

# Efficient Second-Order Iterative Methods for IR Drop Analysis in Power Grid \*

Yu Zhong

Dept. of Electrical and Computer Engineering  
Univ. of Illinois at Urbana-Champaign  
Urbana, IL, 61801  
e-mail: yuzhong@uiuc.edu

Martin D. F. Wong

Dept. of Electrical and Computer Engineering  
Univ. of Illinois at Urbana-Champaign  
Urbana, IL, 61801  
e-mail: mdfwong@uiuc.edu

**Abstract**— Due to the extremely large sizes of power grids, IR drop analysis has become a computationally challenging problem both in terms of runtime and memory usage. It has been shown in [5] that first-order iterative algorithms based on node-by-node and row-by-row traversals of the power grid have both accuracy and runtime advantages over the well-known Random-Walk method. In this paper, we propose second-order iterative algorithms that can significantly reduce the runtime. The new algorithms are extremely fast, and we prove that they guarantee converge to the exact solutions. Experimental results show that our algorithms outperform the Random-Walk algorithm in [2] and algorithms in [5]. For a 25-million node problem, while the Random-Walk algorithm takes 2 days with maximum error of 6.1 mV, the fastest algorithm in [5] takes 50 minutes, and our second-order row-based algorithm takes 32 minutes to get an exact solution. Moreover, we can get a solution with maximum error 2 mV in 10 minutes.

## I. INTRODUCTION

A reliable power grid is an important part of high performance VLSI design. With the rapid increase in the complexities of VLSI circuits, accurate and efficient analysis of power grid is becoming a critical issue in nanometer design.

Let us consider a power grid in steady state, which can be solved by DC analysis. The power grid model is illustrated in Figure 1, which consists of wire resistances, VDD pads, and current sources that represent the currents drawn by logic gates and functional blocks. The DC analysis problem can be formulated as:

$$\mathbf{Ax} = \mathbf{b} \quad (1)$$

where  $\mathbf{A}$  is the conductance matrix for the interconnection resistors,  $\mathbf{x}$  is the vector of node voltages, and  $\mathbf{b}$  is a vector of independent sources. Solving this set of linear equations can become prohibitively expensive for extremely large power grids with millions of nodes. Several methods have been proposed to achieve an acceptable runtime with reduced accuracy [1, 2, 4].

Due to the structure of power grids, iterative methods turn out to be good solution methodologies. Instead of constructing a large matrix as in traditional iterative methods, first-order node-based and row-based methods have been proposed recently [5]. They have both accuracy and runtime advantages over the well-known Random-Walk method in [2], which was the state-of-the-art before [5]. In this paper, we present second-order iterative algorithms to further improve the rate of convergence and shorten the runtime. Due to the special characteris-

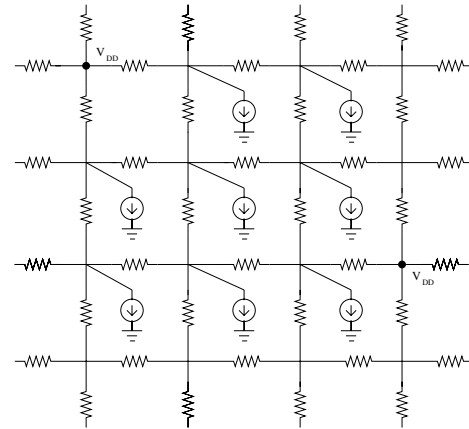


Fig. 1. A typical DC circuit model for power grid.

tics of the power grid problem, the main challenge here is to derive the important iterative coefficients used in the second-order method. The iterative coefficients have a great influence on the convergence and the convergence rate of the iterative algorithms. We prove the convergence of our second-order methods. Experimental results show that our algorithms outperform not only the Random-Walk method but also the first-order node-based and row-based methods in [5]. For a 25-million node problem, our second-order row-based algorithm takes 32 minutes to get an exact solution while the Random-Walk-based algorithm takes 2 days with maximum error of 6.1 mV.

The rest of the paper is organized as follows. We first review the first-order node-based and row-based methods in [5] in Section 2. Then, we present two second-order iterative algorithms in Section 3. In Section 4, we provide theoretical foundations for the new iterative algorithms. Finally, in Section 5, we demonstrate that our algorithms outperform the existing algorithms in terms of both runtime and accuracy.

