# Chapter 11
# Multirate Shooting Method with Frequency Sweep for Circuit Simulation


**Kai Bittner and Hans Georg Brachtendorf**

**Abstract** We introduce multirate shootings methods to compute the response of radio frequency (RF) circuits with frequency modulated stimuli. The multirate technique is based on reformulating the system of ordinary differential algebraic equations (DAE) by partial differential equations (PDE). The PDE is semi-discretized by Rothe's method, i.e. by first discretizing the initial value problem. The resulting periodic boundary value problems are then solved by shooting techniques. Second, the instantaneous frequency is an additional unknown and concurrently estimated.

## 11.1 Introduction

The multirate simulation technique (see e.g. [1–4]) has been introduced to circuit simulation to handle RF signals with widely separated time scales in an efficient way, by reformulating the circuit equations as PDEs in different time scales. A semi-discretization of the multirate PDEs leads to a series of Periodic Steady State (PSS) problems, which are usually solved by waveform relaxation methods (Harmonic Balance, Finite Difference, Galerkin discretization), which approximate the periodic solution over a whole period. However, for some problems this can lead to convergence problems of nonlinear solvers (e.g. Newton) and large problem sizes with prohibitive memory and time requirements.

Here, we consider shooting (e.g. [5–11]) as an alternative approach for PSS. The advantage of shooting methods is that they can handle most problems for which a transient analysis is feasible. Furthermore, the size of linear and nonlinear problems to be solved is determined by the size of the circuit and not by the waveform, which can reduce memory requirements essentially and a speedup might be possible at least for some problems. Our goal here is to give a complete

K. Bittner (✉) · H. G. Brachtendorf
University of Applied Sciences of Upper Austria, Hagenberg, Austria
e-mail: Kai.Bittner@fh-hagenberg.at; Hans-Georg.Brachtendorf@fh-hagenberg.at



U. Langer et al. (eds.), *Scientific Computing in Electrical Engineering*,
Mathematics in Industry 28, https://doi.org/10.1007/978-3-319-75538-0_11

description of the shooting method in circuit simulation as an alternative to other PSS solvers.

We develop a shooting method based on multistep methods for circuit simulation in Sects. 11.2 and 11.3. Then shooting is applied to the multirate method in Sect. 11.4. Various approaches for an optimal frequency sweep are presented in Sect. 11.5. A numerical test in Sect. 11.6 concludes the results.

## 11.2 Circuit Equations and Multistep Methods

Consider the circuit equations in the charge/flux oriented modified nodal analysis (MNA) formulation, which yields a mathematical model in the form of a system of differential-algebraic equations (DAEs):

$$\frac{d}{dt} q\big(x(t)\big) + \underbrace{i\big(x(t)\big) + s(t)}_{g(x,t)} = 0, \qquad x(0) = x_0 \tag{11.1}$$

Here $x(t) \in \mathbb{R}^n$ is the vector of node potentials and specific branch currents and $q(x) \in \mathbb{R}^n$ is the vector of charges and fluxes. The vector $i(x) \in \mathbb{R}^n$ comprises static contributions, while $s(t) \in \mathbb{R}^n$ contains the contributions of independent sources.

The DAEs in (11.1) are usually solved by time integration formulas for stiff systems. Here, we consider implicit linear multistep methods, which approximate the solution at a discrete time step $t_k$ based on approximations at previous time steps $t_\ell < t_k$ as follows. Let the approximations $x_\ell \approx x(t_\ell)$, $0 \le \ell < k$, be already computed. The approximation $x_k \approx x(t_k)$ is found as the solution of the nonlinear system

$$F(x_k) := \sum_{\ell=0}^{s_k} \alpha_\ell^k \, q(x_{k-\ell}) + \beta_\ell^k \, g(x_{k-\ell}, t_{k-\ell}) = 0, \qquad s_k \le k. \tag{11.2}$$

Usually the trapezoidal rule or Gear's backward difference formulas (BDF) are used in circuit simulation. A further choice, in particular for high Q oscillators, are the trigonometric BDF formulas from [12], which avoid artificial energy loss by numerical damping.

The nonlinear system is solved by Newton's method, where we need the Jacobian

$$DF(x_k) = \alpha_0^k \, C(x_k) + \beta_0^k \, G(x_k).$$

The computation of the Jacobians $C(x) := q'(x)$ and $G(x) := i'(x)$ is usually implemented in a circuit simulator, together with the evaluation of $q(x)$ and $g(x, t)$.

## 11.3 Periodic Steady States and Shooting

To determine the periodic steady state (PSS) of a circuit means to solve the periodic boundary value problem

$$\frac{d}{dt}q\big(x(t)\big) + i\big(x(t)\big) + s(t) = 0, \qquad x(0) = x(P) \tag{11.3}$$

instead of the initial value problem (11.1), where the source term $s(t)$ is required to be $P$-periodic to assure existence of a periodic solution. One approach to solve a boundary value problem are shooting methods [5–9]. The principal idea is to solve initial value problems and to adapt the initial values so that the boundary conditions are fulfilled. That is, a nonlinear equation for the boundary value problem has to be solved. However, the computation of the corresponding Jacobian matrix is not trivial in general. An approach for circuit simulation based on the backward Euler method was given in [10]. A generalization to BDF and trapezoidal rule can be found in [11]. Here we give a short description of the shooting method for periodic steady states of circuits, using multistep methods as introduced in Sect. 11.2.

