A NOVEL DIELECTRIC SIMULATION PLATFORM FOR DESIGNING HIGH VOLTAGE DEVICES: EVALUATION OF SURFACE CHARGING USING HIGH PERFORMANCE AND CLOUD COMPUTING

Carsten Trinitis[1,2*], Frank Messerer[3], Martin Schulz[1], Amir Bouslama[2], Bartosz Dobrzelecki[4], Andreas Blaszczyk[1,5]

[1]Technical University of Munich, School of Computation, Information, and Technology, Garching, Germany
[2]Technical University of Munich, School of Computation, Information, and Technology, Campus Heilbronn, Heilbronn, Germany
[3]Technical University of Munich, Professorship of High Voltage Engineering, München, Germany
[4]Rescale.com, Bydgoszcz, Poland
[5]Andreas Blaszczyk Consulting, Zurich, Switzerland
[*]Email: Carsten.Trinitis@tum.de

The first part of this paper presents a new electric fields simulation procedure that includes effects of surface charge accumulated during a discharge. The algorithm is based on a priori defined seed points, which specify surface locations where discharges arrive and the charge accumulation starts until the saturation stage is achieved. The concept of user-defined seed points is explained for a simple rod-barrier-plane arrangement and applied to simulation of a complex medium voltage switchgear. The second part of the paper is focused on performance characteristics of solving large equation systems based on the indirect integral formulation used in the surface charging simulation procedure. The fully populated matrix with a dimension in the range up to 1 million unknowns creates a bottleneck regarding resources available for engineers. In the paper we discuss the newest technologies like clusters, GPUs, and cloud computing that allow us to efficiently solve dielectric models in a typical engineering environment.

Keywords : DIELECTRIC SIMULATIONS, SURFACE CHARGE ACCUMULATION, LINEAR EQUATIONS SOLVER, HIGH PERFORMANCE COMPUTING (HPC), CLOUD COMPUTING

1. Introduction

In high voltage engineering, traditional dielectric simulation is based on the computation of electric fields. Such a computation prescribes electric potentials on active and grounded electrodes where no boundaries of dielectric materials are affected by surface charge accumulation. The results are valid for the initial stage, which we call “background field stage” (or Stage 1) and enable engineers to identify the critical spots where the calculated field strength is above the discharge inception limit. For many applications, like gas insulated high voltage switchgear (GIS) and large power transformers, keeping the calculated stresses below specified limits is essential to correctly design the device[(1)(2)].

When considering HVDC applications, it is necessary to find solutions for the surface charging problem on dielectric materials. For HVDC GIS this has been successfully done in the past[(3)].

For medium voltage applications, like switchgear and dry type transformers, an inception of a discharge does not necessarily lead to a breakdown. However, the propagating streamers depose electric charge on insulator surfaces within nanoseconds, which significantly changes the initial background field. The “background field stage” is followed by the “saturation stage” (Stage 2), which specifies the maximum possible surface charge that can be accumulated during the partial discharge. The knowledge of the electric field in the saturation stage can be crucial in judging whether a breakdown will occur or if the device can withstand the effects of partial discharges[(4)].

In this paper we present a simplified simulation procedure aimed at computing the saturation stage for complex geometries of engineering applications. Section 2 includes the basic formulation and explains the concept of user defined “seed points” illustrated on a simple example of rod-barrier-plane arrangement. Based on these “seed points” it is possible to iteratively define the location of saturation boundary condition without explicit simulation of discharge propagation. In this way, the modelling of comprehensive physical discharge phenomena can be reduced to a simple electrostatic computation. The new method has been implemented as an extension to the charge simulation (CSM) and boundary element methods (BEM), which have been successfully used since the 1960s[(5)]. The 3D BEM implementation was used to compute surface charging in a complex engineering model shown in Section 3. The seed point based approach turned to be robust enough to handle engineering models, however, the bottleneck is the multiple solution of large, fully populated systems of linear equations whose dimension is increased by the unknown saturation charge. Therefore, in Section 4 we discuss the current performance characteristics as well as their possible improvement using different HPC-environments like clusters, GPUs and cloud computing.

