Exploring the Exponential Integrators with Krylov Subspace Algorithms for Nonlinear Circuit Simulation

Xinyuan Wang†, Hao Zhuang‡, Chung-Kuan Cheng†‡
†CSE and ‡ECE Departments, UC San Diego, La Jolla, CA, USA
‡ANSYS, San Jose, CA, USA
email: xiw193@eng.ucsd.edu, hao.zhuang@ansys.com, ckcheng@ucsd.edu

Abstract— We explore Krylov subspace algorithms to calculate \( \varphi \) functions of exponential integrators for circuit simulation. Higham [1] pointed out the potential numerical stability risk of \( \varphi \) functions computation. However, for the applications to circuit analysis, the choice of methods remains open. This work inspects the accuracy of matrix exponential and vector product with Krylov subspace methods, and identifies the proper approach to achieving numerically stable solutions for nonlinear circuits. Empirical results verify the quality of the proposed methods using various orders of \( \varphi \) functions. Furthermore, instead of Newton-Raphson (NR) iterations in conventional methods, an iterative residue correction algorithm is devised for nonlinear system analysis. The stability and efficiency of our methods are illustrated with experiments.

I. INTRODUCTION

SPICE-like circuit simulation [2], [3], [4] is critical during the cycle of integrated circuits (IC) designs. Given a circuit netlist and device models, the circuit’s behaviors can be formulated as differential algebraic equations (DAE). The solution of DAE is computed with numerical integration step by step until the end of whole simulation time span. Therefore, the DAE integration process is one key component in deciding the efficiency and accuracy in SPICE.

The progress of Krylov subspace methods [5], [6] for matrix exponential and related integration functions (so called \( \varphi \) functions) have recently triggered researchers’ interests. Many works are proposed to solve DAE using exponential integrators [7], [8], [9] due to its improvements in higher order approximation solution and stable explicit formulation over conventional linear multi-step methods [2], [3], [4]. However, for the application of matrix exponential to circuit analysis there exists potential numerical issue when evaluating \( \varphi \) functions appearing in exponential integrators [1]. Sidje proposed a direct computation method for \( \varphi \) functions with an augmented matrix [10], providing an efficient way to evaluate the integrated solution. Newton-Raphson (NR) iterations are typically adopted to obtain solutions of the nonlinear function in conventional integration methods. In each NR iteration, circuit simulator needs to linearize and solve the system, which fails to fully utilize the explicit nature of exponential integration method [6].

In this paper, we apply exponential integrators to circuit simulation with a shift-and-invert Krylov subspace (rational) method [11], which is preferred for faster convergence rate when computing matrix exponential and vector product (MEVP) compared to standard Krylov subspace and invert Krylov subspace [8], [12], [13]. The solution expressed as matrix exponential is computed with \( \varphi \) function direct solver [10]. A residual term is used in a compensation iteration for the solution convergence and error control during the time marching. The contributions of this paper are as follows.

- We adopt rational Krylov subspace method for exponential integrators. The rational Krylov subspace method improves the convergence rate. In the meantime, we skip the regularization process [8], [14].
- We characterize the numerical behaviors using different orders of \( \varphi \) functions. The solution errors are related to the order of \( \varphi \) functions, time step sizes and dimension of Krylov subspace. The characterization allows us to balance between computation cost and accuracy.
- We skip Newton-Raphson iterations. Since the matrix exponential integration method is explicit, we skip the Newton-Raphson iterations. The framework with rational Krylov subspace perform matrix factorization only once at each time step. The convergence of the solution is checked by compensation iteration with a correction term evaluated by results from Krylov subspace.

The rest of this paper is organized as follows. Section II introduces the background of circuit time-domain simulation and exponential integrators. Section III presents calculation utilizing \( \varphi \) functions for exponential integration, as well as a compensation iteration algorithm for solution convergence. Section IV illustrates the MEVP computation using rational Krylov subspace method. Section V provides numerical results to validate our method. Section VI concludes this paper.

II. BACKGROUND

Given a circuit netlist and device models, a general formulation of circuit simulation is shown as follows,

\[
\frac{dq(x)}{dt} + f(x) = Bu(t)
\]

where vector \( x \in \mathbb{R}^{n \times 1} \) lists nodal voltages and branch currents. Vector \( q \in \mathbb{R}^{n \times 1} \) and function \( f \in \mathbb{R}^{n \times 1} \) denote the charge/flux and current/voltage terms, respectively. Vector \( u(t) \) represents all the external excitations at time \( t \); \( B \) is an incident matrix that inserts the input signals to the system; and \( n \) is the size of the system.

