier series. The analytic waveform is

$$v_A^*(t) = \frac{1}{2}(1+\tanh(10^3(x-0.4)))\exp(-((x-0.4)/0.2)^2) \tag{4.61}$$

on the interval $[0,T]$ with $T = 1.5$. While this function is not periodic, the periodicity error $v_A^*(T) - v_A^*(0)$ is of order 10^{-13} (-260dB), and is sufficiently small for the demonstration. The function $v_A^*(t)$ has a very rapid transition at $t = 0.4$ as it can be seen from Figure 4.11. The iterative grid selection algorithm (discussed in Chapter 5) resulted in a non-uniform grid which was used to construct the time-map function also shown in Figure 4.11. When the time-map function is evaluated on a uniform grid $\hat{\rho}_u^M$ in pseudo-time, it generates the non-uniform grid ρ^M in real time. The smoothing effect of this non-uniform grid on the function $v_A^*(t)$ is also shown in Figure 4.11.

To compute the Fourier coefficients in the standard approximation, the function $v_A^*(t)$ is evaluated at a uniform set of M points ρ_u^M, and an inverse FFT is performed on this vector of values. To compute the pseudo Fourier coefficients, the function $v_A^*(t)$ is evaluated on the set of non-uniform points ρ^M, and again an inverse FFT is performed on this vector of values.

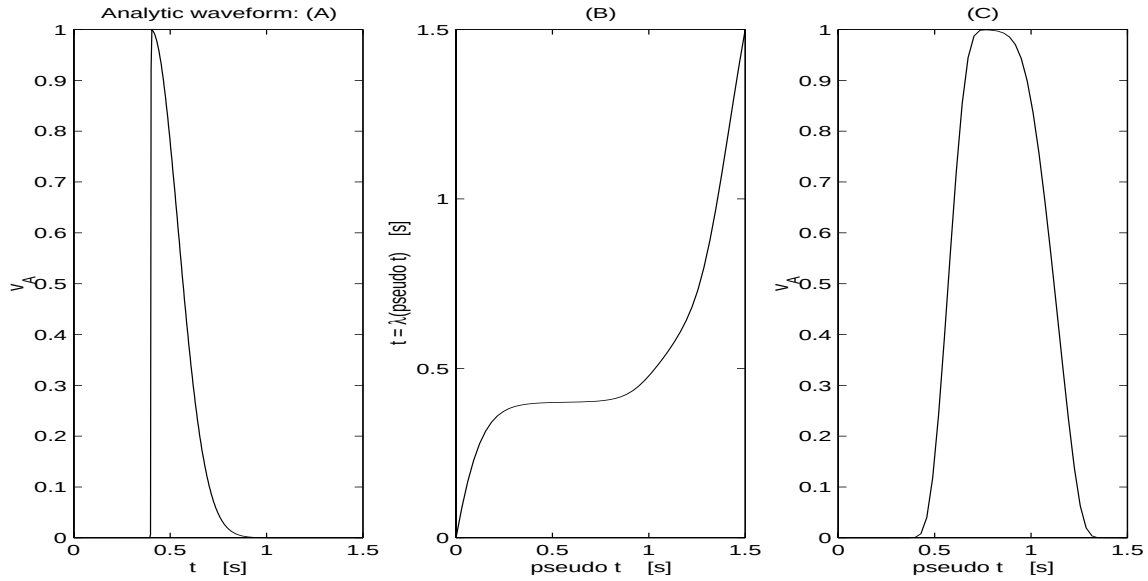

Figure 4.11: The analytic waveform $v_A^*(t)$: (A) plot of v_A^* in real time; (B) time-map function; (C) plot of v_A^* in pseudo time.

To compute the error in the standard and pseudo Fourier series approximations, two different error measures were used. The first one is simply the magnitude of the K-th Fourier coefficient of the series, i.e. $\varepsilon_1 = |V_K|$ for the standard and $\hat{\varepsilon}_1 = |\hat{V}_K|$ for the pseudo Fourier series approximation. The magnitude of this highest-frequency Fourier coefficients is well known to be a measure of the truncation error in the Fourier representation.

The second error measure is computed in the time domain, and it is the normalized L_2 norm of the vector of time-domain errors in the Fourier interpolants at the M midpoints of the grids, i.e. $\varepsilon_2 = \|v_A(t_n) - v_A^*(t_n)\|_2$ where $v_A(t)$ is the standard Fourier interpolant, and $t_n = (t_m + t_{m+1})/2$ where $t_m \in \rho_u^M$ (uniform grid); $\hat{\varepsilon}_2 = \|\hat{v}_A(t_n) - v_A^*(t_n)\|_2$ where $\hat{v}_A(t)$ is the pseudo Fourier interpolant, and $t_n = (t_m + t_{m+1})/2$ where $t_m \in \rho^M$ (non-uniform grid).

A plot of these two error measures in dB is given in Figure 4.12. The spectacular error convergence of the pseudo Fourier series approximation in the operating range (small and moderate K) is clearly evident. While the standard Fourier series achieves a convergence rate exponent of about $p = 1.1$, the convergence exponent for the pseudo Fourier series is about $\hat{p} = 20$. The plot also shows that the two error measures are equivalent.

A plot of the standard and pseudo Fourier interpolants at $M = 12$ is shown in Figure 4.13. It is clearly visible that even for a small M the pseudo Fourier series approximation is much better than the standard Fourier series approximation.

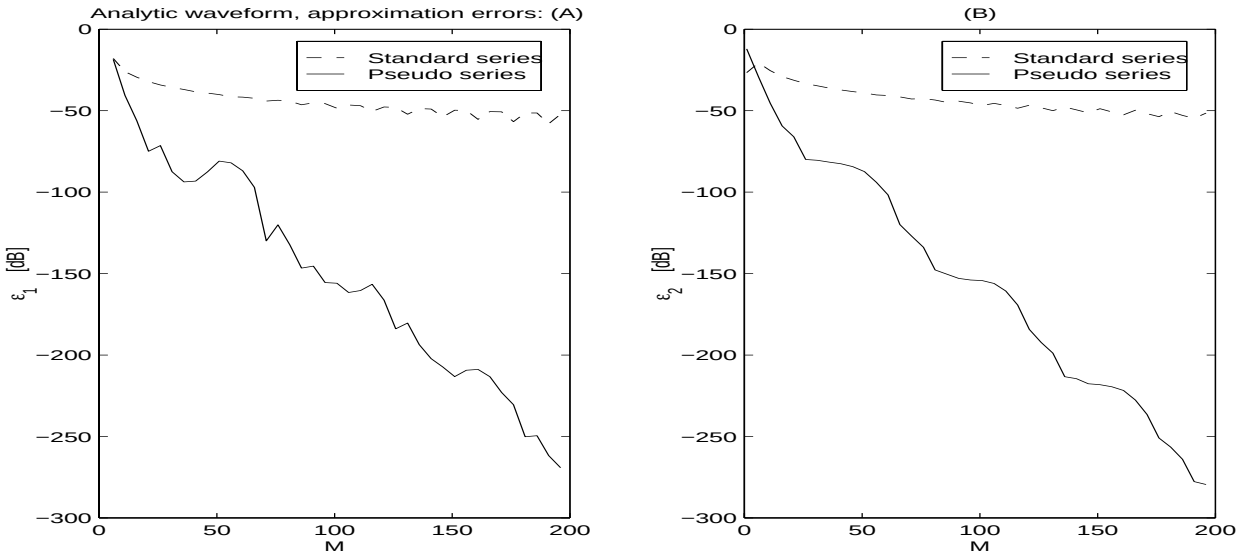

Figure 4.12: Errors in standard and pseudo Fourier series approximations of the analytic waveform $v_A^*(t)$: (A) frequency-domain error measure ε_1; (B) time-domain error measure ε_2.

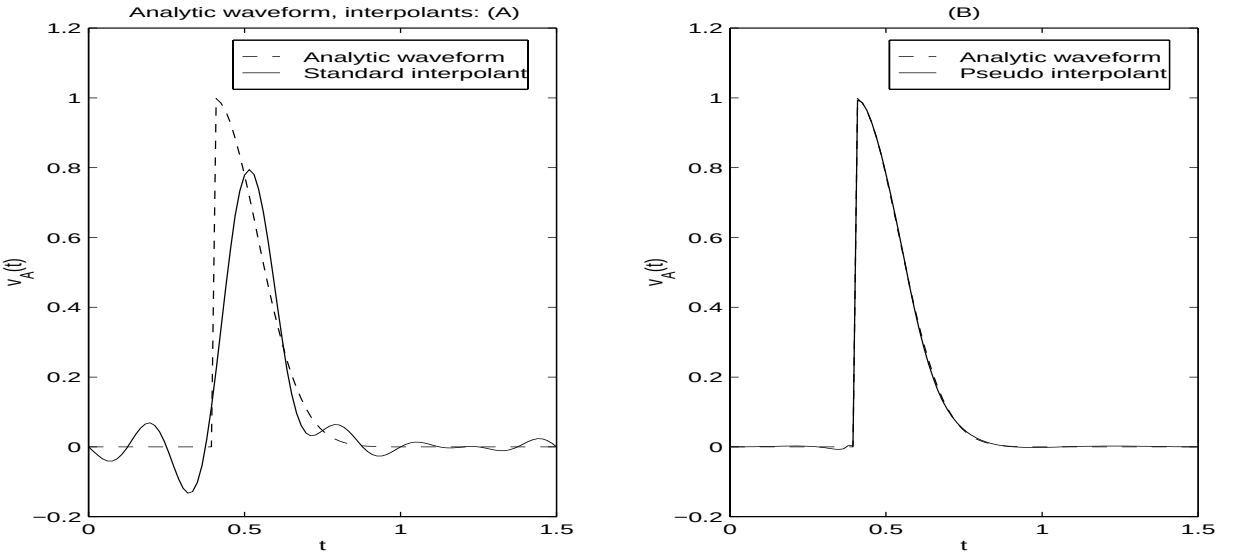

Figure 4.13: Fourier interpolants of the analytic waveform $v_A^*(t)$, $M = 12$: (A) standard interpolant; (B) pseudo interpolant.

Chapter 5

Grid Selection Strategies for the Time-Mapped Harmonic Balance

In Chapter 4 we introduced the Time-Mapped Harmonic Balance (TMHB) Method, a fast Krylov-subspace spectral method for accurate steady-state simulation of circuits with rapid transitions. The TMHB features a non-uniform grid which resolves the sharp features in the signals.

The success of the Time-Mapped Harmonic Balance method is critically dependent on the selection of an appropriate non-uniform grid. In this Chapter, after a brief overview of the TMHB method in the next section, we will present several different selection strategies, and discuss some issues related to their use in practice as a part of the TMHB algorithm.

The relative merits of these strategies when used in TMHB simulation of several circuits are given in Chapter 6.

5.1 TMHB Method Overview

Consider a circuit described with N nonlinear differential equations:

$$\dot{q}(v(t)) + i(v(t)) + u(t) = 0 \tag{5.1}$$

where $v(t) \in R^N$ is the vector of node voltages, $q(v(t)) \in R^N$ the vector of node charges (or fluxes), $i(v(t)) \in R^N$ the vector of resistive node currents, and $u(t) \in R^N$ the vector of input sources. The periodic steady-state solution of (5.1) satisfies the two-point constraint $v(T) = v(0)$.

The TMHB method utilizes a non-uniform grid of $2K+1 = M$ time-points in contrast to the uniform grid used in standard HB. This non-uniform grid in real time t is related to a uniform grid in pseudo-time $\hat{t}$ via the time-map function λ such that $t = \lambda(\hat{t})$, $\lambda(0) = 0$, and $\lambda(T) = T$.

The time-map function is constructed starting from a selected grid of S non-uniform real time time-points, and is a sum of a linear part and a T-periodic part:

$$\lambda(\hat{t}) = \hat{t} + \sum_{k=-J}^{J} \Phi_k e^{j2\pi k f \hat{t}} \tag{5.2}$$

where $2J+1 = S$. The $\lambda(\cdot)$ function (5.2) is then evaluated at M uniform pseudo time-points to yield the M-point non-uniform grid in real time. The construction guarantees the ability to compute $\lambda'(\hat{t})$ with spectral accuracy required by the TMHB method. The strict monotonicity of $\lambda(\hat{t})$ can be ensured by applying an exponential filter to the time-map function.

In the TMHB the problem (5.1) is transformed into:

$$\frac{1}{\lambda'(\hat{t})}\frac{d}{d\hat{t}} q(v(\lambda(\hat{t})) + i(v(\lambda(\hat{t}))) + u(\lambda(\hat{t})) = 0 \tag{5.3}$$

The solution waveforms in TMHB are approximated with truncated pseudo Fourier series:

$$v(t) = v(\lambda(\hat{t})) = \sum_{k=-K}^{k=K} \hat{V}_k e^{j2\pi k f \hat{t}} \tag{5.4}$$

As in the state-of-the-art standard HB method [7, 10, 11, 15], the TMHB method uses a matrix-implicit Krylov-subspace approach to compute the pseudo Fourier coefficients $\hat{V}$ of the solution with $O(NM\log M)$ complexity (which is the complexity of the FFTs used to compute the matrix-vector product in the GMRES solver for the linear problem at each Newton iteration).

To compute the real time Fourier coefficients, the TMHB introduces a non-uniform oversampled grid in pseudo time such that $\lambda(\cdot)$ maps this oversampled grid in pseudo time to a uniform oversampled grid in real time. Since $\hat{t} = \lambda^{-1}(t)$, (5.4) can be rewritten as

$$v(t) = \sum_{k=-K}^{k=K} \hat{V}_k e^{j2\pi k f \lambda^{-1}(t)}. \tag{5.5}$$

The summation in (5.4) is then evaluated to give the solution waveforms at the oversampled uniform grid in real time. Finally, since the $v(t)$'s are now known on a uniform grid in real time, we can use the FFT to compute the real Fourier coefficients V.

5.2 Grid Selection Strategy Requirements

Grid selection techniques for finite-difference methods are discussed in several mathematical references [21, 22]. The techniques are typically classified as either a priori or adaptive. The grid selection strategies considered in this section are a priori methods, in which the S-point non-uniform

grid used in the construction of the time-map function is selected beforehand, and remains fixed during the iterative computation of the solution. In contrast, adaptive grid selection strategies repeatedly update both the grid and the approximate solution until a convergence criteria is met. In the adaptive grid selection for a fixed-order finite-difference numerical method [22], the grid is adjusted to a measure of the local truncation error of the method in the approximate solution at each iteration. However, the TMHB method is a spectral method, which can be viewed as a variable-order finite-difference method (with the order equal to the size K of the Fourier basis), and a local truncation error estimate requires a costly summation of the solution's truncated pseudo Fourier series approximation. A further complication is the need for an expensive spectrally accurate interpolation from the old grid to the new at each iteration of the adaptive method, for which the FFT cannot be used due to the non-uniform spacings of the grids.

Adaptive grid selection techniques have been used in mapped spectral methods applied to problems in fluid dynamics [71, 76, 82]. However, in all of these references a fixed functional parameterized form of the mapping function is considered, with the mapping parameters determined by minimizing a particular functional of the solution. The references deal with explicit small PDE problems, and the adaption is performed iteratively at each time evolution step of the solution.

By nature, all a priori grid selection techniques require some knowledge of the solution behavior (a solution guess). All selection strategies considered here obtain this information by solving the problem (5.1) using a shooting-Newton method with a low-order integration scheme, whose cost is kept low by a loose convergence tolerance.

The shooting-Newton method [4] reformulates the two-point constraint as

$$\varphi(v(0), 0, T) - v(0) = 0 \tag{5.6}$$

where φ is the state-transition function for (5.1). Equation (5.6) is solved with a Newton method.

