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

Yu Zhong and Martin D. F. Wong
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

Outline

- Introduction
- First-order iterative methods
- Second-order node-based method
- Second-order row-based method
- Experimental Results
- Conclusions

Introduction

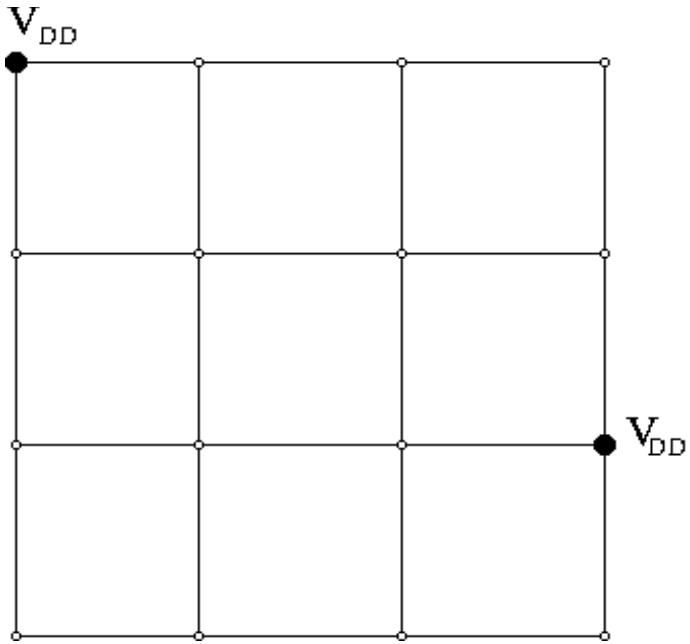
- Power delivery issues are getting significant
  - Increasing complexity of VLSI circuits
  - Increasing power (current) consumption
  - Decreasing supply voltage
  - Reduced noise margin and increased gate delay
- Power grid network must be modeled and analyzed accurately.
- Difficult since power grid networks are very large.

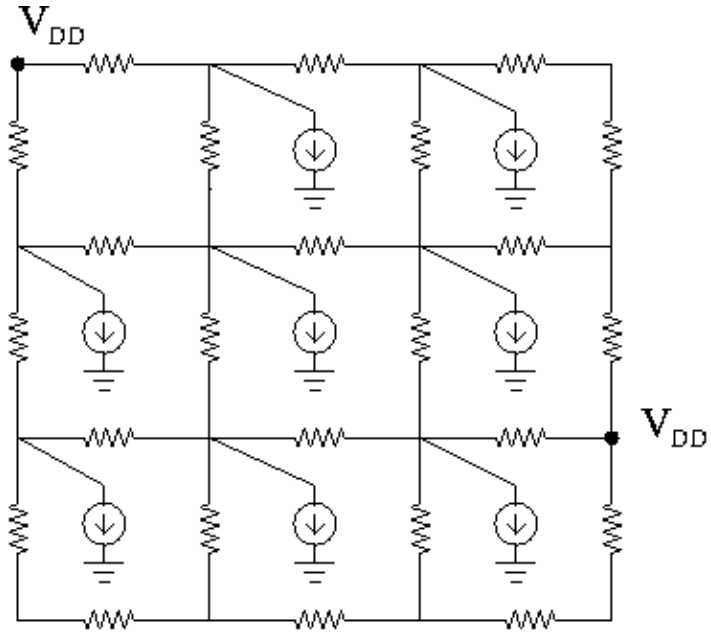
Introduction

- Large scale simulation
  - Usually millions of elements in power grid
  - Even DC analysis is difficult
  - SPICE-level accuracy simulation is required
  - Runtime is slow
- Memory inefficiency
  - e.g. 1 million node → 1 trillion elements in matrix
  - Short of memory
- Tradeoff between runtime and accuracy

Model of Power Grid

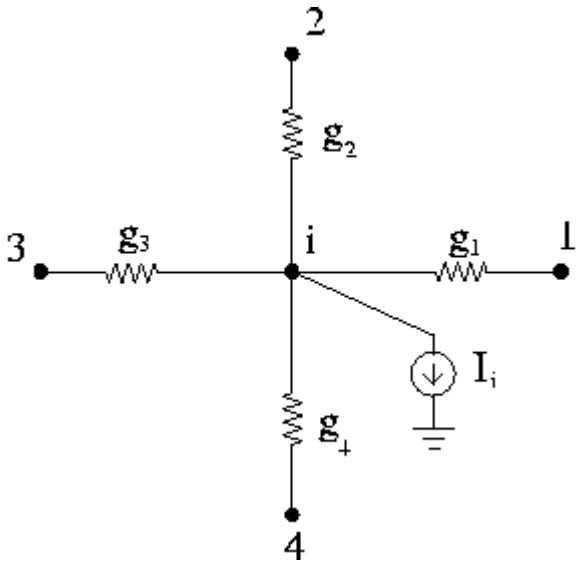
- DC model for power grid.