2. Surface Charging Simulation Procedure

2.1 Formulation For the interface between gas and solid insulating materials, we apply the following flux continuity equation when calculating the electrostatic field:

$$\varepsilon_{Ins} E_{nIns} - \varepsilon_{Gas} E_{nGas} = \sigma_s \qquad (1)$$

where ε_{Ins} and ε_{Gas} are permittivities, E_{nIns} and E_{nGas} are normal components of field strength in insulator and air, respectively; σ_s is the surface charge density accumulated in gas on the insulator. For the initial background field (Stage 1), we assume $\sigma_s = 0$. We also use Equation (*1*) with $\sigma_s = 0$ when computing the saturation stage (Stage 2), but only for dielectric surfaces not affected by charge accumulation. For all other dielectric surfaces we formulate a saturation boundary condition as follows:

$$\varepsilon_{Air} E_{nGas} = 0 \qquad (2)$$

$$\varepsilon_{Ins} E_{nIns} - \sigma_{sat} = 0 \qquad (3)$$

where σ_{sat} is the unknown density of saturation charge accumulated on insulator surface. Equations (*2*) and (*3*) ensure that no more charge carriers can be accumulated on the dielectric surface. Consequently, the saturation stage has been achieved. However, for most engineering applications finding the location of the saturation boundary condition is not straightforward. Therefore, we introduce the concept of "seed points" to enable a semi-automatic localization of the surface patches affected by charge accumulation.

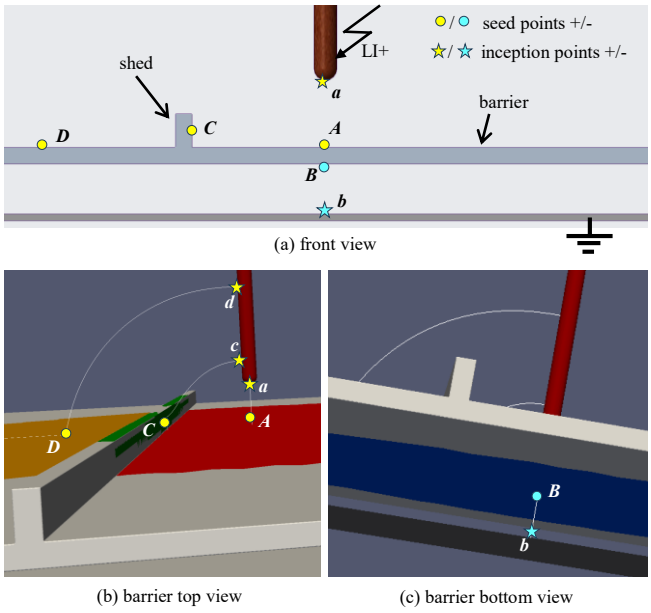
2.2 Seed Points A "seed point" is defined as a point on a dielectric surface where a propagating discharge arrives or (in case of electrodeless inception) starts from. Examples for seed points are shown in Fig. 1. for a simple arrangement of a rod-barrier-plane used for measurements of surface charging[6]. It consists of a cylindrical rod with 7 mm diameter to which a positive voltage is applied. The spherical tip of the rod is facing a dielectric barrier with dimensions 600x600x5 mm. For illustration purposes we have introduced a vertical "shed" as an additional feature of the barrier. Below the barrier a grounded plane is placed.

The seed points from Fig. 1 have been defined based on field lines that represent the possible discharge trajectory. They start from critical points on the rod or from protrusions (if available) at the grounded plane. They are denoted as discharge *inception points* ***a***, ***b***, ***c***, ***d***. The field lines hit the surface of the barrier in the corresponding seed points ***A***, ***B***, ***C***, ***D***. An important feature of a seed point is its polarity, which is the same as the polarity of the discharge they belong to (for seed and inception points in Fig. 1, yellow and blue colors denote positive and negative polarity, respectively). Around the seed points surface patches representing the accumulated charge are defined (see in Fig. 1b red, green, and yellow patches for positive polarity, in Fig. 1c blue patch for negative polarity). These patches determine the surface regions where the saturation boundary condition is applied – see Step 3 in Subsection 2.3.

The arrangement in Fig. 1 has been primarily used for the measurement of the surface charge along a flat barrier (without the shed). In such a case, the seed point ***A*** is considered for positive discharge from the rod or a combination of seed points ***A*** and ***B*** if an additional negative discharge is triggered from a protrusion at the grounded plane. Without the shed, estimating of surface patches for charge accumulation (red and blue) is straightforward: they expand across the whole top and bottom surfaces of the barrier or are limited to a certain radius from the rod axis if streamer propagation limits are applied. The measured charge is comparable with the simulation results[6][6].