The cost of the solution guess computation is kept low by loosening the error tolerance of the shooting method, as well as approximating the Jacobian J_φ of the state-transition function using a forward-difference formula. Note that the solution guess also serves as an initial guess for the TMHB Newton iterations.

As the solution guess is given at the discrete times selected by the local time-step control mechanism of the transient simulation over one period at the last iteration of the shooting-Newton method, local quadratic interpolation is used to get the solution values at arbitrary time.

The S-point non-uniform grid used in the construction of the time-map function should be selected based on two major considerations: reduction of the truncation error (accuracy consideration) and preserving stability.

One approach in satisfying the accuracy requirement is to concentrate the grid time-points in the high-gradient regions of the solution waveforms. The solution waveform $v(t)$ in real time then becomes the smoother waveform $v(\lambda(\hat{t}))$ when viewed in pseudo-time, as illustrated in Fig. 5.1. The greater solution accuracy of the TMHB stems from the reduction in the global truncation error in the pseudo Fourier series approximation of this smoother waveform vs. the standard Fourier series approximation of the original waveform on the uniform standard HB grid.

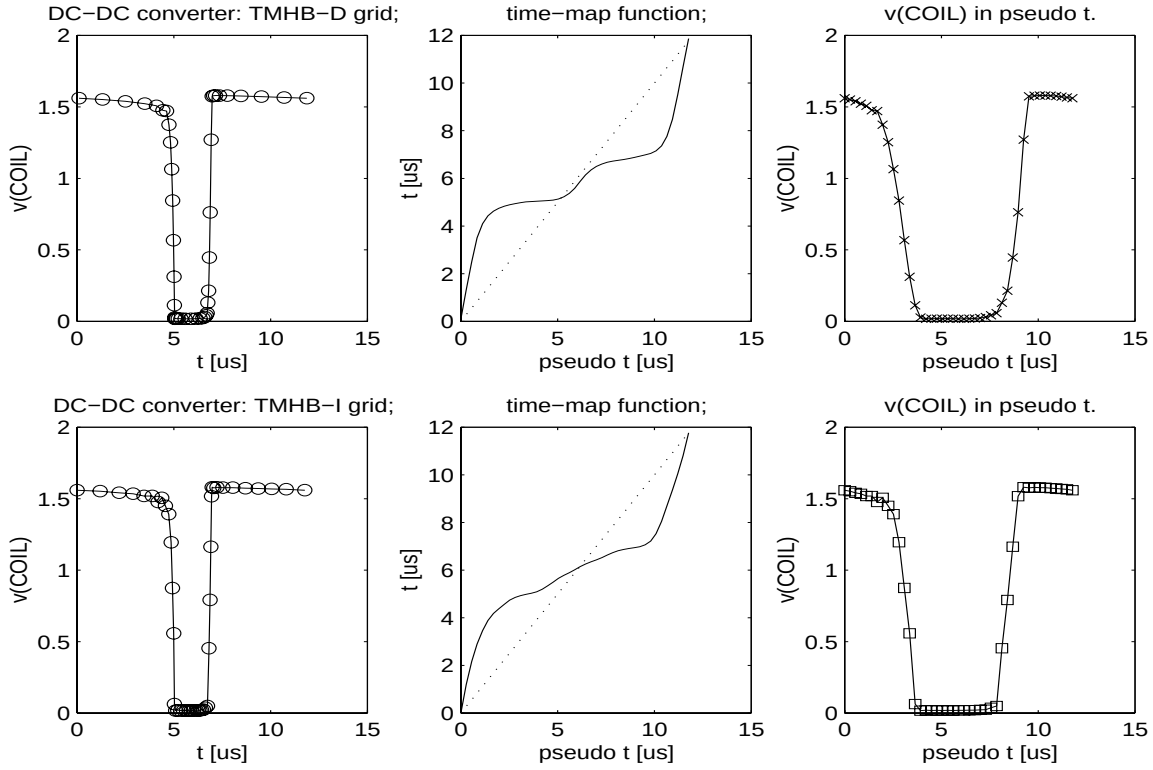

Figure 5.1: DC-DC converter circuit: grid and time-map function for direct strategy (top row) and iterative grid selection strategy (bottom row).

The stability requirement is in general difficult to handle quantitatively, but in practice, bounding the ratio of the neighboring grid spacings is a known rule-of-thumb approach that works well:

$$\frac{\alpha}{\alpha+1} \leq \frac{h_{m+1}}{h_m} \leq \frac{\alpha+1}{\alpha} \tag{5.7}$$

where $\alpha > 0$ determines the grid "rigidity", and $h_m = t_{m+1} - t_m$ with t_m and t_{m+1} two neighboring grid points. Note that rigid grids (larger α) can suffer from "point exhaustion" if the number of grid points S is not large enough. Finally, note that both bounding the grid spacings and using a large enough number of grid points greatly enhances the strict monotonicity of the time-map construction (5.2) and virtually eliminates the need for filtering.

5.3 Direct Grid Selection

The shooting-Newton solution guess generates a non-uniform grid in which the points are distributed such that the local truncation error of the integration method is kept under the specified (loose) tolerance. In general this means smaller time-steps in the regions where the waveforms undergo rapid transitions, smoother waveforms in the pseudo-time uniform grid, and a reduced global truncation error in the TMHB pseudo Fourier series approximations. Unfortunately, in most cases the grids generated by the shooting-Newton method are unusable since their time-steps change too rapidly. This not only causes non-monotonicity in the constructed time-map function and necessitates the use of filtering which renders a much less effective time-map function, but more importantly, instability and convergence problems in the TMHB iterations.

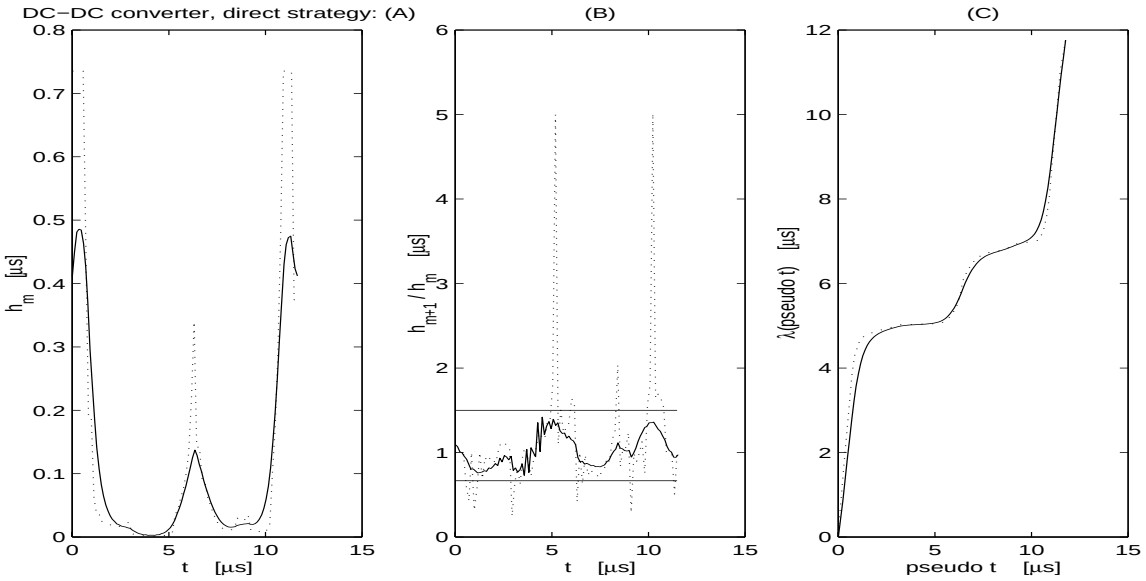

Figure 5.2: DC-DC converter circuit, direct strategy, before (dashed) and after (solid) the smoothing: (A) time-steps; (B) ratios of neighboring time-steps; (C) time-map function.

This instability can be alleviated by bounding the time-steps to obey (5.7). We accomplish this by "smoothing" the S-point shooting-Newton grid with a discrete exponential kernel:

$$\tilde{h}_i = \frac{1}{\gamma} \sum_{j=1}^{S} h_j \left(\frac{\alpha}{\alpha+1} \right)^{|i-j|} \tag{5.8}$$

where h_j are the time-steps in the shooting-Newton grid, and $\tilde{h}_i$ the time-steps from the "smoothed" grid. Note that the kernel is defined for $-\frac{S}{2} \leq (i-j) \leq \frac{S}{2}$ and is periodic with period S. The scaling factor γ is the sum of the kernel over one period.

The direct selection strategy is illustrated in Figure 5.2, with $\alpha = 2$. The dashed lines are the quantities before the smoothing, i.e. straight out of the shooting-Newton initial solution guess run, and the solid lines are the same quantities after the direct selection strategy (i.e. the "smoothing" of the shooting-Newton grid with the periodic exponential kernel). The horizontal bounds in the plot of the ratios of neighboring time-steps are $\alpha/(\alpha+1) = 2/3$ and $(\alpha+1)/\alpha = 3/2$.

The described direct grid selection strategy is simple and fast. Its disadvantages are the indirect control over the number of grid points S via the shooting-Newton tolerances, and the asymmetric nature of the generated grids (somewhat alleviated by smoothing) due to the one-directionality of the shooting-Newton time-step control.

5.4 Iterative Grid Selection

The iterative grid selection was developed from ideas in [21], in which PDE IVPs are solved by adaptively adjusting the spatial grid during the time evolution of the problem. In the iterative grid selection, an initial uniform grid ρ_u^S evolves into the final non-uniform grid ρ^S. During this process the S time points $t_m \in \rho_u^S$ are re-distributed in such a way that the shooting-Newton solution guess of (5.1) is uniformly resolved. The solution guess guides the iterations such that in the regions where the solution has large gradients, the points should be concentrated, and in the regions where the solution varies slowly, the points should be spread out.

We first introduce the grid resolution n defined as $n_m = \frac{T}{t_{m+1}-t_m}$. For the grid resolutions to yield an acceptable grid iterate we need

$$\sum_{m=1}^{S} \frac{1}{n_m} = 1 \ . \tag{5.9}$$

A monitor function R specifies the desired resolution and should indicate increased resolution in the regions where the shooting-Newton solution has rapid transitions. A choice for this function that works well for a number of circuit examples is the L_∞ norm of the weighted rate of the change in the solution guess:

$$R_m = \frac{T}{t_{m+1}-t_m} \max_j \left(\frac{v_j(t_{m+1}) - v_j(t_m)}{F_j} \right) \tag{5.10}$$

where v_j is the shooting-Newton solution guess waveform of the j-th circuit equation, and F_j are the waveform weights, set to the maximal peak-to-peak voltage (or current) amplitude among all waveforms. Local quadratic interpolation is used to compute the solution guess values at arbitrary time.

For accuracy we require that the grid resolution is proportional to the monitor function

$$\frac{n_m}{R_m} = \frac{n_{m+1}}{R_{m+1}} \; . \tag{5.11}$$

For stability we obey (5.7) by replacing n_m in (5.11) with a centered difference smoothing $n_m - \alpha(\alpha+1)(n_{m+1} - 2n_m + n_{m-1})$ resulting in

$$\begin{aligned} \frac{n_m - \alpha(\alpha+1)(n_{m+1} - 2n_m + n_{m-1})}{R_m} &= \\ \frac{n_{m+1} - \alpha(\alpha+1)(n_{m+2} - 2n_{m+1} + n_m)}{R_{m+1}} & \; . \end{aligned} \tag{5.12}$$

Equation (5.9), and (5.12) written for $m = 1, 2, ..., (S-1)$ form a system of S nonlinear equations that are solved for n using a damped Newton method.

The Jacobian J_x can be easily written down

$$J_x = \begin{bmatrix} b_1 & c_1 & d_1 & & & & a_1 \\ a_2 & b_2 & c_2 & d_2 & & & \\ & a_3 & b_3 & c_3 & d_3 & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & a_{S-2} & b_{S-2} & c_{S-2} & d_{S-2} \\ d_{S-1} & & & & a_{S-1} & b_{S-1} & c_{S-1} \\ -\frac{1}{n_1^2} & -\frac{1}{n_2^2} & \cdots & \cdots & \cdots & \cdots & -\frac{1}{n_M^2} \end{bmatrix} \tag{5.13}$$

where

$$a_m = -\alpha(\alpha+1)R_{m+1}, \tag{5.14}$$

$$b_m = (2\alpha^2 + 2\alpha + 1)R_{m+1} + \alpha(\alpha+1)R_m, \tag{5.15}$$

$$c_m = -(2\alpha^2 + 2\alpha + 1)R_m - \alpha(\alpha+1)R_{m+1}, \tag{5.16}$$

$$d_m = \alpha(\alpha+1)R_m \tag{5.17}$$

and is computed exactly.

The l-th Newton iteration is the linear problem $J_x^{(l)}(\Delta n)^{(l+1)} = -x(n^{(l)})$ which is efficiently solved by Gaussian elimination since the Jacobian J_x is banded and S chosen independently of M.

The initial guess for the Newton iterations is $n_m = \frac{S}{T}$ corresponding to an S-point uniform grid. At the l-th newton iteration the resolution iterate $n^{(l)} = n^{(l-1)} + \kappa(\Delta n)^{(l)}$ is used to compute a potential grid iterate $\{t_1^{(l)}, t_2^{(l)}, ..., t_S^{(l)}\}$ where $t_{m+1}^{(l)} = t_m^{(l)} + \frac{T}{n_m^{(l)}}$. The damping factor κ is decreased and $n^{(l)}$ recomputed until the potential grid iterate satisfies the strict monotonicity property. The

l-th iteration is then deemed successful, the monitor function recomputed on the new grid iterate, and the iterations continued until $||\Delta n||$ and $||x(n)||$ are smaller than specified tolerances.

The iterative selection strategy is illustrated in Figure 5.3, with $\alpha = 2$ and $S = 50$. The dashed lines are the quantities corresponding to the initial uniform ρ^S grid, and the solid lines are the same quantities after a converged iterative selection strategy. The horizontal bounds in the plot of the ratios of neighboring time-steps are $\alpha/(\alpha+1) = 2/3$ and $(\alpha+1)/\alpha = 3/2$.

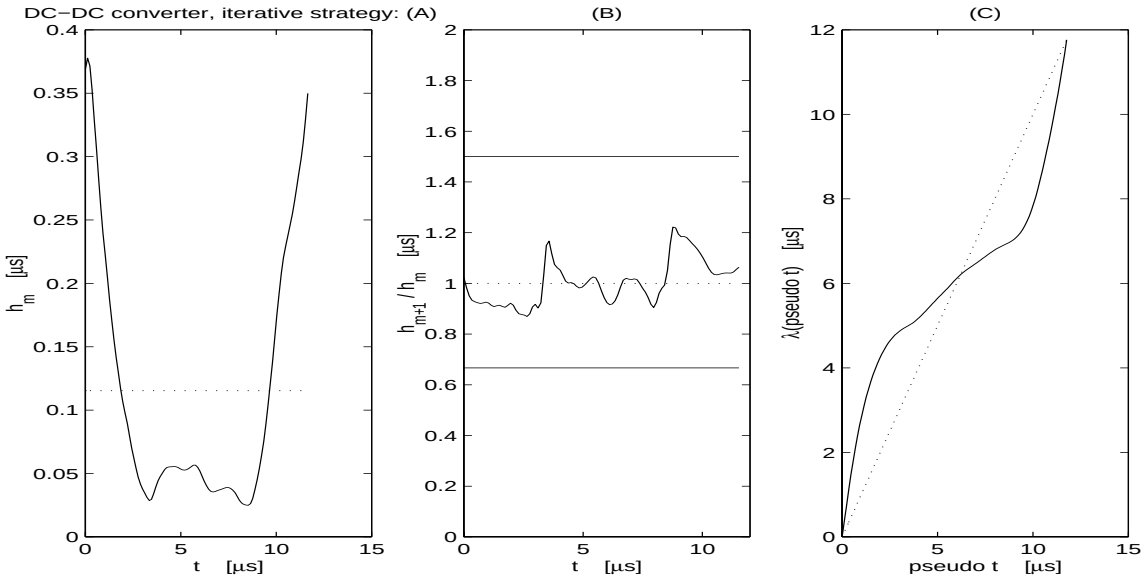

Figure 5.3: DC-DC converter circuit, iterative strategy, initial grid (dashed) and final grid (solid): (A) time-steps; (B) ratios of neighboring time-steps; (C) time-map function.

