A Parallel Floating Random Walk Solver for Reproducible and Reliable Capacitance Extraction

Jiechen Huang*, Shuailong Liu†, and Wenjian Yu*
*Dept. Computer Science & Tech., BNRist, Tsinghua University, Beijing, China. †Exceeda Inc., Beijing, China.

***Abstract*—The floating random walk (FRW) method is a popular and promising tool for capacitance extraction, but its stochastic nature leads to critical limitations in reproducibility and physics-related reliability. In this work, we present FRW-RR, a parallel FRW solver with enhancements for Reproducible and Reliable capacitance extraction. First, we propose a novel parallel FRW scheme that ensures reproducible results, regardless of the degree of parallelism (DOP) or machine used. We further optimize its parallel efficiency and enhance the numerical stability. Then, to guarantee the physical properties of capacitances and reliability for downstream tasks, we propose a regularization technique based on constrained multi-parameter estimation to postprocess FRW's results. Experiments on actual IC structures demonstrate that, FRW-RR ensures DOP-independent reproducibility (with at least 12 decimal significant digits) and physics-related reliability with negligible overhead. It has remarkable advantages over existing FRW solvers, including the one in [1].**

***Index Terms*—Capacitance extraction, floating random walk method, parallel computing, reproducibility, physical properties.**

I. Introduction

As the process technology evolves, rule-based or pattern-matching-based (2.5-D) methods no longer suffice for accurate parasitic capacitance extraction in the design and verification of integrated circuits (ICs). In contrast, 3-D electric field solvers are increasingly preferred and expected to handle larger and more complex structures [2]. To meet this growing demand, leveraging parallel computing in field solvers has become a common strategy to mitigate time-to-market pressure in IC design. Hence, parallel-scalable and reliable field solvers for capacitance extraction are now more crucial than ever.

Among field-solver methods, the *floating random walk* (FRW) method offers better scalability to larger structures than classical finite difference method (FDM), finite element method (FEM) and boundary element method (BEM) [2], [3]. The FRW method is built upon stochastic process and Monte Carlo (MC) integration to produce unbiased capacitance estimates [4]. It avoids the memory-intensive solution of large linear equation systems, which is problematic for classical methods that rely on geometry discretization. Besides, the FRW method is embarrassingly parallelizable due to the independence of MC samples. Parallel implementations of FRW-based capacitance extraction have been extensively explored on shared-memory and distributed platforms [1], [5]–[7], and with accelerators like GPUs [8], [9] and FPGAs [10]. However, *numerical reproducibility*, i.e., the ability to ensure identical results for the same input, is a practical issue for parallel FRW solvers. It reflects a basic challenge in scientific computing, especially in parallel random simulations [11]. Reproducible capacitance extraction is essential for the stability of entire toolchain and is also driven by actual user demand during testing and debugging. An efficient and reproducible parallel FRW scheme was proposed in [1]. However, it lacks flexibility in that it can only reproduce results when the degree of parallelism (DOP), i.e., the number of threads/processes, is fixed.

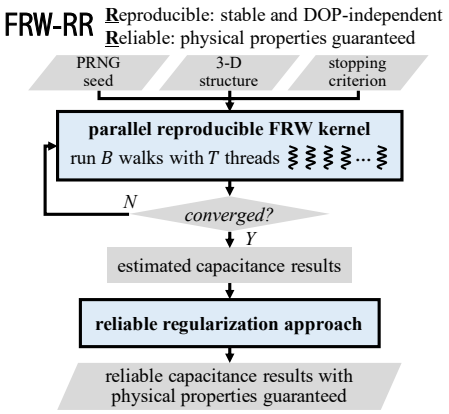

Fig. 1. The overall flow of our parallel FRW solver for reproducible and reliable capacitance extraction.

Another direction of research on the FRW method focuses on improving its accuracy for extracting actual IC structures. In [12]–[14], specific techniques were proposed to enable the FRW method to efficiently and accurately handle complicated dielectrics in advanced process technologies. Reliability of results is also an aspect of accuracy. The FRW method is reliable in terms of unbiasedness, i.e., the random error converges as the number of walks increases [15]. However, its inherent randomness harms another *physics-related reliability* in ensuring that capacitance results adhere to physical laws. According to electrostatics, capacitances in a multi-conductor system form a matrix with special properties, e.g., symmetry and negativity of off-diagonal elements [16]. These properties are *a priori* knowledge for improving result accuracy. They are also required by post-extraction circuit simulation [17] and the macromodel-aware capacitance extraction method [18]–[20]. However, no literature addresses how to guarantee this physics-related reliability for the FRW-produced capacitance results.

It should be pointed out that, the weakness of FRW method on the aforementioned reproducibility and physics-related reliability has been seen as its major disadvantage compared to the classical methods for capacitance extraction [21]. To overcome this, we aim to develop a parallel FRW solver with improved

This work is supported by Beijing Natural Science Foundation (Z220003) and Beijing Science and Technology Plan (Z221100007722025).

978-3-9826741-0-0/DATE25/© 2025 EDAA

reproducibility and guaranteed physics-related reliability. Our contributions are as follows.

- A parallel scheme for FRW-based capacitance extraction is developed, where DOP-independent reproducibility is achieved via fine-grained random number reseeding and is further enhanced with numerically stable summation algorithm. A special random number generator [22] and load balancing skills are also employed to preserve the high parallel efficiency of the FRW solver.
- A regularization approach based on constrained multi-parameter estimation is proposed to guarantee the physics-related reliability of FRW-based extraction. Its computational cost is minimized by converting the solution into a special least squares problem.
- With the above techniques, we present FRW-RR, an improved parallel FRW solver for reproducible and reliable capacitance extraction. Its workflow is shown in Fig. 1. Experiments on actual IC structures show that, FRW-RR offers better reproducibility than the baseline from [1] for both fixed and varied-DOP scenarios, without sacrificing efficiency. Besides, the regularization approach guarantees the physics-related reliability and reduces capacitance error by 21% on average, with negligible overhead.

