The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

ANISOTROPY RESPECTING CONSTITUTIVE NEURAL NETWORKS

WOUTER J. SCHUTTERT, MOHAMMED IQBAL ABDUL RASHEED AND BOJANA ROSIĆ

Applied Mechanics and Data Analysis (AMDA)
University of Twente
Horst Complex, 7522NB Enschede, The Netherlands
e-mail: w.j.schuttert@utwente.nl, www.utwente.nl/en/et/ms3/research-chairs/amda/

Key words: Stochastic anisotropy, Manifold Learning, SPD matrix, Constitutive NN

Summary. Modeling symmetric positive definite (SPD) material properties, such as thermal conductivity, under uncertainty often leads to a substantial computational burden. Neural networks can help mitigate these costs, yet standard architectures do not inherently preserve SPD properties. To address this, we propose novel neural network layers that map the space of SPD matrices to a linear space using logarithmic maps. We evaluate the performance of these networks by comparing different mapping strategies based on validation losses and uncertainty propagation. Our approach is applied to a steady-state heat conduction problem in a patched cube with anisotropic and uncertain thermal conductivity, modeled as a spatially homogeneous, tensor-valued random variable. The results indicate that the logarithmic mapping of tensor eigenvalues significantly improves learning performance, highlighting its utility in handling tensor data in neural networks. Furthermore, the formulation facilitate separation of strength and orientational information.

1 INTRODUCTION

Thermal material properties of composites and biological tissues [1, 2, 3] are often of anisotropic nature, meaning that material characteristics are directionally dependent. As these are not directly measurable, they need to be estimated or modelled as unknown. In addition, thermal characteristics are also known to vary due to variations in manufacturing process, presence of defects or natural phenomena. To be able to predict material behavior, one has to model thermal characteristics as uncertain which means that the properties such a thermal conductivity, considered in this paper, have to be modelled as a random variable. In contrast to the isotropic case in which thermal conductivity can be modelled as a scalar-valued random variable, anisotropy results in modeling of a tensor valued random variable. [4] proposed a stochastic tensor model using an exponential map and spectral decomposition, which is employed in this work to model SPD matrices as random variables.

To predict the temperature dependency on the conductivity, one has to propagate the uncertainties through the physics based-model, often numerically solved by the finite element method. In contrast to sampling based methods such as Monte Carlo, its quasi-variants and deterministic integration rules, the use of surrogate based modelling has shown great promise [5]. In particular, neural networks and their deep forms [6] have gained great popularity due to the ability of representing high nonlinearity in a scalable setting [7]. Most of existing classical neural networks do not incorporate physics-based parameters in their setting. In other words, these networks do not respect constitutive relations for the corresponding material model. Some that do respect such relations are i.e. [8] which proposes used a non-negative activation function and Chomsky decomposition to uphold symmetry and positive definiteness in Lagrangian dynamics. [9] where a physics based loss function is formulated to directly train neural networks, forgoing the need for traditional solvers. Further, specifically for material mechanics, some constitutive artificial networks use invariants for energy-based computations, but such implementations do not work when specific directional information is required, i.e. [10, 11]. Recently, [12] proposed a network architecture using eigenvalue activation functions and rotation-invariant matrices to model symmetric strain matrices.

In this paper, we propose incorporation of a new stochastic model for SPD valued random variables on a manifold to the neural network architecture, extending work done by [13, 14]. We propose a neural network that can handle both strength and directional information in the stochastic tensor. In other words, the neural networks can handle specific anisotropy for varying values of material constants, changing the anisotropy class. The adapted neural network architecture consists of a material model part and the classical MLP network used to represent the parameter-quantity of interest (QoI) map. The method is demonstrated on a 3D steady state thermal problem with anisotropic random thermal conductivity. Novel neural network layers are introduced to map from the SPD manifold to Euclidean space.

The paper is structured as follows. The first section introduces the thermal problem and tensor valued random variable. The second section describes the neural network architecture and different transformation layers studied. In the third section, results of different layers are compared by analyses of the validation losses and uncertainty propagation performance. The final section offers conclusions and discussion.