The described iterative grid selection strategy has the advantage of choosing S independently of M and the number of grid points in the shooting-Newton grid. Its disadvantage is its higher complexity when compared to the direct selection strategy. The complexity can also rise rapidly for large S due to the Gaussian elimination step in Newton's method.

5.5 Optimization-Based Grid Selection

Two different cost functions were used in setting up an optimization-based grid selection strategy. The "design parameters" for both cost functions were the S grid points t_m. The first cost function was simply the magnitude of the last pseudo Fourier coefficient associated with the current grid iterate and the shooting-Newton guess waveform with sharpest transitions. The magnitude of this coefficient is a measure of the truncation error in the pseudo Fourier approximation.

The second cost function was the time-domain L_2 error between the shooting-Newton guess waveform and the pseudo Fourier interpolant associated with the current grid iterate. Note that this cost function computation required a computation of a time-map function and Newton solves to compute its inverse at the points of comparison. Computation of its gradient by finite differences required $S+1$ cost function evaluations.

The strategy was first set up as an unconstrained optimization using the Nelder-Mead non-gradient optimization method [57, 56], and the BFGS Quasi-Newton gradient-based method with mixed quadratic and cubic line searches [57, 58]. Penalty terms were added to the cost function in order to ensure the monotonicity of the grid. A second setup used two versions of a constrained SQP method [58] with grid monotonicity and stability constraints, one without, and one with user supplied gradients of the cost function and the constraints.

The expected advantage of the optimization-based grid selection strategy is reaching an optimal grid. The disadvantage is its complexity (highest among the considered strategies), and the lack of robustness due to the possibility of falling into local minima traps.

Applying the optimization-based strategy on sample waveforms produced inconsistent results. The runs that used the first cost function (magnitude of the last pseudo Fourier coefficient) re-distributed the grid points such that there was a sharp "dip" in the pseudo spectrum at the last frequency, and did not lead to a reduction in the truncation error, or clustering of the points in the regions with sharp transitions. The runs using the second cost function typically ended up trapped in a local minimum, and clustering of the grid points in regions that did not need it. The line searches often generated infeasible intermediate grids causing a breakdown in the optimization. Overall, the results were inconsistent, and the optimization-based strategy was not robust enough to be integrated in TMHB and used for TMHB runs.

5.6 Practical Issues and Limitations of the Grid Selection Strategies

Each of the presented grid selection strategies were developed and initially tested on single sample circuit waveforms with sharp features. This testing phase eliminated the optimization-based strategy due to inconsistent results. The direct and the iterative strategies proved robust and consistent enough to be integrated with the TMHB algorithm and applied to actual circuits.

When either the direct or iterative grid selection strategy is integrated with the TMHB method, the a priori nature of the strategy limits its effectiveness. The errors present in the computed shooting-Newton solution guess with a loose tolerance cause errors in the clustering of time-points

by the grid selection strategy. Recall that tightening the tolerance of the shooting-Newton run would cause its run-time to become a significant part of the total run-time for the TMHB method.

In particular, the inexact shooting-Newton solution guess has been observed to cause: (1) an unnecessary and an excessive increase in grid resolution leading to point exhaustion in the remainder of the grid domain; (2) an insufficient increase in grid resolution in certain regions; and (3) an erroneous time-shift in the positioning of the grid resolution increase (when the rapid transitions in the computed waveform are located at different times than in the actual steady-state waveform). These errors were observed to be more noticeable in larger circuits. All of these effects lead to less effective grids, and translate into a decreased ability of the TMHB method to achieve significantly better solution accuracy than the standard HB method.

The observation of the stability requirement (5.7) for bounding the neighboring time-steps, helps tremendously in producing a grid which is used to construct an effective smooth time-map function which observes the strict monotonicity requirement without a need for filtering. This in turn helps the Newton and GMRES iterations in the TMHB method to be efficient and to converge to the correct solution. The smoothing nature of this time-step bounding also helps in alleviating the aforementioned problems caused by the errors in the shooting-Newton guess, limiting the loss of solution accuracy.

When the circuit is reasonably large, the multiple transitions problem can present a significant setback to the grid selection strategies. In explanation, the many solution waveforms may have rapid transitions that occur at many different times throughout the solution domain $[0, T]$. The grid selection strategy is then required to generate a grid which increases and decreases its resolution many times throughout the solution domain. This is impossible to do with a limited number of points. In addition, the stability requirement limits the peak resolution increases. As a result, the strategies do not generate grids that lead to effective time-map functions.

In the case of direct grid selection strategy, decreasing the amount of smoothing by reducing the grid rigidity (stability) parameter α is one way to reduce this problem. In the course of our research, we tested this approach on several circuits. The reduction of the stability parameter α indeed in many cases lead to a more effective time-map function and increased TMHB accuracy. However, it was impossible to determine a universal near-optimal value for α across all circuits. Instead, a conservative and safe default value of $\alpha = 2$ was selected.

In the iterative grid selection strategy, one may consider using a different monitor function R. In particular, the waveform weights F_j can be (1) all set to a constant value; (2) equal to one constant for the voltage and another (much smaller) constant for the current waveforms; (3) set to the maximal peak-to-peak amplitude among voltage and current waveforms separately, with the

current peak-to-peak amplitude scaled by a constant factor. One can also use a different norm, such as the L_2 norm.

These differing scaling and averaging schemes for the monitor function end up emphasizing different waveforms in guiding the iterations of the iterative grid selection, and lead to different final grids. Unfortunately, if the circuit is large, it is hard to evaluate whether the right waveforms are emphasized. Sometimes the waveforms having the steepest features can in fact have a very small peak-to-peak amplitude. This indicates that the waveform is perhaps simply a noisy DC voltage, whose high frequency Fourier coefficients are so small that they can be of the order of the accuracy of the computation. Such a waveform is not useful in the computation of the monitor function and may cause less effective grid selection. These waveforms can be identified by setting a threshold value for the peak-to-peak amplitude below which the waveform is ignored in computing the monitor function. Choosing this threshold value is, unfortunately, circuit dependent.

In practice, all these hands-on monitor function schemes worked very well for a particular circuit, but poorly for the remaining circuits. The selected default for a monitor function (5.10) uses an L_∞ norm and waveform weights set to the maximal peak-to-peak amplitude among all voltage and current waveforms. It is a conservative choice that does not produce the best results in individual circuits, but is universally good for all considered circuits.

Note that the interpolation of the solution guess waveforms to obtain in-between values in the iterative grid selection strategy did not show to lessen the effectiveness of this strategy. In particular, both linear and quadratic interpolation work equally well for this purpose.

The direct iterative strategy also sometimes showed some sensitivity to the integration method used in the shooting-Newton run. In particular, a low second order (trapezoidal or BDF-2) scheme sometimes can produce a large constant error in the TMHB solution. One way to eliminate this error is to switch to using a higher order integration scheme.

Chapter 6

Time-Mapped Harmonic Balance Results

In this Chapter we compare the performance of the TMHB method with standard state-of-the-art matrix-implicit Krylov-subspace harmonic balance [7, 10, 11, 15] on several circuits.

The TMHB method is meant to be used on circuits whose solution waveforms undergo rapid transitions. Most highly nonlinear circuits will exhibit such waveforms. If the circuit is linear and driven by a sinusoidal source, all signals in the circuit will also be pure sinusoids. In such a case, the standard HB algorithm gives the exact solution since the truncation errors for the standard Fourier series approximations of all solution waveforms are all zero for $M \geq 1$ (see Chapter 2).

Note that the rate of change in the pure sinusoids in a linear circuit is not constant throughout the solution domain $[0, T]$. As a result, the TMHB grid selection algorithms will in general generate mildly non-uniform grid for linear circuits, and the pseudo Fourier series representation of the TMHB solution waveforms will have a non-zero truncation error. Similarly, we can deduce that for mildly nonlinear circuits, i.e. circuits with smoothly varying waveforms, TMHB will not be in general more advantageous than the standard HB method.

Four strongly nonlinear circuits were simulated with the HB and TMHB methods: a diode rectifier, a DC-DC converter, a BiCMOS switching mixer, and a BiCMOS IF preamplifier circuit driven into distortion. Equation and element statistics for these circuits are given in Table 6.1.

Both the standard HB and TMHB methods, as well as the direct and iterative grid selection strategies were implemented in Mica, Motorola's SPICE-like circuit simulator, in the computer language C [55]. All computer runs were done on Sun Ultra-2 300MHz UNIX workstations.

Both the standard HB and the TMHB methods in all runs used the same shooting-Newton solution guess (see Section 5.2) as an initial guess for the Newton iterations. Two variants of the TMHB method were considered: TMHB-D, utilizing the direct grid selection strategy, and TMHB-I, using the iterative grid selection strategy. The grid rigidity parameter α in the grid

Circuit	N	R	C	L	VSRC	ISRC	VCVS	DIO	BJT	MOS	TOTAL
Diode Rectifier	**6**	2	3	1	1			1			8
DC-DC Converter	**9**	4	2	1	2			1	1		11
Switching Mixer	**105**	39	47	2	3		2	14	10	5	122
IF Preamplifier	**289**	138	147		6	1			26	25	343

Table 6.1: Circuit statistics: N is the number of circuit equations; R, C, L numbers of resistors, capacitors, and inductors; VSRC, ISRC, and VCVS numbers of voltage, current, and voltage-controlled voltage sources; DIO number of diodes; BJT and MOS numbers of bipolar and MOS transistors; TOTAL the total number of elements.

selection schemes was fixed at the default value $\alpha = 2$, and the number of grid points S in the iterative scheme was $S = 50$.

The "exact" solution for each of the circuits was computed using a standard HB method with a very large number of harmonics (as permitted by the memory limits of the computer hardware). In particular, the number of harmonics used in the exact HB run was $K = 5000$ for the diode rectifier and the DC-DC converter, and $K = 1000$ for the switching mixer and IF preamplifier circuits.

6.1 Diode Rectifier

The first circuit is a simple but strongly nonlinear diode rectifier, shown in Figure 6.1. The circuit is powered by a 50Hz sinusoidal voltage source $v_{in}(t)$. The diode rectifier was first simulated with both the standard HB and the TMHB-I method at same number of harmonics, $K = 10$. Plots of two solution waveforms exhibiting rapid transitions, $i_{VIN}(t)$ and $v_3(t)$, are given in Figures 6.2 and 6.3 respectively. It can be seen that the TMHB solution waveforms more closely match the exact solution than the standard HB solution waveforms. In other words, even at this small number of harmonics, the TMHB method computes a more accurate solution (about 10dB more accurate in L_∞ norm sense) than the standard HB method.

The pointwise errors in a computed solution waveform $v(t)$ or in the computed Fourier coefficients V_k of its approximation, in the time and frequency domains respectively, are also of interest. The time-domain pointwise error ε_t is computed as:

$$\varepsilon_t(t) = |v^*(t) - v(t)| \tag{6.1}$$

where $v^*(t)$ is the exact solution waveform, and $v(t)$ is the waveform computed by the HB or the

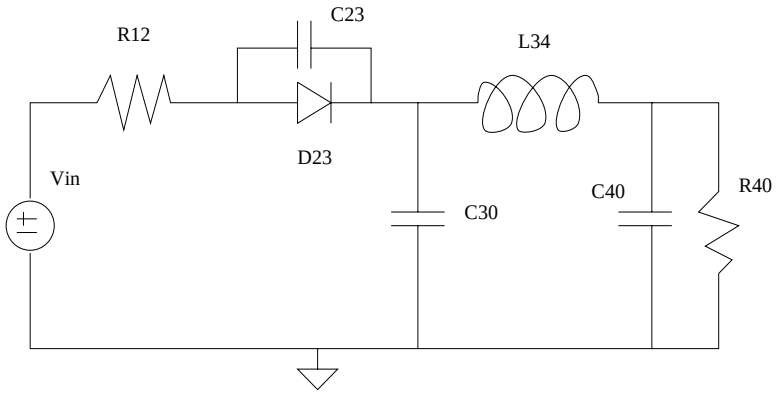

Figure 6.1: Diode rectifier circuit.

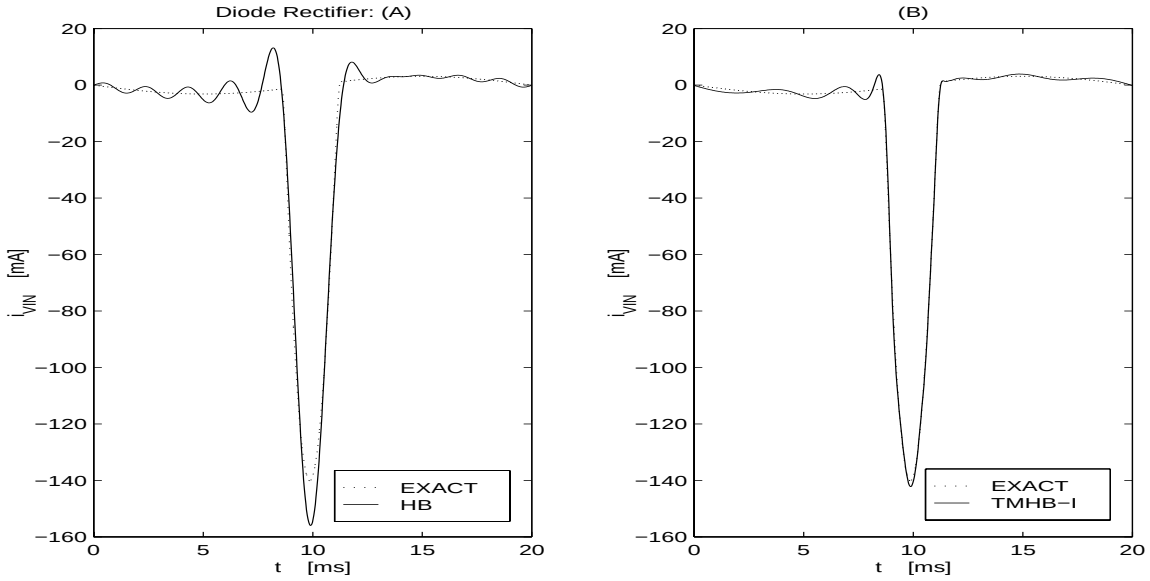

Figure 6.2: Diode rectifier circuit, i_{VIN} computed with: (A) standard HB; (B) TMHB-I, at same number of harmonics $K = 10$.