In the sequel, we assume that a periodic steady state exists, and that the solution depends smoothly on the consistent initial value, which is true for many practical problems. Let $x(t;\xi)$ be the solution of the initial value problem with initial value $\xi$, i.e.,

$$\frac{d}{dt}q\big(x(t;\xi)\big) + i\big(x(t;\xi)\big) + s(t) = 0, \qquad x(0;\xi) = \xi. \tag{11.4}$$

Further, $\Phi(\xi) := x(P;\xi)$ is the value of the solution after one period. To find a PSS one has to determine an initial value $\xi \in \mathbb{R}^n$ such that $\Phi(\xi) - \xi = 0$, i.e., we have to solve a nonlinear system. For the application of Newton's method we need the Jacobian $\frac{d}{d\xi}\big(\Phi(\xi) - \xi\big) = \Phi'(\xi) - \boldsymbol{I}$. Numerical differentiation is prohibitive expensive for larger circuits. Therefore, we consider an alternative approach.

Since $\Phi(\xi) = x(P;\xi)$ cannot be determined exactly, we replace $\Phi(\xi)$ by the approximation

$$\widetilde{\Phi}(\xi) := x_N := x_N(\xi),$$

where $x_k := x_k(\xi)$ is the solution from the multistep method (11.2) with $x_0 = \xi$ and $t_N = P$. While we need a consistent initial value $\xi$ in (11.4), we can avoid this requirement if we use one or more (depending on the index of the DAE) backward Euler steps (BDF1) at the begin of the time integration [11, Lemma 4.2]. However, projecting the initial guess for Newton's method onto a consistent solution can improve the convergence of the shooting method [13].

The Jacobian $\widetilde{\Phi}'(\xi)$ is determined as follows. By differentiating (11.2) one obtains

$$\sum_{\ell=0}^{s_k} \Big(\alpha_\ell^k\, C(x_{k-\ell}) + \beta_\ell^k G(x_{k-\ell})\Big)\frac{dx_{k-\ell}}{d\xi} = 0. \tag{11.5}$$

This leads to the recursion

$$\frac{dx_k}{d\xi} = -\Big(\alpha_0^k\, C(x_k) + \beta_0^k\, G(x_k)\Big)^{-1} \left( \sum_{\ell=1}^{s_k} \Big(\alpha_\ell^k\, C(x_{k-\ell}) + \beta_\ell^k G(x_{k-\ell})\Big) \frac{dx_{k-\ell}}{d\xi} \right), \tag{11.6}$$

for $k = 1, \ldots, N$ with $\frac{dx_0}{d\xi} = \boldsymbol{I}$.

Computing $\widetilde{\Phi}'(\xi) = \frac{dx_N}{d\xi}$ using direct solvers will be rather expensive. The good news are that $C(x_k)$ and $G(x_k)$ are sparse and that the LU-decomposition of the sparse matrix $\alpha_0^k\, C(x_k) + \beta_0^k\, G(x_k)$ has to be computed anyway in order to solve (11.2).[1] The bad news are that the inverse matrices are dense and thus the matrices $\frac{dx_k}{d\xi}$ are dense, too. That is, the computational complexity for computing $\widetilde{\Phi}'(\xi)$ amounts to $\mathcal{O}(n^{1+\gamma}\, N)$, if we assume the computational cost for the sparse forward-backward-substitutions (during Transient analysis) to be $\mathcal{O}(n^\gamma)$ for some $\gamma > 1$. Additionally, we have to solve a linear system with the dense matrix $\widetilde{\Phi}'(\xi) - \boldsymbol{I}$, for the outer Newton iteration to determine $\xi$, which requires $\mathcal{O}(n^3)$ operations. However, this direct computation needs only limited memory of order $\mathcal{O}(n^2)$ (independent of $N$), if it is done immediately during the transient analysis.

However, we can attempt to solve the linear system by an iterative method, e.g., GMRES. This requires an efficient matrix vector multiplication $\widetilde{\Phi}'(\xi)\, y$ for any given $y \in \mathbb{R}^n$. Using the recursion (11.6) we can do this, without knowing the matrix $\widetilde{\Phi}'(\xi)$ itself. We define $y_k := \frac{dx_k}{d\xi} y$, which yields $y_N = \widetilde{\Phi}'(\xi)\, y$ by the recursion

$$\Big(\alpha_0^k\, C(x_k) + \beta_0^k\, G(x_k)\Big) y_k = \sum_{\ell=1}^{s} \Big(\alpha_\ell^k\, C(x_{k-\ell}) + \beta_\ell^k G(x_{k-\ell})\Big) y_{k-\ell}, \qquad y_0 = y. \tag{11.7}$$

This approach requires to store all matrices $C(x_k)$ and $G(x_k)$, $k = 0, \ldots, N$, since the vector $y = \widetilde{\Phi}(\xi) - \xi$ is only available after the transient analysis is complete. This results in a memory consumption $\mathcal{O}(N\, n^\gamma)$ for some small $\gamma > 1$. The computational cost of one matrix vector multiplication would be essentially the same as for the transient analysis (without device evaluation), i.e., $\mathcal{O}(N\, n^\gamma)$. This has to be multiplied by the number of iterations $K$ needed by the iterative solver.