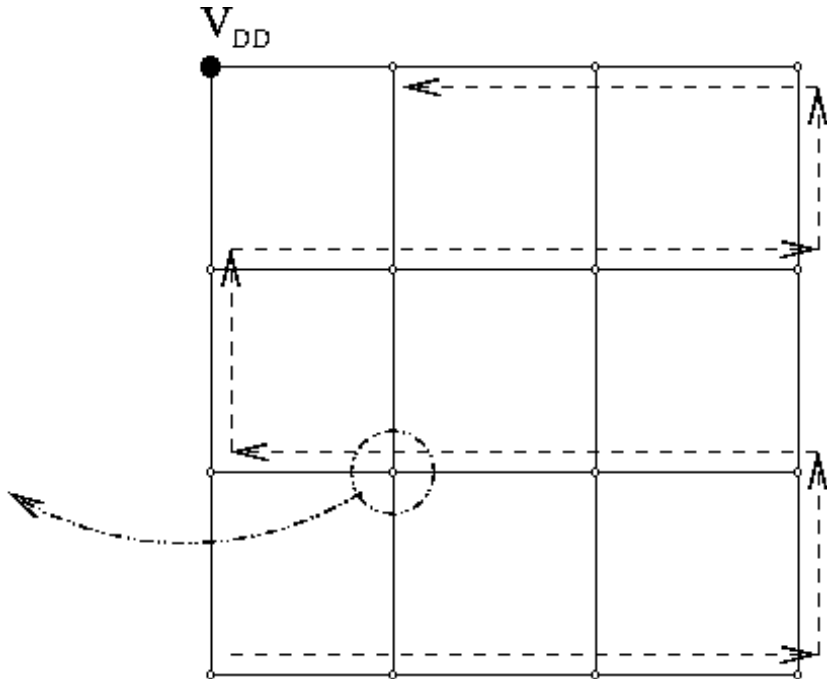




How to solve voltage at each node in the power grid ?

First-order Node-based Methods





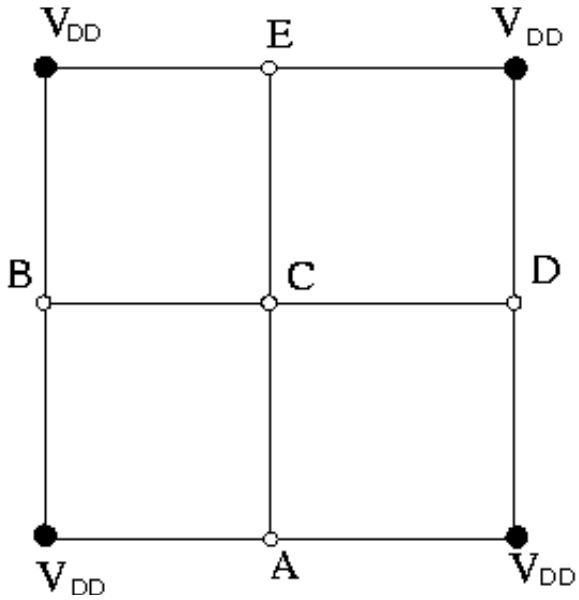
Apply Kirchoff's current law at node i

$$x_i = \frac{(g_1 x_1 + g_2 x_2 + g_3 x_3 + g_4 x_4 - I_i)}{(g_1 + g_2 + g_3 + g_4)}$$

First-order Node-based Methods

- A simple example of the generic first-order node-based method

$$V_{DD} = 1.8 \qquad I = 0.1 \qquad g = 1$$



Initial values for A, B, C, D and E are 0.
First iteration:

V_A: 1.1667=(1/3)(1.8+1.8+0)-0.1/3
V_B: 1.1667=(1/3)(1.8+1.8+0)-0.1/3
V_c: 0.5583=(1/4)(1.1667+1.1667+0+0)-0.1/4
V_D: 1.3528=(1/3)(0.5583+1.8+1.8)-0.1/3
V_E: 1.3528=(1/3)(0.5583+1.8+1.8)-0.1/3

Repeat until convergence!
Final results:
V_A=1.7375, V_B=1.7375, V_C=1.7213, V_D =1.7375, V_E=1.7375