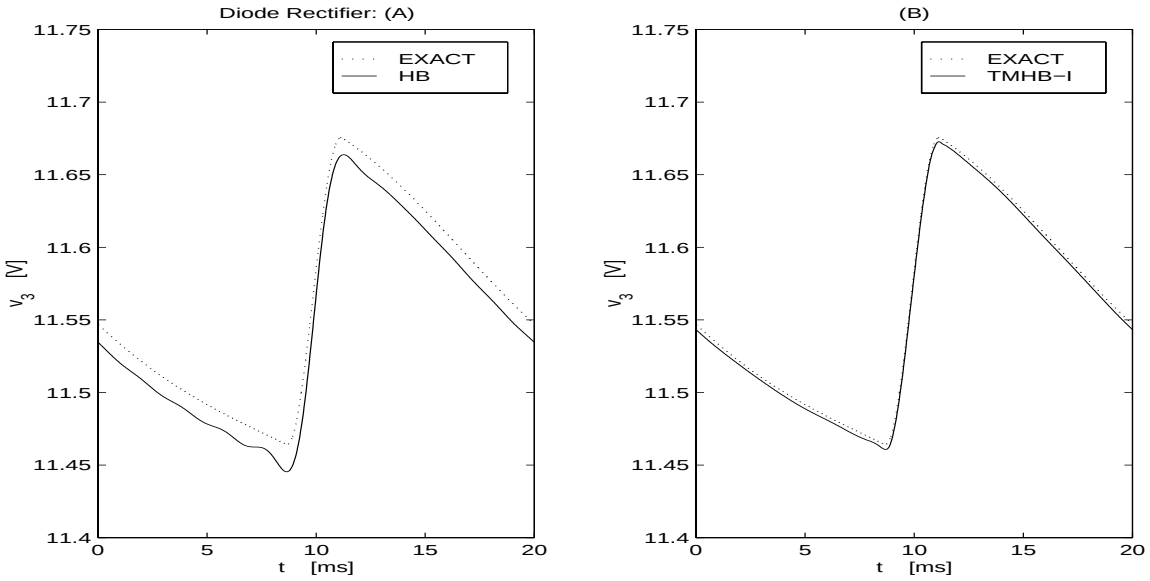

Figure 6.3: Diode rectifier circuit, v_3 computed with: (A) standard HB; (B) TMHB-I, at same number of harmonics $K = 10$.

TMHB method. The frequency-domain pointwise error ε_f is computed as:

$$\varepsilon_f(kf) = |V_k^* - V_k| \tag{6.2}$$

where V_k^* is the k-th Fourier coefficient of the exact solution, and V_k is the k-th Fourier coefficient of the solution computed by HB or TMHB.

Figure 6.4 shows plots the pointwise errors in the computed $i_{VIN}(t)$ solution waveforms by the HB and TMHB methods in the time and frequency domain, at same number of harmonics $K = 10$. In the first plot it is visible how the TMHB method diminishes the peaks in the time domain error in the middle of the solution interval, which is where the waveforms exhibit the rapid transitions. The second plot demonstrates that the TMHB computes each Fourier coefficient more accurately than the standard HB.

The circuit was next repeatedly simulated with both the HB and the TMHB methods for an increasing number of harmonics K. Figure 6.5 shows the L_∞ norm of the frequency-domain pointwise error ε_f, in dB, for the computed Fourier coefficients of the i_{VIN} waveform, versus the number of harmonics K. The plot shows orders of magnitude improvement in the accuracy of the TMHB solutions compared to the standard HB solution. For example, at $K = 220$ the TMHB solution is about 60dB or 3 orders of magnitude more accurate than the standard HB solution.

Moreover, Figure 6.5 demonstrates the superior error convergence properties of the TMHB. The vertical distance between the HB and TMHB convergence curves widens linearly in log space

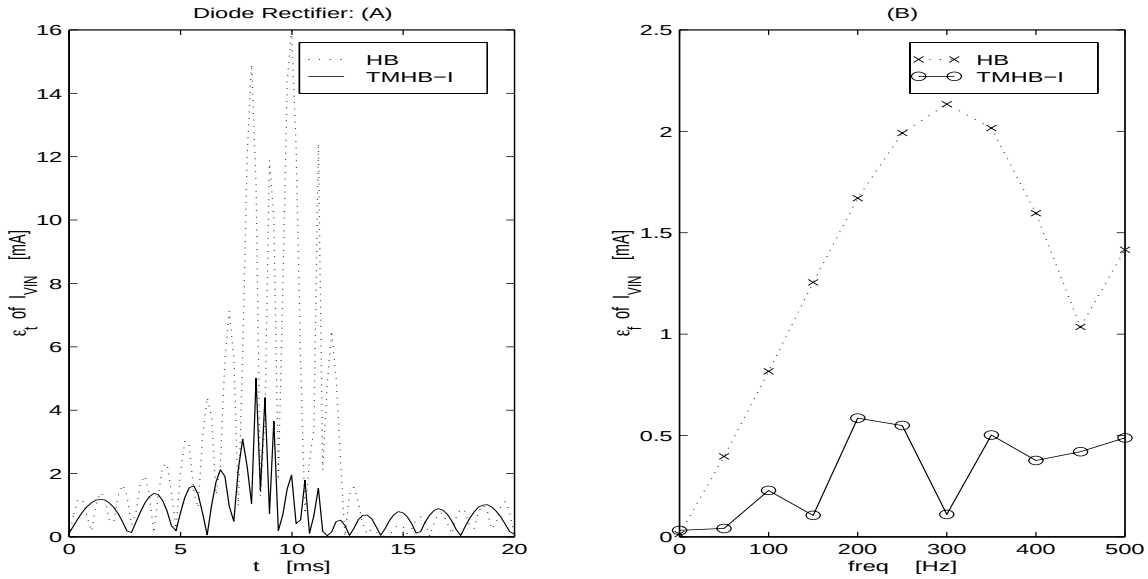

Figure 6.4: Diode rectifier circuit, pointwise errors in i_{VIN}: (A) time-domain errors; (B) frequency domain errors. Both HB and TMHB-I runs used $K = 10$ harmonics.

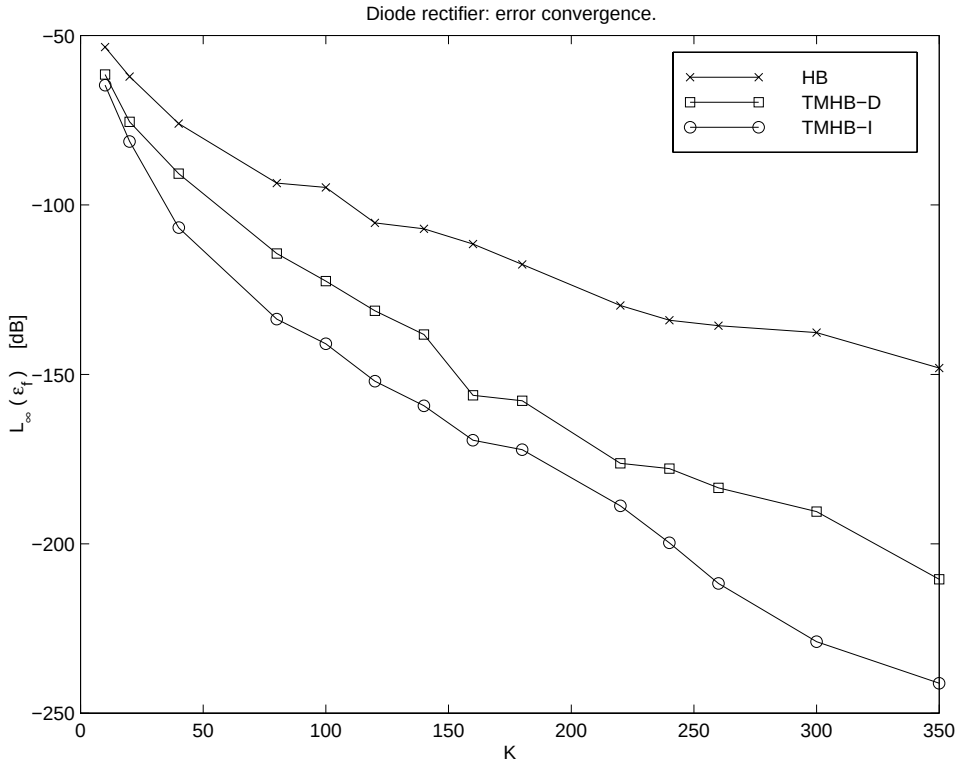

Figure 6.5: Diode rectifier circuit, TMHB error convergence: L_∞ norm of the frequency-domain pointwise error in i_{VIN}, in dB.

- i.e. exponentially fast in linear space with increasing number of harmonics K. This is a confirmation of our expectations for the error convergence properties of the pseudo Fourier series approximations and the TMHB method. The TMHB error decreases with $O(M^{-\hat{p}})$ versus the $O(M^{-p})$ error convergence of the standard HB method, with $\hat{p} > p$.

The plot also indicates that the iterative grid selection strategy in TMHB-I produces a better time-map function and helps the TMHB algorithm reduce the solution error more than TMHB-D which uses the direct selection strategy.

Note that an even more superior convergence profile is obtained if a different monitor function is used in the iterative grid selection. In particular, using an L_2 norm in the monitor function

$$R_m = \left(1 + \frac{T}{t_{m+1} - t_m} \sum_{j=1}^{N} \left(\frac{v_j(t_{m+1}) - v_j(t_m)}{F_j}\right)^2\right)^{\frac{1}{2}} \tag{6.3}$$

and setting $F_j = 1$ for the voltage waveforms and $F_j = 10^{-3}$ for the current waveforms (effectively scaling up the current waveforms), produces a better non-uniform grid and time-map function in the iterative strategy. By scaling up the currents the fast-varying i_{VIN} waveform plays a much greater role in the distribution of the time-points. The net result is that, for example, at $K = 220$, the TMHB-I method using the new monitor function achieves an additional 2 orders of magnitude gain over the previous result, effectively computing the solution 5 orders of magnitude (100dB) more accurately than the standard HB method.

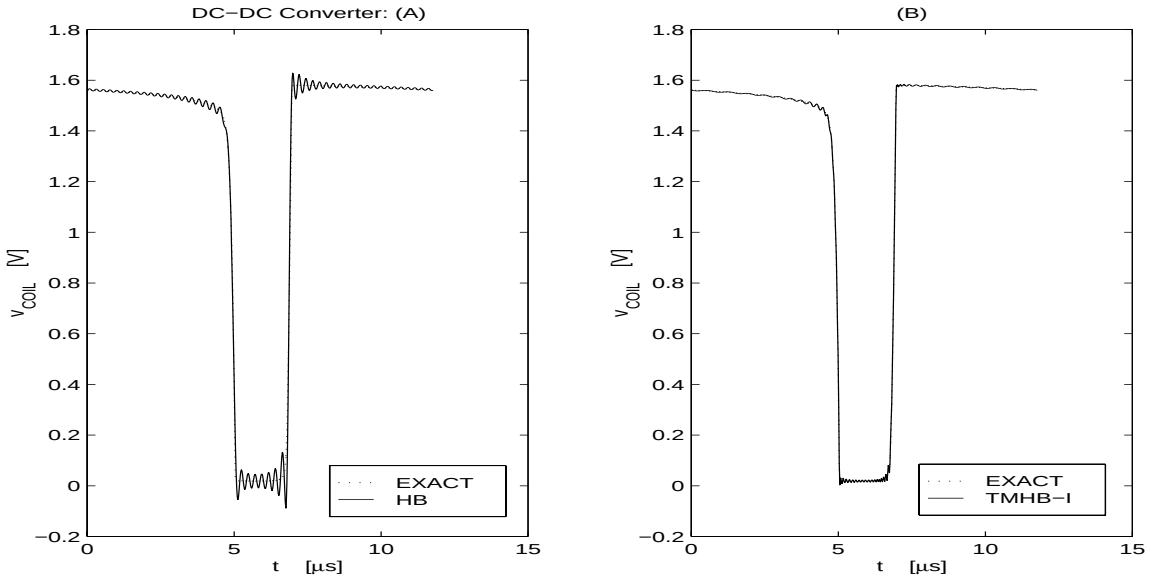

Figure 6.6: DC-DC converter circuit, v_{COIL} computed with: (A) standard HB; (B) TMHB-I, at same number of harmonics $K = 50$.

6.2 DC-DC Converter

The second simulated circuit is a DC-DC converter, powered with an 85kHz input. This circuit was first simulated with the standard HB and the TMHB-I method, both using $K = 50$ harmonics. Plots of the computed solution waveform $v_{COIL}(t)$ is given in Figure 6.6. The plots illustrate the 20dB improvement in L_∞ norm accuracy using the TMHB-I method at this number of harmonics.

The time and frequency domain pointwise error in v_{COIL} as computed by HB and TMHB-I methods at the same number of harmonics $K = 50$, are shown in Figure 6.7. It can be seen from the first plot how the TMHB method diminishes the error peaks at the times of the two sharp edges in the waveform, i.e. lessens the prominence of the Gibbs effect due to its use of the pseudo Fourier approximations. In the frequency domain (second plot), TMHB computes each individual Fourier coefficient with a much smaller error than the standard HB method. The improvement in accuracy is more pronounced for the Fourier coefficients corresponding to the higher frequencies - these are the coefficients of the Fourier basis functions that are used to "build" the sharpest features in the waveform, such as the two rapid transitions in v_{COIL}.

The circuit was next repeatedly simulated for increasing numbers of harmonics with both HB and TMHB methods to capture the asymptotic behavior of the error. Figure 6.8 plots the L_∞ norm of the frequency domain error in the v_{COIL} waveform, versus the number of harmonics K. The plot is another confirmation of the superior error convergence properties of the TMHB method. At $K = 600$ the TMHB-I error is 100dB(5 orders of magnitude) smaller than the standard HB error.

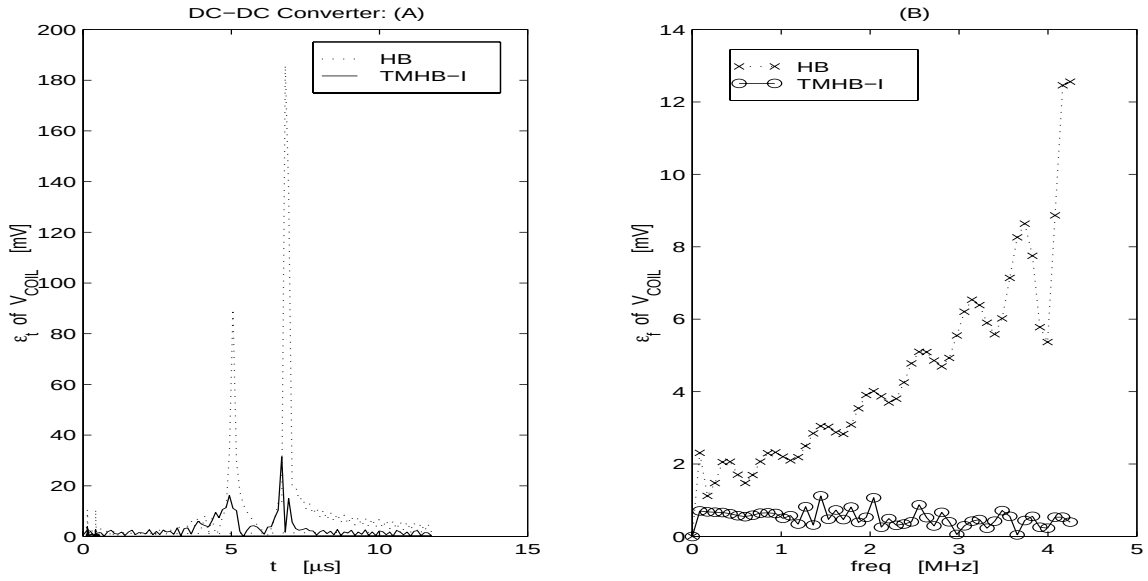

Figure 6.7: DC-DC Converter circuit, pointwise errors in v_{COIL}: (A) time-domain errors; (B) frequency domain errors. Both HB and TMHB-I runs used $K = 50$ harmonics.