The iteration count $K$ can be expected to be small in many cases due to the following statement (see e.g. [14, Prop. 4]). We assume that the matrix $\widetilde{\Phi}'(\xi)$ is diagonizable, which is the typical case. The error for the residual $r_m$ after $m$ iteration steps can be estimated as

$$\|r_m\| \le c\, \|r_0\| \min_{p \in \Pi_{m-1},\, p(0)=1} \max_{\lambda \in \sigma} |p(\lambda)|$$

[1] We assume that a direct sparse solver is used to solve (11.2) and (11.6), which is reasonable in circuit simulation. For an iterative solver we have only to determine a preconditioner once for the multiple solves, i.e., similar considerations apply.

where $c$ is a constant depending on the matrix, $\Pi_{m-1}$ are the polynomials of degree less than $m$ and $\sigma$ is the set of eigenvalues of the system matrix $\widetilde{\Phi}'(\xi)$. If the eigenvalues are clustered around few values one can choose polynomials with the zeros in this clusters to prove a fast decay of the residuals.

In a circuit many components of an initial value are damped out over a period of a periodic signal, which corresponds to small eigenvalues of $\widetilde{\Phi}'(\xi)$, while only few eigenvalues may be away from zero (e.g. due to an oscillator). Thus, the Jacobian $\widetilde{\Phi}'(\xi) - \boldsymbol{I}$ may have mainly eigenvalues close to $-1$ with only a few exceptions, resulting in a fast convergence of GMRES, even without preconditioning.

The direct solver is suitable if $n$ is not too large and accuracy of the linear solver is important, while for larger circuits the iterative solver might be favored.

## 11.4 Multirate Shooting Method

To separate different time scales, the circuit equation (11.1) can be replaced by partial differential equations [1–4], namely

$$\tfrac{\partial}{\partial\tau} q\big(\hat{x}(\tau,t)\big) + \omega(\tau)\, \tfrac{\partial}{\partial t} q\big(\hat{x}(\tau,t)\big) + i\big(\hat{x}(\tau,t)\big) + \hat{s}\big(\tau,t\big) = 0 \tag{11.8}$$

where $\omega(\tau)$ is an estimate of the (scaled) angular frequency. The bivariate function $\hat{x}(\tau,t)$ is related to the univariate solution $x(t)$ of (11.1) as follows. For any solution $\hat{x}(\tau,t)$ of (11.8) we get by $x_\theta(t) = \hat{x}\big(t, \Omega_\theta(t)\big)$, $\Omega_\theta(t) = \theta + \int_0^t \omega(s)\,ds$ a solution of

$$\tfrac{d}{dt} q\big(x(t)\big) + i\big(x(t)\big) + \hat{s}\big(t, \Omega_\theta(t)\big) = 0.$$

Thus, if we choose $\hat{s}$ such that $s(t) = \hat{s}\big(t, \Omega_0(t)\big)$, then the solution of (11.8) provides also a solution of (11.1), i.e., $x(t) = x_0(t) = \hat{x}\big(t, \Omega_0(t)\big)$.

The multirate equation (11.8) are usually solved under periodicity conditions $\hat{x}(\tau,t) = \hat{x}(\tau, t+P)$ in $t$ and initial conditions $\hat{x}(0,t) = X_0(t)$ in $\tau$. The source term has then to be periodic, too, i.e., $\hat{s}(\tau,t) = \hat{s}(\tau, t+P)$. The term $\omega(\tau)$ can be used to adapt to frequency modulated signals. In [3] it was shown that we can improve the smoothness of $\hat{x}$ in $\tau$, if $P$ and $\omega(\tau)$ are chosen such that $\frac{\omega(\tau)}{P}$ equals the instantaneous frequency.

Following [3, 4] we use Rothe's method for semi-discretization. Using Gear's BDF[2] method of order $s$ with respect to $\tau$ one obtains

$$\sum_{i=0}^{s} \tilde{\alpha}_i^k q\big(X_{k-i}(t)\big) + \omega_k \frac{d}{dt} q\big(X_k(t)\big) + i\big(X_k(t)\big) + \hat{s}\big(\tau_k, t\big) = 0 \tag{11.9}$$

[2]Other multistep method (e.g. trapezoidal rule) can be used, too.