## II. PREVIOUS ITERATIVE METHODS

In this section, we give a brief review of the iterative algorithms in [5]. It is shown in [5] that the system matrix  $\mathbf{A}$  in Equation (1) is symmetric and positive definite. This property ensures the convergence of traditional first-order iterative methods.

Note that applying the traditional iterative methods to solve large power grid problem is not practical, because the construction, storage, and computation of such a large matrix is expen-

\*This work was partially supported by the National Science Foundation under grant CCR-0306244.



---

**Second-Order Node-based method**

```

initialize  $\mathbf{x}_0 = \mathbf{x}^{(0)}$ 
initialize  $\mathbf{x}_1 = \mathbf{x}^{(1)}$ 
new  $\mathbf{x}_2$ 
set  $|\epsilon| \rightarrow 0$ 
if  $|x_2^{(k+1)} - x_2^{(k)}| < |\epsilon|$ , return
     $k = k + 1$ ;
/*solve  $x_i^{(n+1)} = (1 - \beta + \alpha)x_i^{(n)} - \alpha x_i^{(n-1)}$  */
for node  $i=1$  to  $N^2$ 
     $x_2[i] = (1 - \beta + \alpha)x_1[i] - \alpha x_0[i]$ 
     $x_0[i] = x_1[i]$ 
/*symmetric improved node-based method.*/
for node  $i=1$  to  $N^2$ 
    forward improved node-based method
for node  $i=N^2$  down to  $1$ 
    backward improved node-based method
/*Solve  $x^{(n+1)}$  */
for node  $i=1$  to  $N^2$ 
     $x_2[i] = x_2[i] + \beta x_1[i]$ 
     $x_1[i] = x_2[i]$ 

```

---

Fig. 5. The algorithm for the second-order node-based method

of the improved node-based method, while the second half iteration is the improved node-based method taken in reverse order as in Figure 4. We will explain the reason that we choose the symmetric improved node-based method instead of the original improved node-based method in Section 4.

For each node  $i$ , Equation (7) can be written as

$$x_i^{(n+1)} = \beta \bar{x}_i^{(n+1)} + (1 - \beta + \alpha)x_i^{(n)} - \alpha x_i^{(n-1)} \quad (8)$$

Suppose we know the iterative coefficients, then we can solve the voltage  $x_i^{(n+1)}$  at node  $i$  in iteration  $n + 1$  by two parts: the symmetric improved node-based iteration  $\bar{x}_i^{(n+1)}$ , and the influence of its own previous two iterations  $x_i^{(n)}$  and  $x_i^{(n-1)}$ .

The overall algorithm is as follows. Suppose a power grid has  $N$  rows and  $N$  columns, the total node number is  $N^2$ . Begin with any initial values (e.g.,  $V_{DD}$  for VDD power grids) in iterations 0 and 1 at the nodes to be solved. For each node  $i$ , we first compute the influence of the past two iterations  $(1 - \beta + \alpha)x_i^{(n)} - \alpha x_i^{(n-1)}$  as in Equation (8). Then in the first half iteration, apply the original improved node-based method to each node in forward order  $1, 2, \dots, N^2$ . Later in the second half iteration, apply the original improved node-based method with  $\omega_1$  to each node in reverse order  $N^2, N^2 - 1, \dots, 1$ . Now we have a solution of the symmetric node-based method for each node. Finally, we combine the influence of the past two iterations  $(1 - \beta + \alpha)x_i^{(n)} - \alpha x_i^{(n-1)}$  with the symmetric node-based part together to obtain the updated solution  $x_i^{(n+1)}$  of second-order node-based method at node  $i$  as in Equation (8). Figure 5 shows the algorithm for the second-order node-based method. Though compared to the original improved node-based method, we must do at least twice as much work per iteration in the second-order method, the faster convergence speed justifies the additional work.

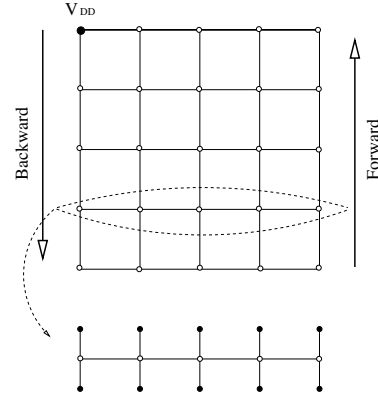


Fig. 6. Symmetric row-based method.

