A Fast Method for Steady-State Memristor Crossbar Array Circuit Simulation

Rui Xie[1], Mingyang Song[1], Junzhuo Zhou[1], Jie Mei[1] and Quan Chen*[1]

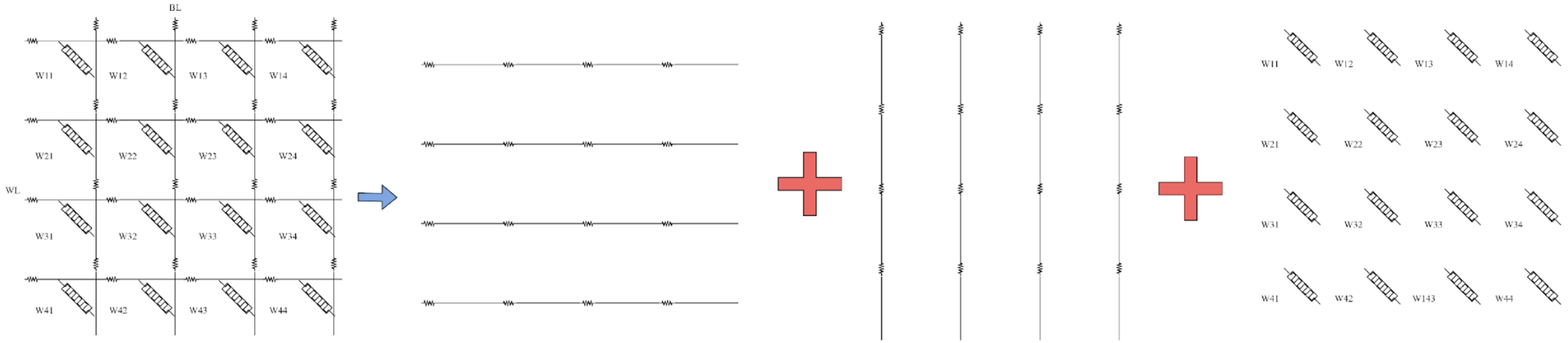
[1]School of Microelectronics, Southern University of Science and Technology

IEEE International Conference on Integrated Circuits, Technologies and Application, Nov. 2021
(Zhuhai, Guangdong Province, China)

Motivation

- Existing MCA by SPICE, use **Newton iteration**
- **Huge matrix size**, a 1024×1024 MCA leads to a matrix size >10^6
- Lack of **efficient preconditioners** for large scale MCA circuits

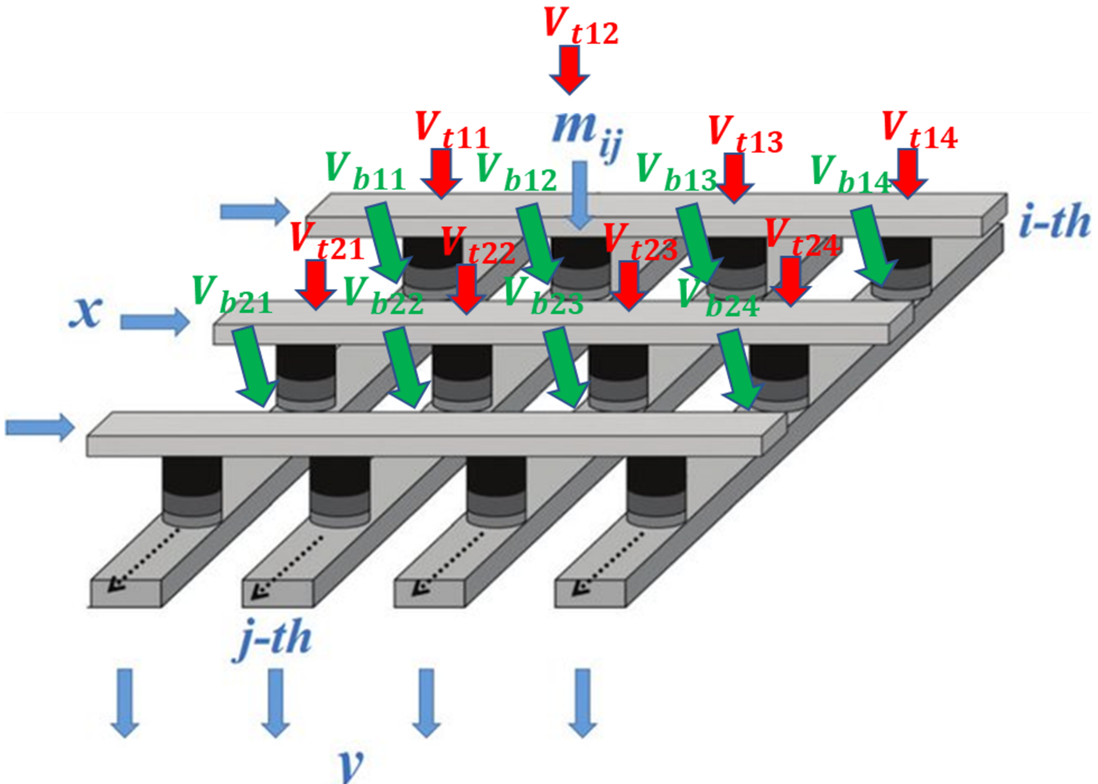
Division of Memristor Crossbar Array (MCA)