With the definition

$$g_k(x,t) := \tilde{\alpha}_0^k q(x) + i(x) + \hat{s}\big(\tau_k, t\big) + \sum_{i=1}^{s} \tilde{\alpha}_i^k q\big(X_{k-i}(t)\big),$$

$X_k$ is the solution of the periodic boundary value problem

$$\omega_k \frac{d}{dt} q\big(x(t)\big) + g_k(x(t),t) = 0, \qquad x(t) = x(t+P). \tag{11.10}$$

The new problem (11.10) is closely related to the original periodic steady state problem of the circuit, only modified by the additional 'source term' $\sum_{i=1}^{s} \tilde{\alpha}_i^k q\big(X_{k-i}(t)\big)$. Analogous to (11.5) one obtains

$$\sum_{\ell=0}^{s_m} \Big(\omega_k \alpha_\ell^m\, C(x_{m-\ell}) + \beta_\ell^m \big(G(x_{m-\ell}) + \tilde{\alpha}_0^k\, C(x_{m-\ell})\big)\Big) \tfrac{dx_{m-\ell}}{d\xi} = 0. \tag{11.11}$$

Thus, $X_k$ can be approximated by the shooting method from Sect. 11.3. The only additional problem is to compute $\sum_{i=1}^{s} \tilde{\alpha}_i^k q\big(X_{k-i}(t)\big)$ at the transient time steps for $t_{k,i}$ for $X_k$. This requires to store the values $q_\ell = q\big(X_{\ell,i}\big) \approx q\big(X_\ell(t_{\ell,i})\big)$. These values can be used to approximate $q\big(X_\ell(t_{k,i})\big)$, $k > \ell$, e.g. by interpolation.

## 11.5 Frequency Sweep and Smoothness Conditions

As pointed out in [3, 15, 16] the function $\omega(\tau)$ can be chosen in order to obtain a smoother solution, which accelerates the simulation due to larger time steps in $\tau$. The observation that a modification of $\omega(\tau)$ for $\tau < \tau_k$ results in a phase shift of $\hat{x}(\tau_k, \cdot)$ leads to the proposition of a smoothness condition of the form

$$\big\| \tfrac{\partial}{\partial \tau} \hat{x}(\tau, \cdot) \big\|_{L^2} \to \min, \tag{11.12}$$

(cf. [3, 15, 16]) or

$$\big\| \tfrac{\partial}{\partial \tau} q\big(\hat{x}(\tau, \cdot)\big) \big\|_{L^2} \to \min, \tag{11.13}$$

(cf. [15–17]) which should reduce changes with respect to $\tau$. Here, we use the norm $\|x\|_{L^2}^2 := \int_0^P \sum_{k=1}^n |x_k(t)|^2\, dt$ and the corresponding inner product $\langle x, y\rangle := \int_0^P \sum_{k=1}^n x_k(t)\, y_k(t)\, dt$. In many cases a (near) optimal choice of $\omega(\tau)$ is known in advance, e.g., from the (instantaneous) frequency of sources in a driven circuit.

Often, a good choice of $\omega(\tau)$ is not known in advance, but central for the success of the simulation, e.g., for the simulation of high Q oscillators without numerical damping [12] or voltage controlled oscillators (VCO) (in a Phase-Locked

Loop (PLL)) [3, 4]. For the existence and uniqueness of an optimal $\omega(\tau)$ we refer to [18]. There have been several approaches to include the above smoothness conditions into simulation methods, e.g. if finite differences or collocation or Galerkin methods (Harmonic Balance, spline wavelets) are applied to solve the periodic problem (11.10). During our investigations it turned out that the application of these approaches to the shooting method is not straightforward. In this section we will develop frequency sweep following methods for the shooting methods, based on established methods for other periodic solvers.

### *11.5.1 An Explicit Approach*

We first refer to an approach of Houben [17] based on condition (11.13), which leads to the equality

$$\omega(\tau) = -\frac{\left\langle \frac{\partial}{\partial t} q\big(\hat{x}(\tau,\cdot)\big), i\big(\hat{x}(\tau,\cdot)\big) + \hat{s}\big(\tau,\cdot\big)\right\rangle}{\left\| \frac{\partial}{\partial t} q\big(\hat{x}(\tau,\cdot)\big)\right\|_{L^2}^2}. \tag{11.14}$$

Apparently, we can determine $\omega(\tau)$ only after $\hat{x}(\tau,\cdot)$ is known. However in the Rothe discretization (11.9) we can use the guess

$$\omega_k \approx \omega(\tau_{k-1}) = -\frac{\left\langle \frac{d}{dt} q(X_{k-1}), i(X_{k-1}) + \hat{s}\big(\tau_{k-1},\cdot\big)\right\rangle}{\left\| \frac{d}{dt} q(X_{k-1})\right\|_{L^2}^2}, \tag{11.15}$$

based on the solution of the previous time step. From the shooting method we know not only approximations $X_{k-1,\ell} \approx X_{k-1}(t_\ell)$, but also $g_{k-1,\ell} = i\big(X_{k-1}(t_\ell)\big) + \hat{s}\big(\tau_{k-1}, t_\ell\big)$ and $q_{k-1,\ell} = q\big(X_{k-1}(t_\ell)\big)$. Approximations $Dq_{k-1,\ell} = \frac{d}{dt} q\big(X_{k-1}(t_\ell)\big)$ for the derivatives can be computed using finite differences, as it is done in the BDF method anyway. Numerical integration leads to the formulation

$$\omega_k = -\frac{\sum_{\ell=0}^{N} w_{k,\ell} \langle Dq_{k-1,\ell}, g_{k-1,\ell}\rangle}{\sum_{\ell=0}^{N} w_{k,\ell} \| Dq_{k-1,\ell}\|_{L^2}^2}, \tag{11.16}$$

where the $w_{k\ell}$ are quadrature weights for the grid $\{t_{k,\ell}\}$. The original approach in [17] uses the method of lines, but it works with Rothe's method as well.