After adding the shed, the definition of additional seed points may be required to enable surface charging on the shed and the upper barrier area left of the shed. In Fig. 1 seed points ***C*** and ***D*** have been added based on lateral inception at the rod. If the lateral inception in points ***c*** and ***d*** does not occur, the seed points ***C*** and ***D*** cannot be defined and the corresponding surface patches will not be charged. However, it may happen that the saturation charge calculated for ***A*** and ***B*** creates critical fields at convex and concave corners of the shed. Due to new inception points at the shed corners the positive discharge may "jump" over the shed and extend the charge accumulation along the whole barrier. Therefore, an analysis of the intermediate state (with seed points ***A*** and ***B*** only) is highly recommended. Based on this analysis a "manual" specification of seed points ***C*** and ***D*** (without field lines) can be performed. The accurate position of a seed point does not influence the final saturation charge except that seed points placed at not charged locations may be cancelled together with the corresponding surface patch – see Steps 3 and 5 in Subsection 2.3.

The analysis of the intermediate state provides an additional insight into possible discharge development and allows for shape optimization. A shape of the shed, which is more suitable to stop propagation of creeping discharges, could be designed.



(a) front view

(b) barrier top view (c) barrier bottom view

Fig. 1. Seed points and surface patches for saturation boundary condition defined in a rod barrier plane arrangement

2.3 Iterative Computation of Saturation Charge The iterative algorithm is presented in Fig. 2. The sequence of six steps can be explained as follows:

Step 1. The background field computation delivers results that are directly used in the next two steps. The geometrical model prepared for Stage 1 is reused for all subsequent solver iterations in Steps 4 and 6. Only boundary conditions are subject to changes.

Step 2. At least one seed point, as introduced above, must be defined together with its polarity to continue the procedure.

Step 3. The initial calculation of surface patches for the saturation boundary condition is conducted based on the normal field component E_{nGas} (*1*). This component of the electric field vector is positive if it points from gas to solid insulation and is negative in the opposite case. The polarity of the seed point must be

the same as the sign of E_{nGas} at the location of the seed point. Otherwise, the seed point must be rejected (this cannot happen if the seed point is defined by a field line). The polarity check is continued for all surface elements surrounding the seed point. Consequently, only elements with the sign of E_{nGas} matching the seed point polarity will be included into the calculated surface patch. The calculation is finished if no more surface elements can be attached to the patch. This calculation ensures that the surface patch for each seed point is contiguous. There are no detached areas separated from the seed points. However, it may happen that for two different seed points the same surface patch will be calculated. For all elements belonging to the calculated surface patches, the boundary condition (*1*) is replaced by the saturation boundary condition expressed by equations (*2*) and (*3*). Optional features that may be used in the calculation of surface patches are the distance limit for discharge propagation[6] and the threshold for the normal field value (exclude surfaces where E_{nGas} magnitude is below the predefined threshold) .

Step 4. Compared with Step 1 the number of unknowns is increased by the number of unknown saturation charges included in Equation (*3*). This computation may become longer and more resource demanding, particularly during the very first iteration.

Step 5. The polarity of the unknown surface charge is not predefined when formulating Equation (*3*). Consequently, it may happen that the resulting polarity of σ_{sat} is opposite to the polarity of the seed point associated with the surface patch. On the other hand, we assume that the charge carriers of another polarity than the propagating discharge are not available and cannot be accumulated. Therefore, the resulting σ_{sat} of opposite polarity is invalid. The solution proposed by our algorithm is based on removing the saturation boundary condition from all invalid locations. In these locations we go back to boundary condition (*1*). As a consequence of this removal, it may happen that the original surface patch becomes discontiguous. Similar to step 3, we remove the saturation charge boundary condition for the surface patches detached from any seed point and go back on these surfaces to boundary condition (*1*).