For a linear circuit, Eq. (1) could be reduced via modified nodal analysis (MNA) as

\[
C\ddot{x}(t) + Gx(t) = Bu(t)
\]

where \( C(x) \in \mathbb{R}^{n \times n} \) is a matrix of capacitance and inductance from \( \frac{\partial C}{\partial x} \). \( G(x) \in \mathbb{R}^{n \times n} \) represents the conductance and the incidence between voltages and currents. Suppose that \( C \) is invertible, we can rewrite (2) as

\[
\dot{x}(t) = g(x, u, t) = Jx(t) + C^{-1}Bu(t)
\]

where \( J \) denotes the Jacobian matrix of \( g(x, u, t) \)

\[
J = -C^{-1}G
\]

Given an initial vector \( x_0 \) at time \( t_0 \) and time step \( h \), the analytical solution \( x_{k+1} \) of Eq. (2) can be written in matrix exponential expression [2],

\[
x_{k+1} = e^{hJ}x_k + \int_0^h e^{(h-\tau)J}b(t_k + \tau)d\tau
\]
where \( b(t) = C^{-1}Bu(t) \). For SPICE-like simulation, we treat input \( u(t) \) as a vector of piece-wise-linear (PWL) functions. The integral in Eq. (4) can be derived into matrix exponential operators as

\[
x_{k+1} = x_k + (e^{hJ} - I)J^{-1}g_k + (e^{hJ} - hJ - I)J^{-2}b_k
\]

where the second term computes the circuit response to the step input \( g_k = C^{-1}Bu(t_k) \), and the third term the ramp input \( b_k = C^{-1}Bu(t_k + h) \).

Equation (5) can be further written in exponential Euler type [7]

\[
x_{k+1} = x_k + h\varphi_1(hJ)g_k + h^2\varphi_2(hJ)b_k
\]

(6)

The \( \varphi_s \) functions are the convolution of the \( s-1 \) th order term in time domain [7], [15], i.e.

\[
\varphi_s(z) = \frac{1}{(s-1)!} \int_0^1 e^{(1-\theta)z} \theta^{s-1} d\theta \text{ for } s \geq 1
\]

(7)

We have the recursion as

\[
z\varphi_s(z) = \varphi_{s-1}(z) - \frac{1}{(s-1)!}I
\]

(8)

We adopt an effective method [10] to compute the MEVP in Eq. (6) with an augmented matrix \( \tilde{A} \). Given matrix \( A \in \mathbb{C}^{n \times n} \), input vector \( e \in \mathbb{C}^{n \times 1} \), and \( e_i \in \mathbb{C}^{p \times 1} \) as an i-th unit vector, we define

\[
\tilde{A} = \begin{pmatrix} A & Ce_i^\top \\ 0 & E \end{pmatrix} \in \mathbb{C}^{(n+p) \times (n+p)}, \quad E = \begin{pmatrix} e_2^\top \\ \vdots \\ e_p^\top \end{pmatrix} \in \mathbb{C}^{p \times p}
\]

The matrix exponential of the augmented matrix with time \( \tau \) has the format

\[
\exp(\tau\tilde{A}) = \begin{pmatrix} \varphi_0(\tau A) & P \\ 0 & e^{\tau J} \end{pmatrix}
\]

(9)

For any order of \( \varphi \) functions \( 1 \leq s \leq p \),

\[
F(1 : n, s) = \tau^s \varphi_s(\tau A)e
\]

(10)

The above derivation provides multiple ways of evaluating the exponential terms in Eq. (6), which can be written in different orders of \( \varphi \) functions. The derivation will be discussed in Section III-A.