First-order Node-based Methods

$$\bar{x}_i^{(k+1)} = \Big(\sum_{j<i} g_{ij} x_j^{(k+1)} + \sum_{j>i} g_{ij} x_j^{(k)} - I_i\Big) / \sum_j g_{ij}$$

Generic first-order node-based method:

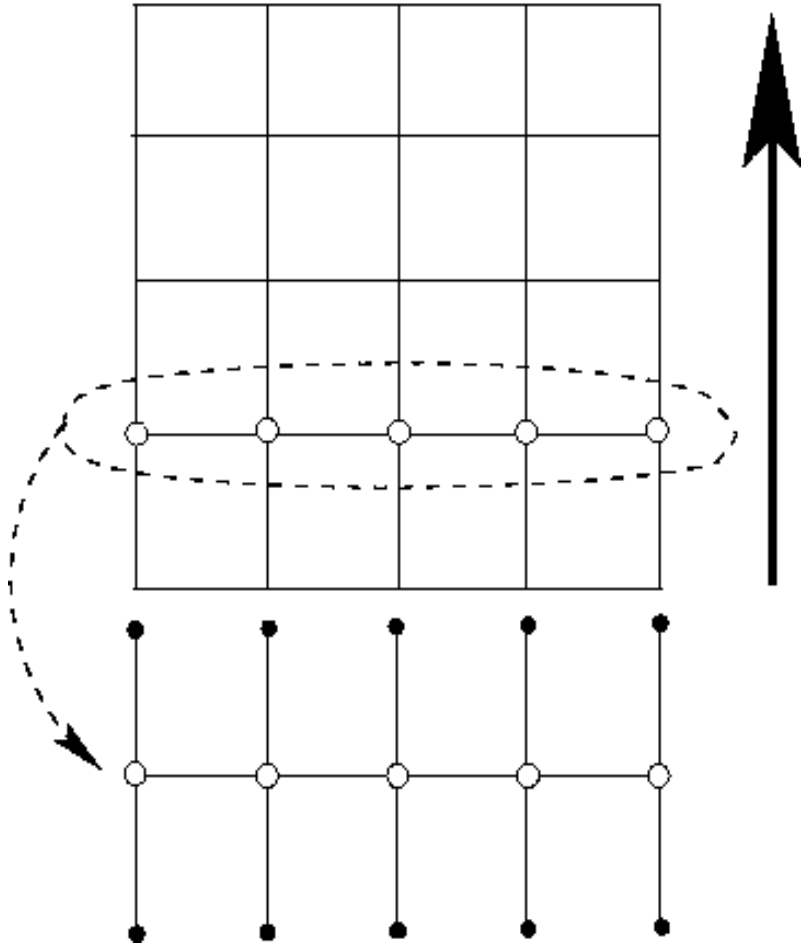
$$x_i^{(k+1)} = \bar{x}_i^{(k+1)}$$

Improved first-order node-based method:

$$x_i^{(k+1)} = \omega \bar{x}_i^{(k+1)} + (1-\omega) x_i^{(k)}$$

We can improve the convergence rate of the generic first-order node-based method by choosing an appropriate extrapolation factor ω.

First-order Row-based Methods



Generic first-order row-based method:

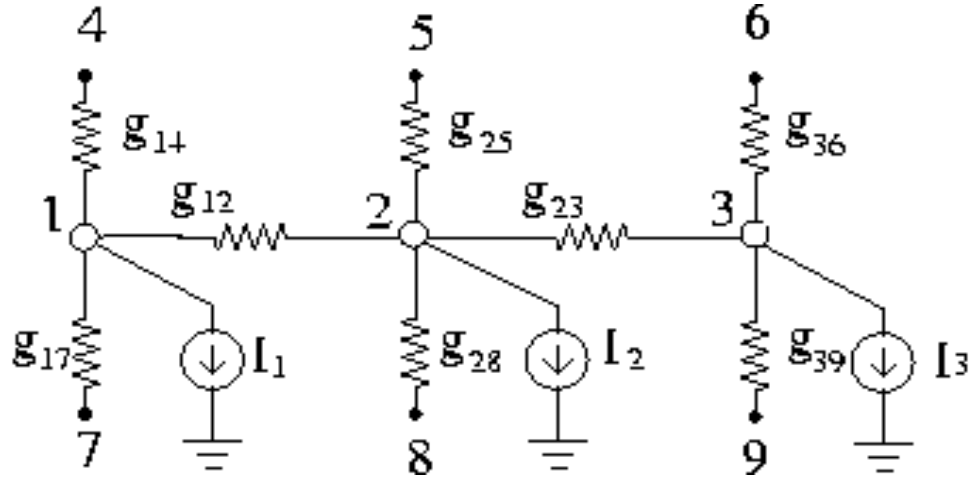
1. Group the nodes on the same row.