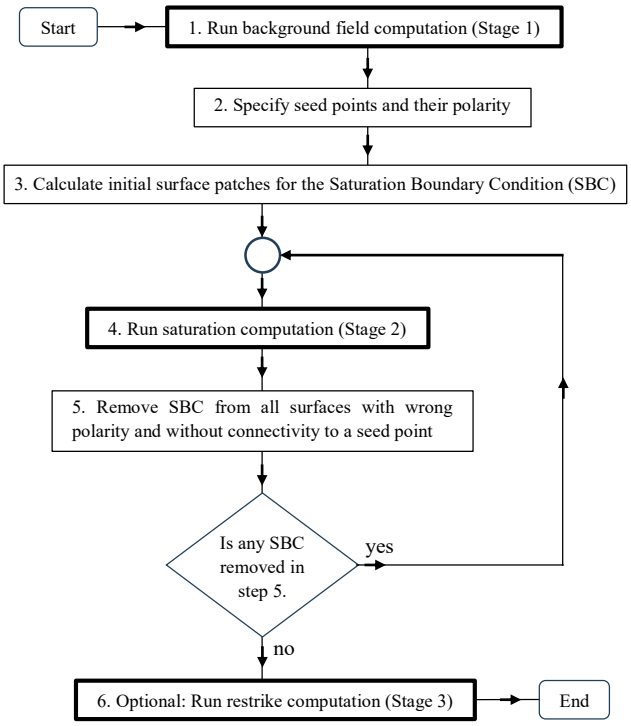


Fig. 2. Surface charging simulation procedure

Step 6. In the restrike stage only Equation (*1*) is formulated for all dielectric surfaces. However, in contrast to the background field stage a non-zero the value of σ_s may be used. It happens for all locations where the saturation boundary condition was applied in Step 4. For these locations we assume $\sigma_s = \sigma_{sat}$. In addition, the potential of active electrodes is changed either to zero or to an opposite polarity. It allows an evaluation of surface charge impact for alternating loads.

The main computational effort is associated with steps 1, 4 and 6 which include solving an electrostatic problem. Step 4 may need to be repeated a few times (in most cases no more than 5 times). In addition, the user may need to calculate intermediate charging states including several variants of seed point definition (step 2) and repeat the whole procedure several times. A good illustration for repeated computations is provided by the example discussed in Subsection 2.2 where several combinations of seed points make sense: A, AB, AC, ABC, ABCD, etc. It depends on goals of the engineering analysis as well as on geometrical dimensions, applied voltage, material properties etc. All these computations may become a bottleneck for large, real-life models as shown in the next sections

3. BEM Implementation Applied to a Real Device

The algorithm described in Section 2 can be implemented by utilizing any numerical technology including the widely used finite elements method (FEM[1]). The focus of our implementation is an integral approach based on *virtual* charges. It has its origins in the theory of electric images invented a long time before digital computers became available[7]. The first computer implementation of Lord Kelvin's concept was created in the 1960s at the Technical University Munich and is known as the Charge Simulation Method (CSM)[8]. The next milestone was the development of the electrostatic solver based on the Boundary Element Method (BEM)[9] in the 1980s and a BEM simulation suite for DC charging simulations in 2001[10]. The BEM-extension for surface charging is the most recent development[11]. An overview of all these integral implementations was presented at ISH2023[5]. More details are included on the Website of the Elfi-project, which also provides a framework for the simulation platform presented in this paper[12].

For testing the applicability and performance of the surface charging procedure, we have selected a switchgear component as shown in Fig. 3. A positive impulse of 125 kV is applied to the bus bar on the left side whereas the right side and all parts below the insulator are grounded. Based on field lines starting from inception points at location ***a*** the seed points ***A*** and the surface patches denoted by grey color are calculated. The resulting saturation charge suppresses the initial inception at ***a*** but increases the stress at points ***b*** and ***c***, which may lead to an inception at the edge ***c*** and finally to a breakdown between ***a*** and ***b***. The surface charging simulation procedure has been applied to different shape variants of the support insulator (with different field conditions around the critical edges). The inception voltage at point ***c*** computed for these variants in the saturation stage has shown a good correlation to the

[1] See e.g. https://www.comsol.com

breakdown voltage measured in prototype tests.

The dimension of the fully populated matrix created by the electrostatic BEM solver is 462,000. It corresponds to the number of virtual charges located at the corner nodes of triangular elements generated for this model. Due to additional unknown saturation charges, this dimension grows by 12 % in the first iteration of Stage 2 (reaching the level of 518,000). For the fourth and final iteration, the growth amounts to only 1.4% since the unknown saturation charges have been removed in Iterations 1 to 4 and the matrix dimension has decreased. The memory required for a fast solution is in the range of 1 Terabyte and the number of parallel processors must be above 50 to ensure acceptable computation times.