$MCA(\mathrm{n} \times n\ crossbar)$ $\quad V_t \quad V_b \quad Memristor$

Modified Nodal Analysis Equations

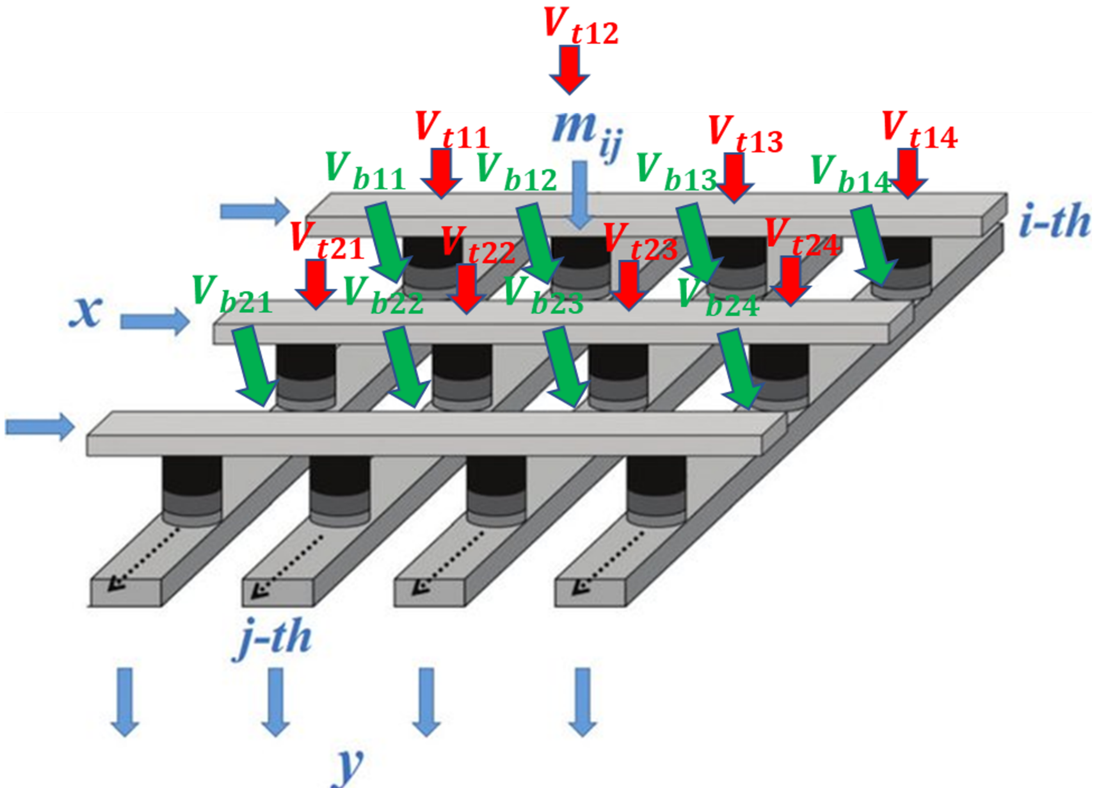
- $\begin{cases} G_t * V_t + I_t(V_t, V_b) - Y_t = 0 \\ G_b * V_b - I_b(V_t, V_b) - Y_b = 0 \end{cases}$
- $\begin{cases} G_t = I \otimes G \\ G_b = G \otimes I \end{cases}$
- I is identity matrix.



Huang, A., Zhang, X., Li, R., & Chi, Y. (2018). Memristor neural network design. *Memristor and Memristive Neural Networks*, 1-35.

Modified Nodal Analysis Equations

- $$G = \begin{bmatrix} 2g & -g & \cdots & 0 & \\ -g & 2g & & & \\ \vdots & & \ddots & \vdots & \\ & & & 2g & -g \\ 0 & & \cdots & -g & 2g \end{bmatrix}_{n\times n}$$



Huang, A., Zhang, X., Li, R., & Chi, Y. (2018). Memristor neural network design. *Memristor and Memristive Neural Networks*, 1-35.

Modified Nodal Analysis Equations—cont.

- $\begin{cases} G_t * V_t + I_t(V_t, V_b) - Y_t = 0 \\ G_b * V_b - I_b(V_t, V_b) - Y_b = 0 \end{cases}$
- $\begin{cases} G_t = I \otimes G \\ G_b = G \otimes I \end{cases}$
- I is identity matrix.

- $V_t = \begin{bmatrix} v_{t1,1} \\ v_{t1,2} \\ \vdots \\ v_{t1,n-1} \\ v_{t1,n} \\ v_{t2,1} \\ \vdots \\ v_{t2,n} \\ \vdots \\ v_{tn,n} \end{bmatrix}, \quad V_b = \begin{bmatrix} v_{b1,1} \\ v_{b1,2} \\ \vdots \\ v_{b1,n-1} \\ v_{b1,n} \\ v_{b2,1} \\ \vdots \\ v_{b2,n} \\ \vdots \\ v_{bn,n} \end{bmatrix}$

Newton Raphson Method

- $F = \begin{bmatrix} G_t & 0 \\ 0 & G_b \end{bmatrix} * \begin{bmatrix} V_t \\ V_b \end{bmatrix} + \begin{bmatrix} I_t \\ -I_b \end{bmatrix} - \begin{bmatrix} Y_t \\ Y_b \end{bmatrix}$

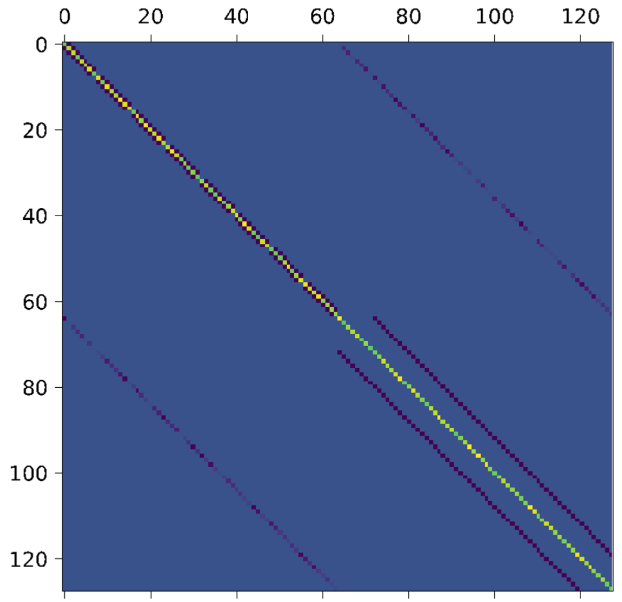
-
- $F' = \begin{bmatrix} G_t & 0 \\ 0 & G_b \end{bmatrix} + \begin{bmatrix} \frac{\partial I_t}{\partial V_t} & \frac{\partial I_t}{\partial V_b} \\ \frac{-\partial I_b}{\partial V_t} & \frac{-\partial I_b}{\partial V_b} \end{bmatrix} = J$