2. Solve the whole row together in linear time.

3. Do iteration of row-based traversals until converges.

First-order Row-based Methods

- One row can be solved in linear time because the system matrix is a tri-diagonal matrix.



Linear equations for nodes 1, 2, and 3 on the one row

$$\begin{bmatrix} g_{12}+g_{14}+g_{17} & -g_{12} & \\ -g_{12} & g_{23}+g_{25}+g_{12}+g_{28} & -g_{23} \\ & -g_{23} & g_{36}+g_{23}+g_{39} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -I_1+g_{14}x_4+g_{17}x_7 \\ -I_2+g_{25}x_5+g_{28}x_8 \\ -I_3+g_{36}x_6+g_{39}x_9 \end{bmatrix}$$

First-order Row-based Methods

- Similar to the improved first-order node-based method, we may improve the convergence speed of the generic first-order row-based method by choosing an optimal extrapolation factor ω.

- Improved first-order row-based method:

$$x_i^{(k+1)} = \omega \bar{x}_i^{(k+1)} + (1-\omega) x_i^{(k)}$$

where $\bar{x}_i$ represents x_i in generic first-order row-based method, which represents the voltage vector on the row i.

Second-order Iterative Methods

- Upgrade a first-order iterative method to obtain a second-order iterative method.
- First-order iterative method:

$$\bar{\mathbf{x}}^{(n+1)} = G\mathbf{x}^{(n)} + \mathbf{k}_1$$

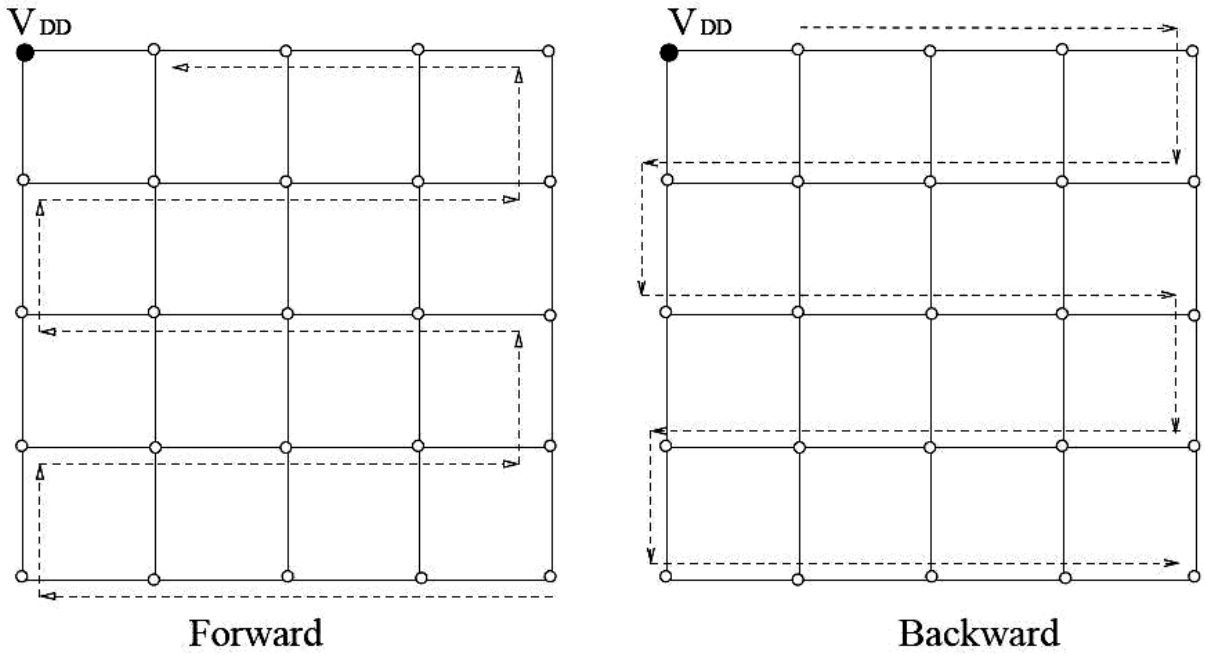
- Consider a second-order iterative method defined by

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

where $\bar{x}^{(n+1)}$ denotes the first-order iteration, and α and β are important coefficients to be decided, which influence the speed of convergence.