Note that the L_∞ norm error as well as the errors for each individual harmonic for each of the remaining waveforms in the circuit again show similar superior convergence properties. Figure 6.8 also indicates that the TMHB-I method is more successful at simulating the DC-DC converter than the TMHB-D method.

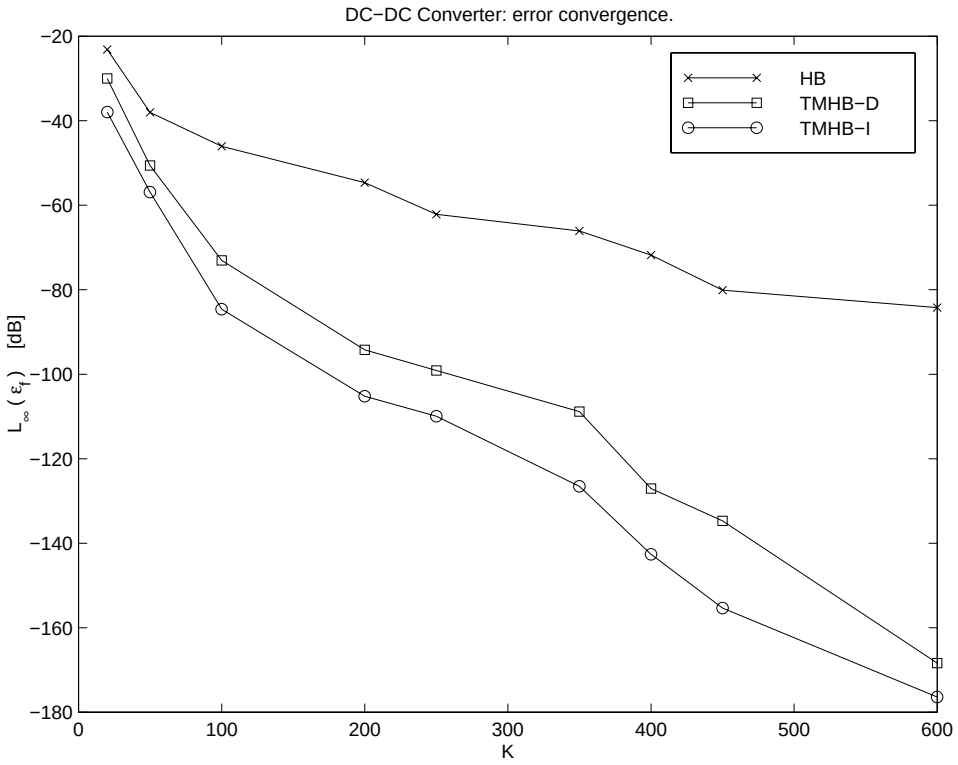

Figure 6.8: DC-DC Converter circuit, TMHB error convergence: L_∞ norm of the frequency-domain pointwise error in v_{COIL}, in dB.

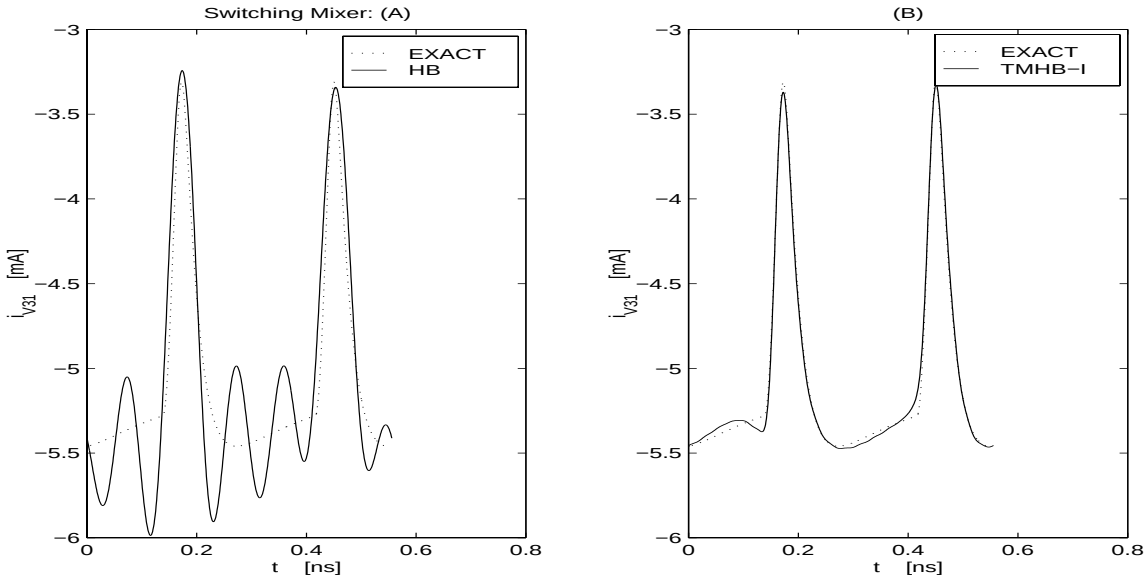

Figure 6.9: Switching Mixer circuit, i_{V31} computed with: (A) standard HB; (B) TMHB-I, at same number of harmonics $K = 6$.

6.3 Switching Mixer

The third considered circuit is a Gilbert cell switching mixer in BiCMOS technology. Its LO (local oscillator source) was a 1.8GHz square wave, with a rising edge at $\frac{T}{4}$, a falling edge at $\frac{3T}{4}$, and both rise and fall times for the edges equal to 5% of T. For the simulation experiments the RF inputs were kept at zero. The idea was to obtain a transient steady state operating point with high accuracy.

The mixer was first simulated with both the standard HB and TMHB methods at a very small number of harmonics $K = 6$. Since the circuit is relatively large, larger numbers of harmonics not only increase run times, but can also cause out of memory problems in computer systems with insufficient RAM. It was therefore of particular interest to see whether the TMHB can compute a significantly more accurate solution than the HB method when only a few harmonics are considered.

Plots comparing the standard HB and TMHB-I computed $i_{V31}(t)$ waveforms with the exact solution are given in Figure 6.9. The TMHB-I solution is dramatically better than the standard HB solution despite the use of only $K = 6$ harmonics (or equivalently, $M = 2K + 1 = 13$ time points). The L_∞ norm of the error for the TMHB method was 18dB lower than the standard HB.

The mixer was next simulated with HB and TMHB methods at $K = 50$ harmonics. Figure 6.10 shows plots of the time and frequency domain pointwise errors in i_{V31}. The time domain error plot shows the peaking of the errors around the times when the waveform has large gradients. The

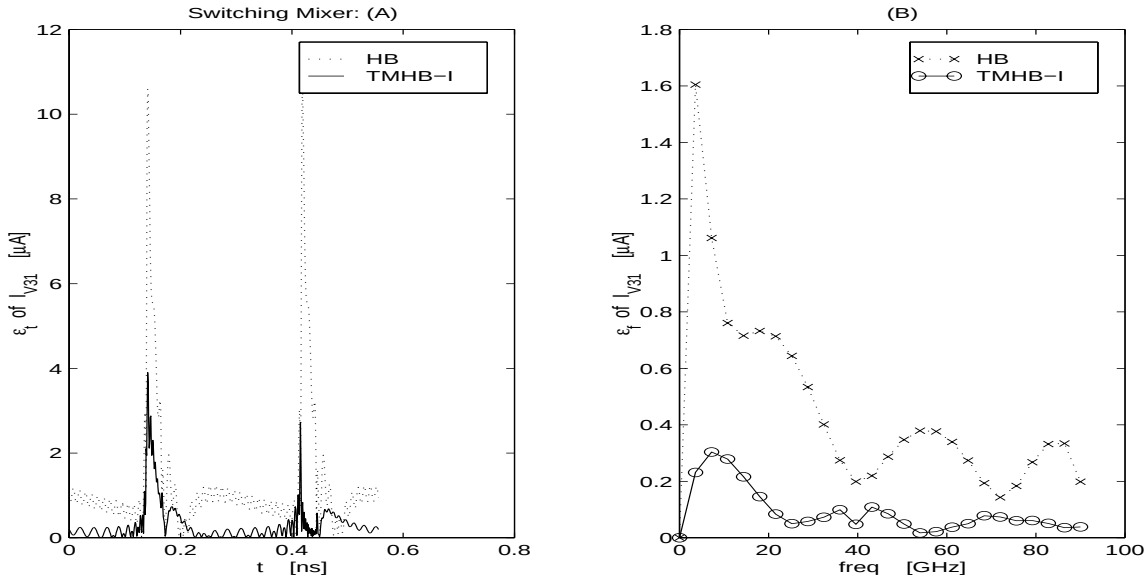

Figure 6.10: Switching Mixer circuit, pointwise errors in i_{V31}: (A) time-domain errors; (B) frequency domain errors. Both HB and TMHB-I runs used $K = 50$ harmonics.

TMHB method is seen to reduce this maximal error, as well as reduce the error throughout the rest of the solution domain. The second, frequency domain error plot shows again how the TMHB computes each individual Fourier coefficients more accurately than the standard HB method. Note that only the coefficients with even k are plotted as the odd k coefficients are all zero (the waveform has a frequency equal to twice the excitation frequency). It is interesting to note that the greatest error reduction is now for the low frequency Fourier coefficients.

Our final simulation experiment for the switching mixer circuit consists of repeated HB and TMHB runs for increasing numbers of harmonics K. Figure 6.11 summarizes the results. While the TMHB error convergence is better than the standard HB, it is not as impressive as in the case of the DC-DC converter circuit. As it was discussed in Section 5.6, larger circuits present several special problems to the grid selection algorithms due to the large number of waveforms. In the case of the switching mixer, the multiple transitions problem, as well as the erroneous time-shift in the transitions in the computed shooting-Newton guess have been found to be the cause of the somewhat limited success of TMHB in this circuit.

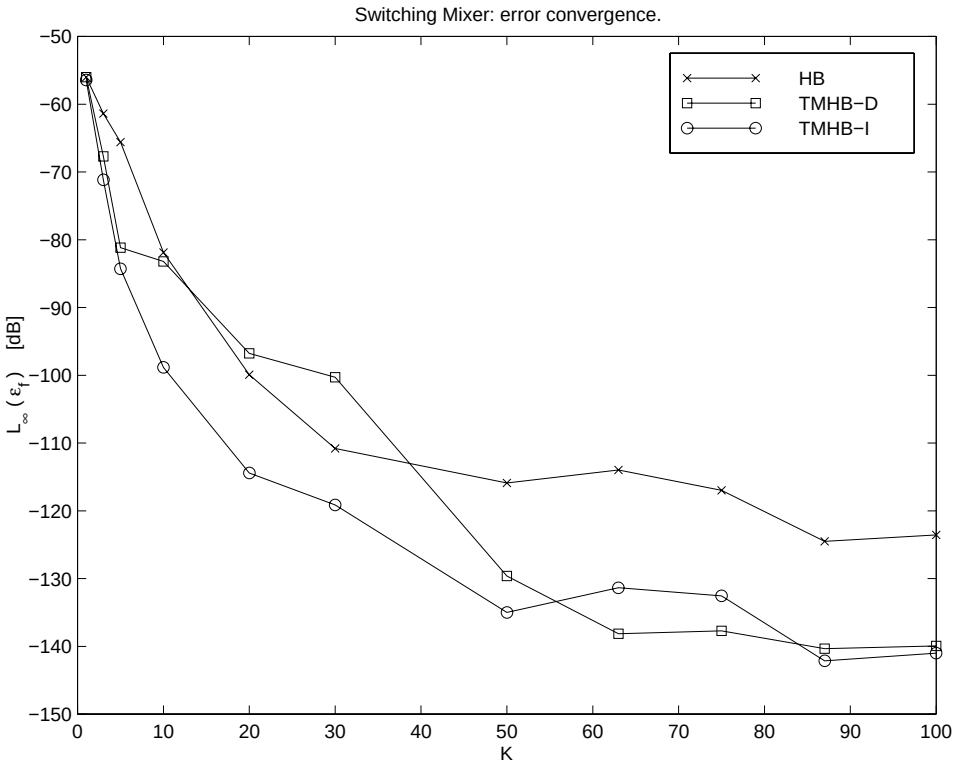

Figure 6.11: Switching Mixer circuit, TMHB error convergence: L_∞ norm of the frequency-domain pointwise error in i_{V31}, in dB.

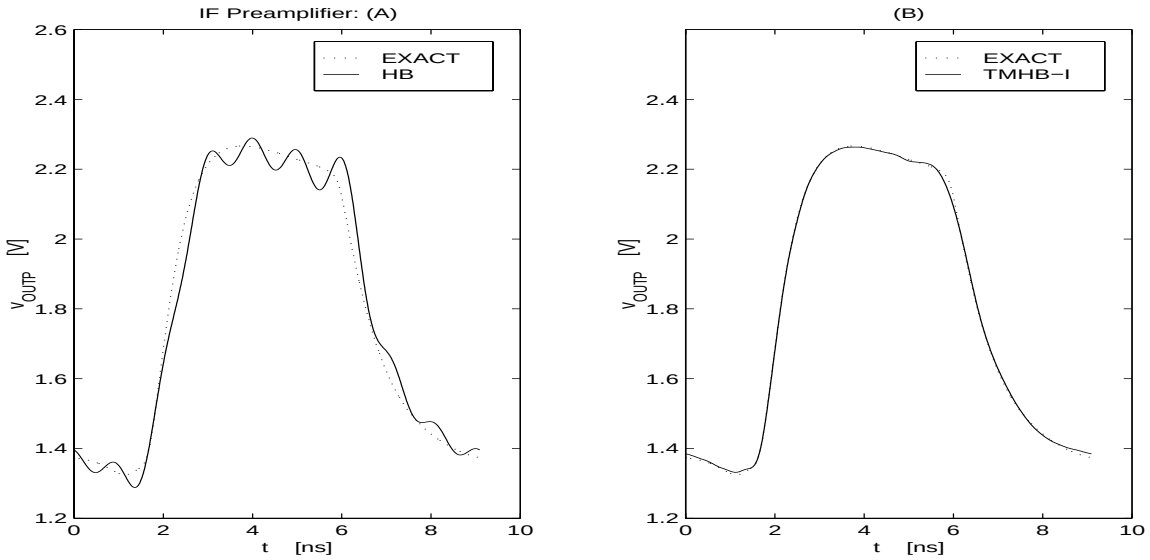

Figure 6.12: IF Preamplifier circuit, v_{OUTP} computed with: (A) standard HB; (B) TMHB-I, at same number of harmonics $K = 9$.

6.4 IF Preamplifier

The fourth and final simulated circuit is a BiCMOS IF (intermediate frequency) preamplifier, which was driven into distortion with a 0.1V peak-to-peak 110MHz sinusoidal source. This circuit generated the largest number of equations $N = 289$. The first simulation experiment was a standard HB and a TMHB run both at $K = 9$ harmonics. As in the case of the switching mixer, due to the size of the circuit, it was of particular interest to see if TMHB is a superior method for small number of harmonics.

Plots comparing the distorted output solution waveform v_{OUTP} for the standard HB method and the TMHB-I method are shown in Figure 6.12. The L_∞ norm of the error of the TMHB method was 19 dB lower than the standard HB, which these plots illustrate quite well qualitatively. Figure 6.13 shows plots of the time and frequency domain pointwise errors in v_{OUTP}. A reduction of the error in each computed Fourier coefficient using the TMHB method is evident from the frequency domain plot.