Newton Raphson Method

- $F = \begin{bmatrix} G_t & 0 \\ 0 & G_b \end{bmatrix} * \begin{bmatrix} V_t \\ V_b \end{bmatrix} + \begin{bmatrix} I_t \\ -I_b \end{bmatrix} - \begin{bmatrix} Y_t \\ Y_b \end{bmatrix}$
-
- $F' = \begin{bmatrix} G_t & 0 \\ 0 & G_b \end{bmatrix} + \begin{bmatrix} \frac{\partial I_t}{\partial V_t} & \frac{\partial I_t}{\partial V_b} \\ \frac{-\partial I_b}{\partial V_t} & \frac{-\partial I_b}{\partial V_b} \end{bmatrix} = J$

Newton Raphson Method

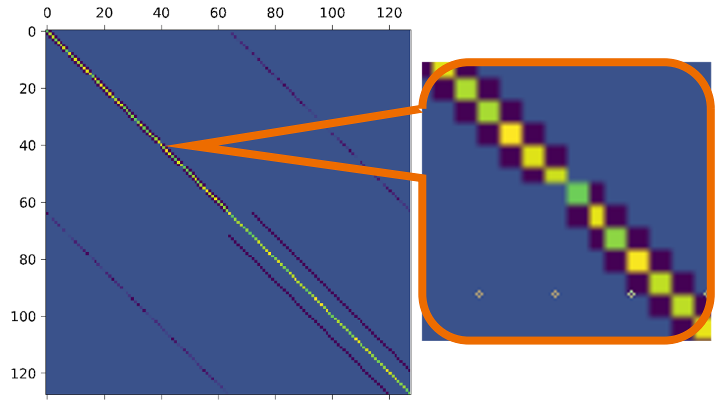
- $F = \begin{bmatrix} G_t & 0 \\ 0 & G_b \end{bmatrix} * \begin{bmatrix} V_t \\ V_b \end{bmatrix} + \begin{bmatrix} I_t \\ -I_b \end{bmatrix} - \begin{bmatrix} Y_t \\ Y_b \end{bmatrix}$
-
- $F' = \begin{bmatrix} G_t & 0 \\ 0 & G_b \end{bmatrix} + \begin{bmatrix} \frac{\partial I_t}{\partial V_t} & \frac{\partial I_t}{\partial V_b} \\ \frac{-\partial I_b}{\partial V_t} & \frac{-\partial I_b}{\partial V_b} \end{bmatrix} = J$



Schematic diagram of the shape of matrix J when $n = 8$. Note that dimension of the matrix here is $(2 * n^2) \times (2 * n^2)$

$G_t = I \otimes G$

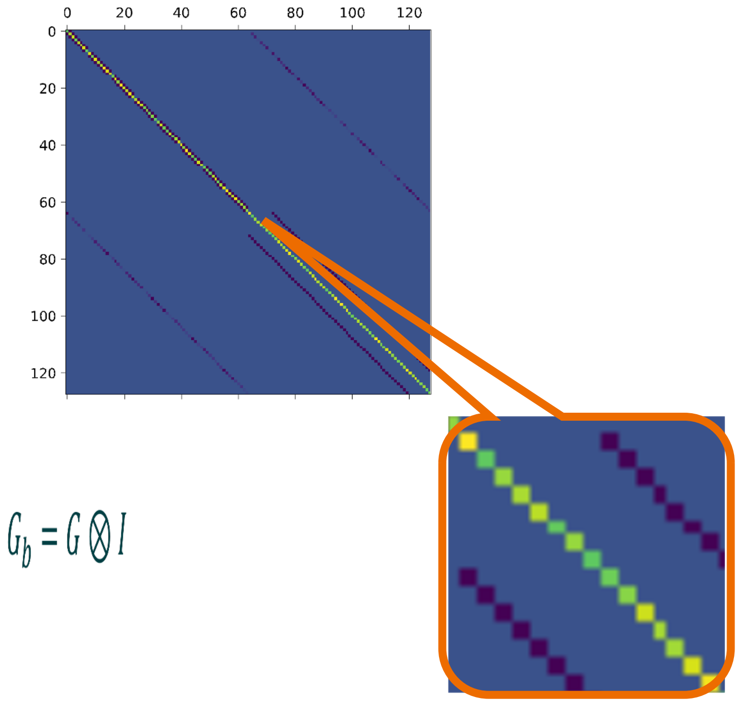
$$G_t = \begin{bmatrix} \overset{G}{\boxed{\begin{matrix} 2g & -g & \cdots & 0 & 0 \\ -g & 2g & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 2g & -g \\ 0 & 0 & \cdots & -g & 2g \end{matrix}}} & & \\ \begin{matrix} \vdots & \vdots & \vdots & \vdots & \vdots \end{matrix} & \begin{matrix} 2g & -g \\ -g & 2g \end{matrix} & \\ & & \ddots \end{bmatrix}_{n^2 \times n^2}$$



Schematic diagram of the shape of matrix J when $n = 8$. Note that dimension of the matrix here is $(2 * n^2) \times (2 * n^2)$