### III. INTEGRATION IN CIRCUIT SIMULATION

For nonlinear system Eq. (1) could be expressed at time \( t \) with formulation

\[
C(x)\dot{x}(t) + G(x)x(t) = F(x) + Bu(t)
\]

(11)

where the elements of matrixes \( C(x) \) and \( G(x) \) are functions of state \( x(t) \). Vector \( F(x) \) represents the offset of the nonlinear device models. Starting from \( x_k \) at time \( t_k \) and given time step \( h \), \( x_{k+1} = x(t_k + h) \) can be solved explicitly with Eq. (12) where \( C(x), G(x) \) and \( F(x) \) are linearized at \( x_k \).

\[
C_k\dot{x}_{k+1} + G_kx_{k+1} = F(x_k) + Bu(t_k + h)
\]

(12)

A. Evaluation of Exponential Integrators

We evaluate the terms in Eq. (6) with different orders of \( \varphi \) functions. Basically, a \( \varphi \) function can be derived from the augmented matrix method directly. An alternative way is to use Eq. (8) to derive with lower ordered \( \varphi \) functions.

\[
h\varphi_1(hJ)g_k = \varphi_0(hJ)g_k - g_k
\]

(13)

\[
h^2\varphi_2(hJ)b_k = h\varphi_1(hJ)b_k - h\tilde{b}_k
\]

(14)

where the new vectors

\[
\tilde{g}_k = J_{k-1}^{-1}g_k = x_k - G^{-1}_k(F(x_k) + Bu(t))
\]

\[
\tilde{b}_k = J_{k-1}^{-1}b_k = -G^{-1}_kB(u(t + h) - u(t))
\]

\[
\tilde{\tilde{b}}_k = J_{k-1}^{-1}\tilde{b}_k = G^{-1}_kC_kG^{-1}_kB(u(t + h) - u(t))
\]

In circuit system, the dimension of matrix \( J \) can be above millions. Direct computation of matrix exponential is infeasible. One efficient way is to approximate the product of matrix function and vector through Krylov subspace based algorithms [5], [16]. The standard Krylov subspace is constructed using \( J_k \) directly as \( K_m(J, v) := \text{span}\{v, Jv, \ldots, J^{m-1}v\} \) with dimension of \( m \). The MEVP is calculated with a \( m \times m \) Hessenberg matrix \( H_m \) obtained by Arnoldi process as \( e^{hJ} \approx |v\rangle v_m e^{hJ} e_1 \), where \( v_m \) is an \( n \times m \) orthonormal basis and \( e_1 \) is the first unit vector. Therefore, the computing complexity of matrix exponential will be reduced drastically, while the accuracy is maintained via high order polynomial approximations [5].

Standard Krylov subspace may not converge fast enough for stiff circuit systems. Since \( H_m \) tends to approximate large magnitude eigenvalues of \( J \) [11], larger dimension of subspace is needed to achieve accuracy. The performance of standard Krylov subspace has been described in [17] on power delivery networks. Besides, in the process of generating standard Krylov subspace we have to compute inverse of \( C \) as part of matrix \( J \). For singular \( C \), extra regularization step will be required [8], [14]. Thus, we adopt the rational Krylov subspace as described in Sec. IV-A. We also notice that inverse of \( C \) is incorporated in the vector \( g_k \) and \( b_k \) which may suffer the same problem. Eq. (13) and (14) enable multiple methods for calculating the exponential terms and \( C^{-1}_k \) is canceled out in \( \tilde{g}_k \), \( \tilde{b}_k \) and \( \tilde{\tilde{b}}_k \). Matrix exponential integrators breaks away from conventional linear multi-step integrators, which is limited by the Dahlquist barrier. The adopted Krylov subspace method is able to obtain high order precision and take longer step sizes [2], [4], [6], [9], [18].

B. Approximation Theory and Compensation Iteration for Convergence

Eq.(11) enforces KCL and KVL laws at time \( t_k \) and \( t_k + h \) with solutions \( x_k \) and \( x_{k+1} \) respectively. In order to evaluate the undershoot or overshoot, a term \( \Delta x_{k+1} \) is used to express the difference between solution \( x_{k+1} \) from Eq. (12) and real solution at \( t_k + h \) which satisfies

\[
C_{k+1}(\dot{x}_{k+1} + \Delta \dot{x}_{k+1}) + G_{k+1}(x_{k+1} + \Delta x_{k+1}) = F(x_{k+1}) + Bu(t_k + h)
\]

(15)

which is equivalent to the following relation

\[
C_{k+1}\Delta \dot{x}_{k+1} + G_{k+1}\Delta x_{k+1} = -(C_{k+1}\dot{x}_{k+1} + G_{k+1}x_{k+1}) + F(x_{k+1}) + Bu(t_k + h)
\]

(16)

Therefore, \( \Delta x_{k+1} \) could be approximated similarly to how the differential equation is solved with exponential integrators. The right
hand side of the above equation is defined as the negative residue 
\( r_{k+1} \) of Eq.(12).