Although the method is simple and easy to implement, also in a shooting method, it has limits. Since the computation uses only data from the previous time step, the accuracy of this approach is limited. In many circuits an accurate estimate for the optimal $\omega_k$ is essential for the efficiency of the multirate algorithm. In the sequel we will consider methods, where $\omega_k$ is determined in the Newton iteration of the shooting method for $X_k$.

### 11.5.2 An Additional Equation

Treating $\omega_k$ in Eq. (11.10) as an unknown requires two things. We need derivatives with respect to $\omega_k$ for the Jacobian, and an additional equation and its derivatives. Since our shooting equation depends now also on $\omega$, we replace $\widetilde{\Phi}(\xi)$ by $\widetilde{\Phi}(\xi,\omega)$. We obtain the derivative with respect to $\omega$ as $\frac{\partial}{\partial\omega}\Big(\widetilde{\Phi}(\xi,\omega)-\xi\Big) = \frac{\partial}{\partial\omega}\widetilde{\Phi}(\xi,\omega) = \frac{dx_N}{d\omega}$. Obviously the initial value is independent of $\omega$, i.e., $\frac{dx_0}{d\omega}=0$. Differentiating the discretized version of Eq. (11.10) with respect to $\omega$ yields

$$\sum_{\ell=0}^{s_m}\alpha_\ell^m\Big(q\big(x_{m-\ell}\big)+\omega C(x_{m-\ell})\frac{dx_{m-\ell}}{d\omega}\Big)+\beta_\ell^m\Big(G_k(x_{m-\ell})+\tilde{\alpha}_0^k\,C(x_{m-\ell})\Big)\frac{dx_{m-\ell}}{d\omega}=0. \tag{11.17}$$

Thus $\frac{\partial}{\partial\omega}\widetilde{\Phi}(\xi,\omega)$ can be computed during the transient simulation of the shooting using the recursion over $\frac{dx_k}{d\omega}$ similar to (11.6). Pulch [15, 16] suggests the following approach. Based on the Gâteaux derivative, he shows that the smoothness condition (11.12) is equivalent to

$$0=\big\langle\tfrac{\partial}{\partial\tau}\hat{x}(\tau,\cdot),\,\tfrac{\partial}{\partial t}\hat{x}(\tau,\cdot)\big\rangle. \tag{11.18}$$

The challenge is to incorporate this equation into the shooting method, which is done as follows. First we semi-discretize by replacing $\frac{\partial}{\partial\tau}\hat{x}(\tau_k,\cdot)$ and $\frac{\partial}{\partial t}\hat{x}(\tau_k,\cdot)$ by $\frac{X_k-X_{k-1}}{\tau_k-\tau_{k-1}}$ and $X_k'$, respectively. Thus condition (11.18) is substituted by

$$\begin{aligned}0&=\big\langle X_k-X_{k-1},X_k'\big\rangle\\&=\tfrac{1}{2}\int_0^P\tfrac{d}{dt}\|X_k(t)\|^2\,dt-\Big(\underbrace{X_{k-1}(P)}_{X_{k-1}(0)}{}^T\,X_k(P)-X_{k-1}(0)^T\,X_k(0)\Big)+\int_0^P X_{k-1}'(t)^T\,X_k(t)\,dt\\&=\tfrac{1}{2}\Big(\|X_k(P)\|^2-\|X_k(0)\|^2\Big)+X_{k-1}(0)^T\Big(X_k(0)-X_k(P)\Big)+\int_0^P X_{k-1}'(t)^T\,X_k(t)\,dt,\end{aligned}$$

i.e., the solution $(x,\omega_k)$ of the periodic problem (11.10) shall satisfy

$$\tfrac{1}{2}\Big(\|x(P)\|^2-\|x(0)\|^2\Big)+X_{k-1}(0)^T\Big(x(0)-x(P)\Big)+\int_0^P X_{k-1}'(t)^T\,x(t)\,dt=0. \tag{11.19}$$

Although $x$ is $P$-periodic, we cannot assume $x(P)=x(0)$ during the Newton iteration of the shooting method, i.e., none of terms above can be neglected.

By numerical integration we approximate (11.19) by

$$\Psi(\xi,\omega):=\tfrac{1}{2}\Big(x_N^Tx_N-x_0^Tx_0\Big)+\tilde{x}_{k-1,0}^T\Big(x_0-x_N\Big)+\sum_{i=0}^{N}w_i\;\tilde{x}_{k-1,i}^T\,x_i=0, \tag{11.20}$$

where the $w_k$ are quadrature weights and $\tilde{x}_{k-1,i}$ are approximations of $X'_{k-1}(t_{k,i})$. Now we have to solve the system $\widetilde{\Phi}(\xi,\omega)=0$, $\Psi(\xi,\omega)=0$ with $n+1$ unknowns and $n+1$ equations.