$G_b = G \otimes I$

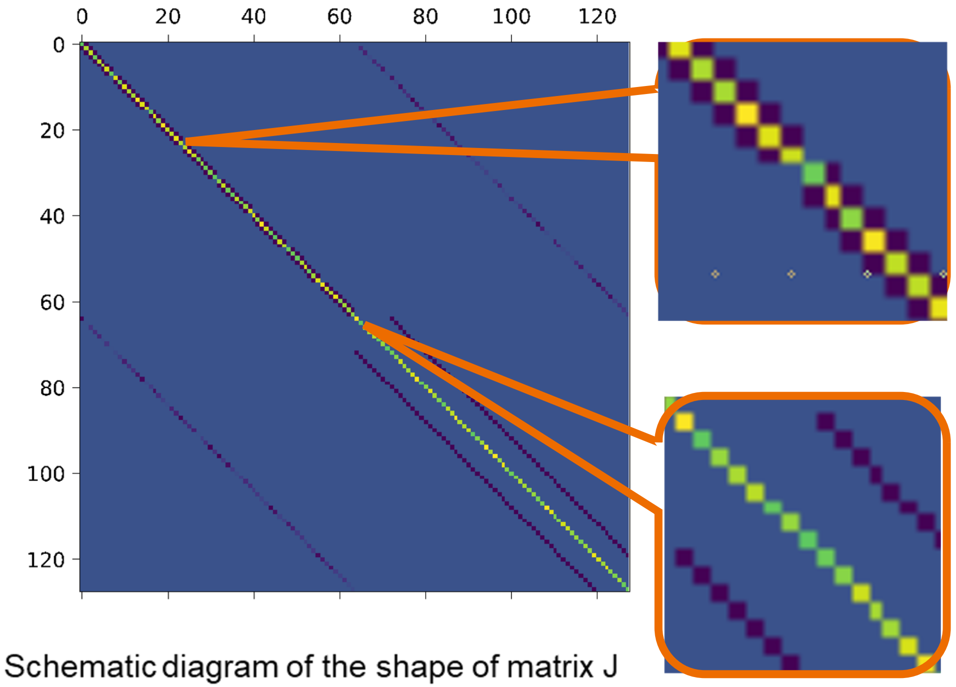
$$G_b = \begin{bmatrix} 2g & \cdots & -g & 0 & 0 & & & \\ \vdots & 2g & \cdots & -g & 0 & & & \\ -g & \vdots & \ddots & \cdots & \ddots & & & \\ 0 & -g & \cdots & 2g & \cdots & -g & & \\ 0 & 0 & -g & \cdots & 2g & & \ddots & \\ \vdots & \vdots & \vdots & \ddots & \vdots & 2g & & \\ & & & & & & 2g & \\ & & & & & & & \ddots \end{bmatrix}_{n^2\times n^2}$$



$G_b = G \otimes I$

Modified Nodal Analysis Equations—cont.

- The four blocks in the nonlinear Jacobian matrix are now all **diagonal**;
- Assuming **equal conductance** for all segments at the same layer[1].



Schematic diagram of the shape of matrix J when $n = 8$. Note that dimension of the matrix here is $(2 * n^2) \times (2 * n^2)$

[1]Liu, H., Wei, M., & Chen, Y. (2018). Optimization of non-linear conductance modulation based on metal oxide memristors. Nanotechnology Reviews, 7(5), 443-468.

Preconditioner Generate

- $J = \begin{bmatrix} G_t + \frac{\partial I_t}{\partial V_t} & \frac{\partial I_t}{\partial V_b} \\ \frac{-\partial I_b}{\partial V_t} & G_b + \frac{-\partial I_b}{\partial V_b} \end{bmatrix}$
-

- $P = \begin{bmatrix} G_t + a_1 I & -a_1 I \\ -a_2 I & G_b + a_2 I \end{bmatrix}$

- a_1 and a_2 are the **mean** of diagonal of $\frac{\partial I_t}{\partial V_b}$ and $\frac{-\partial I_b}{\partial V_b}$.
- Notice that $\frac{\partial I_t}{\partial V_t}$ and $\frac{\partial I_t}{\partial V_b}$ are opposite, as well as $\frac{-\partial I_b}{\partial V_t}$ and $\frac{-\partial I_b}{\partial V_b}$.

Preconditioner Generate

- $J = \begin{bmatrix} G_t + \frac{\partial I_t}{\partial V_t} & \frac{\partial I_t}{\partial V_b} \\ \frac{-\partial I_b}{\partial V_t} & G_b + \frac{-\partial I_b}{\partial V_b} \end{bmatrix}$
-
- $P = \begin{bmatrix} G_t + a_1 I & -a_1 I \\ -a_2 I & G_b + a_2 I \end{bmatrix}$

- a_1 and a_2 are the **mean** of diagonal of $\frac{\partial I_t}{\partial V_b}$ and $\frac{-\partial I_b}{\partial V_b}$.
- Notice that $\frac{\partial I_t}{\partial V_t}$ and $\frac{\partial I_t}{\partial V_b}$ are opposite, as well as $\frac{-\partial I_b}{\partial V_t}$ and $\frac{-\partial I_b}{\partial V_b}$.

Block Matrices Inversion

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} -M^{-1}DB^{-1} & M^{-1} \\ B^{-1} + B^{-1}AM^{-1}DB^{-1} & -B^{-1}AM^{-1} \end{bmatrix}$$

Preconditioner Generate

Dealing with $\boldsymbol{M}$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} -M^{-1}DB^{-1} & M^{-1} \\ B^{-1}+B^{-1}AM^{-1}DB^{-1} & -B^{-1}AM^{-1} \end{bmatrix}$$

Dealing with $\boldsymbol{M^{-1}}$

Preconditioner Generate

Dealing with M

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} -M^{-1}DB^{-1} & M^{-1} \\ B^{-1} + B^{-1}AM^{-1}DB^{-1} & -B^{-1}AM^{-1} \end{bmatrix}$$

- $M = (C - DB^{-1}A)$
  - $= (-a_2 I - (G_b + a_2 I)(-a_1 I)^{-1}(G_t + a_1 I))$
  - $= (-a_2 I + \frac{1}{a_1}(G \otimes I + a_2 I)(I \otimes G + a_1 I))$
  - $= (-a_2 I + \frac{1}{a_1}(G_2 \otimes I)(I \otimes G_1))$

Dealing with M^{-1}

Preconditioner Generate

Dealing with M

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} -M^{-1}DB^{-1} & M^{-1} \\ B^{-1} + B^{-1}AM^{-1}DB^{-1} & -B^{-1}AM^{-1} \end{bmatrix}$$

- $M = (C - DB^{-1}A)$
  - $= (-a_2 I - (G_b + a_2 I)(-a_1 I)^{-1}(G_t + a_1 I))$
  - $= (-a_2 I + \frac{1}{a_1}(G \otimes I + a_2 I)(I \otimes G + a_1 I))$
  - $= (-a_2 I + \frac{1}{a_1}(G_2 \otimes I)(I \otimes G_1))$

Dealing with M^{-1}

- Assumption: $20a_2 < \frac{1}{a_1}(2g +$
- $a_2)(2g + a_1)$
- $M^{-1} \cong \widehat{M}^{-1} = \widehat{G_2}^{-1} \otimes G_1{}^{-1}$