II. Background

A. Desired Properties of Capacitance Extraction Result

Consider a system of conductors $\{i|1 \le i \le N\}$. The electric charges $\mathbf{Q}=(Q_i)_{N\times 1}$ and potentials $\mathbf{V}=(V_i)_{N\times 1}$ are linearly related to each other via the *capacitance matrix* $\mathbf{C}=(C_{ij})_{N\times N}$:

$$\mathbf{Q} = \mathbf{C}\mathbf{V}. \tag{1}$$

This form of definition is known as the *coefficients of induction* or Maxwell capacitance matrix [23]. In practical and bounded-domain problems, $\mathbf{C}$ has three physics-related properties [16]:

Property 1 (sign property): $C_{ii} \ge 0$, $C_{ij} \le 0$, $\forall (i,j), i \ne j$.

Property 2 (symmetry): $C_{ij} = C_{ji}$, $\forall (i,j)$.

Property 3 (zero row-sum): $\sum_{j=1}^{N} C_{ij} = 0$, $\forall i$.

Many applications must use capacitance matrices with the above properties. For example, in circuit simulation, they are prerequisites of techniques like pGRASS-Solver [17]. In hierarchical capacitance extraction [18]–[20], *macromodels* are built from solved capacitance matrices of local layouts, and are valid only when those matrices satisfy the three properties [20].

Capacitance matrices computed via numerical methods can hardly satisfy these properties, a problem largely overlooked in the literature. Yang *et al.* [20] identified this a "reliability" issue and fixed it to some extent for FDM solvers by reformulating the equations. We also adopt the term "reliability" to mean the alignment with physical properties.

B. The FRW Algorithm for Capacitance Extraction

Field solvers extract capacitance typically by solving Dirichlet boundary value problems of electrostatic field, i.e., assigning $\mathbf{V}$ and finding $\mathbf{Q}$. The basic idea of FRW is to evaluate the charge on a master conductor i using Gauss's law

$$Q_i = -\oint_{G_i} \varepsilon(\mathbf{r}) \nabla \phi(\mathbf{r}) \cdot \mathbf{n}(\mathbf{r}) \mathrm{d}s, \tag{2}$$

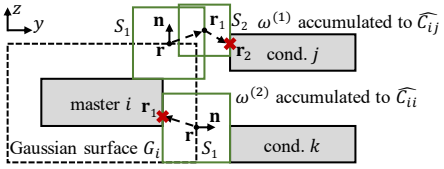

Fig. 2. A cross-section view of two walks in FRW-based capacitance extraction.

where G_i is the Gaussian surface enclosing i, while $\varepsilon(\mathbf{r})$, $\phi(\mathbf{r})$ and $\mathbf{n}(\mathbf{r})$ denote the dielectric permittivity, electric potential and unit normal at point $\mathbf{r}$. Next, the unknown $\phi(\mathbf{r})$ is estimated with Markov chain Monte Carlo integration, where the transition probability $P(\mathbf{r}_1|\mathbf{r})$ satisfies

$$\phi(\mathbf{r}) = \oint_{S_1} P(\mathbf{r}_1|\mathbf{r}) \phi(\mathbf{r}_1) \mathrm{d}s. \tag{3}$$

S_1 is the surface of *transition domain*, typically a cube centered at $\mathbf{r}$. Transitions continue until hitting an *absorbing boundary*, i.e., conductor surface with given ϕ. With (2) and (3), the Markov chain can be formulated, yielding a random variable $X_i = \omega \phi_T$. Here ω is a weight value depending on the first two states, and ϕ_T is the potential at the destination. It is proven

$$\mathbb{E}(X_i) = Q_i. \tag{4}$$

Refer to [15] for a detailed construction of the FRW procedure.

As ϕ_T is always one of the conductor potentials, X_i clearly depends on the excitation $\mathbf{V}$. Let $X_{ij} = (X_i|\mathbf{V}=\mathbf{e}_j)$, where $\mathbf{e}_j$ is the one-hot vector with 1 on the j-th position. We have

$$\mathbb{E}(X_{ij}) = \mathbb{E}(X_i|\mathbf{V}=\mathbf{e}_j) = (Q_i|\mathbf{V}=\mathbf{e}_j) = C_{ij}. \tag{5}$$

A walk from G_i yields N samples $(x_{i1}, \ldots, x_{iN})$, corresponding to $(X_{i1}, \ldots, X_{iN})$. If the walk ends on conductor k, only x_{ik} is non-zero, so only one actual accumulation is needed. After M walks, we output unbiased estimates $(\widehat{C}_{i1}, \ldots, \widehat{C}_{iN})$ where $\widehat{C}_{ij} = \frac{\sum_{m=1}^{M} x_{ij}^{(m)}}{M}$ with an asymptotic error $\sqrt{\frac{\mathrm{Var}(X_{ij})}{M}}$. They constitute the i-th row of capacitance matrix. Fig. 2 depicts two examples of random walk paths.

Existing FRW solvers lack reproducibility and reliability in practical use. In common parallel schemes, workloads are partitioned by allocating accuracy criteria [1], [6], [7]. The idea is that if T threads respectively reach a statistical error of $\epsilon\sqrt{T}$,

Algorithm 1: The parallel FRW algorithm reproducible with a fixed degree of parallelism (DOP), from [1].