For Newton's method one needs the derivatives of $\Psi(\xi,\omega)$, which are

$$\frac{\partial}{\partial\xi}\Psi(\xi,\omega)=\left(x_N-\tilde{x}_{k-1,0}\right)^T\frac{dx_N}{d\xi}-\left(x_0-\tilde{x}_{k-1,0}\right)^T+\sum_{i=0}^{N}w_i\;\tilde{x}_{k-1,i}^T\frac{dx_i}{d\xi},$$

$$\frac{\partial}{\partial\omega}\Psi(\xi,\omega)=\left(x_N-\tilde{x}_{k-1,0}\right)^T\frac{dx_N}{d\omega}+\sum_{i=0}^{N}w_i\;\tilde{x}_{k-1,i}^T\frac{dx_i}{d\omega}.$$

This does not increase the computational cost essentially, since $\frac{dx_i}{d\xi}$ is already computed during the recursion (11.6) (with the modification from (11.11)) and $\frac{dx_i}{d\omega}$ is determined in the recursion (11.17). The only extra effort is to add up the terms $\frac{dx_i}{d\xi}\,X'_{k-1}(t_{k,i})$ and $\frac{dx_i}{d\omega}\,X'_{k-1}(t_{k,i})$ during the computation.

If we replace (11.12) by (11.13) one obtains by an analogous argument

$$\Psi_q(\xi,\omega):=\tfrac{1}{2}\left(q_N^Tq_N-q_0^Tq_0\right)+\tilde{q}_{k-1,0}^T\left(q_0-q_N\right)+\sum_{i=0}^{N}w_i\tilde{q}_i^Tq_i=0, \qquad (11.21)$$

as well as the derivatives with respect to $\xi$ and $\omega$.

### 11.5.3 A Discrete Smoothness Criterion

We start from the smoothness criterion (11.12), which we discretize instead of formulating an equivalent equation, namely as

$$\|X_k(t)-X_{k-1}(t)\|_{L^2}^2\to\min. \qquad (11.22)$$

A similar criterion was introduced in [3] for waveform relaxation methods. Using numerical integration, condition (11.22) becomes

$$\sum_{\ell=0}^{N}w_\ell\left|x_\ell(\xi,\omega)-\tilde{x}_{k-1,\ell}\right|^2\to\min, \qquad (11.23)$$

with suitable quadrature weights $w_\ell$ and approximations $\tilde{x}_{k-1,\ell}$ of $X_{k-1}(t_{k,\ell})$. This optimization condition has to be solved under the condition that $\Phi(\xi,\omega)-\xi=0$ (remember $x_0(\xi,\omega)=\xi$ and $x_N(\xi,\omega)=\Phi(\xi,\omega)$ with $t_N=P$). Using a Lagrange

multiplier approach we obtain

$$\tfrac{1}{2}\sum_{\ell=0}^{N} w_\ell \left|x_\ell(\xi,\omega)-\tilde{x}_{k-1,\ell}\right|^2 + \lambda^T\big(\Phi(\xi,\omega)-\xi\big) \to \min. \tag{11.24}$$

To establish a Gauss-Newton type method, we linearize the problem as follows. For a given initial guess $(\xi,\omega)$ we use the linear approximation

$$\begin{aligned} x_\ell(\xi-d_\xi,\omega-d_\omega) &\approx x_\ell(\xi,\omega) - \tfrac{\partial x_\ell}{\partial \xi}(\xi,\omega)\,d_\xi - d_\omega\,\tfrac{\partial x_\ell}{\partial \omega}(\xi,\omega)\\ \Phi(\xi-d_\xi,\omega-d_\omega)-(\xi-d_\xi)&\\ &\approx \Phi(\xi,\omega)-\xi-\Big(\tfrac{\partial \Phi}{\partial \xi}(\xi,\omega)-I\Big)d_\xi - d_\omega\,\tfrac{\partial \Phi}{\partial \omega}(\xi,\omega). \end{aligned}$$

Substituting this into (11.24) and setting the derivatives with respect to $d_\xi$, $d_\omega$, and $\lambda$ to zero we obtain the equations

$$\begin{aligned} -\sum_{\ell=0}^{N} w_\ell\Big(\tfrac{\partial x_\ell}{\partial \xi}(\xi,\omega)\Big)^T\Big(x_\ell(\xi,\omega) - \tfrac{\partial x_\ell}{\partial \xi}(\xi,\omega)\,d_\xi - d_\omega\,\tfrac{\partial}{\partial \omega}x_\ell(\xi,\omega) - \tilde{x}_{k-1,\ell}\Big)&\\ -\Big(\tfrac{\partial}{\partial \xi}\Phi(\xi,\omega)-I\Big)^T\lambda = 0&\\ -\sum_{\ell=0}^{N} w_\ell\Big(\tfrac{\partial x_\ell}{\partial \omega}(\xi,\omega)\Big)^T\Big(x_\ell(\xi,\omega) - \tfrac{\partial x_\ell}{\partial \xi}(\xi,\omega)\,d_\xi - d_\omega\,\tfrac{\partial}{\partial \omega}x_\ell(\xi,\omega) - \tilde{x}_{k-1,\ell}\Big)&\\ -\Big(\tfrac{\partial}{\partial \omega}\Phi(\xi,\omega)\Big)^T\lambda = 0&\\ \Phi(\xi,\omega)-\xi-\Big(\tfrac{\partial \Phi}{\partial \xi}(\xi,\omega)-I\Big)d_\xi - d_\omega\,\tfrac{\partial \Phi}{\partial \omega}(\xi,\omega) = 0.& \end{aligned}$$