Rescale

$$\begin{bmatrix} \frac{1}{\hat{g}_{1,1}^{2}} * G_1^{-1} & \cdots & \frac{1}{g_{1,n}^{2}} * G_1^{-1} \\ \vdots & \ddots & \vdots \\ \frac{1}{\hat{g}_{n,1}^{2}} * G_1^{-1} & \cdots & \frac{1}{g_{n,n}^{2}} * G_1^{-1} \end{bmatrix} \begin{bmatrix} v_{1,1} \\ v_{1,2} \\ \vdots \\ v_{1,n-1} \\ v_{1,n} \\ v_{2,1} \\ \vdots \\ v_{2,n} \\ \vdots \\ v_{n,n} \end{bmatrix}$$

Rescale

$$\begin{bmatrix} \frac{1}{\hat{g}_{1,1}^2} * G_1^{-1} & \cdots & \frac{1}{g_{1,n}^2} * G_1^{-1} \\ \vdots & \ddots & \vdots \\ \frac{1}{\hat{g}_{n,1}^2} * G_1^{-1} & \cdots & \frac{1}{g_{n,n}^2} * G_1^{-1} \end{bmatrix} \boxed{\begin{bmatrix} v_{1,1} \\ v_{1,2} \\ \vdots \\ v_{1,n-1} \\ v_{1,n} \\ v_{2,1} \\ \vdots \\ v_{2,n} \\ \vdots \\ v_{n,n} \end{bmatrix}}$$

Rescale

$$\begin{bmatrix} \frac{1}{\hat{g}_{1,1}^{2}} * G_1^{-1} & \cdots & \frac{1}{g_{1,n}^{2}} * G_1^{-1} \\ \vdots & \ddots & \vdots \\ \frac{1}{\hat{g}_{n,1}^{2}} * G_1^{-1} & \cdots & \frac{1}{g_{n,n}^{2}} * G_1^{-1} \end{bmatrix} \boxed{\begin{bmatrix} v_{1,1} \\ v_{1,2} \\ \vdots \\ v_{1,n-1} \\ v_{1,n} \\ v_{2,1} \\ \vdots \\ v_{2,n} \\ \vdots \\ v_{n,n} \end{bmatrix}} \Rightarrow \hat{V} = \begin{bmatrix} v_{1,1} & \cdots & v_{1,n} \\ \vdots & \ddots & \vdots \\ v_{n,n} & \cdots & v_{n,n} \end{bmatrix}_{n*n} = \begin{bmatrix} V_1 & \cdots & V_n \end{bmatrix}$$

Rescale

$$
\begin{bmatrix}
\frac{1}{\hat{g}_{1,1}^{2}} * G_1^{-1} & \cdots & \frac{1}{g_{1,n}^{2}} * G_1^{-1} \\
\vdots & \ddots & \vdots \\
\frac{1}{\hat{g}_{n,1}^{2}} * G_1^{-1} & \cdots & \frac{1}{g_{n,n}^{2}} * G_1^{-1}
\end{bmatrix}
\boxed{\begin{bmatrix}
v_{1,1} \\ v_{1,2} \\ \vdots \\ v_{1,n-1} \\ v_{1,n} \\ v_{2,1} \\ \vdots \\ v_{2,n} \\ \vdots \\ v_{n,n}
\end{bmatrix}}
\Rightarrow
\hat{V} = \boxed{\begin{bmatrix}
v_{1,1} & \cdots & v_{1,n} \\
\vdots & \ddots & \vdots \\
v_{n,n} & \cdots & v_{n,n}
\end{bmatrix}_{n*n}}
= \begin{bmatrix} V_1 & \cdots & V_n \end{bmatrix}
$$

Rescale

$$
\begin{bmatrix}
\frac{1}{\hat{g}_{1,1}^{2}} * G_1^{-1} & \cdots & \frac{1}{g_{1,n}^{2}} * G_1^{-1} \\
\vdots & \ddots & \vdots \\
\frac{1}{\hat{g}_{n,1}^{2}} * G_1^{-1} & \cdots & \frac{1}{g_{n,n}^{2}} * G_1^{-1}
\end{bmatrix}
\begin{bmatrix}
v_{1,1} \\ v_{1,2} \\ \vdots \\ v_{1,n-1} \\ v_{1,n} \\ v_{2,1} \\ \vdots \\ v_{2,n} \\ \vdots \\ v_{n,n}
\end{bmatrix}
$$

$$
\hat{V} = \begin{bmatrix}
v_{1,1} & \cdots & v_{1,n} \\
\vdots & \ddots & \vdots \\
v_{n,n} & \cdots & v_{n,n}
\end{bmatrix}_{n*n} = \begin{bmatrix} V_1 & \cdots & V_n \end{bmatrix}
$$

$$
\frac{1}{\hat{g}_{j,1}^{2}} * G_1^{-1} V_1 + \frac{1}{g_{j,2}^{2}} * G_1^{-1} V_2 + \ldots + \frac{1}{g_{j,n}^{2}} * G_1^{-1} V_n = \begin{bmatrix} G_1^{-1} V_1 & \cdots & G_1^{-1} V_n \end{bmatrix}
\begin{bmatrix}
\frac{1}{\hat{g}_{j,1}^{2}} \\ \vdots \\ \frac{1}{\hat{g}_{j,n}^{2}}
\end{bmatrix}
$$