Input: Structure with N conductors, master conductor i, error tolerance ϵ, T threads, seeds $s_1 \ldots s_T$.
Output: A row in capacitance matrix $(\widehat{C}_{i1}, \ldots, \widehat{C}_{iN})$.

```
Construct Gaussian surface enclosing i;
Init T threads with private PRNGs and accumulators;
Seed the T PRNGs respectively with s_1 ... s_T;
ε' ← ε√T;                          // allocate stopping criterion;
In parallel run T threads
    repeat
        (ω, conductor k) ← a random walk;
        accumulate(ω, conductor k);
    until estimated error < ε';
Merge T accumulators to compute (Ĉ_i1, ..., Ĉ_iN);
```

the merged result's error will be ϵ. In [1], this scheme was made reproducible by fixing the pseudo-random number generator (PRNG) seeds in each thread (see Alg. 1). Note that the random number streams are deterministic within each thread. However, if T changes, the merged result will change unpredictably as each thread converges to a new error criterion. Moreover, this reproducibility is fragile due to the random scheduling of parallel processors (see Sec. III-A). As for reliability, since capacitance estimates are computed independently, they hardly satisfy the physics-related properties. Although recent studies improved the accuracy of FRW method [12]–[14], none addressed this critical issue of physics-related reliability.

III. A Parallel FRW Algorithm with DOP-Independent Reproducibility

A. Problem Analysis and Definition

We first analyze numerical reproducibility more deeply and define the metric for reproducibility. Pursuing reproducibility in parallel computing involves three levels. First, properly seeding PRNGs can fix the random streams [1]. Beyond this, unordered scheduling and non-associativity of floating-point arithmetic must be considered. At the second level, using numerically stable algorithms can mitigate cancellation and round-off error propagation [24], [25]. Finally, full bitwise reproducibility can be realized via dedicated hardware or algorithms [11]. We will focus on the first two levels: controlling random streams and enhancing numerical stability. We do not ensure bitwise exactness, because 1) it is not worth the overhead and 2) limited precision is sufficient for capacitance [1]. Compared to the approach that works for fixed-DOP scenarios [1], our approach aims at higher flexibility and numerical precision.

To assess the reproducibility, we consider the number of matched digits between two results of the same input. In practice, we execute P extractions/runs for the same structure and parameters, possibly with **different degrees of parallelism (DOP) or machines**. To compare the i-th and j-th runs, we define reproducibility index (RI) as d_{ij}, which means every capacitance matches in at least d_{ij} decimal significant digits. For bitwise identical results, d_{ij} should be ∞ but is set to 17[1] for convenience. A total of $P(P-1)/2$ pairwise comparisons can be made. Then, we have the minimum and average reproducibility indices (RIs) for this experiment:

$$\mathrm{RI}_{\min} = \min\{d_{ij}, \forall i \neq j\},\ \mathrm{RI}_{\mathrm{avg}} = \frac{\sum_{i=1}^{P}\sum_{j=i+1}^{P} d_{ij}}{P(P-1)/2}. \quad (6)$$

B. Achieving DOP-Independent Reproducibility

Our basic idea of achieving DOP-independent reproducibility is illustrated in Fig. 3. Compared to Alg. 1, threads are still allocated the private PRNGs and accumulators, but they cannot converge isolatedly. We introduce a global *checkpoint* after every batch of B walks, where stopping criterion is examined. The problem then reduces to reproducing the exact state at each checkpoint. In serial computing, successive B walks are trivially reproducible with a fixed initial seed. However, if they are scattered across T threads, **fine-grained reseeding** is needed. We assign each walk a **unique** ID (s, u, v), indicating the v-th walk in the u-th batch, and s is a user-defined seed. When a thread fetches a walk ID (s, u, v), it seeds its PRNG accordingly (e.g., using $s+uB+v$ as a unique seed), ensuring deterministic sampling. Thus, the merged result of a batch will be independent of the execution order or DOP. Also, at each checkpoint, the IDs of executed walks are predetermined, so a reproducible convergence is ensured.

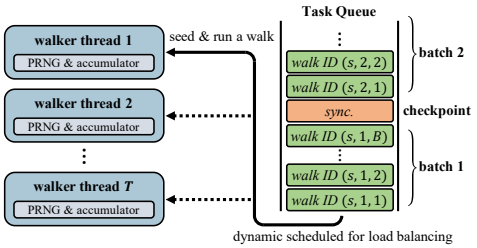

Fig. 3. The proposed parallel scheme with DOP-independent reproducibility, where every walk is assigned a unique ID and every batch of B walks is parallelized and dynamically scheduled across T threads.

With the above derivation, we present the parallel FRW algorithm with DOP-independent reproducibility in Alg. 2. The core idea is fine-grained reseeding (Line 6). The actual seed value can be tailored to specific PRNGs. Line 5 means to parallelize B walks across T threads and then synchronize. Besides, we fix the implementation of PRNGs, so reproducibility is also independent of the machine used. Note that this algorithm also suits multi-process computing, although we present it in a multi-threaded scenario for simplicity.

Algorithm 2: The parallel FRW algorithm with DOP-independent reproducibility.

Input: Structure with N conductors, master conductor i, error tolerance ϵ, T threads, global seed s.
Output: A row in capacitance matrix $(\widehat{C}_{i1}, \ldots, \widehat{C}_{iN})$.

1 Construct Gaussian surface enclosing i;
2 Init T threads with private PRNGs and accumulators;
3 $u \leftarrow 1$;
4 **repeat**
5 **Run T threads to parallelize loop: for $v \leftarrow 1$ to B**
6 Seed PRNG in current thread with (s, u, v);
7 $(\omega, \text{conductor } k) \leftarrow$ a random walk;
8 accumulate$(\omega, \text{conductor } k)$;
9 Merge T accumulators to compute $(\widehat{C}_{i1}, \ldots, \widehat{C}_{iN})$;
10 $u \leftarrow u + 1$;
11 **until** *estimated error* $< \epsilon$ *// the global checkpoint*;

C. Optimizations for Efficiency and Numerical Stability

In the proposed Alg. 2, the randomness of parallel FRW is effectively managed to reproduce results. However, its efficiency and numerical stability still need careful considerations.