In spite of its high dimensionality, the model shown in Fig. 3 does not include all geometrical features. Some of them had to be skipped due to limited computational resources. An example is a rounding of insulator edges and corners, which is essential for accuracy of inception and surface charging computations. This rounding (R=0.2 mm) has been limited to selected edges and corners only (close to the charged surface patches and close to the inception point ***c***). Another feature not included in the simplified model from Fig. 3 is the representation of all three phases instead of the two neighboring ones. Having all three phases is convenient but usually such models must be reduced due to insufficient computational resources. After including the missing features, the dimension of the new model grows to 1,250,000. Such a dimension with a fully populated matrix creates a new challenge for engineering simulations.

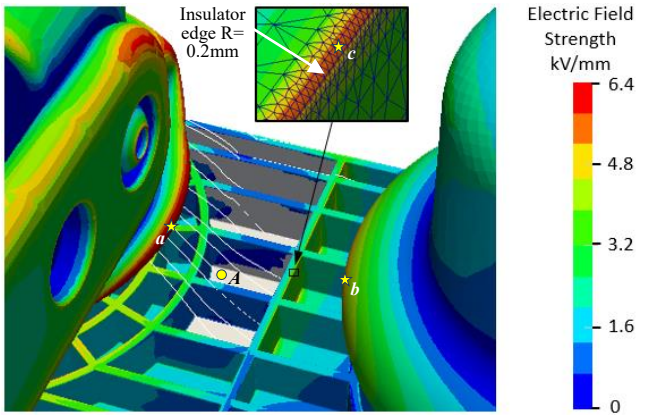


Fig. 3. Results of surface charge simulation for Safering switchgear component designed by ABB Electrification, Norway[13][4]

4. Performance Characteristics

For our performance studies we have generated three models with different dimensions: small (S), medium (M) and large (L). The small problem corresponds to the simple rod plane barrier arrangement shown in Fig. 1 with a dimensionality of 48,200 unknowns. The medium and large models are defined by two variants of the switchgear component introduced in Section 3: the first variant created within the initial study[4] with a dimensionality of 462,000 unknowns, and the second variant created explicitly for this study with a dimensionality of 1,250,000 unknowns. For all models the corresponding boundary element meshes have been generated using the CAD-system Creo[2] – even for the largest model the mesh generation can be performed within one minute on a standard laptop PC. The minimum memory required to efficiently solve these models amounts to approximately 9.3 GB, 860 GB, and 6.3 TB for S, M and L, respectively. All dimensions and memory requirements correspond to the basic background field computation (Stage 1). The unknown saturation charge increases the matrix dimension in Stage 2 by 10-20% and as a consequence also the memory requirement by 20-50%. However, this growth can be mitigated in future implementations by proper handling of non-fully populated matrices introduced by equations (*2*) and (*3*).

4.1. Experimental Results

The test runs were carried out on the following hardware configurations:

- A standard laptop PC with 128GB RAM, an eight core Intel W-10885M (Comet Lake) CPU running at 2.4GHz.
- A compute node with 2TB RAM and two Intel Xeon Gold 4548N (Sapphire Rapids) CPU running at 2.8GHz.
- Up to 64 cluster nodes with 512-2034GB RAM and a 128 core AMD Epyc Genoa 9554 CPU running at 3.1GHz.

Table 1 shows the configurations (the core-node distribution always guaranteed that communication is minimal and the entire matrix fits into main memory), and the resulting runtimes:

Table 1. Configuration and Runtimes for S, M, and L

Computer type	# of cores	Model size	Elapsed time, s			
			Build	Solve	Calc	Total
Laptop Xeon 128GB	8	S	185	35	195	378
Sapphire Rapids 2TB	8	S	56	17	56	129
	32	S	22	8	20	50
	8	M	5412	1721	5620	12753
	32	M	1782	894	1903	4579
Viper Cluster	8	S	71	13	68	152
	32	S	21	4	20	45
	512	S	1	9	3	13
	8	M	7720	1717	7630	17067
	32	M	1923	433	1923	4279
	512	M	129	47	134	310
	4096	M	21	124	48	193
	512	L	953	262	950	2165
	4096	L	139	457	225	852

[2] For details on Creo see https://www.ptc.com.

Our BEM based field simulation basically runs three phases[16]:

1. The first phase, depicted as "Build" in Table 1., represents the generation of the dense BEM coefficient matrix.
2. The second phase, "Solve", involves solving the resulting linear equation system using a GMRES solver[14].
3. The third phase, "Calc", refers to the computation of potential and electric field strength based on the solution vector obtained in the second phase.