Rescale

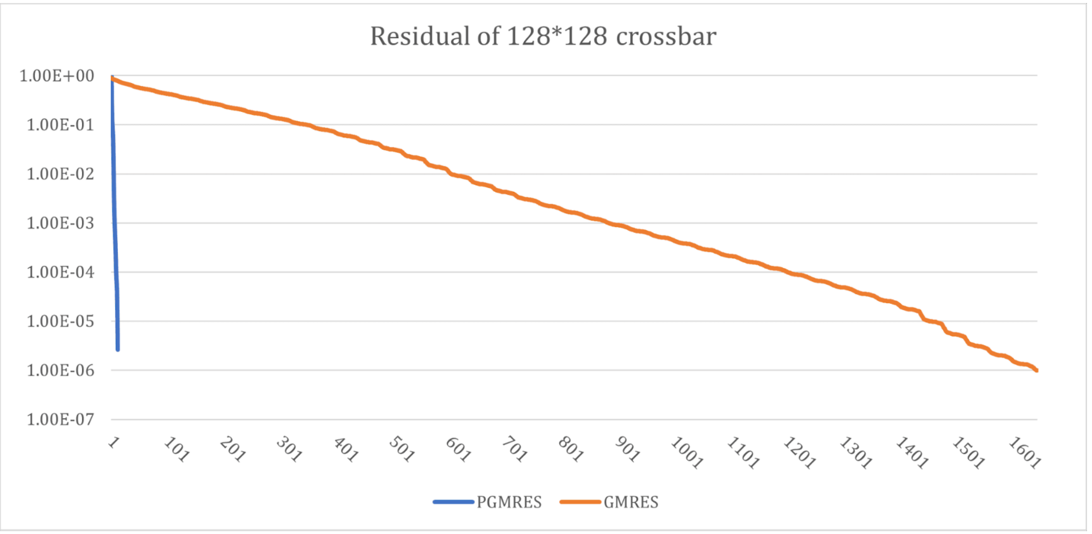
$$\begin{bmatrix} \frac{1}{\hat{g}_{1,1}^{2}} * G_1^{-1} & \cdots & \frac{1}{g_{1,n}^{2}} * G_1^{-1} \\ \vdots & \ddots & \vdots \\ \frac{1}{\hat{g}_{n,1}^{2}} * G_1^{-1} & \cdots & \frac{1}{g_{n,n}^{2}} * G_1^{-1} \end{bmatrix} \boxed{\begin{bmatrix} v_{1,1} \\ v_{1,2} \\ \vdots \\ v_{1,n-1} \\ v_{1,n} \\ v_{2,1} \\ \vdots \\ v_{2,n} \\ \vdots \\ v_{n,n} \end{bmatrix}}$$

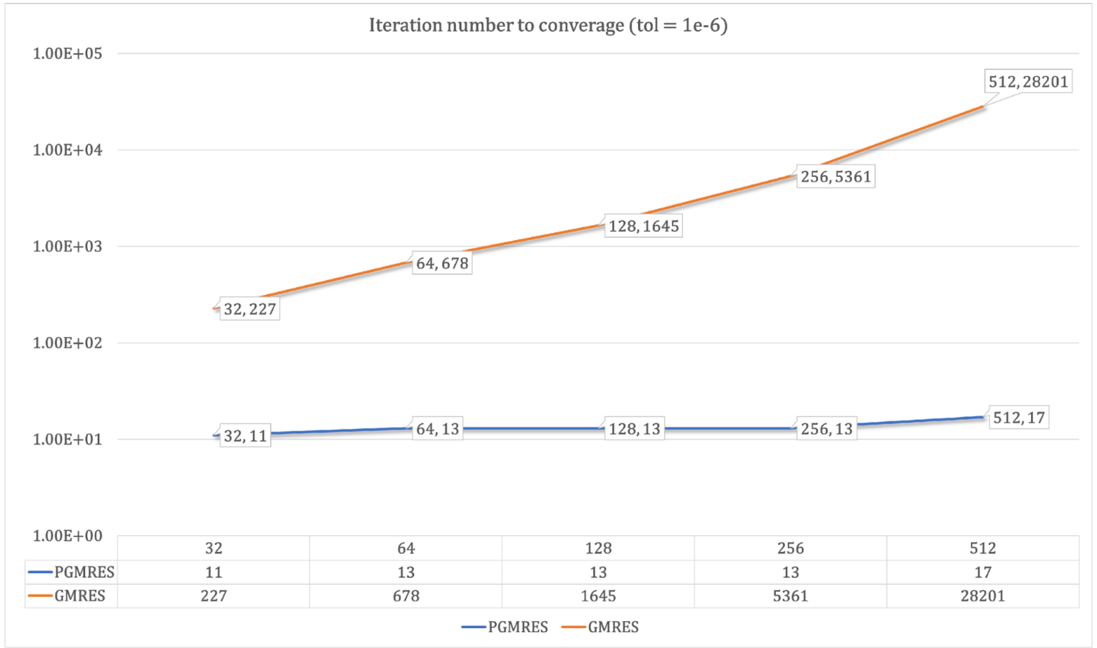
$$\Rightarrow \hat{V} = \boxed{\begin{bmatrix} v_{1,1} & \cdots & v_{1,n} \\ \vdots & \ddots & \vdots \\ v_{n,n} & \cdots & v_{n,n} \end{bmatrix}_{n*n}} = \begin{bmatrix} V_1 & \cdots & V_n \end{bmatrix}$$

$$\frac{1}{\hat{g}_{j,1}^{2}} * G_1^{-1} V_1 + \frac{1}{g_{j,2}^{2}} * G_1^{-1} V_2 + \ldots + \frac{1}{g_{j,n}^{2}} * G_1^{-1} V_n = \begin{bmatrix} G_1^{-1} V_1 & \cdots & G_1^{-1} V_n \end{bmatrix} \begin{bmatrix} \frac{1}{\hat{g}_{j,1}^{2}} \\ \vdots \\ \frac{1}{\hat{g}_{j,n}^{2}} \end{bmatrix}$$