Load balancing is more challenging in our scheme with synchronization than in the isolated scheme of Alg. 1. First, we should allocate enough workload per batch by choosing $B \gg T$, to boost parallel resource utilization. Second, consid-

[1]In double-precision floating-point format, bitwise distinct numbers share at most 16 decimal significant digits [25], so $d_{ij} = 17$ signifies bitwise equality.

ering that walk lengths are random and highly divergent, we use a task queue with dynamic scheduling. With these tricks, we achieve a near-linear parallel speedup.

Counter-based RNGs (CBRNGs) [22] are cryptography-based and stateless PRNGs. We use CBRNGs to remove the overhead of reseeding. Mersenne Twister [26], the most popular PRNG, does not suit our need of frequent reseeding because: 1) seeding its 624-words state costs too high, and 2) its superlong period of 2^{19937} is wasted. CBRNGs offer ideal statistical property and cost-free seeding [22], which significantly improves the efficiency of Alg. 2.

Numerically stable summation. Since weight values are accumulated and merged in random order, the non-associativity of floating-point arithmetic causes instability. We apply the well-known *Kahan summation*, also referred as *compensated summation*, which tracks round-off errors for higher accuracy [24], [25]. This makes bitwise reproducibility much more probable, though not guaranteed.

Multi-level parallelism. Running Alg. 2 with too many threads can be inefficient, because the batch workload may be insufficient, leaving some threads idle. If there are multiple master conductors, we can divide the T threads for concurrent extractions. This multi-level parallelism allows more scalable performance, with reproducibility unaffected.

IV. Enhancing Reliability of FRW Results with Physical Properties Guaranteed

Physics-related reliability is defined as ensuring capacitance matrix's properties in Sec. II-A, and will be referred to as "reliability" hereafter. Modified FDM equations can ensure some properties [20], but this is infeasible for the FRW method since its results are independently estimated by construction. From another perspective, however, FRW's key feature is the statistical confidence intervals of capacitance estimates. This inspires us to regularize the capacitance matrix using *constrained multi-parameter estimation* [27], where the statistical nature of FRW method can be fully leveraged.

Before describing our approach, it is worth discussing its necessity. In fact, desired properties can be achieved via some naive adjustments, e.g., replacing the diagonals with the sums of off-diagonal terms. However, we observed that errors of the off-diagonal terms can accumulate and even result in totally wrong self-capacitances. Using constrained estimation technique is more reliable and practical.

A. Problem Formulation

Suppose the FRW method extracts for master conductors 1 to N_m and outputs a matrix $\widehat{\mathbf{C}} = (\widehat{C}_{ij})_{N_m \times N}$. We call it an *observation* of the underlying *parameter* $\mathbf{C}$, i.e., the true capacitance matrix. In conventional FRW solvers, $\widehat{\mathbf{C}}$ is treated as the final result, which is already an unbiased estimate. However, we seek a better estimate $\widehat{\mathbf{C}}^*$ by incorporating the *a priori* knowledge $\mathcal{S}(\mathbf{C}) \geq 0$, which is the abstraction of Property 1-3 in Sec. II-A. The problem is stated in the form of constrained multi-parameter estimation:

$$\begin{aligned} &\text{With an observed } \widehat{\mathbf{C}} \text{ of parameter } \mathbf{C} \text{ s.t. } \mathcal{S}(\mathbf{C}) \geq 0, \\ &\text{find a new estimate } \widehat{\mathbf{C}}^* \text{ s.t. } \mathcal{S}(\widehat{\mathbf{C}}^*) \geq 0. \end{aligned} \tag{7}$$

The $\widehat{\mathbf{C}}^*$ is reliable by our definition. Moreover, since constrained estimates are known with a lower Cramér-Rao variance bound than unconstrained ones [27], the $\widehat{\mathbf{C}}^*$ is also more stastically **accurate**. Solving (7) can be seen as a physics-aware postprocessing or fine-tuning step for the FRW method.

B. Reliable Regularization Approach for Capacitance Matrix

We choose the constrained *maximum likelihood estimate* (MLE) as our new estimate. MLE is often considered **optimal** because maximum likelihood is a necessary condition to reach Cramér-Rao bound [27]. Actually, in our case, other derivations like *weighted least squares* and *minimizing Kullback–Leibler divergence* lead to exactly the same solution. It should be noted that our derived estimator can still be proven **unbiased**.

To solve the problem, we need to characterize $\widehat{\mathbf{C}}$ first. Since the sample X_{ij}'s distribution is intricate, we instead consider the asymptotic distribution of sample mean according to central limit theorem:

$$\widehat{C}_{ij} \sim \mathcal{N}(C_{ij}, \frac{\mathrm{Var}(X_{ij})}{M}),\ M \to \infty. \tag{8}$$

To simplify analysis, we use an estimated variance σ_{ij}^2:

$$\sigma_{ij}^2 = \frac{\sum_{m=1}^{M}(x_{ij}^{(m)} - \widehat{C}_{ij})^2}{M(M-1)} \approx \frac{\mathrm{Var}(X_{ij})}{M}. \tag{9}$$

Next, we inspect the independence between $\widehat{C}_{ij}$'s. Between different rows, it is natural. Within rows, $(\widehat{C}_{ij}, \widehat{C}_{ik})$ are uncorrelated since

$$\mathrm{Cov}(\widehat{C}_{ij}, \widehat{C}_{ik}) = \mathbb{E}(\widehat{C}_{ij}\widehat{C}_{ik}) - C_{ij}C_{ik} = \frac{-C_{ij}C_{ik}}{M} \to 0. \tag{10}$$

We further assume their independence because only $\frac{1}{M}$ of their samples are dependent. Therefore, $\widehat{\mathbf{C}}$ is fully characterized by its distribution conditional on the underlying parameter.