\[
r_{k+1} = C_{k+1}\dot{x}_{k+1} + G_{k+1}x_{k+1} - (F(x_{k+1}) + Bu(t_k + h))
\]

(17)

Let us treat the difference as a ramp input [19], i.e.

\[
\Delta u(t_k + h) \approx -\frac{1}{h}r_{k+1}
\]

(18)

Then, \( \Delta x_{k+1} \) will be added to original solution as a compensation term

\[
x_{k+1} = x_{k+1} + \Delta x_{k+1}
\]

\[
\approx x_{k+1} + h^2\varphi^2(hJ)C_{k-1}\Delta u(t_k + h)
\]

(19)

The process will be repeated until the solution converges. All the parameters with subscript \( k+1 \) in above derivation are evaluated by device models according to the updated parameters with subscript \( \Delta t \). Therefore, the compensation term is the response to the change of the system parameters from state \( x_k \) to state \( x_{k+1} \). Sec. IV-C provides an iteration process showing how the correction term works to achieve convergence.

IV. Exponential Integrators with Rational Krylov Subspace

In this section, we adopt the rational Krylov subspace approach to calculate the matrix exponential. We then compare the error using different orders of \( \varphi \) function. Finally, we use one order of calculation to describe the simulation algorithm.

A. Computation of MEVP via Rational Krylov Subspace Method

To improve the efficiency, a rational Krylov subspace basis is designed to confine the spectrum of \( J \). Instead of using \( J \) directly, \( (I - \gamma J)^{-1} \) is used to generate the Krylov subspace [11] which is implemented as \((G + \frac{\gamma}{\gamma})^{-1}(\frac{\gamma}{\gamma})\)

\[
K_m((I - \gamma J)^{-1}, v) := \text{span}\{v, (I - \gamma J)^{-1}v, \ldots, (I - \gamma J)^{-(m-1)}v\}
\]

(21)

where \( \gamma \) is a predefined parameter. With the shift-and-invert of \( J \) all its eigenvalues with small eigenvalues become the large ones and limited by one. The rational Krylov subspace basis \( V_m \) is constructed with Arnoldi process. The corresponding \( H_m \) effectively approximates small magnitude eigenvalues of \( J \), which leads to a fast and accurate computation of matrix exponential and vector product. The relation between \( J \) and \( H_m \) is

\[
(I - \gamma J)^{-1}V_m = V_mH_m + h_{m+1,m}v_m^\top e_m^\top
\]

(22)

where \( e_m \) is the m-th unit vector. The matrix exponential can be approximated as

\[
e^hJv \approx ||v||V_mh^\frac{I - H_m^{-1}}{\gamma}e_1
\]

(23)

More generally, the computation of exponential integrators \( h^s\varphi_s(hJ)v \) as discussed in Sec. III-A is through [6]

\[
h^s\varphi_s(hJ)v \approx h^s||v||V_m\varphi_s(h\frac{I - H_m^{-1}}{\gamma})e_1
\]

(24)

Algorithm 1: Arnoldi process for rational Krylov subspace

<p>| Input: | C, G, v, h, γ |</p>
<table>
<thead>
<tr>
<th>Output:</th>
<th>H_m, V_m</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( v_1 = \frac{v}{|v|} )</td>
</tr>
<tr>
<td>2</td>
<td>for ( j = 1 : m ) do</td>
</tr>
<tr>
<td>3</td>
<td>Solve ((G + \frac{\gamma}{\gamma})w = \frac{\gamma}{\gamma}v_j ) and obtain ( w );</td>
</tr>
<tr>
<td>4</td>
<td>for ( i = 1 : j ) do</td>
</tr>
<tr>
<td>5</td>
<td>( h_{i,j} = w^\top v_i );</td>
</tr>
<tr>
<td>6</td>
<td>( w = w - h_{i,j}v_i );</td>
</tr>
<tr>
<td>7</td>
<td>end</td>
</tr>
<tr>
<td>8</td>
<td>( h_{j+1,j} =</td>
</tr>
<tr>
<td>9</td>
<td>( v_{j+1} = \frac{w}{</td>
</tr>
<tr>
<td>10</td>
<td>if (</td>
</tr>
<tr>
<td>11</td>
<td>( m = j );</td>
</tr>
<tr>
<td>12</td>
<td>break;</td>
</tr>
<tr>
<td>13</td>
<td>end</td>
</tr>
</tbody>
</table>

The evaluation of Eq. (24) can be directly derived with augmented matrix in Eq. (9). Therefore we can express the solution in Eq. (6) as

\[
x_{k+1} = x_k + h^s||v||V_m\varphi_s(h\frac{I - H_m^{-1}}{\gamma})e_1
\]

\[+ h^s||h_{m+1,m}(G + \frac{C}{\gamma})v_{m+1}^\top e_m^\top H_m^{-1}e_1||^2
\]

(25)

where \( m_1 \) and \( m_2 \) represent dimensions of Krylov subspace for each exponential term.