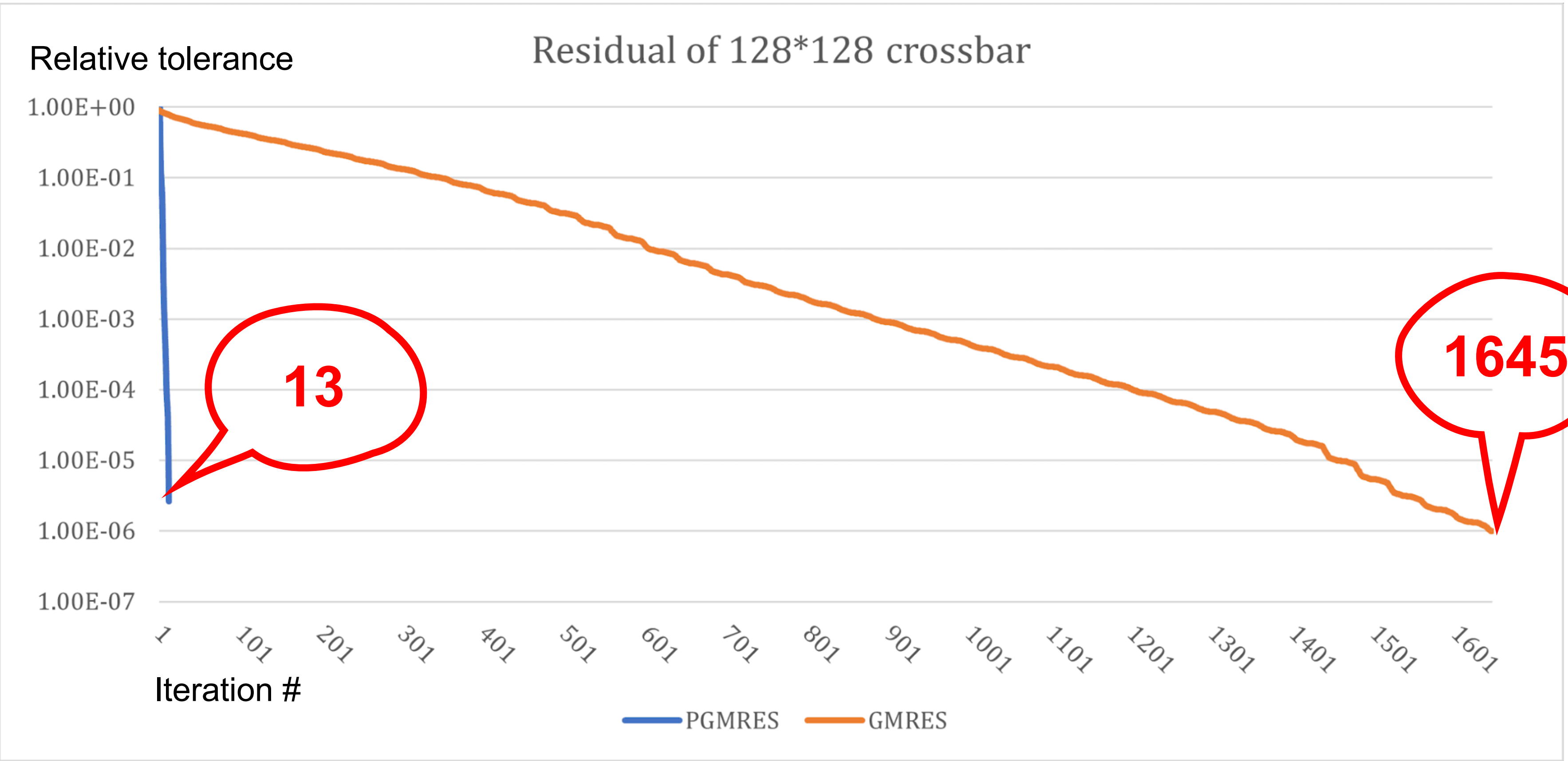
$$\left(G_2^{-1} \otimes G_1^{-1} \right) v = G_1^{-1} \hat{V} G_2^{-1}$$