$$p(\widehat{\mathbf{C}}|\mathbf{C}) = \prod_{1 \leq i \leq N_m, 1 \leq j \leq N} f(\widehat{C}_{ij}|C_{ij}, \sigma_{ij}^2) \tag{11}$$

where f is probability density function of Gaussian distribution. The MLE solution is given by

$$\max_{\mathbf{C}} \ln p(\widehat{\mathbf{C}}|\mathbf{C}) \Leftrightarrow \min_{\mathbf{C}} \sum_{i,j} \frac{(C_{ij} - \widehat{C}_{ij})^2}{\sigma_{ij}^2};\ \text{s.t. } \mathcal{S}(\mathbf{C}) \geq 0. \tag{12}$$

To simplify the solution, we remove the first constraint in $\mathcal{S}$ (Property 1). Violating the sign property means that an estimate has an error larger than itself, which is very rare. In practice, these wrong capacitances (if existing) are minimal and negligible. We ensure Property 1 by simply discarding positive coupling capacitances and compensating on the diagonal.

Now, (12) reduces to a quadratic optimization with linear equality constraints. Let N_c be the length of optimization variable, i.e., the number of capacitances. Obviously, $N_m < N_c \leq N_m N$. Directly solving (12) takes $O(N_c^2)$ time, which is considerable for large structures. We develop an $O(N_m^2 + N_c)$ technique that remarkably speeds up the solution. First, for coupling conductors never hit by random walks, $\widehat{C}_{ij} = \sigma_{ij} = 0$. Thus, $\widehat{C}_{ij}^*$ (and $\widehat{C}_{ji}^*$ if existing) is directly solved as 0 and

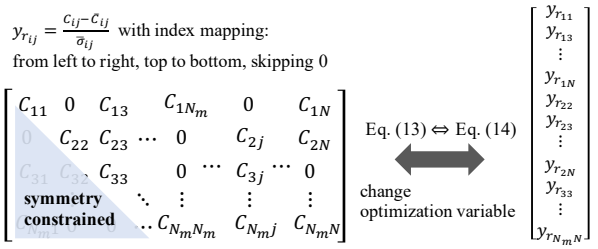

Fig. 4. By substituting optimization variable and rearranging (13) into (14), we convert a constrained quadratic optimization into a least-norm problem.

removed from the optimization. Then, $C_{ij}, \forall i > j$ are substituted with C_{ji} to satisfy Property 2. The optimization variables reduce to those above the diagonal. We rearrange (12) into

$$\begin{aligned} &\min_{\mathbf{C}} \sum_{i=1}^{N_m}\sum_{j\geq i} \frac{(C_{ij}-\overline{C}_{ij})^2}{\overline{\sigma}_{ij}^2}, \text{ s.t. } \sum_{j<i} C_{ji} + \sum_{j\geq i} C_{ij} = 0, \forall i \\ &\text{where } (\overline{C}_{ij}, \overline{\sigma}_{ij}^2) = (\widehat{C}_{ij}, \sigma_{ij}^2), j = i \text{ or } j > N_m, \\ &(\overline{C}_{ij}, \overline{\sigma}_{ij}^2) = (\frac{\sigma_{ji}^2 \widehat{C}_{ij} + \sigma_{ij}^2 \widehat{C}_{ji}}{\sigma_{ij}^2 + \sigma_{ji}^2}, \frac{\sigma_{ij}^2 \sigma_{ji}^2}{\sigma_{ij}^2 + \sigma_{ji}^2}), i < j \leq N_m. \end{aligned} \tag{13}$$

We change variable to a vector $\mathbf{y}$, with $y_{r_{ij}} = \frac{C_{ij}-\overline{C}_{ij}}{\overline{\sigma}_{ij}}$ and index mapping shown in Fig. 4. Eq. (13) is further rewritten as

$$\min_{\mathbf{y}} \|\mathbf{y}\|_2, \text{ s.t. } \mathbf{A}\mathbf{y} = \mathbf{b}, \tag{14}$$

where the linear equality constraint exactly corresponds to Property 3. This is a least-norm problem of an underdetermined linear equation. We can prove that $\mathbf{A} \in \mathbb{R}^{N_m \times N_c}$ has full row rank, so (14) has a closed-form solution:

$$\mathbf{y}^* = \mathbf{A}^\dagger \mathbf{b} = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{b}. \tag{15}$$

The desired $\widehat{\mathbf{C}}^*$ can be recovered from $\mathbf{y}^*$, and we can prove it remains an unbiased estimate of $\mathbf{C}$.

By inspecting the structure of $\mathbf{A}$, we can avoid explicitly constructing it. Let $\widetilde{\mathbf{A}} = \mathbf{A}\mathbf{A}^T \in \mathbb{R}^{N_m \times N_m}$:

$$\widetilde{\mathbf{A}}_{ii} = \sum_{j<i} \overline{\sigma}_{ji}^2 + \sum_{j\geq i} \overline{\sigma}_{ij}^2, \widetilde{\mathbf{A}}_{ij} = \overline{\sigma}_{ij}^2, \mathbf{b}_i = -\sum_{j<i} \overline{C}_{ji} - \sum_{j\geq i} \overline{C}_{ij}. \tag{16}$$

Since $\widetilde{\mathbf{A}}$ is sparse, solving with Cholesky factorization [28] takes about $O(N_m^2)$. Multiplying by $\mathbf{A}^T$ can be seen as scaling each element by specific $\overline{\sigma}_{ij}$, which can be done in $O(N_c)$.

Finally, our regularization approach is presented as Alg. 3, which achieves the optimal MLE for the constrained estimation in (7) if there is no positive $\widehat{C}_{ij}^*$ (see Line 6). Alg. 3 can serve as a postprocessing step in FRW-based capacitance extraction, with an acceptable cost of $O(N_m^2 + N_c)$.