2 PATCHED CUBE WITH ANISOTROPIC TENSOR VALUED RANDOM THERMAL CONDUCTIVITY

Let be given the steady state heat conduction problem described by:

$$\nabla \cdot (\boldsymbol{C} \nabla T) = Q \text{ with b.c. a.e. on } \mathcal{G}. \tag{1}$$

Here, T [K] is the temperature, $\boldsymbol{C} \in \mathrm{Sym}^+(d)$ [W/(m*K)] is the thermal conductivity tensor with $\mathrm{Sym}^+(d)$ being the space of symmetric positive definite matrices, Q [W/m^3] is a volumetric source term and $\mathcal{G} \subset \mathbb{R}^3$ is the bounded domain of interest in Euclidean space.

To model the uncertainty associated with the anisotropic thermal conductivity $\boldsymbol{C}$, we model the tensor as a spatially homogeneous random variable. The tensor valued random variable $\boldsymbol{C}(\omega) \in \mathrm{Sym}^+(d)$ is defined in a probability space given by a triplet $(\Omega, \mathcal{F}, \mathbb{P})$ in which Ω is the sample space of elementary outcomes ω, $\mathcal{F}$ represents the sigma algebra, and $\mathbb{P}$ is the probability measure. The model is adopted from [4, 15] based on the spectral decomposition of the deterministic thermal conductivity tensor, where:

$$\boldsymbol{C} = \boldsymbol{Q} \boldsymbol{\Lambda} \boldsymbol{Q}^T \text{ with } \boldsymbol{\Lambda} \in \mathrm{Diag}{+}(d) \text{ and } \boldsymbol{Q} \in \mathrm{SO}(d). \tag{2}$$

The decomposition splits the strength and orientation information from each axis in the form of eigenvalues $\boldsymbol{\Lambda}$ and eigenvectors $\boldsymbol{Q}$. Diag+(d) is a Lie group of positive definite diagonal matrices, SO(d) is the Lie group of proper rotations. The spectral components allow for modelling of $\boldsymbol{\Lambda}(\omega)$ and $\boldsymbol{Q}(\omega)$ separately. Directly altering values in either SO(d) and Diag+(d) is challenging due to their manifold projections. To model variation in either of the quantities, we use an exponential map from the tangent space, or Lie algebra, towards the respective Lie group. The Lie algebra of Diag+(d) is $\mathfrak{diag}(d)$ and that of SO(d) is $\mathfrak{so}(d)$, both of which are unconstrained vector spaces, making stochastic modelling simpler. Thus, we introduce the mapping:

$$(\boldsymbol{Y}, \boldsymbol{W}) \longmapsto (\boldsymbol{\Lambda}, \boldsymbol{Q}) = (\exp \boldsymbol{Y}, \exp \boldsymbol{W}) \longmapsto \boldsymbol{C} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T, \tag{3}$$

in which $\boldsymbol{Y} \in \mathfrak{diag}(d)$ is a diagonal matrix and $\boldsymbol{W} \in \mathfrak{so}(d)$ is a skew symmetric matrix, representing the strength and orientation matrix respectively. Extending Eq. (3) to the stochastic counterpart, one may introduce:

$$\boldsymbol{\Lambda}(\omega) = \exp(\boldsymbol{Y}(\omega)) = \text{ diag } (\exp(y_i(\omega))) \quad \text{with} \quad y_i(\omega) \sim \mathcal{N}(\mu_i, \sigma_i), \quad i = 1, ..., d \tag{4}$$

for varying strength. By use of exponential mapping, we ensure the positive definite property of the tensor. Furthermore, we introduce the stochastic rotational $\boldsymbol{R}(\omega)$ such that:

$$\boldsymbol{C}(\omega) = \boldsymbol{R}(\omega)\overline{\boldsymbol{Q}}\boldsymbol{\Lambda}(\omega)\overline{\boldsymbol{Q}}^T\boldsymbol{R}(\omega)^T \quad \forall \omega \in \Omega. \tag{5}$$

in which $\overline{\boldsymbol{Q}}$ is the deterministic counterpart of the eigenvector and $\boldsymbol{R}(\omega)$ is modelled as,