Numerical Results

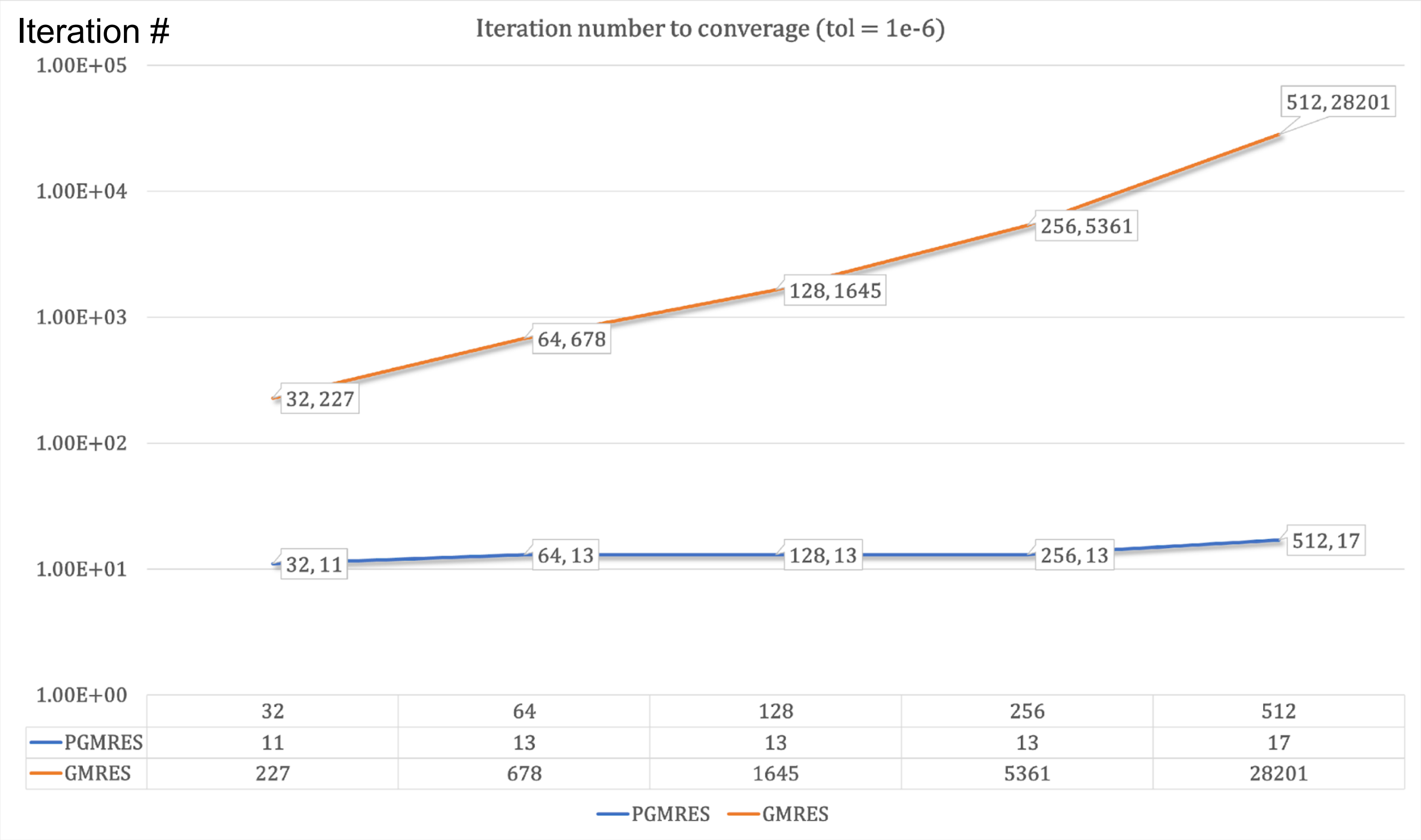




Numerical Results



Numerical Results



	32	64	128	256	512
PGMRES	11	13	13	13	17
GMRES	227	678	1645	5361	28201

Numerical Results

	Before Preconditioned		Jacobi Preconditioner		ILU Preconditioner		Our Preconditioner	
Crossbar Dimension (n)	Steps	CPU time consumption/s	Steps	CPU time consumption/s	Steps	CPU time consumption/s	Steps	CPU time consumption/s
16*16	67	0.01596	60	0.00897	3	0.00598	11	0.00299
32*32	227	0.05785	203	0.03092	15	0.01396	11	0.02194
64*64	678	0.30377	551	0.14319	111	0.14561	13	0.09275
128*128	1645	1.97858	1610	0.95511	308	1.24064	13	0.57907
256*256	7526	28.51413	5361	23.41768	589	11.05555	13	4.77912
512*512	22801	331.31852	19273	324.20583	3120	309.91122	17	225.12859

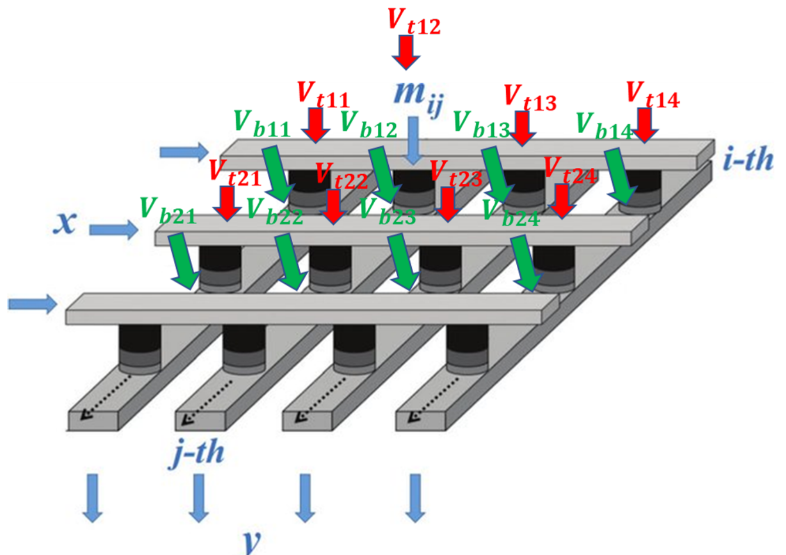
Conclusion--Modified Nodal Analysis Equations

- $\begin{cases} G_t * V_t + I_t(V_t, V_b) - Y_t = 0 \\ G_b * V_b - I_b(V_t, V_b) - Y_b = 0 \end{cases}$
- $\begin{cases} G_t = I \otimes G \\ G_b = G \otimes I \end{cases}$

$$J\left(\vec{V}_n\right) * \left(\vec{V}_{n+1} - \vec{V}_n\right) = \left(-F\left(\vec{V}_n\right)\right)$$

$$J = F' = \begin{bmatrix} G_t & 0 \\ 0 & G_b \end{bmatrix} + \begin{bmatrix} \frac{\partial I_t}{\partial V_t} & \frac{\partial I_t}{\partial V_b} \\ \frac{-\partial I_b}{\partial V_t} & \frac{-\partial I_b}{\partial V_b} \end{bmatrix}$$

$$\boxed{P = \begin{bmatrix} G_t + a_1 I & -a_1 I \\ -a_2 I & G_b + a_2 I \end{bmatrix}}$$



Huang, A., Zhang, X., Li, R., & Chi, Y. (2018). Memristor neural network design. *Memristor and Memristive Neural Networks*, 1-35.

a_1 and a_2 are the **mean** of diagonal of $\frac{\partial I_t}{\partial V_b}$ and $\frac{-\partial I_b}{\partial V_b}$.

Notice that $\frac{\partial I_t}{\partial V_t}$ and $\frac{\partial I_t}{\partial V_b}$ are opposite, as well as $\frac{-\partial I_b}{\partial V_t}$ and $\frac{-\partial I_b}{\partial V_b}$

Conclusion--Block Matrices Inversion

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} -M^{-1}DB^{-1} & M^{-1} \\ B^{-1} + B^{-1}AM^{-1}DB^{-1} & -B^{-1}AM^{-1} \end{bmatrix}$$

$$M = (C - DB^{-1}A)$$

$$M = -a_2 I + \frac{1}{a_1}(G_2 \otimes G_1)$$

omitting a_2

$$\hat{M} = \frac{1}{a_1}(G_2 \otimes G_1)$$

$$M^{-1} \cong \hat{M}^{-1} = G_2^{-1} \otimes G_1^{-1}$$

Rearranging v

$$\left(G_2^{-1} \otimes G_1^{-1}\right) v = G_1^{-1}\hat{V}G_2^{-1}$$

Reference

[1] Chua, L. (1971). Memristor-the missing circuit element. IEEE Transactions on circuit theory, 18(5), 507-519.

[2] Zhang, F., & Hu, M. (2020, November). CCCS: customized spice-level crossbar-array circuit simulator for in-memory computing. In Proceedings of the 39th International Conference on Computer-Aided Design (pp. 1-8). [2] A. N. Expert, A Book He Wrote, His Publisher, 1989.

[3] Truong, S. N., Van Pham, K., Yang, W., & Min, K. S. (2016, October). Memristor circuits and systems for future computing and bio-inspired information processing. In 2016 IEEE Biomedical Circuits and Systems Conference (BioCAS) (pp. 456-459). IEEE.

[4] Ferronato, M.. Preconditioning for sparse linear systems at the dawn of the 21st century: History, current developments, and future perspectives. International Scholarly Research Notices, 2012.

[5] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino and S. Rogers, "A Memristor Device Model," in IEEE Electron Device Letters, vol. 32, no. 10, pp. 1436-1438, Oct. 2011,

Thanks for listening!