### B. Second-Order Row-based Method

Now consider how to upgrade the first-order row-based method to a second-order method. We still group the nodes on the same row and solve them together depending on the boundary conditions of the adjacent rows in linear time. Similar to the second-order node-based method, we want to combine the improved row-based method in [5] with a second-order method to improved its rate of convergence.

Here we choose some a first-order method called symmetric improved row-based method which is obtained from the improved row-based method in Equation (5). In each iteration of the symmetric improved row-based method, the first iteration is the same as the improved row-based method, while the second half iteration is the improved row-based method taken in reverse order as Figure 6.

In second-order row-based method, for each row  $i$ , Equation (7) can be written as

$$\mathbf{x}_i^{(n+1)} = \beta \bar{\mathbf{x}}_i^{(n+1)} + (1 - \beta + \alpha)\mathbf{x}_i^{(n)} - \alpha \mathbf{x}_i^{(n-1)} \quad (9)$$

where  $\bar{\mathbf{x}}_i^{(n+1)}$  denotes the symmetric improved row-based method. Suppose we know the iterative coefficients, then we can solve the voltages  $\mathbf{x}_i^{(n+1)}$  on row  $i$  in iteration  $n + 1$  by two parts: the symmetric improved row-based iteration  $\bar{\mathbf{x}}_i^{(n+1)}$ , and the influence of its own previous two iterations  $\mathbf{x}_i^{(n)}$  and  $\mathbf{x}_i^{(n-1)}$ .

The overall algorithms is as follows. Suppose a power grid has  $N$  rows and  $N$  columns, the total number of rows is  $N$ . Begin with any initial values (e.g.,  $V_{DD}$  for VDD power grids) in iterations 0 and 1 at the nodes to be solved. For each row  $i$ , we first compute the influence of the past two iterations  $(1 - \beta + \alpha)\mathbf{x}_i^{(n)} - \alpha \mathbf{x}_i^{(n-1)}$  as in Equation (9). Then in the first half iteration, apply the first-order improved row-based method to each row in forward order  $1, 2, \dots, N$ . Later in the second half iteration, apply the first-order improved row-based method to each row in reverse order  $N, \dots, 2, 1$ . Now we have a solution of the symmetric improved row-based method for each node. Finally, we combine the influence of the past two iterations  $(1 - \beta + \alpha)\mathbf{x}_i^{(n)} - \alpha \mathbf{x}_i^{(n-1)}$  with the symmetric improved row-based part together to obtain the updated solution  $\mathbf{x}_i^{(n+1)}$  of second-order row-based method for row  $i$ . Figure 7 shows the algorithm for the second-order row-based method.

---

**Second-Order Row-based method**

```

initialize  $\mathbf{x}_0 = \mathbf{x}^{(0)}$ 
initialize  $\mathbf{x}_1 = \mathbf{x}^{(1)}$ 
new  $\mathbf{x}_2$ 
set  $|\epsilon| \rightarrow 0$ 
if  $|x_2^{(k+1)} - x_2^{(k)}| < |\epsilon|$  return
     $k = k + 1$ ;
/*solve  $x^{(n+1)} = (1 - \beta + \alpha)x^{(n)} - \alpha x^{(n-1)}$ .*/
for node  $i=1$  to  $N^2$ 
     $x_2[i] = (1 - \beta + \alpha)x_1[i] - \alpha x_0[i]$ 
     $x_0[i] = X_1[i]$ 
/*symmetric improved row-based method.*/
for row  $j=1$  to  $N$ 
    forward improved row-based method
for row  $j=N$  down to  $1$ 
    backward improved row-based method
/*Solve  $x^{(n+1)}$ .*/
for node  $i=1$  to  $N^2$ 
     $x_2[i] = x_2[i] + \beta x_1[i]$ 
     $x_1[i] = x_2[i]$ ;

```

---

Fig. 7. The algorithm for the second-order row-based method

## IV. THEORETICAL FOUNDATION

In this section, we derive the iterative coefficients and provide convergence analysis.