For abbreviation we introduce $U, A \in \mathbb{R}^{N\times N}$, $v, c, z, b \in \mathbb{R}^N$ and $\rho, \eta \in \mathbb{R}$

$$\begin{aligned} U &:= \sum_{\ell=0}^{N} w_\ell\Big(\tfrac{\partial x_\ell}{\partial \xi}(\xi,\omega)\Big)^T\tfrac{\partial x_\ell}{\partial \xi}(\xi,\omega), & A &:= \tfrac{\partial \Phi}{\partial \xi}(\xi,\omega)-I; \\ v &:= \sum_{\ell=0}^{N} w_\ell\Big(\tfrac{\partial x_\ell}{\partial \xi}(\xi,\omega)\Big)^T\tfrac{\partial x_\ell}{\partial \omega}(\xi,\omega), & c &:= \sum_{\ell=0}^{N} w_\ell\Big(\tfrac{\partial x_\ell}{\partial \xi}(\xi,\omega)\Big)^T\Big(x_\ell(\xi,\omega)-\tilde{x}_{k-1,\ell}\Big), \\ z &:= \tfrac{\partial \Phi}{\partial \omega}(\xi,\omega), & b &:= \Phi(\xi,\omega)-\xi; \\ \rho &:= \sum_{\ell=0}^{N} w_\ell\Big(\tfrac{\partial x_\ell}{\partial \omega}(\xi,\omega)\Big)^T\tfrac{\partial x_\ell}{\partial \omega}(\xi,\omega), & \eta &:= \sum_{\ell=0}^{N} w_\ell\Big(\tfrac{\partial x_\ell}{\partial \omega}(\xi,\omega)\Big)^T\Big(x_\ell(\xi,\omega)-\tilde{x}_{k-1,\ell}\Big) \end{aligned} \tag{11.25}$$

such that the above linear system becomes (in block matrix notation)

$$\begin{pmatrix} U & v & A^T \\ v^T & \rho & z^T \\ A & z & \mathbf{0} \end{pmatrix}\cdot\begin{pmatrix} d_\xi \\ d_\omega \\ \lambda \end{pmatrix} = \begin{pmatrix} c \\ \eta \\ b \end{pmatrix}.$$

By a Schur complement elimination we obtain the solutions

$$d_\omega = \frac{\eta - v^T\tilde{b} - \tilde{z}^T\big(c - U\tilde{b}\big)}{\rho - 2v^T\tilde{z} + \tilde{z}^T U\tilde{z}} \qquad \text{and} \qquad d_\xi = \tilde{b} - d_\omega\,\tilde{z}.$$

for Newton updates of $\xi$ and $\omega$, where $A\tilde{b} = b$ and $A\tilde{z} = z$.

This requires to solve a linear system with two right hand sides, which can be done nearly as efficient as in the original shooting. However, the computation of $U$ is rather expensive since it requires $N + 1$ matrix-matrix multiplications. A faster way is to compute first the vectors $\zeta_\ell = \frac{\partial x_\ell}{\partial \xi}(\xi, \omega)\,\tilde{z}$. Then $\tilde{z}^T U\tilde{z}$ is computed by

$$\tilde{z}^T U\tilde{z} = \textstyle\sum_{\ell=0}^{N} w_\ell\,\zeta_\ell^T\zeta_\ell, \tag{11.26}$$

which needs only $N+1$ inner products and $N+1$ matrix vector products. The value of $\tilde{z}^T U\tilde{b}$ can be computed analogously. However, we need to store the Jacobians $\frac{\partial x_\ell}{\partial \xi}(\xi, \omega)$. That is, if we follow the memory saving approach (with a direct shooting solver) we will compute $U$ directly using the formula in (11.25). For the time saving approach (with GMRES in shooting) we will use the stored data to do a fast computation based on (11.26) (cf. Sect. 11.3).

## 11.6 Numerical Test

The described methods have been implemented in C++ and incorporated in our circuit simulator LinzFrame. We have tested the method on a PLL (containing 145 MOSFETs and 80 unknowns), leading to a DAE of index 1. Here we show the multirate simulation of the locking phase using the frequency sweep method from Sect. 11.5.2. In Fig. 11.1 one can see that the reference and feedback signal are in-phase after ca. 200 µs. This is reflected by the charge pump output in Fig. 11.2 which measures the phase difference of both signals and is low pass filtered to control the VCO. The instantaneous frequency estimate in Fig. 11.2 provides information on the frequency modulation of the signals.