The residue of solution is derived with Eq. (22) for matrix exponential to approximate the truncated error of Krylov subspace. For a homogeneous system \( C\dot{x} = -Gx \) with \( x(t) \) in Eq.(23), we have the residue

\[
r_m = C\dot{x} + Cx
\]

\[= ||v||CV_m\frac{1}{\gamma}H_m^{-1} + GV_m)e^h\frac{I - H_m^{-1}}{\gamma}e_1
\]

\[= -||v||h_{m+1,m}(G + \frac{C}{\gamma})v_{m+1}^\top e_m^\top H_m^{-1}e_1 ||^2
\]

(26)

The derivation can be applied to Eq. (25), we have residue for each exponential term as

\[
r(\varphi_s, m) = h^s||g_k||h_{m+1,m}(G + \frac{C}{\gamma})v_{m+1}^\top e_m^\top H_m^{-1}e_1 \frac{1}{\gamma}e_1
\]

(27)

B. Comparison among Exponential Integrators

Algorithm 1 shows the Arnoldi process of creating rational Krylov subspace for exponential integrators. According to [5], [18], the Krylov subspace method with dimension \( m \) approximates the exponential integrators in Eq. (24) up to the \( (m - 1) \) th degree of the Taylor expansions. When the residue is below a given error tolerance, Arnoldi algorithm terminates with the dimension \( m \) as shown in line 10 of Algorithm 1.

An RC mesh circuit [18] is used to check the numerical difference of evaluating the same exponential integrator with different \( \varphi \) functions. We set the initial state \( x(0) \) of circuit and input \( u(t) \) all zeros, \( u(t) \) is piece-wise linear input as \( PWL([0, 0, 0.4, T, I]) \) where \( T \) is the whole time span and \( I \) is peak current. From the derivation in previous sections, the solution with step \( h \) is

\[
x(h) = h^s\varphi_s(hJ)(C^{-1}du_1)
\]
C. Integration Algorithm for Circuit Simulation

The integration framework of transient simulation with matrix exponential based integration method is shown in Algorithm 2. The LU decomposition is performed on \((G_k + \frac{\Delta}{h})\) for rational Krylov subspace. Eq.(25) is used to compute solution with step \(h\) and the exponential integrators are evaluated separately with \(\varphi\) functions. Lines 5-10 show the compensation iteration for circuit nonlinear elements as discussed in Sec. III-B. The residue term \(r_{k+1}\) is element-wisely compared to an error bound \(Err\). Once the relation \(r_{k+1} \leq Err\) is not satisfied, compensation term is computed and added to \(x_{k+1}\) until solution converges.

The framework also incorporates an adaptive step method. If the solution cannot converge within \(\text{Iter}_{\text{max}}\) iterations, time step \(h\) is shrunk and solution has to be recalculated. Once solution converges with a small number of iterations, step \(h\) will be increased for next step \(x_{k+2}\) to accelerate the simulation process.