Small problems like the S example (with 48,200 unknowns) can still be run on the laptop system within reasonable time, on the large cluster it did not make sense to run this problem on 4,096 nodes as the communication overhead became too high.

The M and L problems did not fit into 128GB main memory, i.e. the simulation runs were carried out on a 32 core Sapphire Rapids node and on the larger Viper cluster.

All binaries were compiled with the GNU Fortran and GNU C compiler v11.4.1 and OpenMPI v5.0 .

4.2 Discussion of the Results

With several Terabytes of main memory being available on the Viper cluster, a problem with more than 1 million unknowns could be run for the first time ever. For the M and L problems, the following phenomena can be observed:

The matrix generation and field calculation phases scale up with 4,096 vs. 512 cores, however, the solution phase performs best for 512 cores. However, the overall runtime is the shortest for 4,096 cores. Therefore, the optimal number of cores for the GMRES solver phase is not necessarily the one with the highest number of cores. In addition, with increasing number of cores the fraction of solving with regard to overall runtime becomes larger. Thus, porting the solver to a GPU architecture is a promising option, see Section 5.2 .

5. Discussion of the HPC Environment

5.1 Cloud Computing

Our simulation platform is versatile and can be deployed across various environments. For educational purposes, powerful laptops are sufficient to handle small-scale problems. Dedicated workstations and clusters equipped with thousands of cores and terabytes of memory can address large problems. However, such traditional on-premises high-performance computing (HPC) clusters are typically accessible only to academic institutions and large enterprises with the capacity to develop in-house HPC expertise. In contrast, cloud HPC resources can offer a more widely accessible and sustainable alternative for simulation platforms requiring extensive computational resources beyond the capabilities of individual workstations.

HPC as a Service (HPCaaS) platforms, such as e.g. Rescale3, streamline software deployment and version management.

Such vendor-agnostic cloud HPC platforms provide unified access to diverse hardware resources from various cloud service providers (CSPs) across multiple regions. Users can select the optimal configuration for their workload, from a single-node machine with 64 cores and up to 4TB RAM to a cluster with up to 480 cores and 200Gbps interconnect. These clusters are provisioned on-demand and exist only for the duration of the computation, aligning with peaks in computational resource demand.

Most importantly data, often representing valuable intellectual property, is encrypted both at rest and in transit using user-specific keys.

5.2 GPU Port

The current implementation of our electrostatic field simulation is based on the Message Passing Interface (MPI)[4] and running on CPU only. It has, however, the potential to benefit from GPU offloading, enhancing computational efficiency and performance. To migrate the simulation to a single GPU initially would require rewriting its numerical components (i.e., Build-Solve-Calc) as GPU kernels. For that, the reference sequential version running on a single CPU core serves as the foundation to ensure that data is stored contiguously, facilitating efficient GPU storage and access.

For a GPU implementation, another area to consider is whether all simulation data should reside on the GPU simultaneously throughout the different phases or only the parts of the data relevant to the current computation step. Specifically, should all data reside on the GPU at all times, or would it be more convenient for data parts that are not in use to be offloaded from the GPU during each computation phase.

Limiting the data parts stored on the GPU at once would allow for solving potentially larger problems by making more GPU storage available but would require more data transfers in the process.

First implementations for a GPU based design already exist: These are able to run the "Solve" part on the GPU using GMRES or other solvers utilizing a high-performance numerical linear algebra library. There are already existing algebraic GPU frameworks and libraries like Ginkgo[15] that could be integrated into the simulation system. Such libraries offer out-of-the-box solver kernels, which substitute rewriting the algebraic solver, GMRES. These solutions would still require a contiguous system matrix, making them more compatible with the sequential model initially. However, this is a strong limiting factor, as it means parallel matrix construction through the MPI module could not be used conveniently in such a manner.

As a workaround, more recent approaches propose building a distributed copy of the matrix on the host in a parallelized way with the MPI module and assembling these blocks to the full matrix on the GPU. Given a mapping between each matrix block and its location in the matrix, it is possible to transfer the given block by using a relative block index and memory offset directly to a well-defined memory location within the matrix allocated on the GPU. Such a process results in a complete matrix once all blocks are transferred.

[3] For details see https://rescale.com

[4] https://mpi-forum.org

6. Conclusion and Outlook

In this paper, we introduced a novel simulation approach taking into account effects of surface charge accumulated during a discharge.