The final set of simulation experiments was a repeated sequence of HB and TMHB runs for increasing number of harmonics K. A summary of the results is given in Figure 6.14. The plots show that the TMHB-I method has the best error convergence profile. The size of the circuit, as in the case of the switching mixer, caused some problems to the grid selection strategies, which is why the TMHB convergence profiles are not quite as spectacular as in the case of the two smaller circuits, the diode rectifier and DC-DC converter. In particular, the multiple transitions

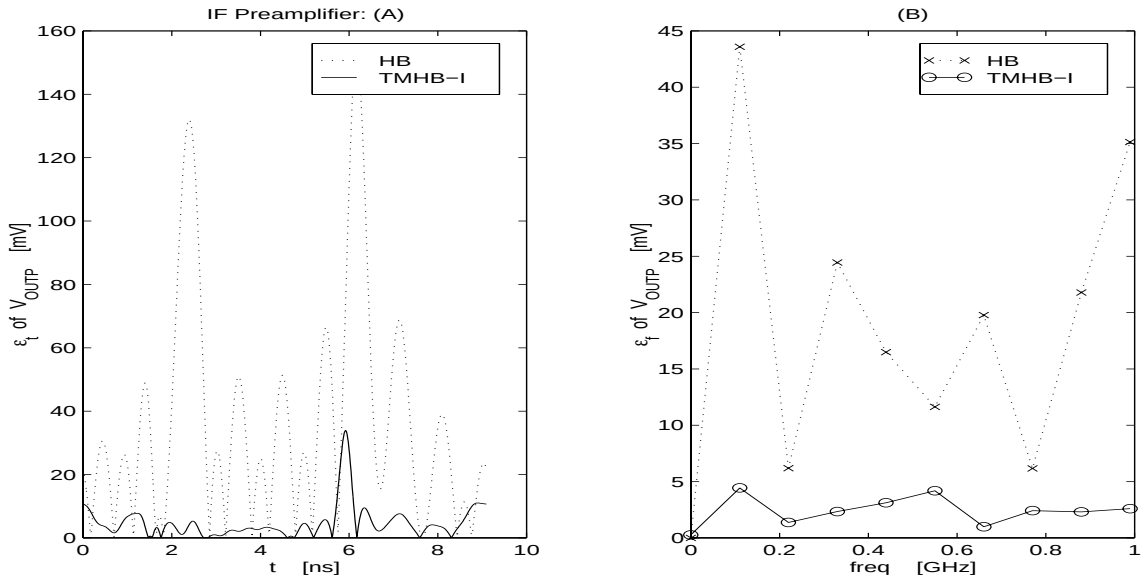

Figure 6.13: IF Preamplifier circuit, pointwise errors in v_{OUTP}: (A) time-domain errors; (B) frequency domain errors. Both HB and TMHB-I runs used $K = 9$ harmonics.

problems was found to be a cause in the degraded performance of both TMHB-D and TMHB-I. In addition, in the case of TMHB-D, using the a low second order (trapezoidal) integration method in the shooting-Newton solution guess run caused a large constant error which is in evidence in Figure 6.14. This problem was resolved when a higher order scheme (BDF-5) was used, as seen from the plot.

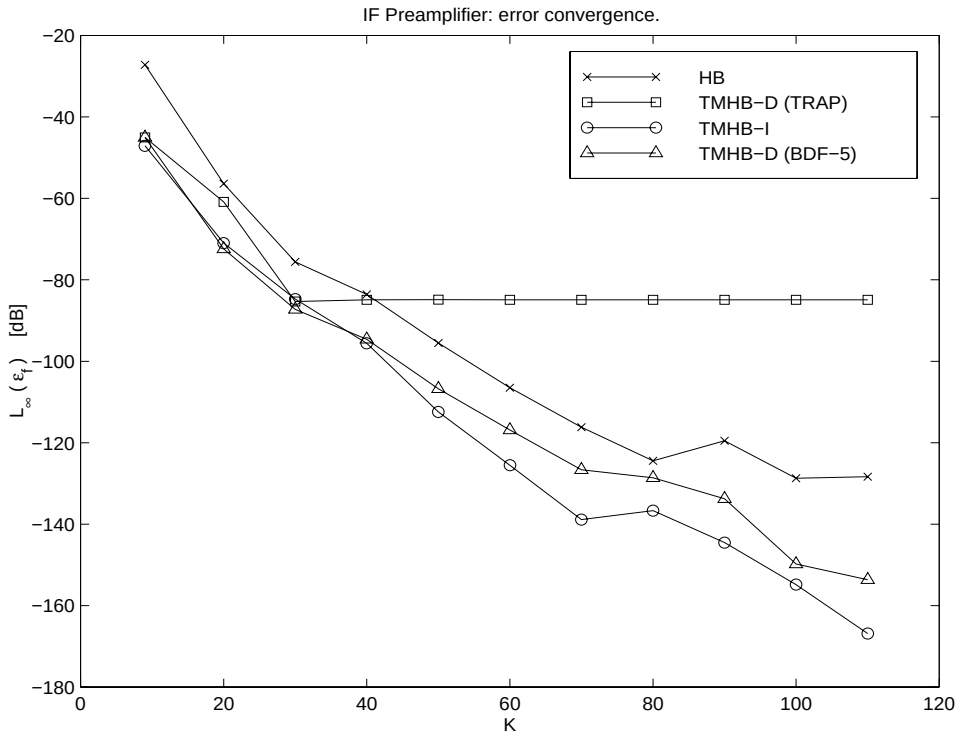

Figure 6.14: IF Preamplifier circuit, TMHB error convergence: L_∞ norm of the frequency-domain pointwise error in v_{OUTP}, in dB.

6.5 Runtime Efficiency and Storage Requirements of TMHB

A logical way to measure the runtime efficiency of the TMHB method is to compare standard HB and TMHB runs achieving same accuracy in the solution. Tables 6.2 and 6.3 summarize these findings. The accuracy E was the L_∞ norm of the frequency domain pointwise error in the computed Fourier coefficients for the waveforms used for the error convergence profile plots in the previous Sections.

The total CPU times include the time spent in the grid selection strategy, as well as a complete unmap of all solution waveforms in the circuit. The simpler direct grid selection strategy is much faster than the iterative grid selection strategy. On the other hand, the cost of the iterative grid selection strategy is independent of the number of harmonics M, and linearly depends only on the selected size of the iterated grid S (which was kept fixed at $S = 50$ in our experiments) and the size of the circuit N.

A complete unmap of all solution waveforms, as discussed in Subsection 4.3.1, is in general unnecessary in practice as only a few waveforms are of interest. A partial unmap of only the few needed waveforms can generate significant total CPU time savings for larger circuits with hundreds of waveforms.

		Standard HB					TMHB-D				
Circuit	E	K	T	T_L	I_L	I_N	K	T	T_L	I_L	I_N
Diode Rectifier	-200	650	**43.2**	33.0	273	16	330	**40.8**	21.1	243	16
DC-DC Converter	-100	1000	**1080**	1053	2487	14	250	**1740**	1658	7160	12
Switching Mixer	-130	150	**67.3**	21.8	37	8	50	**56.4**	22.7	88	9
IF Preamplifier	-155	170	**1065**	861	417	18	100	**626**	509	370	15

Table 6.2: Comparison of the standard HB and TMHB-D methods at same achieved solution accuracy. E is the achieved accuracy in dB. K is number of harmonics, T is total CPU time, T_L is linear solve time, I_L is number of GMRES iterations, I_N is number of Newton iterations. All times are in seconds.

Table 6.2 compares the runtime statistics for the standard HB runs and the TMHB-D runs achieving identical solution accuracy. It can be seen that the TMHB-D is comparable in efficiency to the standard HB method for the diode rectifier circuit, while it is less efficient for the DC-DC converter. This is caused by the larger number of GMRES linear iterations associated with the TMHB-D method.

The cause of this increase in linear iterations was traced to the nature of the time-map functions $\lambda(\hat{t})$ generated by the direct grid selection strategy. In particular, this time-map function increases the grid density in the regions of rapid transitions to a much larger degree than in the case of the time-map function generated by the iterative grid selection strategy. The very large increase in grid density corresponds to the "flat" segments of $\lambda(\hat{t})$, i.e. to a very small first derivative $\lambda'(\hat{t})$ of the time-map function. The large spread in these derivative values leads to a large condition number of the diagonal matrix of these derivatives Λ, which in turn causes condition number increase in the Jacobian matrix J for the linear problem in TMHB.

The situation improves for the larger circuits, the switching mixer and the IF preamplifier. Due to the multiple transitions problem in larger circuits (see Section 5.6), the time-map function constructed from the non-uniform grid generated by the direct strategy does not yield excessive grid density increases as in the case of the two smaller circuits. For the IF preamplifier, the TMHB-D method (using a BDF-5 integration method for the shooting-Newton solution guess run) is 1.7 times faster than standard HB.

		Standard HB					TMHB-I				
Circuit	E	K	T	T_L	I_L	I_N	K	T	T_L	I_L	I_N
Diode Rectifier	-200	650	**43.2**	33.0	273	16	240	**27.2**	6.84	187	14
DC-DC Converter	-100	1000	**1080**	1053	2487	14	180	**177**	156	2112	12
Switching Mixer	-130	150	**67.3**	21.8	37	8	45	**62.8**	13.1	73	9
IF Preamplifier	-155	170	**1065**	861	417	18	90	**662**	514	441	17

Table 6.3: Comparison of the standard HB and TMHB-I methods at same achieved solution accuracy. E is the achieved accuracy in dB. K is number of harmonics, T is total CPU time, T_L is linear solve time, I_L is number of GMRES iterations, I_N is number of Newton iterations. All times are in seconds.

Table 6.3 compares the runtime statistics for the standard HB runs and the TMHB-I runs achieving identical solution accuracy. Significant runtime speedups are seen for three of the four simulated circuits. For both the diode rectifier and the IF preamplifier, a speedup of 1.6 is achieved. For the DC-DC converter the speedup is a factor of 6.

The total CPU times T for the HB and TMHB-I methods in reaching a specific accuracy in v_{COIL} from the DC-DC converter circuit are shown in Figure 6.15. The accuracy measure was again the L_∞ norm of the frequency domain pointwise error in the computed Fourier coefficients. For less stringent accuracies , the total CPU times for the TMHB method are comparable to the

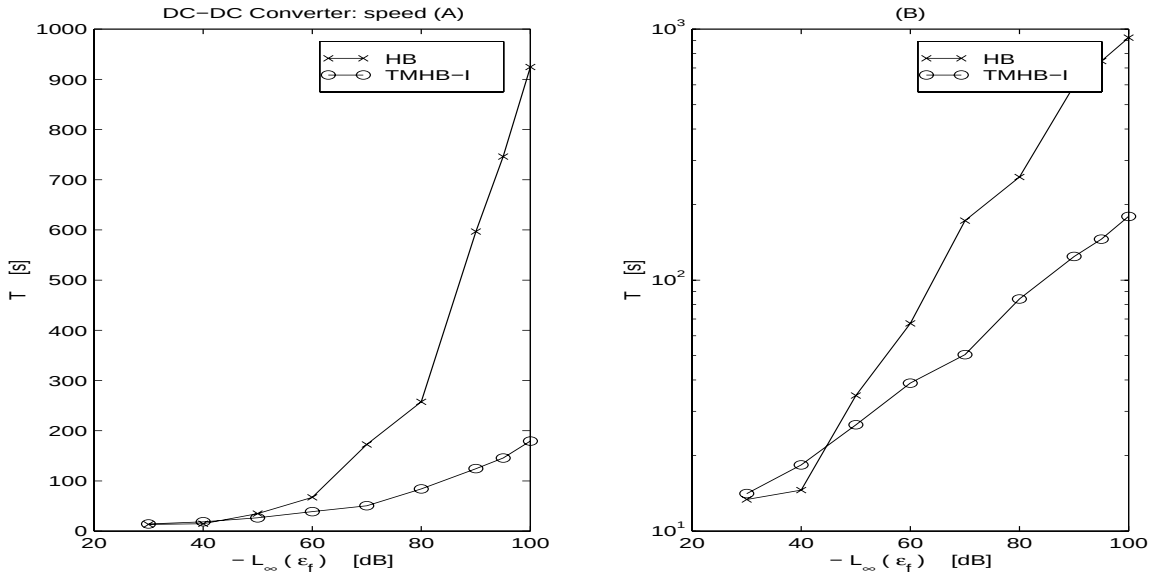

Figure 6.15: DC-DC Converter circuit: total CPU time T for HB and TMHB-I to reach a specific solution accuracy in v_{COIL}. (A) linear plot; (B) log plot.

HB CPU times due to the TMHB overhead in grid selection and waveform unmap. The situation is drastically different for accuracies better than -50dB: the TMHB becomes up to several times faster than the HB method. In addition the speedup factor grows with increases in required accuracies.

The TMHB is expected to retain the $O(NM\log M)$ complexity of standard HB, while reducing the error $O\left(\left(\frac{1}{M}\right)^{\hat{p}}\right)$ versus the $O\left(\left(\frac{1}{M}\right)^{p}\right)$ error convergence of the standard HB method, with $\hat{p} > p$. The roughly linear dependence of the CPU time T on M, means that $\log(T)$ will roughly behave as $\log(M)$. Therefore, when $log(T)$ is plotted against the error in dB which has dependence $\log\left(\frac{1}{M}^{p}\right) = -p\log(M)$, the plot is a straight line with a slope proportional to p. The steeper slope of the $log(T)$ line corresponding to TMHB in plot (B) of Figure 6.15 confirms the complexity and convergence properties of TMHB.

The memory storage requirements for the TMHB method are the same as for the standard HB method, growing linearly with M due to the storage of the Krylov subspace vectors in the GMRES linear solver. Since the TMHB method can achieve same solution accuracy as the standard HB method with a smaller number of harmonics, it follows that significant memory savings can be achieved by using the TMHB method. In particular, from Table 6.3, we can measure the memory savings roughly as the ratio of the needed numbers of harmonics K for the standard HB and the TMHB-I method respectively. For example, the memory savings range from a factor of 1.9 for the IF preamplifier, to a factor of 5.5 for the DC-DC converter.

Figure 6.16 shows the required numbers of harmonics K needed by the HB and the TMHB-I

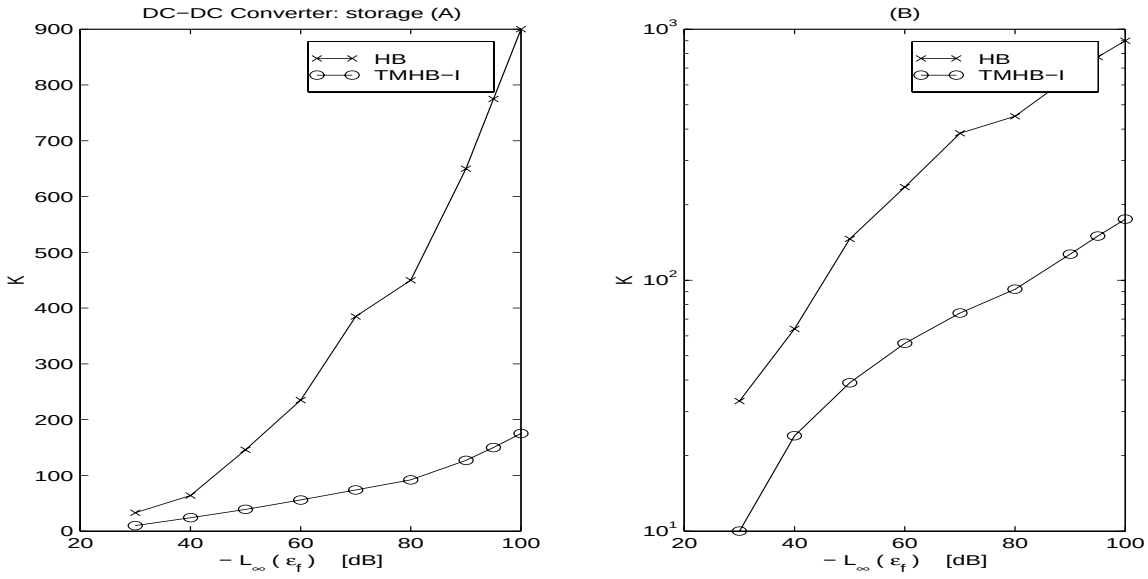

Figure 6.16: DC-DC Converter circuit: numbers of harmonics K required of HB and TMHB-I to reach a specific solution accuracy in v_{COIL}. (A) linear plot; (B) log plot.

methods, versus the reached accuracy in the v_{COIL} waveform for the DC-DC Converter circuit. Since the storage requirements are proportional to K the plot demonstrates that the TMHB method storage requirements at same solution accuracy are not only smaller than those of the HB, but also grow less rapidly for higher accuracy computations.