Second-order Node-based Method

- Choose the *symmetric improved node-based method* as the first-order method, then upgrade it to the second-order node-based method.



In each iteration, first half iteration is the same as one iteration of the improved node-based method, while the second half iteration is the improved node-based method taken in reverse order.

Second-order Node-based Method

- For each node *i,* the second order iterative method can be rewritten as

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

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

Second-order Node-based Method

Second-Order Node-based method

initialize $\mathbf{x_0} = \mathbf{x}^{(0)}$

initialize $\mathbf{x_1} = \mathbf{x}^{(1)}$

/*$x^{(1)}$ is the 1st iteration of improved node-based method */

new $\mathbf{x_2}$

set $|\epsilon| \ll 1$

if $max_i|x_1[i] - x_0[i]| < |\epsilon|$, **return**

/*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 $x_1[i]$

for node i=N^2 **down to 1**

backward improved node-based method $x_1[i]$

/*Solve $x^{(n+1)}$.*/

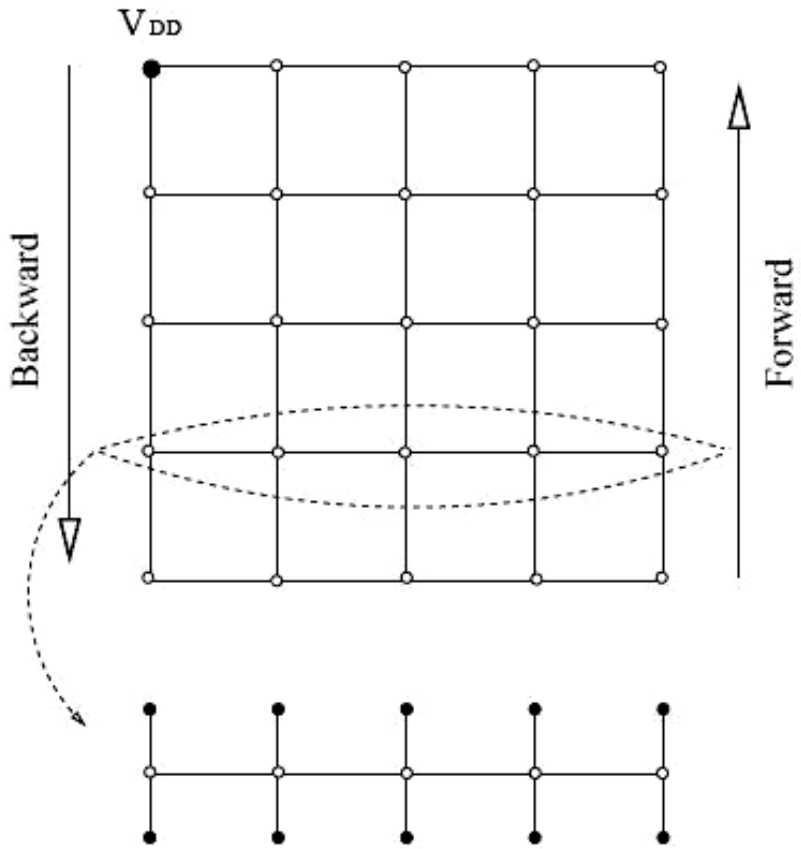
for node i=1 to N^2

$x_2[i] = x_2[i] + \beta x_1[i]$

$x_1[i] = x_2[i]$

Second-order Row-based Method

- To upgrade the first-order row-based method to a second-order method, we choose a first-order symmetric improved row-based method.



In each iteration, the first half is the same as the improved row-based method, while the second half iteration is the improved row-based method taken in reversed order.

Second-order Row-based Method

- For each row i, the second-order row-based method can be expressed as

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

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 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 $\mathbf{x}_i^{(n-1)}$.

Second-order Row-based Method

Second-Order Row-based method

initialize $\mathbf{x_0} = \mathbf{x}^{(0)}$

initialize $\mathbf{x_1}$ $x^{(1)}$

/*$x^{(1)}$ is the 1st iteration of row-based method.*/

new $\mathbf{x_2}$

set $|\epsilon| \ll 1$

if $max_i|x_1[i] - x_0[i]| < |\epsilon|$ **return**

/*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 $x_1[i]$

for row j=N **down to 1**

backward improved row-based method $x_1[i]$

/*Solve $x^{(n+1)}$.*/

for node i=1 to N^2

$x_2[i] = x_2[i] + \beta x_1[i]$

$x_1[i] = x_2[i]$;

Consistency of 2nd-order Method

Given a first-order iterative method

$$\mathbf{x}^{(n+1)} = G\mathbf{x}^{(n)} + \mathbf{k}_1$$

Consider the second-order method

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

Lemma 1: If the 1st-order iterative method converges to the exact solution, the 2nd-order method will converge to the same solution for any $\alpha \neq 0$ and $\beta \neq 0$.