C. More Practical Considerations

The proposed approach has been shown to be theoretically optimal, but a potential concern is that the three properties are not always equally important in practice. Hence, we discuss some possible adjustments to Alg. 3, while it is important to note that our numerical experiments in Sec. V are still based on the **original** version of Alg. 3.

In applications like touchscreen design, certain coupling capacitances are of particular interest. In that case, Property 3 is not compulsory. We can prove that: without the constraint of Property 3, the solution of (12) is exactly $\overline{C}_{ij}$ in (13), which is an intuitive symmetrization formula. It only affects two related values and is less likely to induce unexpected error. In such cases, we can optionally simplify Alg. 3 to only Line 1 and 2.

As for IC design, self-capacitances are usually considered more critical. Due to robustness concern, drastic changes to self-capacitances should be avoided. We can achieve this by increasing the weights of C_{ii}'s in (12), e.g., scaling it by a large factor. In that case, the regularized results are no longer MLEs, but still satisfy unbiasedness and the desired properties.

Algorithm 3: The reliable regularization approach for FRW results based on constrained MLE.

Input: Unreliable result $\widehat{\mathbf{C}} \in \mathbb{R}^{N_m \times N}$, estimated σ_{ij}^2's.
Output: Reliable result $\widehat{\mathbf{C}}^*$ ensuring the Properties 1-3.
1 Ignore all zeros and their symmetric positions in $\widehat{\mathbf{C}}$;
2 Compute $(\overline{C}_{ij}, \overline{\sigma}_{ij}^2)$ in (13) $\forall i \leq j$;
3 Construct $\widetilde{\mathbf{A}}, \mathbf{b}$ in (16);
4 Solve $\mathbf{y}^* = \mathbf{A}^T(\widetilde{\mathbf{A}}^{-1}\mathbf{b})$ with Cholesky factorization;
5 Recover $\widehat{\mathbf{C}}^*$ from $\mathbf{y}^*$ by $\widehat{C}_{ij}^* = \overline{\sigma}_{ij} y_{r_{ij}}^* + \overline{C}_{ij}$;
6 Delete positive $\widehat{C}_{ij}^*$ (extremely rare) and add it to $\widehat{C}_{ii}^*$;

V. Numerical Results

We have implemented the proposed algorithms in C++, with multi-threading based on OpenMP. The following variant algorithms are tested. **FRW-R**: the FRW solver with DOP-independent reproducibility (Alg. 2) and optimizations in Sec. III-C. **FRW-NK**: FRW-R without the Kahan summation. **FRW-NC**: FRW-R without CBRNG, using Mersenne Twister instead. **FRW-RR**: the FRW solver with both reproducibility (Alg. 2) and reliability (Alg. 3). In all tests, we set $B = 10000$ for Alg. 2, and the weight scaling discussed in Sec. IV-C is not applied. We have also implemented the algorithm in [1], i.e., Alg. 1, as the baseline method.

Six interconnect structures from IC design are tested, whose details are described in Table I. Recall that N_m is the number of master conductors, N is the number of all conductors, and N_c is the number of non-zero capacitances. The stopping criterion for self-capacitance error is set to 0.1% for Cases 1 and 2, and 1% for all others. Unless otherwise stated, the experiments are conducted on a Linux server with Intel Xeon 8375C CPUs.

TABLE I
Details about the test cases.

Case	N_m	N	N_c	Description
1	3	4	12	Parallel-wire structure obtained from [5]
2	3	4	12	Parallel-wire structure obtained from [5]
3	38	40	866	Voltage-controlled oscillator (VCO) design
4	129	131	10335	Analog-to-digital converter (ADC) design
5	653	657	15778	Static random-access memory (SRAM) design
6	48384	48386	926503	A large structure

A. Validating DOP-Independent Reproducibility

To validate the improved reproducibility, we add a second machine with AMD EPYC 7V12 CPUs for experiment. The algorithms are run on these two machines with two test modes:

- Fixed DOP (set $T = 16$): Repeat 32 extractions on each machine.
- Varied DOP: Run 32 extractions with $T = 1, 2, \ldots, 32$ threads on each machine.

In each mode, 64 extraction results are obtained, leading to 64×63/2=2016 pairwise comparisons to evaluate the reproducibility indices (RIs) in (6). Recall that for each comparison, RI denotes the minimum number of matched decimal digits in the capacitances. The minimum and average RIs for Cases 1-6 are listed in Table II.

TABLE II
THE REPRODUCIBILITY RESULTS OF THE ALGORITHMS (MEASURED BY RIS IN (6)) AFTER EXECUTING 64 REPEATED EXTRACTIONS ON TWO MACHINES.

Mode	Case	Alg. 1 [1]		FRW-NK		FRW-R		FRW-RR	
		RI_{min}	RI_{avg}	RI_{min}	RI_{avg}	RI_{min}	RI_{avg}	RI_{min}	RI_{avg}
Fixed DOP	1	13	14.0	13	13.1	**17**	**17.0**	**17**	**17.0**
	2	14	14.1	12	13.1	**17**	**17.0**	**17**	**17.0**
	3	12	12.7	11	11.6	**13**	**13.8**	**13**	13.7
	4	11	11.0	9	9.0	**13**	**13.0**	12	12.0
	5	11	11.0	10	10.0	**13**	**13.0**	**13**	**13.0**
	6	11	11.0	**17**	**17.0**	**17**	**17.0**	**17**	**17.0**
Varied DOP	1	0	1.2	11	12.4	16	16.9	**17**	**17.0**
	2	1	2.0	12	12.4	**17**	**17.0**	**17**	**17.0**
	3	0	0.2	10	11.3	**13**	**13.7**	**13**	13.5
	4	0	0.0	11	11.0	**13**	**13.0**	12	12.0
	5	0	0.0	9	9.0	**13**	**13.0**	12	12.0
	6	0	0.0	**17**	**17.0**	**17**	**17.0**	**17**	**17.0**

From Table II, we see that the reproducibility of Alg. 1 [1] is lost when the degree of parallelism (DOP) changes, while the proposed algorithms maintain DOP-independent reproducibility. Comparing the FRW-NK and FRW-R, we also see that the Kahan algorithm significantly enhances numerical stability. Recall that an RI of 17 indicates bitwise identical results, which is more probable with the Kahan algorithm. Moreover, the regularization step (Alg. 3) in FRW-RR has a slight impact on reproducibility. Results for FRW-NC are omitted because simply changing PRNGs does not affect reproducibility.