$$\boldsymbol{R}(\omega) = \exp(\boldsymbol{W}(\omega)) \text{ where } \boldsymbol{W}(\omega) = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} (\omega) \quad \text{with} \quad \phi = ||\boldsymbol{w}||. \tag{6}$$

in which ϕ is the rotation angle and $\boldsymbol{w}$ is the Euler vector or random angle variables which are drawn from a von Mises Fisher distribution (vMF), with mean direction vector μ and concentration parameter κ, which are akin to the mean and standard deviation of a normal distribution [16]. vMF is a generalization of a Gaussian distribution on the unit sphere. Introducing the stochastic model for the tensor into Eq. (1) one further obtains,

$$\nabla \cdot (\boldsymbol{C}(\omega)\nabla T(\boldsymbol{x}, \omega)) = Q \text{ with b.c. a.e. on } \mathcal{G} \times \Omega. \tag{7}$$

In which the temperature field also becomes stochastic. To quantify uncertainty, one has to build the surrogate model that is sampling efficient. For this purpose, the previous model is first spatially discretized by the FEM, and further sampled in a stochastic domain by a simple Monte Carlo procedure. In this manner, the data set is collected based on which T is constructed as described in the following section.

3 ANISOTROPY RESPECTING NEURAL NETWORK LAYERS

To map the varying input parameter $\boldsymbol{\Lambda}(\omega)$ and $\boldsymbol{R}(\omega)$ to the QoI $T(x, \omega)$, we propose the NN architecture as shown in Fig. (1). The network consists of two parts: the part that introduces the anisotropy notation and the second part that maps the tensor C to the QoI. Fig. (1) illustrates the proposed network architecture: an input node receives a tensor sampled from Eq. (5), a transformation layer maps the tensor to $\mathbb{R}^{n_1}$, followed by standard hidden layers, and an output layer that produces the temperature field.

The standard for converting tensor to inputs for neural networks is to use a vectorize operation. But in this work we formulate alternatives which better preserve the information carried by

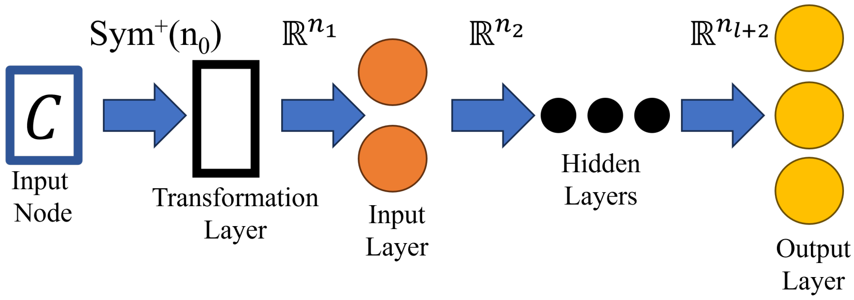

Figure 1: Neural network architecture

the tensor. It is possible to reformulate the tensor representation using a logarithmic mapping on the eigenvalues [13], inspired by the Log-Euclidean metric, thereby transforming them to the Lie algebra $\mathfrak{diag}$, a vector space. The tensor is reconstructed as:

$$\boldsymbol{Q} \log(\boldsymbol{\Lambda}) \boldsymbol{Q}^T = \mathbf{X} \Longrightarrow [x_{11}\ x_{12}\ x_{22}\ x_{13}\ x_{23}\ x_{33}]^T \tag{8}$$

This transformation, referred to as the log-eigenvalue operation (LogEig), results in $X \in \mathrm{Sym}(d)$, a linear space.

3.1 Strength-Orientational (StrOr) Layer

An alternative approach involves feeding the network separate information about strength and orientation. We introduce an eigenvalue/eigenvector-based transformation layer:

$$\log(\boldsymbol{\Lambda}), \boldsymbol{Q} \Longrightarrow [y_1\ y_2\ y_3\ q_{11}\ q_{21}\ q_{31}\ q_{12}\ q_{22}\ q_{23}\ q_{13}\ q_{23}\ q_{33}]^T \tag{9}$$

Similar to Equation 8, the log operation maps the eigenvalues to their Lie Algebra. For the q entries, this representation maintains the direct interpretation of vectors on the unit sphere hereby seperation the input channels for the strength and orientational (StrOr) information.