Convergence of 1st-order Method

Lemma 2: An iterative method

$$\mathbf{x}^{(n+1)} = G\mathbf{x}^{(n)} + \mathbf{k}_1$$

converges if and only if *S(G)*<1, where *S(G)* is the spectral radius of matrix *G*.

- Spectral radius $S(G) = \max_{\lambda \in S_G} |\lambda|$ where S_G is the set of all eigenvalues of G.
- The smaller S(*G*) is, the faster the iterative method converges. Thus, one should minimize S(*G*) to maximize the rate of convergence

Convergence of 2nd-order Method

Consider a general form of 2nd-order method:

$$\mathbf{x}^{(n+1)} = G_1\mathbf{x}^{(n)} + G_2\mathbf{x}^{(n-1)} + k_2$$

Observe that

$$\begin{pmatrix} \mathbf{x}^{(n)} \\ \mathbf{x}^{(n+1)} \end{pmatrix} = \begin{pmatrix} 0 & I \\ G_2 & G_1 \end{pmatrix} \begin{pmatrix} \mathbf{x}^{(n-1)} \\ \mathbf{x}^{(n)} \end{pmatrix} + \begin{pmatrix} 0 \\ k_2 \end{pmatrix}$$

The iterative matrix of the 2nd-order method is

$$\hat{G} = \begin{pmatrix} 0 & I \\ G_2 & G_1 \end{pmatrix}$$

- A necessary and sufficient condition that the iterative method converges for all initial conditions is that $S(\hat{G}) < 1$.
- If we minimize $S(\hat{G})$, the rate of convergence is maximized.

Convergence of 2nd-order Method

Lemma 3: The eigenvalues λ of $\hat{G}$ are related to the eigenvalues μ of *G* as follows

$$\mu + \frac{1-\beta+\alpha}{\beta} = \frac{\lambda^2+\alpha}{\beta\lambda}$$

- Suppose the eigenvalues μ of G are real numbers that lie within the interval $[\mu_{\min}, \mu_{\max}]$, where $\mu_{\max} < 1$. We can compute α and β to minimize $S(\hat{G})$, which maximize the rate of convergence of the second order method.

Iterative Coefficients

Theorem 1: For any first-order iterative method

$$\mathbf{x}^{(n+1)} = G\mathbf{x}^{(n)} + \mathbf{k}_1$$

which has real eigenvalues, if we know the eigenvalue bounds $\mu_{\min}$ and $\mu_{\max}$ of its iterative matrix ***G***, the optimal coefficients in the second order method are

$$\alpha = |\lambda|^2$$

$$\beta = \frac{2(1+|\lambda|^2)}{2-(\mu_{\max}+\mu_{\min})}$$

where λ satisfies $2|\lambda| = \frac{(\mu_{\max}-\mu_{\min})}{2-(\mu_{\max}+\mu_{\min})}(1+|\lambda|^2)$

Why Symmetric Iterative Method?

- Problem is formulated as linear equation $\mathbf{Ax} = \mathbf{b}$, where **A** is a symmetric and positive definite matrix.
- Consider improved first-order node-based method:

  $$x_i^{(k+1)} = \omega \bar{x}_i^{(k+1)} + (1-\omega) x_i^{(k)}$$

  It converges with any initial solution when $0 < \omega < 2$.

  But the eigenvalues of its iterative matrix *G* are not all real if $\omega > 1$.
- In Theorem 1, we assume that the 1st-order method have real eigenvalues, to obtain the optimal α and β. So we cannot use the original improved node-based method directly as the 1st-order method, because the eigenvalues of its iterative matrix *G* are not all real .

Why Symmetric Iterative Method?

- Symmetric improved first-order node-based method converges with any initial condition when $0 < \omega < 2$. And the eigenvalues of its iterative matrix G are real and nonnegative for all real ω .
- So we use symmetric improved node-based method as the first-order iterative method, to obtain optimal iterative coefficients α and β by Theorem 1.