Using three benchmark models with up to more than one million unknowns, extensive simulation runs were carried out on different hardware, including a large cluster with several Terabytes if main memory using more than 4,000 CPU cores. The possibility of deploying the simulation on a cloud system as an alternative for industrial users was also discussed. Regarding GPUs, single-GPU solvers present a significant challenge in terms of storage, as GPU device global memory is usually fixed and limited compared to CPU storage. For example, the NVIDIA H100 NVL Tensor has only 96 GB of device memory, which can be used for solving at most a problem size of around 150,000, under the assumption that no further storage is needed for the computation. Therefore, a distributed or multi-GPU approach becomes necessary for larger problem sizes. In such a setting, new parallelization techniques and algorithms would need to be designed. This would also involve rewriting the "Build" part, which is subject to future work.

7. Acknowledgments

The authors would like to thank Erwin Laure and Markus Rampp for providing access to and support on their Viper cluster at Max Planck Computing and Data Facility (MPCDF) Garching, and Hartwig Anzt and Tobias Ribizel for providing access to their Compute Nodes at TUM Campus Heilbronn.

References

(1) N. De Kock, M. Mendik, Z. Andjelic, A. Blaszczyk: "Application of 3D boundary element method in the design of EHV GIS components", IEEE Magazine on Electrical Insulation., Vol.14, No. 3, pp. 17–22 (1998)
(2) Z. Andjelic, E. Henriksen, G. Jorendal, H. Nordman, and G. Bertagnolli,"3D Simulation in Transformer Design," Proc. 10th Int. Symp. High Voltage Eng. (ISH'97), Montreal, Aug. 25-29, (1997)
(3) F. Messerer, M. Finkel, W. Boeck: „Surface charge accumulation on HVDC-GIS-spacer, ISEI 2002, pp. 421-425, Boston, MA, USA.
(4) A. Blaszczyk, E. Morelli, P. Homayonifar: 'Surface Charging Models for Prediction of Dielectric Withstand in Medium Voltage Range', IEEE Trans. on Magnetics, Vol. 57, (2021)
(5) A. Blaszczyk, F. Messerer, C. Trinitis: "Charge Simulation Method and surface charging computation for design of high voltage devices" Proc. of 23rd International Symposium on High Voltage Engineering, ISH Glasgow, pp. 402-408 (2023)
(6) H. K. Meyer: 'Dielectric barriers under lightning impulse stress', PhD. Diss., No. 106, NTNU Trondheim (2019)
(7) W. Thomson (Lord Kelvin): "Geometrical Investigations with Reference to the Distribution of Electricity on Spherical Conductors", Camb. Dublin Math. J. 3, 131 (1848)
(8) H. Steinbigler: "Anfangsfeldstärken und Ausnutzungsfaktoren rotationssymmetrischer Elektrodenanordungen in Luft", Dissertation TH München (1969)
(9) Z. Andjelic: "A contribution to the BEM for calculation and optimization of 3D electrostatic fields", Ph.D. Thesis, Faculty of Electrical Engineering, University of Zagreb (1984).
(10) F. Messerer, W. Boeck, H. Steinbigler, S. Chakravorti; „Enhanced field calculation for HVDC GIS", Gaseous Dielectrics IX, pp. 473-483, Maryland, 2001.
(11) A. Blaszczyk, T. Christen, H.K. Meyer, M. Schueller: "Surface charging formulations for engineering applications. Validation by experiments and transient models": Scientific Computing in Electrical Engineering SCEE 2018, Taormina, Springer Nature (2020)
(12) "Website : Elliptic Fields", http://ellipticfields.com, (accessed 4 March 2025)
(13) E. Attar, et al, "Eco-efficient puffer-type load break switch for medium voltage applications," 25th Int. Conf. on Electricity Distribution, CIRED, Madrid (2019)
(14) Y. Saad, M.H. Schultz: "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems". SIAM Journal on Scientific and Statistical Computing. 7 (3): 856–869. doi:10.1137/0907058. ISSN 0196-5204 (1986)
(15) Website; Ginkgo – a high-performance numerical linear algebra library for many-core systems, https://github.com/ginkgo-project/ginkgo.
(16) Z. Andjelić, B. Krstajić, S. Milojković, A. Blaszczyk, H. Steinbigler, M. Wohlmuth: Integral Methods for the Calculation of Electric Fields. Scientific Series of the International Bureau, Vol. 10, Forschungszentrum Jülich GmbH, Germany, 1992.