Next, we test the parallel efficiency of the proposed scheme. Fig. 5 shows the runtime curves for all FRW variants tested on representative Cases 1 and 3. Comparing FRW-NC and FRW-R, we see that using the CBRNG with cost-free seeding brings a speedup about 2×. The parallel efficiency of our solvers FRW-R and FRW-RR is ideal and comparable to that of Alg. 1 [1].

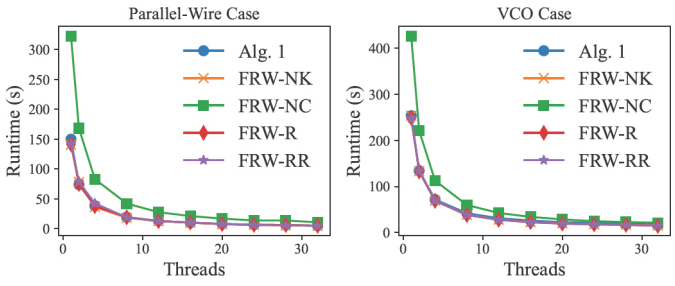

Fig. 5. Total runtime vs. the number of threads, tested on Cases 1 and 3.

B. Validating the Reliable Regularization Approach

To assess the overall accuracy of our FRW solvers, we obtain high-precision reference values with a commercial 3-D capacitance extraction tool. We calculate an average relative error of all capacitances, weighted by the absolute value of each capacitance, denoted as Err_{cap}:

$$\text{Err}_{\text{cap}} = \frac{\sum_{\text{all caps}} |C_{ij}^{(\text{ours})} - C_{ij}^{(\text{ref})}|}{\sum_{\text{all caps}} |C_{ij}^{(\text{ref})}|}. \tag{17}$$

We should also define evaluation metrics for the physics-related reliability, to examine how much the capacitance results deviate from Properties 1-3. As discussed in Sec. IV-B, Property 1 is rarely violated and is therefore omitted. Corresponding to Property 2 and 3, we define Err_2 and Err_3 to represent the "asymmetry error" and "row-sum error" of results, respectively. We also use weighted average relative errors:

$$\text{Err}_2 = \frac{\sum_{1\le i<j\le N_m} |C_{ij} - C_{ji}|}{\sum_{1\le i<j\le N_m} |C_{ij}|}, \quad \text{Err}_3 = \frac{\sum_{i=1}^{N_m} \left|\sum_j C_{ij}\right|}{\sum_{i=1}^{N_m} |C_{ii}|}. \tag{18}$$

The errors of capacitance results and the errors of reliability for Cases 1-6 are listed in Table III. The commercial tool fails on the largest Case 6, so for it Err_{cap} is absent. It is evident that the reliability of FRW-RR's results is strictly guaranteed, and the capacitances are therefore slightly more accurate. Its Err_{cap} is 21% lower than that of FRW-R on average. As for the Alg. 1 and our FRW-R, although capacitances are indeed unbiased and accurate, their deviations from physical properties (Err_2 and Err_3) are considerable. The T_{total} in Table III represents the total runtime. The runtime differences are not significant and can be attributed to random fluctuations in parallel computing. We also list T_{post} to show the time cost by the additional postprocessing step (Alg. 3) in FRW-RR. The overhead of our regularization approach is negligible, even for Case 6 with a large N_c.

TABLE III
THE ERRORS OF RELIABILITY, ERRORS OF CAPACITANCES, AND RUNTIMES FOR CASES 1-6, ALL TESTED WITH 16 THREADS.

Case	Alg. 1 [1]				FRW-R				FRW-RR				
	Err_2	Err_3	Err_{cap}	T_{total}	Err_2	Err_3	Err_{cap}	T_{total}	Err_2	Err_3	Err_{cap}	T_{total}	T_{post}
1	0.1%	7e-5	**0.32%**	9.8s	0.4%	1e-4	0.38%	8.3s	**0**	**1e-16**	0.34%	8.3s	1ms
2	0.3%	7e-4	0.28%	9.5s	0.3%	8e-4	0.27%	8.4s	**0**	**7e-17**	**0.24%**	8.6s	1ms
3	4%	3e-3	1.61%	27s	4%	4e-3	1.83%	26s	**0**	**1e-16**	**1.31%**	27s	1ms
4	2%	7e-3	0.86%	252s	2%	8e-3	1.10%	245s	**0**	**1e-16**	**0.81%**	245s	12ms
5	5%	3e-3	1.78%	178s	5%	4e-3	1.78%	173s	**0**	**2e-16**	**1.29%**	174s	9ms
6	24%	9e-4	-	3001s	13%	8e-4	-	3015s	**0**	**1e-16**	-	3019s	0.7s

VI. CONCLUSIONS

Due to its stochastic nature, the FRW-based capacitance extraction faces issues of reproducibility and physics-related reliability in practical use. In this work, we have resolved these issues and developed an improved capacitance extractor FRW-RR. FRW-RR achieves degree-of-parallelism (DOP)-independent reproducibility, validated by repeated experiments, and demonstrates remarkable advantages over existing method [1]. FRW-RR also guarantees the physical properties of capacitance results through a novel regularization approach based on constrained multi-parameter estimation, and thus improves accuracy by 21% on average in experiments. Importantly, FRW-RR maintains comparable efficiency and parallel scalability with existing FRW schemes.