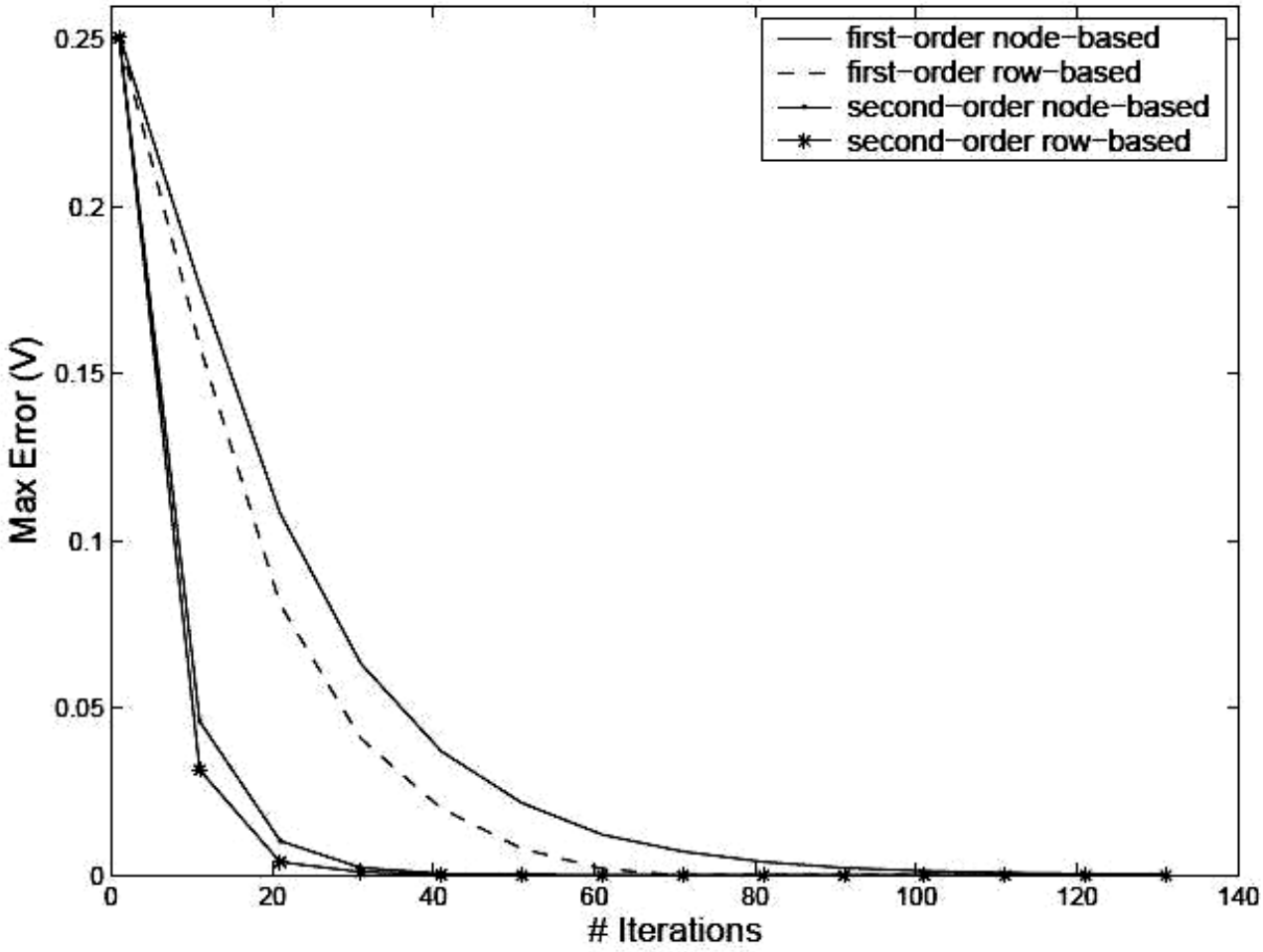
Experimental Results

- Circuit C1 with 251K nodes.

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

Machine: Linux PC with 2.8-GHz CPU and 4-GB RAM.

Experimental Results



Circuit 1 with 251K nodes.

The convergence of the first-order and second-order node-based and row-based methods.