## A. Second-Order Method

Let  $\bar{\mathbf{x}}$  be the solution to the linear system in Equation (1). Consider a second-order iterative method defined by

$$\mathbf{x}^{(n+1)} = G_1 \mathbf{x}^{(n)} + G_2 \mathbf{x}^{(n-1)} + \mathbf{k}_2 \quad (10)$$

In Equation (10), as  $n \rightarrow \infty$ ,  $\mathbf{x}^{(n+1)}$ ,  $\mathbf{x}^{(n)}$ ,  $\mathbf{x}^{(n-1)} \rightarrow \bar{\mathbf{x}}$ . This gives the following consistency condition on  $G_1$  and  $G_2$ .

$$(I - G_2 - G_1)\mathbf{A}^{-1}\mathbf{b} = \mathbf{k}_2 \quad (11)$$

where  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$  is the solution to Equation (1).

**Theorem IV.1** Consider the following second-order method:

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \alpha(\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)}) + \beta(G\mathbf{x}^{(n)} + k_1 - \mathbf{x}^{(n)}) \quad (12)$$

where  $G$  and  $k_1$  are defined in Equation (6). If  $\bar{\mathbf{x}}$  is a solution to Equation (6), it is also a solution to Equation (12) for any  $\alpha$  and  $\beta \neq 0$ .

**Proof:** Compare Equation (12) to the general form of second-order iterative method in Equation (10), we have

$$\begin{aligned} G_1 &= (1 - \beta + \alpha)I + \beta G \\ G_2 &= -\alpha I \\ \mathbf{k}_2 &= \beta \mathbf{k}_1 \end{aligned} \quad (13)$$

If we substitute Equation (13) in Equation (11) to check the consistency condition, then we obtain

$$(I - G)\mathbf{A}^{-1}\mathbf{b} = \mathbf{k}_1 \quad (14)$$

which is satisfied because we assume that the first-order method can solve the linear system (1). Therefore,  $\bar{\mathbf{x}}$  is also a solution of Equation (12).  $\square$

Now let us analyze the convergence for the second-order method in Equation (10), and furthermore, derive the critical coefficients  $\alpha$  and  $\beta$ . Note that the iterative method in Equation (6) is convergent if and only if  $S(G) < 1$ , where  $S(G)$  is the spectral radius of matrix  $G$ ,  $S(G) = \max_{\lambda \in S_G} |\lambda|$  [3].

For second-order method in Equation (10), we observe that

$$\begin{pmatrix} \mathbf{x}^{(n)} \\ \mathbf{x}^{(n+1)} \end{pmatrix} = \begin{pmatrix} 0 & I \\ G_2 & G_1 \end{pmatrix} \begin{pmatrix} \mathbf{x}^{(n-1)} \\ \mathbf{x}^{(n)} \end{pmatrix} + \begin{pmatrix} 0 \\ \mathbf{k}_2 \end{pmatrix} \quad (15)$$

From above, a necessary and sufficient condition that the method converge for all initial conditions  $\mathbf{x}^0$  and  $\mathbf{x}^1$  is that

$$S(\hat{G}) < 1 \quad (16)$$

where

$$\hat{G} = \begin{pmatrix} 0 & I \\ G_2 & G_1 \end{pmatrix} \quad (17)$$

And  $S(\hat{G}) < 1$  if and only if all eigenvalues  $\lambda$  of  $\hat{G}$  are less than unity in modulus. So we must have all roots  $\lambda$  of

$$\det(\hat{G} - \lambda I) = \det(\lambda^2 I - \lambda G_1 - G_2) = 0 \quad (18)$$

are less than unity in modulus. Substitute Equation (13) into (18), we have

$$\det(\lambda^2 I - \lambda(\beta G + (1 - \beta + \alpha)I) + \alpha I) = 0 \quad (19)$$

Rearranging the terms, we obtain

$$\det\left(G + \left(\frac{1 - \beta + \alpha}{\beta}\right)I - \left(\frac{\lambda^2 + \alpha}{\beta\lambda}\right)I\right) = 0 \quad (20)$$

So the eigenvalues  $\lambda$  of  $\hat{G}$  are related to the eigenvalues  $\mu$  of  $G$  as follows

$$\mu + \frac{1 - \beta + \alpha}{\beta} = \frac{\lambda^2 + \alpha}{\beta\lambda} \quad (21)$$