REFERENCES

[1] M. Song, Z. Xu, W. Yu, and L. Yin, "Realizing reproducible and reusable parallel floating random walk solvers for practical usage," in *Proc. DATE*, 2019, pp. 192–197.

[2] W. Yu and X. Wang, *Advanced Field-Solver Techniques for RC Extraction of Integrated Circuits*. Springer, 2014.

[3] W. Yu, Z. Wang, and X. Hong, "Preconditioned multi-zone boundary element analysis for fast 3d electric simulation," *Engineering analysis with boundary elements*, vol. 28, no. 9, pp. 1035–1044, 2004.

[4] Y. L. Le Coz and R. B. Iverson, "A stochastic algorithm for high speed capacitance extraction in integrated circuits," *Solid-State Electron.*, vol. 35, no. 7, pp. 1005–1012, 1992.

[5] W. Yu, H. Zhuang, C. Zhang, G. Hu, and Z. Liu, "RWCap: A floating random walk solver for 3-D capacitance extraction of very-large-scale integration interconnects," *IEEE TCAD*, vol. 32, no. 3, pp. 353–366, 2013.

[6] Z. Xu, W. Yu, C. Zhang, B. Zhang, M. Lu, and M. Mascagni, "A parallel random walk solver for the capacitance calculation problem in touchscreen design," in *Proc. GLSVLSI*, 2016, pp. 99–104.

[7] M. Song, Z. Xu, W. Xue, and W. Yu, "A distributed parallel random walk algorithm for large-scale capacitance extraction and simulation," in *Proc. GLSVLSI*, 2018, p. 189–194.

[8] K. Zhai, W. Yu, and H. Zhuang, "GPU-friendly floating random walk algorithm for capacitance extraction of VLSI interconnects," in *Proc. DATE*, 2013, pp. 1661–1666.

[9] N. D. Arora, S. Worley, and D. R. Ganpule, "FieldRC, a GPU accelerated interconnect RC parasitic extractor for full-chip designs," in *Proc. EDSSC*, 2015, pp. 459–462.

[10] X. Wei, C. Yan, H. Zhou, D. Zhou, and X. Zeng, "An efficient FPGA-based floating random walk solver for capacitance extraction using SDAccel," in *Proc. DATE*, 2019, pp. 1040–1045.

[11] J. Demmel and H. D. Nguyen, "Parallel reproducible summation," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 2060–2070, 2014.

[12] M. Song, M. Yang, and W. Yu, "Floating random walk based capacitance solver for vlsi structures with non-stratified dielectrics," in *Proc. DATE*, 2020, pp. 1133–1138.

[13] M. Visvardis, P. Liaskovitis, and E. Efstathiou, "Deep-learning-driven random walk method for capacitance extraction," *IEEE TCAD*, vol. 42, no. 8, pp. 2643–2656, 2023.

[14] J. Huang and W. Yu, "Enhancing 3-D random walk capacitance solver with analytic surface green's functions of transition cubes," in *Proc. DAC*, 2024.

[15] J. Huang, M. Yang, and W. Yu, "The floating random walk method with symmetric multiple-shooting walks for capacitance extraction," *IEEE TCAD*, vol. 43, no. 7, pp. 2098–2111, 2024.

[16] I. Smolić and B. Klajn, "Capacitance matrix revisited," *Progress In Electromagnetics Research B*, vol. 92, pp. 1–18, 2021.

[17] Z. Liu and W. Yu, "pGRASS-Solver: A graph spectral sparsification-based parallel iterative solver for large-scale power grid analysis," *IEEE TCAD*, vol. 42, no. 9, pp. 3031–3044, 2023.

[18] T. El-Moselhy, I. M. Elfadel, and L. Daniel, "A markov chain based hierarchical algorithm for fabric-aware capacitance extraction," *IEEE Trans. Adv. Packag.*, vol. 33, no. 4, pp. 818–827, 2010.

[19] W. Yu, B. Zhang, C. Zhang, H. Wang, and L. Daniel, "Utilizing macromodels in floating random walk based capacitance extraction," in *Proc. DATE*, 2016, pp. 1225–1230.

[20] M. Yang and W. Yu, "Reliable macromodel generation for the capacitance extraction based on macromodel-aware random walk algorithm," *IEEE TCAD*, vol. 39, no. 4, pp. 946–951, 2020.

[21] Siemens EDA, *Parasitic Extraction Calibre xACT 3D*. Accessed: Sept. 2024. [Online]. Available: https://eda.sw.siemens.com/en-US/ic/calibre-design/circuit-verification/xact-3d/.

[22] J. K. Salmon, M. A. Moraes, R. O. Dror, and D. E. Shaw, "Parallel random numbers: as easy as 1, 2, 3," in *Proc. SC*, 2011, pp. 1–12.

[23] J. Maxwell, *A Treatise on Electricity and Magnetism*, 1873.

[24] W. Kahan, "Pracniques: further remarks on reducing truncation errors," *Commun. ACM*, vol. 8, no. 1, p. 40, 1965.

[25] J.-M. Muller, N. Brunie, F. De Dinechin, C.-P. Jeannerod, M. Joldes, V. Lefevre, G. Melquiond, N. Revol, and S. Torres, *Handbook of floating-point arithmetic*. Springer, 2018.

[26] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, 1998.

[27] T. L. Marzetta, "A simple derivation of the constrained multiple parameter Cramér-Rao bound," *IEEE Trans. Signal Process.*, vol. 41, no. 6, pp. 2247–2249, 1993.

[28] T. A. Davis, *Direct methods for sparse linear systems*. SIAM, 2006.