Experimental Results

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

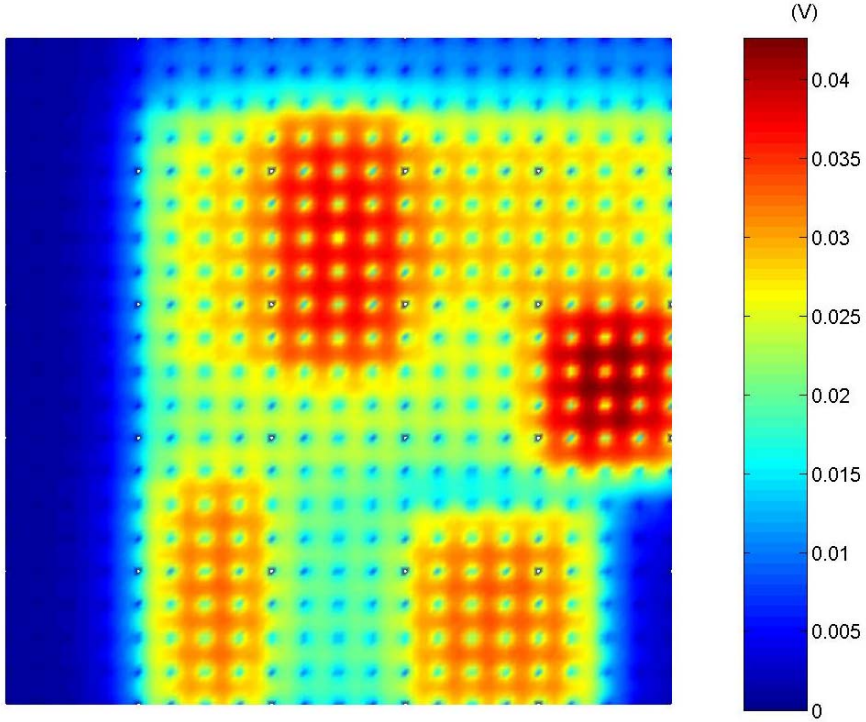
Experimental Results

- Runtime and error comparison

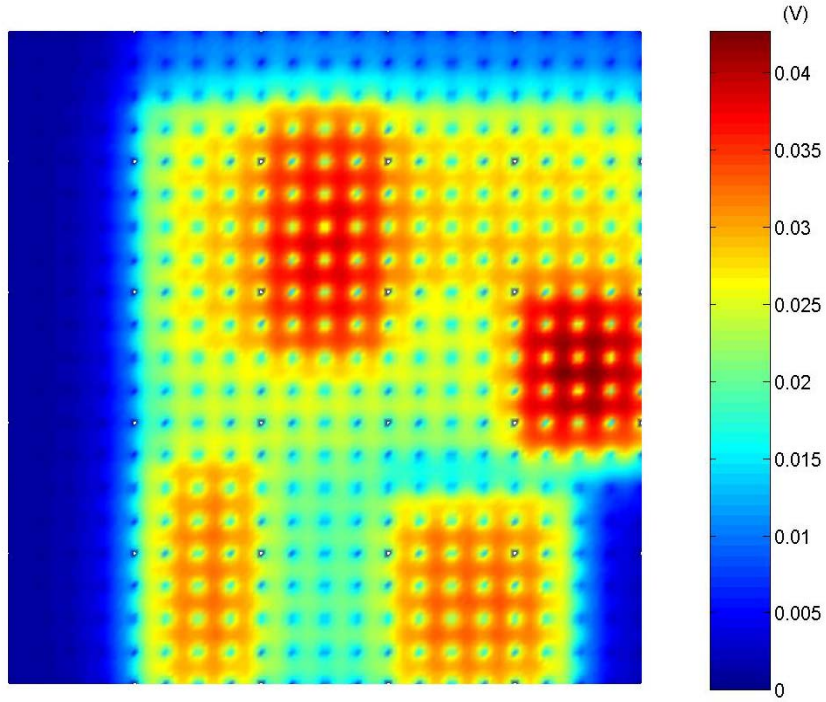
Circuits	#nodes	Random Walk		2nd-Order Row	
		MaxE (mV)	time (min:sec)	MaxE (mV)	time (min:sec)
C1	251K	5.4	4:55	1.8	0:00
C2	251K	6.0	11:04	2.0	0:01
C3	1M	7.5	37:40	1.9	0:13
C4	4M	6.3	152:22	2.0	0:25
C5	16M	4.9	1019:16	1.9	4:42
C6	25M	6.1	2943:56	2.0	9:55

Results (cont′d)

Circuit C5 with 16 million nodes by second-order row-based method



Runtime: 4 min 42 sec
Max error: 1.9mV
Average error: 0.07 mV



Runtime: 15 min 5 sec
no error

Conclusions

- Power grid simulation is a big challenge.
- Second-order Node-based and Row-based methods
  - No need to construct system matrix
  - Based on local grid structure to save memory
  - Significantly faster than first-order node-based and row-based methods
  - Choice of α and β to improve convergence speed
- Experimental results show the advantage in both accuracy and runtime
  - 25 million nodes, runtime is about 30 minutes without error, and 10 minutes with maximum error 2 mV.