Suppose the eigenvalues  $\mu$  of  $G$  are real numbers that lie within the interval  $[\mu_{min}, \mu_{max}]$ , where  $\mu_{max} < 1$ . Rewrite Equation (21) as

$$\left[\mu + \frac{1 - \beta + \alpha}{\beta}\right]^2 = \left[\frac{1}{\beta} \left(|\lambda| + \frac{\alpha}{|\lambda|}\right)\right]^2 \quad (22)$$

It has been shown in [3] that to minimize the number of iterations, we need to minimize  $|\lambda|$ , which shows the way of choosing coefficients  $\alpha$  and  $\beta$ . Then the optimum choice of  $\alpha$  and  $\beta$  satisfies the conditions

$$\begin{aligned} \mu_{min} &= \frac{\beta - 1 - \alpha}{\beta} - \frac{1}{\beta} \left(|\lambda| + \frac{\alpha}{|\lambda|}\right) \\ \mu_{max} &= \frac{1}{\beta} \left(|\lambda| + \frac{\alpha}{|\lambda|}\right) + \frac{\beta - 1 - \alpha}{\beta} \\ &|\lambda| = \frac{d}{|\lambda|} \end{aligned} \quad (23)$$

Therefore,

$$\begin{aligned} \alpha &= |\lambda|^2, \beta = 2(1 + |\lambda|^2)/[2 - (\mu_{max} + \mu_{min})] \\ \{(\mu_{max} - \mu_{min})/[2 - (\mu_{min} + \mu_{max})]\}(1 + |\lambda|^2) &= 2|\lambda| \end{aligned} \quad (24)$$

For any first-order iterative method which has real eigenvalues, if we know the eigenvalue bounds  $\mu_{min}$  and  $\mu_{max}$  of its iterative matrix, we can obtain the coefficients in the second order method by Equation (24).

### B. Symmetric SOR Method

Now we want to show the reason that we used symmetric improved node-based or row-based method in Section 3. It is because the iterative matrix of symmetric method has real eigenvalues, which allows us to use the coefficients  $\alpha$  and  $\beta$  as Equation (24).

Let us check SOR method first. Define matrix  $B$  as

$$B = I - D^{-1}\mathbf{A} \quad (25)$$

where  $D = \text{diag}(\mathbf{A})$ . The diagonal elements of matrix  $B$  are all 0. Thus, we separate  $B$  into  $L$  and  $U$ , one strictly lower and one strictly upper triangular matrices respectively,

$$B = L + U \quad (26)$$

Then we can express SOR method as

$$\mathbf{x}^{(n+1)} = \mathcal{L}_\omega \mathbf{x}^{(n)} + (I - \omega L)^{-1} \omega c \quad (27)$$

where

$$\mathcal{L}_\omega = (I - \omega L)^{-1}(\omega U + (1 - \omega)I) \quad (28)$$

If  $\omega > 1$ , then the eigenvalues of the SOR method are not all real and hence the methods of choosing coefficients in Equation (24) are not applicable.

To make the eigenvalues of the first-order iterative method all real, instead of generic SOR method, we consider the symmetric SOR method as two half iterations. We have  $\mathbf{x}^{(n+1/2)}$  from  $\mathbf{x}^{(n)}$  by forward SOR method

$$\mathbf{x}^{(n+1/2)} = \mathcal{L}_\omega \mathbf{x}^{(n)} + (I - \omega L)^{-1} \omega c \quad (29)$$

and  $\mathbf{x}^{(n+1)}$  from  $\mathbf{x}^{(n+1/2)}$  by the backward SOR method

$$\mathbf{x}^{(n+1)} = \mathcal{U}_\omega \mathbf{x}^{(n+1/2)} + (I - \omega U)^{-1} \omega c \quad (30)$$

where

$$\begin{aligned} \mathcal{L}_\omega &= (I - \omega L)^{-1}(\omega U + (1 - \omega)I) \\ \mathcal{U}_\omega &= (I - \omega U)^{-1}(\omega L + (1 - \omega)I) \end{aligned} \quad (31)$$