We can therefore conclude that TMHB is particularly well suited for high accuracy simulations of large memory hungry circuits.

Chapter 7

Conclusions

Harmonic balance (HB) methods are the frequency-domain algorithms of choice for high accuracy computation of the periodic steady-state of circuits. Matrix-implicit Krylov-subspace techniques have made it possible for these methods to simulate large circuits more efficiently. However, the harmonic balance methods are not so efficient in computing steady-state solutions of strongly nonlinear circuits with rapid transitions. While the time-domain shooting-Newton methods can handle these problems, the low-order integration methods typically used with shooting-Newton methods are inefficient when high solution accuracy is required.

In this dissertation, we embarked on the quest for better, more powerful spectral methods for circuit analysis.

7.1 Contributions of Thesis

We first examined possible enhancements to the standard state-of-the-art preconditioned matrix-implicit Krylov-subspace HB method. We formulated the BDF time-domain preconditioners and showed that they can be quite effective for strongly nonlinear circuits, speeding up the HB runtimes by several times compared to using the frequency-domain block-diagonal preconditioner. Also, an approximate Galerkin HB formulation was derived, yielding a small improvement in accuracy over the standard pseudospectral HB formulation, and about a factor of 1.5 runtime speedup in runs reaching identical solution error.

Next, we introduced and developed the Time-Mapped Harmonic Balance method (TMHB) as a fast Krylov-subspace spectral method that overcomes the inefficiency of standard harmonic balance for circuits with rapid transitions. TMHB features a non-uniform grid and a time-map function to resolve the sharp features in the signals. At the core of the TMHB method is the

notion of pseudo Fourier approximations. The rapid transitions in the solution waveforms are well approximated with pseudo Fourier interpolants, whose building blocks are complex exponential basis functions with smoothly varying frequencies.

The TMHB method features a matrix-implicit Krylov-subspace solution approach of same complexity as the standard harmonic balance method. As the TMHB solution is computed in a pseudo domain, we gave a procedure for computing the real Fourier coefficients of the solution, and we also detailed the construction of the time-map function. The convergence properties of TMHB were analyzed and demonstrated on analytic waveforms.

The success of TMHB is critically dependent on the selection of a non-uniform grid. Two grid selection strategies, direct and iterative, were introduced and studied. Both strategies are a priori schemes, and are designed to obey accuracy and stability requirements. Practical issues associated with their use were also addressed.

Results of applying the TMHB method on several circuit examples demonstrated that the TMHB method achieves up to five orders of magnitude improvement in accuracy compared to the standard harmonic balance method. The solution error in TMHB decays exponentially faster than the standard HB method when the size of the Fourier basis increases linearly. The TMHB method is also up to six times faster than the standard HB method in reaching identical solution accuracy, and uses up to five times less computer memory. The TMHB runtime speedup factor and storage savings favorably increase for stricter accuracy requirements, making TMHB well suited for high accuracy simulations of large strongly nonlinear circuits with rapid transitions.

7.2 Future Work

The grid selection strategies were shown to be the weak link in the TMHB method. More work in this area (particularly in the case of multiple waveforms in larger circuits) may enhance the consistency of the TMHB method and increase its practical value. In particular, more robust optimization schemes minimizing different functionals of the solution [71, 73, 76] deserve additional attention.

The post-processing procedure used in computing the real Fourier coefficients from the pseudo Fourier coefficients computed by the TMHB has $O(\sigma M^2)$ complexity, compared to the $O(NM\log M)$ complexity of the TMHB method. While we argued that in practice this does not cause runtime efficiency problems (see Chapter 4), it may be possible to use a faster alternative algorithm for this post-processing unmap step [78]. A less expensive procedure for computing the real Fourier coefficients can have a major impact on an efficient implementation of an adaptive grid selection TMHB algorithm.

While time-domain steady-state methods, such as the shooting-Newton method, and the finite-difference method, cannot be applied to analyze circuits driven by multi-tone signals, the harmonic balance method can be readily extended to solve these circuits [6, 23, 24, 25, 26, 27, 28, 29]. When the excitation signals have widely-separated frequencies, new methods based on converting the circuit equations to multi-rate partial differential equations have been proposed [19, 20]. Two-dimensional mapping pseudospectral techniques applied to explicit scalar problems have been considered in [74, 83]. However, it is not yet clear whether extending the TMHB method to multi-tone problems is possible. A further study is needed to answer this question.

An efficient analysis of two-tone linear time-varying circuits (e.g. PLLs, mixers, narrow-band amplifiers, switched capacitor filters) has recently been facilitated with Krylov-subspace based time-varying time-domain methods [16], as well as with a Krylov-subspace time-varying extension of the harmonic balance method [18]. Harmonic balance has also been applied to autonomous circuits (e.g. oscillators) [14]. Whether the mapping techniques of the TMHB method can be extended to these problems is an open question.

New spectral methods for special classes of circuits will continue to be developed. For example, we also formulated the Mixed Fourier-Chebyshev (MFC) method, which is much like the Mixed Frequency-Time (MFT) method [2, 4] and is meant to be used for clocked analog circuits. A derivation and a simple example of this method are presented in the Appendix A. A complete evaluation of the anticipated advantages of this new method requires further research.

Appendix A

The Mixed Fourier-Chebyshev Method

The steady-state analysis of clocked analog circuits is extremely computationally expensive because the period of the clock is typically orders of magnitude smaller than the time period of interest. The Mixed Frequency-Time method (MFT) [2, 4] exploits the property of these circuits that their waveforms are similar over the clock cycle intervals. The method thus efficiently computes a solution by integrating over only a few selected cycles.

We developed the Mixed Fourier-Chebyshev method (MFC) as a fully spectral method similar to MFT. The cycle segments of the waveforms are no longer obtained by time integration, but by approximation with truncated Chebyshev series.

To understand the MFC it is helpful to review the Mixed Frequency-Time method (MFT) [2, 4]. Given a quasiperiodic response sampled at S points at rate T_c, as shown in Figure A.1, the method starts by picking J sample times τ_j out of the sample set. The sampled waveform $v(\tau_j)$ is then approximated by a J-term Fourier series.

The MFT method represents coupling of the Fourier delay relationship:

$$v(\tau + T_c) = D(T_c)v(\tau) \tag{A.1}$$

where $D(T_c) \equiv \Gamma^{-1}(T_c)\Gamma$, with the time integration of the J cycles $[\tau, \tau + T_c]$:

$$v(\tau + T_c) = \xi(v(\tau), \tau, \tau + T_c) \tag{A.2}$$

The coupled equations are then solved for $v(\tau_j)$:

$$\xi(v(\tau), \tau, \tau + T_c) - D(T_c)v(\tau) = 0 \tag{A.3}$$

The Mixed Fourier-Chebyshev method features the same Fourier delay relationship as in the MFT method:

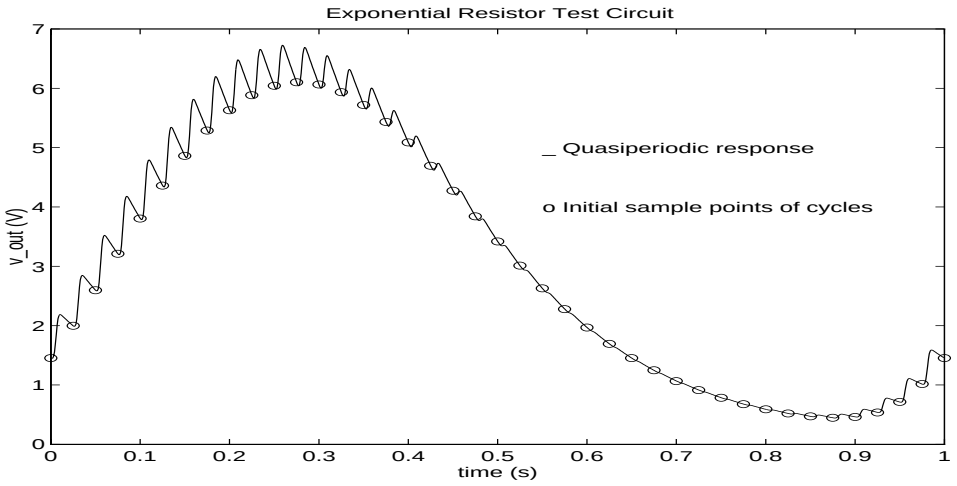

Figure A.1: Sampled quasiperiodic response.

$$\begin{bmatrix} v(\tau_1+T_c) \\ v(\tau_2+T_c) \\ \cdots \\ v(\tau_J+T_c) \end{bmatrix} = D(T_c) \begin{bmatrix} v(\tau_1) \\ v(\tau_2) \\ \cdots \\ v(\tau_J) \end{bmatrix} \tag{A.4}$$

The cycle waveforms are no longer computed by time integration (transient simulation), but with Chebyshev polynomials due to their non-periodicity. The method is thus fully spectral in nature and achieves in theory exponential order convergence.

The approximation of the J cycle waveforms $v_j(t)$ with Chebyshev series is:

$$v_j(t) \approx \sum_{k=0}^{M-1} C_{j,k} T_k(x) - \frac{1}{2} C_{j,0} \tag{A.5}$$

where $x = \mu_j(t)$ (linear map of $[\tau_j, \tau_j + T_c]$ onto $[-1,1]$).

The collocation grid is the standard "roots" interior $(M-1)$-point Chebyshev grid: $x_l = \cos\left(\frac{\pi(l-\frac{1}{2})}{M-1}\right)$

Consider a simple example: an exponential resistor circuit shown in Figure A.2 and described by the following equations

$$\begin{aligned} &C\dot{v} + \tfrac{1}{R}v - f(v - v_{in}) = 0 \\ &f(v) = e^v \\ &v_{in} = a_i sin(2\pi f_i t) + a_c sin(2\pi f_c t) \end{aligned} \tag{A.6}$$

The Mixed Fourier-Chebyshev method for this problem generates a total of MJ equations; the first $(M-1)J$ equations represent the J spectral problems:

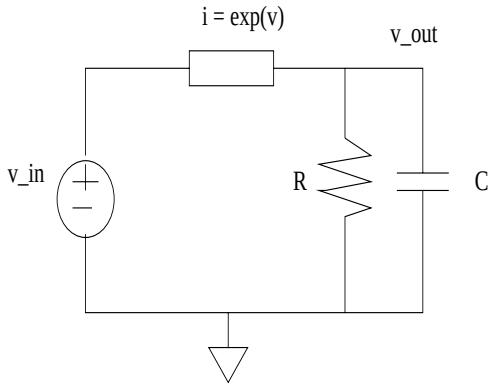

Figure A.2: Exponential resistor circuit.

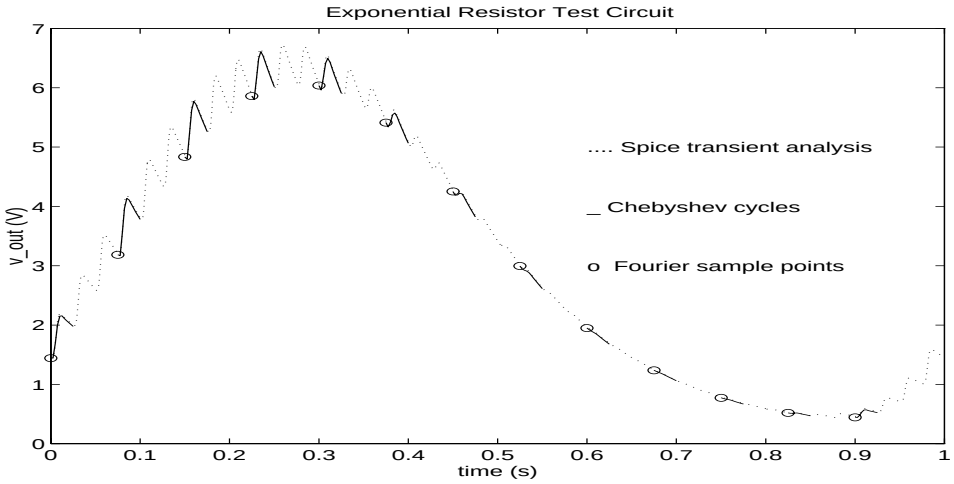

Figure A.3: MFC method results for the exponential resistor circuit.

$$\begin{aligned} &C\left(\tfrac{2}{T_c}\textstyle\sum_{k=0}^{M-1} C_{j,k} T_k'(x_l)\right) + \tfrac{1}{R}\left(\textstyle\sum_{k=0}^{M-1} C_{j,k} T_k(x_l) - \tfrac{1}{2} C_{j,0}\right) - \\ &\quad f\left(\textstyle\sum_{k=0}^{M-1} C_{j,k} T_k(x_l) - \tfrac{1}{2} C_{j,0}\right) = 0 \end{aligned} \tag{A.7}$$

while the last J equations represent the B.C.s via the Fourier delay relationship:

$$\begin{bmatrix} \sum_{k=0}^{M-1} C_{1,k} T_k(1) - \frac{1}{2} C_{1,0} \\ \vdots \\ \sum_{k=0}^{M-1} C_{J,k} T_k(1) - \frac{1}{2} C_{J,0} \end{bmatrix} = D(T_c) \begin{bmatrix} \sum_{k=0}^{M-1} C_{1,k} T_k(-1) - \frac{1}{2} C_{1,0} \\ \vdots \\ \sum_{k=0}^{M-1} C_{J,k} T_k(-1) - \frac{1}{2} C_{J,0} \end{bmatrix} \tag{A.8}$$

This system of equations is then solved for the Chebyshev coefficients $C_{j,k}$ with Newton's method. Note that the Chebyshev (upper) part of Jacobian is block-diagonal, while the Fourier (lower) part is full.

Figure A.3 shows the resulting Fourier sample points and cycle waveforms from applying the

Mixed Fourier-Chebyshev method to the exponential resistor circuit from Figure A.2. There were $J = 13$ total cycles. Robust and fast Newton convergence was observed. The plot verifies the validity of the MFC method by comparing the solution with the steady-state waveform computed via SPICE's transient analysis.

Bibliography

[1] Thomas J. Aprille and Timothy N. Trick. “Steady-State Analysis of Nonlinear Circuits with Periodic Inputs.” *Proceedings of the IEEE*, Vol. 60, No. 1, pp. 108–114, January 1972.

[2] Kenneth S. Kundert, Jacob White, and Alberto Sangiovanni-Vincentelli. “A Mixed Frequency-Time Approach for Distortion Analysis of Switching Filter Circuits”. *IEEE J. of Solid-State Circuits*, Vol. 24, No. 2, April 1989.