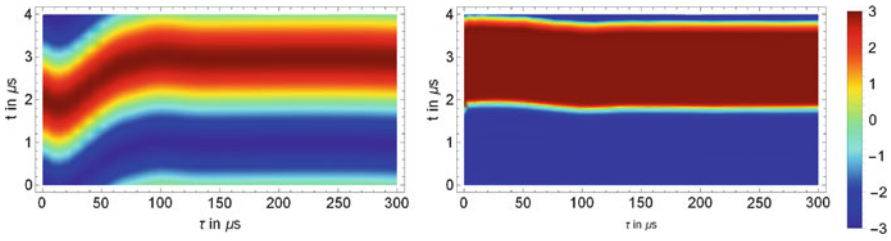


**Fig. 11.1** Reference and feedback signal of the PLL

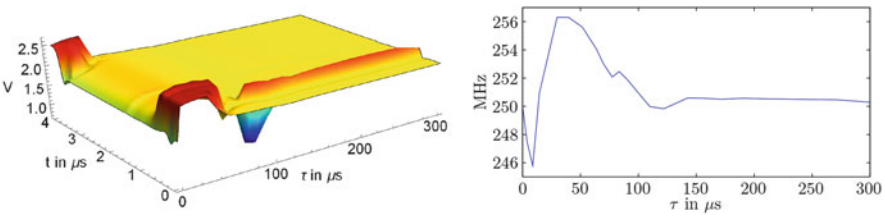


**Fig. 11.2** Charge pump output and instantaneous frequency estimate based on $\omega(\tau)$

It turned out that the shooting method performs much better than e.g. the adaptive spline-Galerkin method from [4] in this locking phase. This is due to the fact that the transient simulation does not rely on a good initial guess for the PSS, which is taken from the previous envelope time step for Galerkin or finite difference schemes. In the locking phase, signals as the charge pump output depicted in Fig. 11.2 will require very small envelope time steps to achieve convergence of Newton's method. However, if all signals are sufficiently smooth (e.g. after locking of the PLL), adaptive Galerkin or FD schemes often perform better since they can employ information on grid and signal shape from the previous envelope time step.

## 11.7 Conclusion

A shooting method to determine PSS of circuits has been developed and implemented. Possible modifications of this method have been introduced to solve sub-problems in the PDE based multirate circuit simulation method for RF circuits. The new method provides an alternative to waveform relaxation methods if the latter fail due to prohibitive time or memory requirements, or convergence problems.

**Acknowledgements** This work has been partly supported by the fp7 project nanoCOPS under grant 619166 and the EFRE project Connected Vehicles under grant IWB2020.

## References

1. Brachtendorf, H.G.: Simulation des eingeschwungenen Verhaltens elektronischer Schaltungen. Shaker, Aachen (1994)
2. Brachtendorf, H.G., Welsch, G., Laur, R., Bunse-Gerstner, A.: Numerical steady state analysis of electronic circuits driven by multi-tone signals. Electr. Eng. **79**(2), 103–112 (1996)
3. Bittner, K., Brachtendorf, H.G.: Optimal frequency sweep method in multi-rate circuit simulation. COMPEL **33**(4), 1189–1197 (2014)
4. Bittner, K., Brachtendorf, H.G.: Adaptive multi-rate wavelet method for circuit simulation. Radioengineering **23**(1), 300–307 (2014)

5. Aprille, T.J., Trick, T.N.: Steady-state analysis of nonlinear circuits with periodic inputs. Proc. IEEE **60**(1), 108–114 (1972)
6. Strohband, P.H., Laur, R., Engl, W.: TNPT - an efficient method to simulate forced RF networks in time domain. IEEE J. Solid-State circuits **12**(3), 243–246 (1977)
7. Skelboe, S.: Computation of the periodic steady-state response of nonlinear networks by extrapolation methods. IEEE Trans. Circuits Syst. **CAS-27**(3), 161–175 (1980)
8. Kakizaki, M., Sugawara, T.: A modified Newton method for the steady-state analysis. IEEE Trans. Comput. Aided Des. **4**(4), 662–667 (1985)
9. Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis, 2nd edn. Springer, New York (1992)
10. Telichevesky, R., Kundert, K.S., White, J.K.: Efficient steady-state analysis based on matrix-free Krylov-subspace methods. In: Proceedings of the 32nd Design Automation Conference, DAC-95, pp. 480–484. ACM, New York (1995)
11. Baiz, A.: Effiziente Lösung periodischer differential-algebraischer Gleichungen in der Schaltungssimulation. Ph.D. thesis, TU Darmstadt (2003). Shaker Verlag, Aachen
12. Brachtendorf, H., Bittner, K.: Grid size adapted multistep methods for high Q oscillators. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **32**(11), 1682–1693 (2013)
13. Brachtendorf, H.G., Laur, R.: On consistent initial conditions for circuit's DAEs with higher index. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **48**(5), 606–612 (2001)
14. Saad, Y., Schultz, M.: GMRES a generalized minimal residual algorithm for solving nonsymmetric linear system. SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986)
15. Pulch, R.: Initial-boundary value problems of warped MPDAEs including minimisation criteria. Math. Comput. Simul. **79**, 117–132 (2008)
16. Pulch, R.: Variational methods for solving warped multirate partial differential algebraic equations. SIAM J. Sci. Comput. **31**(2), 1016–1034 (2008)
17. Houben, S.: Simulating multi-tone free-running oscillators with optimal sweep following. In: Schilders, W., ter Maten, E., Houben, S. (eds.): Scientific Computing in Electrical Engineering 2002. Mathematics in Industry, pp. 240–247. Springer, Berlin (2004)
18. Kugelmann, B., Pulch, R.: Existence and uniqueness of optimal solutions for multirate partial differential algebraic equations. Appl. Numer. Math. **97**, 69–87 (2015)