Because it is shown that the improved node-based method is equivalent to SOR method in [5], so our symmetric improved node-based method is equivalent to symmetric SOR. From Equations (29) and (30) we have

$$\mathbf{x}^{(n+1)} = \mathcal{T}_\omega \mathbf{x}^{(n)} + \omega(2 - \omega)(I - \omega U)^{-1}(I - \omega L)^{-1}c \quad (32)$$

where

$$\mathcal{T}_\omega = \mathcal{U}_\omega \mathcal{L}_\omega \quad (33)$$

It can be shown that if system matrix  $\mathbf{A}$  is a symmetric matrix with positive diagonal elements. For any real  $\omega$  the eigenvalues of  $\mathcal{T}_\omega$  are real and nonnegative. Moreover, if  $0 < \omega < 2$  then  $S(\mathcal{T}_\omega) < 1$ , so symmetric SOR converges with any initial condition.

If we choose  $\omega$  in symmetric node-based method as

$$\omega = \frac{2}{1 + 2\sin(\pi h/2)} \quad (34)$$

where  $h = N^{-1}$ . then

$$S(\mathcal{T}_\omega) \sim 1 - \pi h \quad (35)$$

TABLE I

RUN TIME COMPARISON BETWEEN FIRST AND SECOND-ORDER METHODS FOR CIRCUIT C1.

Method	# Iterations	CPU time(s)
First-Order Node	134	2.69
First-Order Row	81	1.86
Second-Order Node	46	1.67
Second-Order Row	34	1.28

In the symmetric row-based method, choose  $\omega$  as

$$\omega_2 = \frac{2}{1 + \sqrt{2}\pi h} \quad (36)$$

Then

$$S(\mathcal{T}_\omega) \sim 1 - \sqrt{2}\pi h \quad (37)$$

Now consider second-order symmetric node-based and row-based method, then  $G = \mathcal{T}_\omega$ . Because  $G$  has real and non-negative eigenvalues, pick  $\mu_{min} = 0$ ,  $\mu_{max} = S(\mathcal{T}_\omega)$ . Once we know  $S(\mathcal{T}_\omega)$ , we can obtain the coefficients  $\alpha$  and  $\beta$  from Equation (24). Also by Equation (24), the spectral radius of second degree iterative matrix  $S(\hat{G})$  is

$$S(\hat{G}) = \left( \frac{S(\mathcal{T}_\omega)^{1/2}}{1 + (1 - S(\mathcal{T}_\omega))^{1/2}} \right)^2 \quad (38)$$

Compare  $S(\hat{G})$  with the spectral radius  $S(\mathcal{L}_\omega)$  of SOR method [3],

$$S(\mathcal{L}_\omega) = \left( \frac{S(B)}{1 + (1 - S(B)^2)^{1/2}} \right)^2 \quad (39)$$

Usually,  $S(\mathcal{T}_\omega)$  is substantially less than  $S(B)^2$ , so  $S(\hat{G})$  is smaller than  $S(\mathcal{L}_\omega)$ . Then the second degree SSOR method converges much faster than the original SOR method.

## V. EXPERIMENTAL RESULTS

We apply the proposed second-order node-based and row-based methods to a 251,001-node power grid model, named Circuit C1, with the  $V_{DD}$  value of 1.8 V. Our computations are carried out on a Linux PC with 2.8-GHz CPU and 4-GB of RAM. All the algorithms were implemented using C++.

Circuit C1 with 251,001 nodes is analyzed by the four iterative methods, which are separately the first-order improved node-based and row-based methods, the first-order improved node-based and row-based methods.

Figure 8 shows the convergence of the solution by the four iterative methods. We do iterations until the maximum error is zero, which guarantees that the iterative methods all converge to the exact solution.

Still for Circuit C1, Table I shows the runtime comparison between the first-order and second-order methods. It is shown that the second-order node-based method converges almost two times faster than the first-order node-based method. And the second-order row-based method converges faster than the second-order node-based method, as we expected. We can take the results as reference to exact solutions.

To compare runtime, we implemented the well-known Random-Walk algorithm, and applied it to our problems as a

TABLE II  
 RUNTIME COMPARISON. ITERATIVE METHODS HAVE NO ERRORS.