[3] Petrick L. Heron, Chao-Ren Chang, and Michael B. Steer. “Control of Aliasing in the Harmonic Balance Simulation of Nonlinear Microwave Circuits”. *IEEE MTT-S Digest*, 1989.

[4] Kenneth S. Kundert, Jacob K. White, and Alberto Sangiovanni-Vincentelli. *Steady-State Methods for Simulating Analog and Microwave Circuits*. Kluwer Academic Publishers, 1990.

[5] Rowan Gilmore and Michael B. Steer. “Nonlinear Circuit Analysis Using the Method of Harmonic Balance - A Review of the Art. Part I - Introductory Concepts”. *Int. J. on Microwave and Millimeter Wave Computer Aided Engineering*, Vol. 1, No. 1, 1991.

[6] Rowan Gilmore and Michael B. Steer. “Nonlinear Circuit Analysis Using the Method of Harmonic Balance - A Review of the Art. Part II - Advanced Concepts”. *Int. J. on Microwave and Millimeter Wave Computer Aided Engineering*, Vol. 1, No. 1, pp. 159–180, 1991.

[7] P. Heikkilä. *Object-Oriented Approach to Numerical Circuit Analysis*. Ph.D. dissertation, Helsinki University of Technology, January 1992.

[8] A. Brambilla, D. D’Amore, M. Pillan. “Convergence Improvements of the Harmonic Balance Method”. *IEEE Int. Symp. on Circuits and Systems*, Vol. 4, pp. 2482–2485, 1993.

[9] D. D’Amore, P. Maffezzoni, and M. Pillan. “A Newton-Powell Modification Algorithm for Harmonic Balance-Based Circuit Analysis”. *IEEE Trans. on Circuits and Systems, Fundamental Theory and Applications*, Vol. 41, No. 2, February 1994.

[10] H.G. Brachtendorf, G. Welsch, and R. Laur. “Fast Simulation of the Steady-State of Circuits by the Harmonic Balance Technique”. *Proceedings of the IEEE Int. Symposium on Circuits and Systems*, May 1995.

[11] R. Melville, P. Feldmann, and J. Roychowdhury. “Efficient Multi-Tone Distortion Analysis of Analog Integrated Circuits”. *Proceedings of the Custom Integrated Circuits Conference*, May 1995.

[12] R. Telichevesky, K. Kundert, and J. White. "Efficient Steady-State Analysis Based on Matrix-Free Krylov-Subspace Methods". *Proceedings of the IEEE Design Automation Conference*, pp. 480–484, 1995.

[13] R. Telichevesky, K. Kundert, J. Elfadel, and J. White. "Fast Simulation Algorithms for RF Circuits". Preprint, 1995.

[14] Michal Odyniec and William Overstreet. "New Applications of Harmonic Balance Analysis". *Int. J. of Microwave and Millimeter-Wave Computer-Aided Engineering*, Vol. 5, No. 3, pp. 224–233, 1995.

[15] P. Feldmann, R. Melville, and D.E. Long. "Efficient Frequency Domain Analysis of Large Nonlinear Analog Circuits". *Proceedings of the Custom Integrated Circuits Conference*, May 1996.

[16] R. Telichevesky, K. Kundert, and J. White. "Efficient AC and Noise Analysis of Two-Tone RF Circuits". *Proceedings of the IEEE Design Automation Conference*, 1996.

[17] A. Brambilla and D. D'Amore. "A Filter-Based Technique for the Harmonic Balance Method". *IEEE Tran. on Circuits and Systems, Fundamental Theory and Applications*, Vol. 43, No. 2, February 1996.

[18] Peter Feldmann and Jaijeet Roychowdhury. "Computation of Circuit Waveform Envelopes Using an Efficient, Matrix-Decomposed Harmonic Balance Algorithm". *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 1996.

[19] Jaijeet Roychowdhury. "Efficient Methods for Simulating Highly Nonlinear Multi-Rate Circuits". *Proceedings of the IEEE Design Automation Conference*, pp. 269–274, 1997.

[20] Jaijeet Roychowdhury. "MPDE Methods for Efficient Analysis of Wireless Systems". *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 451–454, 1998.

[21] E. A. Dorfi, L. O'C. Drury. "Simple Adaptive Grids for 1-D Initial Value Problems". *J. of Comp. Physics*, Vol. 69, pp. 175–195, 1987.

[22] Uri M. Ascher, Robert M. M. Mattheij, Robert D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Prentice Hall, 1988.

[23] Akio Ushida and Leon O. Chua. "Frequency-Domain Analysis of Nonlinear Circuits Driven by Multi-Tone Signals". *IEEE Trans. on Circuits and Systems*, Vol. CAS-31, No. 9, September 1984.

[24] Vittorio Rizzoli, Claudio Cecchetti, and Alessandro Lipparini. "A General-Purpose Program for the Analysis of Nonlinear Microwave Circuits Under Multitone Excitation by Multidimensional Fourier Transform". *17-th European Microwave Conference*, 1987.

[25] Vittorio Rizzoli, Claudio Cecchetti, Alessandro Lipparini, and Franco Mastri. "General-Purpose Harmonic Balance Analysis of Nonlinear Microwave Circuits Under Multitone Excitation". *IEEE Trans. on Microwave Theory and Techniques*, Vol. 36, No. 12, December 1988.

[26] Vittorio Rizzoli, Claudio Cecchetti, and Alessandro Lipparini. “Numerical Analysis of Intermodulation Distortion in Microwave Mixers”, *IEEE MTT-S Digest*, 1988.

[27] A. Ushida, L.O. Chua, and T. Sugawara. “A Substitution Algorithm”, *Int. J. of Circ. Theory and Applications*, Vol. 15, 1987.

[28] S.C. Chan and K.L. Ho. “Use of Multidimensional Discrete Fourier Transforms in the Harmonic balance method”, *Int. J. Electronics*, Vol. 66, No. 4, pp. 613–617, 1989.

[29] Patrick L. Heron and Michael B. Steer. “Jacobian Calculation Using the Multidimensional Fast Fourier Transform in the Harmonic Balance Analysis of Nonlinear Circuits”, *IEEE Trans. on Microwave Theory and Techniques*, Vol. 38, No. 4, April 1990.

[30] Michael B. Steer, Chao-Ren Chang, and George W. Rhyne. “Computer-Aided Analysis of Nonlinear Microwave Circuits Using Frequency-Domain Nonlinear Analusis Techniques: The State of the Art”. *Int. J. of Microwave and Millimeter-Wave Computer-Aided Engineering*, Vol. 1, No. 2, pp. 181–200, 1991.

[31] Ognen J. Nastov. *Analysis of Two-Dimensional Steady-State and Transient Simulations of Semiconductor Devices.* S.B. Thesis, Massachusetts Institute of Technology, June 1991.

[32] Ognen J. Nastov. *Automated Design Synthesis of CMOS Operational Amplifiers.* S.M. Thesis, Massachusetts Institute of Technology, February 1994.

[33] Ognen J. Nastov. “Spectral Methods for Steady-State Analysis of Nonlinear Circuits”. MIT EECS Ph.D. oral qualifying exam presentation, 1997.

[34] Ognen J. Nastov. “The Use of Wavelets in Solving Boundary Value Ordinary Differential Equations”. MIT EECS Ph.D. area exam paper, 1997.

[35] Ognen J. Nastov and Jacob K. White. “Time-Mapped Harmonic Balance.” Submitted to the *1999 IEEE Design Automation Conference*, 1999.

[36] Ognen J. Nastov and Jacob K. White. “Grid Selection Strategies for the Time-Mapped Harmonic Balance Simulation of Circuits with Rapid Transitions.” Submitted to the *1999 IEEE Custom Integrated Circuits Conference*, 1999.

[37] Ognen J. Nastov and Jacob K. White. “Error Convergence Analysis of the Time-Mapped Harmonic Balance.” Manuscript in progress, 1999.

[38] David Gottlieb and Steven A. Orszag. *Numerical Analysis of Spectral Methods : Theory and Applications* . Society for Industrial and Applied Mathematics, Philadelphia, PA, 1977.

[39] H.-O.Kreiss and J. Oliger. “Stability of the Fourier Method”. *SIAM J. of Numerical Analysis*, Vol. 16, pp. 421–433, 1979.

[40] Robert G. Voigt, David Gottlieb, and M. Yousuff Hussaini (editors). “Spectral Methods for Partial Differential Equations.” *Proceedings of the Symposium on Spectral Methods for Partial Differential Equations, NASA Langley Research Center*, SIAM, Philadelphia, PA, 1982.

[41] C. A. J. Fletcher. *Computational Galerkin Methods*. Springer-Verlag, New York, 1984.

[42] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods in Fluid Dynamics.* Springer-Verlag, Berlin, New York, 1987.

[43] John P. Boyd. *Chebyshev and Fourier Spectral Methods*. Lecture notes in engineering, Springer-Verlag, 1989.

[44] John Strain. "Spectral Methods for Nonlinear Parabolic Systems." *J. of Comp. Physics*, Vol. 122, pp. 1–12, 1995.

[45] Bengt Fornberg. *A Practical Guide to Pseudospectral Methods.* Cambridge University Press, England, 1996.

[46] Laurence W. Nagel. *SPICE2: A Computer Program to Simulate Semiconductor Circuits.* Memorandum No. ERL-M520, Electronics Research Laboratory, University of California, Berkeley, 1975.

[47] Kenneth S. Kundert. *The Designer's Guide to SPICE & SPECTRE*. Kluwer Academic Publishers, 1995.

[48] Germund Dahlquist. *Numerical methods.* Prentice-Hall, Englewood Cliffs, NJ, 1974.

[49] Assem S. Deif. *Advanced Matrix Theory for Scientists and Engineers.* Abacus Press, England, 1982.

[50] Gilbert Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, Wellesley, MA, 1986.

[51] Gilbert Strang. *Linear Algebra and Its Applications*. Harcourt Brace Jovanovich, San Diego, CA, 1988.

[52] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* John Hopkins University Press, Baltimore, 1989.

[53] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag, New York, 1993.

[54] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Co., Boston, MA, 1996.

[55] Al Kelley and Ira Pohl. *A Book on C.* The Benjamin/Cummings Publishing Company, Menlo Park, CA, 1984.

[56] William H. Press et al. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, UK, 1988.

[57] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization.* Academic Press, San Diego, CA, 1981.

[58] Jasbir S. Arora. *Introduction to Optimum Design.* McGraw-Hill, New York, 1989.

[59] William McC. Siebert. *Circuits, Signals, and Systems.* McGraw Hill, MIT Press, Cambridge, MA, 1986.

[60] Jack Smith. *Modern Communication Circuits.* McGraw Hill, New York, 1986.

[61] Phillip E. Allen and Douglas R. Holberg. *CMOS Analog Circuit Design,* Harcourt Brace Jovanovich, Saunders College Publishing, Orlando, Florida, 1987.

[62] Randall L. Geiger, Phillip E. Allen, Noel R. Strader. *VLSI Design Techniques for Analog and Digital Circuits*, McGraw-Hill, New York, 1990.

[63] Robert Northrop. *Analog Electronic Circuits: Analysis and Applications.* Addison-Wesley, Reading, MA, 1990.

[64] Donald O. Pederson. *Analog Integrated Circuits for Communication: Principles, Simulation, and Design.* Kluwer Academic Publishers, Boston, 1991.

[65] Paul R. Gray and Robert G. Meyer. *Analysis and Design of Analog Integrated Circuits*. John Wiley & Sons, Inc., New York, 1993.

[66] Zoher Z. Karu. *Signals and Systems Made Ridiculously Simple.* ZiZi Press, Cambridge, MA, 1995.

[67] Steven A. Orszag. "Comparison of Pseudospectral and Spectral Approximation". *Studies in Applied Mathematics*, Vol. LI, No. 3, September 1972.

[68] David Gottlieb, Liviu Lustman, and Steven A. Orszag. "Spectral Calculations of One-Dimensional Inviscid Compressible Flows". *SIAM J. on Sci. and Stat. Computing*, Vol. 2, No. 3, September 1981.

[69] M. Deville and E. Mund. "Chebyshev Pseudospectral Solution of Second-Order Elliptic Equations with Finite Element Preconditioning". *J. of Comp. Physics*, Vol. 60, pp. 517–533, 1985.

[70] C. Basdevant, M. Deville, P. Haldenwang, J.M. Lacroix, J. Ouazzani, R. Peyret, P. Orlandi, and A. T. Patera. "Spectral and Finite Difference Solutions of the Burgers Equation". *Computers and Fluids*, Vol. 14, No. 1, pp. 23–41, 1986.

[71] Herve' Guillard and Roger Peyret. "On the Use of Spectral Methods for the Numerical Solution of Stiff Problems". *Computer methods in Applied Mechanics and Engineering*, Vol. 66, pp. 17–43, 1988.

[72] M.Y. Hussaini, D.A. Kopriva, A.T. Patera. "Spectral Collocation Methods". *Applied Numerical Mathematics*, Vol. 5, pp. 177–208, 1989.

[73] A. Bayliss, D. Gottlieb, B.J. Matkowsky, and M. Minkoff. "An Adaptive Pseudo-Spectral Method for Reaction Diffusion Problems". *J. of Comp. Physics*, Vol. 81, pp. 421–443, 1989.

[74] A. Bayliss, R. Kuske, and B.J. Matkowsky. “A Two-Dimensional Adaptive Pseudo-Spectral Method”. *J. of Comp. Physics*, Vol. 91, pp. 174–196, 1990.

[75] John P. Boyd. “The Envelope of the Error for Trigonometric and Chebyshev Interpolation”. *J. of Scientific Computing*, Vol. 5, No. 4, December 1990.

[76] Herve’ Guillard, J.M. Male, and Roger Peyret. “Adaptive Spectral Methods with Application to Mixing Layer Computations”. *J. of Comp. Physics*, Vol. 102, pp. 114–127, 1992.

[77] John P. Boyd. “The Arctan/Tan and Kepler-Burgers Mappings for Periodic Solutions with a Shock, Front, or Internal Boundary Layer”, *J. of Comp. Physics*, Vol. 98, pp. 181–193, 1992.

[78] John P. Boyd. “A Fast Algorithm for Chebyshev, Fourier, and Sinc Interpolation onto an Irregular Grid”. *J. of Comp. Physics*, Vol. 103, pp. 243–257, 1992.

[79] John P. Boyd. “The Rate of Convergence of Fourier Coefficients for Entire Functions of Infinite Order with Application to the Weideman-Cloot Sinh-Mapping for Pseudospectral Computations on an Infinite Interval”. *J. of Comp. Physics*, Vol. 110, pp. 360–372, 1994.

[80] A. Pinelli, C. Benocci, and M. Deville. “Chebyshev Pseudospectral Solution of Advection-Diffusion Equations with Mapped Finite Difference Preconditioning”. *J. of Comp. Physics*, Vol. 112, pp. 1-11, 1994.

[81] Alvin Bayliss, Andres Class, and Bernard J. Matkowsky. “Adaptive Approximation of Solutions to Problems with Multiple Layers by Chebyshev Pseudo-Spectral Methods”. *J. of Comp. Physics*, Vol. 116, pp. 160–172, 1995.

[82] Alvin Bayliss, Marc Garbey, and Bernard J. Matkowsky. “Adaptive Pseudo-Spectral Domain Decomposition and the Approximation of Multiple Layers”. *J. of Comp. Physics*, Vol. 119, pp. 132–141, 1995.

[83] L. S. Mulholland, W. Z. Huang, and D. M. Sloan. “Pseudospectral Solution of Near-Singular Problems Using Numerical Coordinate Transformations Based on Adaptivity”. *SIAM J. Sci. Comput.*, Vol. 19, No. 4, pp. 1261-1289, July 1998.