3.2 Strength-Angular (StrAng) Layer

To maintain consistency in representation, we can also use the logarithmic map to both eigenvalues and eigenvectors, mapping them to their respective Lie algebras. While this approach introduces computational complexity for the full eigenvector matrix compared to the diagonal eigenvalue matrix, it is equivariant. The mapping is given as,

$$\log(\boldsymbol{\Lambda}), \log(\boldsymbol{Q}) = \boldsymbol{W} \Longrightarrow [y_1\ y_2\ y_3\ w_1\ w_2\ w_3]^T. \tag{10}$$

While both $\boldsymbol{y}$ and $\boldsymbol{w}$ are represented as elements of $\mathbb{R}^3$, they belong to different spaces: $\boldsymbol{y}$ is in the Lie algebra of diagonal matrices, while $\boldsymbol{w}$ is in $\mathfrak{so}(3)$, the Lie algebra of SO(3). Now are again separated channels but for the strength and angular (StrAng) information. The comparison of these transformation layers now follows in the next section.

4 SURROGATE MODEL FOR THE PATCHED CUBE

The problem tackled in this paper is the steady state heat conduction of a patched cube with an anisotropic and uncertain thermal conductivity. The patched cube consists of a cube with six patches, each situated along an axis, which originate from the center point of the cube

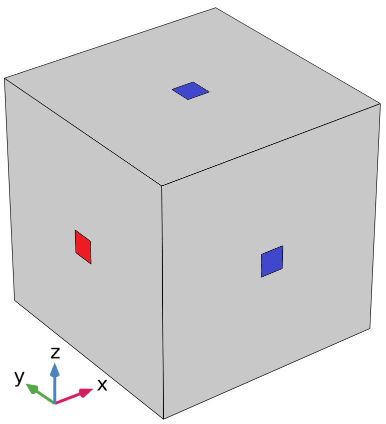

Figure 2: Geometry of the patched cube.

Table 1: Patched Cube Parameters.

Parameter	Value [Unit]
Cube length	0.05 [m]
Patch length	0.005 [m]
Q_h	50 [kW/m^2]
T_b	273.15 [K]

and run normal to the cubes faces. The geometry is shown in Fig. (2), where a patch on the YZ plane indicates a heat source boundary condition Q_h [W/m^2], while the other five patches are treated as temperature boundary condition T_b [K]. All the remaining boundary is insulated. The physics are governed by Eq. (1). To solve the patched cube problem of Eq. (7) we use the deterministic parameters of Table (1). For the varying thermal conductivity, we use

$$\overline{\boldsymbol{C}} = \begin{bmatrix} 11.24 & 5.18 & 1.73 \\ 5.18 & 3.49 & -0.356 \\ 1.73 & -0.356 & 1.78 \end{bmatrix} \text{ where } \overline{\boldsymbol{\Lambda}} = \begin{bmatrix} 14 \\ 0.11 \\ 2.4 \end{bmatrix} \text{ and } \overline{\boldsymbol{Q}} = \begin{bmatrix} 0.892 & -0.416 & 0.174 \\ 0.436 & 0.700 & 0.565 \\ 0.113 & 0.580 & 0.807 \end{bmatrix} \tag{11}$$

as mean values. The variability in the tensors is defined by the standard deviation vector for the three normal distributions, $\sigma_i = [0.8, 0.2, 0.27]$, and the concentration parameter $\kappa = 200$ for the vMF distribution in Eq. (4, 6). These input parameter were chosen with an initial scale difference between the eigenvalues, such is commonly seen with thermoplastic composites [17]. The rotation represent an Euler rotation (XYZ) of 35, 10 and 25 degrees respectively, creating a full rotation matrix. Three datasets with 50000 input-output pairs are generated, first one with random eigenvalues, then one with randomly rotated eigenvectors and lastly with both random eigenvalues and eigenvectors referred to as scale, orientation and scale-orientation random datasets respectively. The steady state patched cube solutions are solved by COMSOL Multiphysics® [18]. The temperature field is discretized with an 11 by 11 by 11 grid over the cube. For each grid a mean and standard deviation of the temperature field is computed using the 50000 samples as a reference value to analyze neural network performance.