Circuits	#nodes	#sources	Random Walk		First-Order		Second-Order	
			MaxE (mV)	time (min:sec)	Node time (min:sec)	Row time (min:sec)	Node time (min:sec)	Row time (min:sec)
C1	251K	441	5.4	4:55	0:02	0:02	0:02	0:01
C2	251K	121	6.0	11:04	0:09	0:04	0:04	0:03
C3	1M	121	7.5	37:40	2:02	1:24	1:08	0:49
C4	4M	441	6.3	152:22	3:54	2:30	2:01	1:35
C5	16M	441	4.9	1019:16	40:09	23:52	21:58	15:05
C6	25M	441	6.1	2943:56	76:50	47:39	42:16	32:20

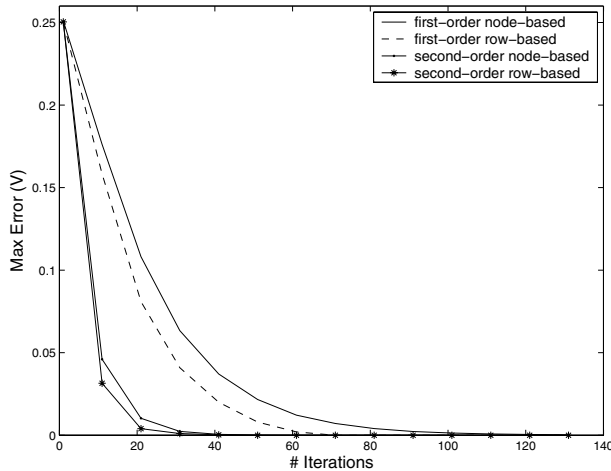


Fig. 8. The convergence of the first-order and second-order node-based and row-based method.

reference. Table II compares the performance of the first and second-order node and row-based method with the Random-Walk program on the same problems. We take the results by the iterative methods as reference of error analysis because they all converged to the exact solutions. There are six different test circuits of power grids in column 1. Column 2 is the number of nodes in those circuits and column 3 is the number of voltage sources. Column 4 shows the maximum error by Random Walk method. The comparison of columns 5, 6, 7, 8, and 9 is the runtime by the Random Walk, the first-order node-based method, the first-order row-based method, the second-order node-based method, and second-order row-based method. Experimental results show that our second-order methods outperform not only the Random-Walk-based algorithm but also the first-order ones. For example, in circuit C6, the runtime of Random Walk method is about 2 days, while the the second-order row-based method takes only 32 minutes without error.

In practice, if the required error margin is larger, the runtime of iterative methods will get even shorter. From Figure 8, we can see the tradeoff between maximum error and runtime. We can cut the runtime into 30% if 2 mV maximum error is permitted. So we give the second-order row-based method some small error bound and compare the error and runtime with random walk again in Table III. In circuit C6, the runtime of Random Walk method is about 2 days, while the the second-order row-based method takes less than 10 minutes with maximum error 2 mV.

TABLE III  
 RUNTIME AND ERROR COMPARISON.

Circuits	#nodes	Random Walk		2nd-Order Row	
		MaxE (mV)	time (min:sec)	MaxE (mV)	time (min:sec)
C1	251K	5.4	4:55	1.8	0:00
C2	251K	6.0	11:04	2.0	0:01
C3	1M	7.5	37:40	1.9	0:13
C4	4M	6.3	152:22	2.0	0:25
C5	16M	4.9	1019:16	1.9	4:42
C6	25M	6.1	2943:56	2.0	9:55

## VI. CONCLUSION

Second-order node-based and row-based iterative methods for power grid DC analysis are presented. They are shown to be faster than the first-order iterative methods in [5] and the well-known Random-Walk algorithm in [2].

## REFERENCES

- [1] J. Kozhaya, S. R. Nassif, and F. N. Najm. A multigrid-like technique for power grid analysis. *IEEE Transactions on Computer-Aided Design*, 21(10):1148–1160, 2002.
- [2] H. Qian, S. R. Nassif, and S. S. Sapatnekar. Random walks in a supply network. *Annual ACM IEEE Design Automation Conference*, pages 93 – 98, 2003.
- [3] D. M. Young. *Iterative Solution of Large Linear System*. Academic Press, New York and London, 1971.
- [4] M. Zhao, R. V. Panda, S. S. Sapatnekar, and D. Blaauw. Hierarchical analysis of power distribution networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(2):159–168, Feb 2002.
- [5] Y. Zhong and M. D. F. Wong. Fast algorithms for IR drop analysis in large power grid. *International Conference on Computer Aided Design*, 2005.