**Algorithm 2**: Integration Kernel for rational Krylov subspace using Compensation Iteration.

**Input**: Circuit netlist, input sources, \(x_k\) at time \(t_k\) and expected time step \(h\).

**Output**: Solution \(x_{k+1}\) at \(t_k + h\).

1. Load the netlist and obtain \(C, G_k\) and \(F(x_k)\) with \(x_k\);
2. Perform \texttt{LU_decompose} of \((G_k + \frac{\Delta}{h})\);
3. Use Algorithm 1 to compute Eq.(13) and Eq.(14);
4. Set iteration number \(i = 0\);
5. While \(r_{k+1} > Err\) by Eq.(17) and \(i < \text{Iter}_{\text{max}}\) do
6. Compute compensation term with Algorithm 1
7. \(\Delta x_{k+1} = -h^2 \varphi_2(h)\varphi_1^{-1} x_h + h\);
8. \(x_{k+1} = x_k + \Delta x_{k+1}\);
9. Update \(r_{k+1}\) at \(x_{k+1}\) with device model;
10. Increase the iteration number \(i = i + 1\);
11. If \(r_{k+1} \leq Err\) when \(i = \text{Iter}_{\text{max}}\) then
12. \(i = 0\); \(h = \mu h\); // Computed solution \(x_{k+1}\) is rejected. Shrink \(h\) by \(\mu = 0.5\).
13. End
14. Else
15. \(x(t_k + h) = x_{k+1}; t_{k+1} = t_k + h; k = k + 1\);
16. If \(i \leq \text{Iter}_{\text{min}}\) then
17. \(h = ah\); // \(i\) is small, \(h\) is increased by \(\alpha > 1\) to accelerate the process. Here \(\alpha = 1.2\).
18. End
19. End

V. NUMERICAL RESULTS

In this section, the numerical results are compared for rational Krylov subspace method which evaluates the exponential terms with different \(\varphi\) functions. We define \(\gamma\) as half of time step and restrict maximum allowed step within \(\text{Ins}\). Table I provides the specification of nonlinear test cases from industry. In order to verify numerical difference among exponential integrators computed with \(\varphi\) functions, all test cases except for D8 are stiff designs with non-singular \(C\) matrices. Extra regularization process is not required when the inverse of \(C\) is incorporated with vectors as in Eq. (3) and (6). Size of the test cases varies from 43 to 40k, represented by \#Node. \#Dev is the number of MOSFETs in each circuit. The next 2 columns in Table I are numbers of non-zero elements of \(C\) and \(G\) in simulation. We can find D4 - D6 has relatively denser matrices. \(T\) is the time span of transient simulation.
TABLE I: Specification of Test Cases

<table>
<thead>
<tr>
<th>Index</th>
<th>Design</th>
<th>#Node</th>
<th>#Dev</th>
<th>nzn(C)</th>
<th>nzn(G)</th>
<th>T (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>voter2S</td>
<td>43</td>
<td>74</td>
<td>345</td>
<td>345</td>
<td>$1 \times 10^{-9}$</td>
</tr>
<tr>
<td>D2</td>
<td>counter</td>
<td>93</td>
<td>220</td>
<td>0.7k</td>
<td>0.7k</td>
<td>$1 \times 10^{-8}$</td>
</tr>
<tr>
<td>D3</td>
<td>add122</td>
<td>101</td>
<td>288</td>
<td>1.1k</td>
<td>1.1k</td>
<td>$1 \times 10^{-8}$</td>
</tr>
<tr>
<td>D4</td>
<td>add20</td>
<td>521</td>
<td>958</td>
<td>7.2k</td>
<td>3.6k</td>
<td>$1.6 \times 10^{-7}$</td>
</tr>
<tr>
<td>D5</td>
<td>memplus</td>
<td>2.8k</td>
<td>7.4k</td>
<td>35k</td>
<td>26k</td>
<td>$4.75 \times 10^{-7}$</td>
</tr>
<tr>
<td>D6</td>
<td>ram2k</td>
<td>4.8k</td>
<td>13.8k</td>
<td>47.6k</td>
<td>47.6k</td>
<td>$6 \times 10^{-7}$</td>
</tr>
<tr>
<td>D7</td>
<td>Inv. chain</td>
<td>11k</td>
<td>24</td>
<td>63k</td>
<td>34k</td>
<td>$1 \times 10^{-9}$</td>
</tr>
<tr>
<td>D8</td>
<td>Power Grid</td>
<td>40k</td>
<td>400</td>
<td>47k</td>
<td>140k</td>
<td>$1 \times 10^{-8}$</td>
</tr>
</tbody>
</table>

Same tolerance is set in Algorithm 1 for checking the accuracy of MEVP computed by rational Krylov subspace. Convergence of nonlinear system is achieved using compensation iteration with a correction term derived from residue in Eq. (17). The exponential integrators are evaluated with $\varphi_0$, $\varphi_1$ and $\varphi_2$ functions separately in the format from Eq. (13) and (14). Computation of $\varphi$ functions is realized by augmented matrix method. Notice in Table II, $\varphi_2$ method is only for Eq. (14) and Eq. (13) is computed with $\varphi_1$ method. The simulation results are listed in Table II. DC represents the DC analysis which includes computation of solution and residue term in compensation iteration. Total time steps and runtime are displayed as well. Iter avg is the average iteration number for each step which reflects convergence rate of circuits. Designs with more complex $C$ and $G$ matrices tend to have larger Iter avg, like D4 - D6. Relatively higher $m$ for Krylov subspace is observed for those cases. For all the test cases, $\varphi_0$ method costs the least running time and $m_0$. The results are consistent with the error distribution of $\varphi$ functions as shown in Fig. 1. To achieve same accuracy, Krylov subspace with $\varphi_0$ requires smaller $m$.

In Fig. 3, the waveform of D4 is extracted to compare with traditional Backward Euler method with Newton-Raphson iteration (BENR). Smaller time step (0.1 ps) is applied to BENR as a reference solution. Solution computed with our proposed algorithm well fits the reference.

We implement the algorithms for circuit transient simulation in MATLAB 2014a and use UMFPACK package for matrix factorization. The experiments are performed on a Linux server with Intel(R) Xeon(R) CPU E5-2640 v3 2.60GHz and 125 GB memory. Device evaluation and matrix stamping are done in C/C++ with BSIM3 model for MOSFET. The interactions are through MATLAB Executable (MEX) external interface with GCC 4.4.7.

VI. CONCLUSION

We propose an efficient algorithmic framework for nonlinear circuit time domain simulation using exponential integrators. The MEVP is computed by rational Krylov subspace. In order to reduce the number of LU decomposition operations, we remove Newton-Raphson iterations. A residue term based compensation iteration algorithm is devised to maintain the convergence.

The computation of $\varphi$ functions is realized with rational Krylov subspace using the augmented matrix. Relative error compared to exact solution is used to illustrate numerical difference among integrated solution with $\varphi$ functions. Results in Sec. V show potential advantage when choosing $\varphi$ function for the computation of exponential integrators in order to achieve low computation cost while ensuring accuracy. For example, with relatively larger time step $\varphi_0$ method is preferred for its dramatic decrease in residue versus increasing $h$. While for tiny $h$, $\varphi_1$ and $\varphi_2$ methods demonstrate potential for lower error. For the future work, we will address the numerical stability when matrix $C$ is singular. In addition, nonlinear integration methods will be explored to accelerate the convergence of the iterations.

VII. ACKNOWLEDGMENTS

We acknowledge the support from NSF CCF-1564302. We also thank the reviewers for their suggestions.

REFERENCES

| Design | DC(s) | $\varphi_0$ method | | | | $\varphi_1$ method | | | | $\varphi_2$ method | | |
|--------|--------|------------------|--|--|--|--|------------------|--|--|--|--|------------------|--|--|--|--|
|        |        | $m_a$ | Step | Tran(s) | Iter | $\alpha_{avg}$ | $m_a$ | Step | Tran(s) | Iter | $\alpha_{avg}$ | $m_a$ | Step | Tran(s) | Iter | $\alpha_{avg}$ |
| D1     | 0.01   | 24.3  | 3896  | 45.3 | 1.93 | 34.1  | 3896  | 65.6 | 1.93 | 377    | 3891  | 72.3 | 1.92 |
| D2     | 0.28   | 23.7  | 344   | 4.4  | 0.78 | 31.0  | 344   | 6.9  | 0.79 | 320    | 344   | 6.9  | 0.79 |
| D3     | 0.02   | 25.0  | 844   | 14.3 | 2.24 | 33.5  | 844   | 19.3 | 2.24 | 379    | 844   | 21.2 | 2.24 |
| D4     | 0.33   | 31.2  | 734   | 36.1 | 3.22 | 42.3  | 734   | 42.8 | 3.22 | 555    | 733   | 53.6 | 3.17 |
| D5     | 1.39   | 24.9  | 1853  | 496.7| 1.89 | 33.1  | 1859  | 568.7| 1.86 | 464    | 1861  | 703.6| 1.91 |
| D6     | 1.75   | 29.1  | 3265  | 1633 | 1.88 | 40.3  | 3191  | 1975 | 1.94 | 47.1   | 3168  | 2071 | 1.97 |
| D7     | 0.13   | 13.9  | 180   | 60.2 | 1.12 | 20.4  | 180   | 98.3 | 1.12 | 18.2   | 180   | 86.3 | 1.12 |
| D8     | 11.78  | 27.9  | 196   | 245.3| 0.93 | *     | *     | *     | C     | *     | *     | C     |

*C* matrix of D8 is singular so evaluation of exponential integrators consisting inverse of *C* is not applicable.