Table 2: Training result of scale random dataset.

Layer	Best Val. Loss	Mean Error SD	SD Error SD	Mean Val. Loss	SD Val. Loss
Vectorize	0.048414	0.06627	0.03153	0.720689	0.422424
LogEig	0.000301	<u>0.00431</u>	0.00259	0.001066	0.000992
StrOr	<u>0.000148</u>	0.00502	0.00271	<u>0.000512</u>	<u>0.000387</u>
StrAng	0.000164	0.00801	<u>0.00151</u>	0.000930	0.001204

Table 3: Training result of orientation random dataset.

Layer	Best Val. Loss	Mean Error SD	SD Error SD	Mean Val. Loss	SD Val. Loss
Vectorize	0.000065	0.0081	0.0087	0.000179	0.000194
LogEig	0.000027	0.0038	0.0040	0.000082	0.000010
StrOr	0.000039	0.0056	0.0077	0.000137	0.000117
StrAng	0.000061	0.0072	0.0081	0.000172	0.000122

Table 4: Training result of scale and orientation random dataset.

Layer	Best Val. Loss	Mean Error SD	SD Error SD	Mean Val. Loss	SD Val. Loss
Vectorize	0.03498	0.0291	0.0266	0.0656	0.0211
LogEig	0.00068	0.0090	0.0075	0.0021	0.0012
StrOr	0.00152	0.0118	0.0157	0.0052	0.0038
StrAng	0.00410	0.0091	0.0080	0.0138	0.0063

The neural networks are set up as shown in Fig. (1) with as transformation layer baseline the Vectorize operation and the three transformation layers LogEig, StrOr and StrAng given in Eq. (8) - 10. The hyperparameters are: 1500 epochs, a batch size of 20, two hidden layers of 1700 and 1500 neurons, a RELU activation function, the mean square error loss, the ADAM optimizer, 250 samples, a learning rate of 1×10^{-4} and 5 experiments with different initial samples. The analyzed metrics include the best validation loss with the corresponding estimated mean and standard deviation of the error of the standard deviation over all the grid points. Additionally, the average and standard deviation of the validation loss over the 5 repeat experiments is provided. The results are visible in Tables 2, 3 and 4.

Looking at the Table 2 it is evident that the logarithmic mapping of the eigenvalues is particularly crucial for the proper representation of the tensor scaling. Interestingly, the StrOr layer shows good performance in terms of validation loss, but it does not directly translate to better uncertainty propagation. For the orientation random dataset, in Table 3 we see that the logarithmic mapping of the eigenvectors seems to matter less. Here most networks performed similarly with a small edge given to the LogEig and StrOR layers. It is worth noting that in this case, the eigenvalue input nodes were disabled which, contrary to the scale results, improved performance. Lastly, Table 4 again demonstrates that the lack of a logarithmic map for the eigenvalues heavily affects the performance of the Vectorize layer. The best performing layers in terms of uncertainty propagation are the LogEig and StrAng transformations.

Overall, the LogEig layer performs best. A possible reason could be that the entries originate from a unified space, that is $\mathrm{Sym}(d)$. A disadvantage of this layer is its lack of separation between strength and orientation. The difference between the StrOr and StrAng layers can be explained by in processing of the eigenvector, as the unit vectors appear to be better interpreted than the angular information, resulting in better approximation. It is important to note that since the best validation loss did not always result in the best uncertainty propagation performance, it is difficult to decide which network is the best to use in practice.

5 CONCLUSIONS

In this paper, several neural network transformation layers were introduced to map from a SPD manifold to a vector space. The layers approximation performance is compared with data from a steady state heat conduction problem with anisotropic random conductivity. The neural network layers proposed in this study have demonstrated consistent and significant improvements over the standard approach of vectorizing second-order tensors. In particular, the LogEig layer Eq. (8) and StrOr layer Eq. (9)) have shown notable enhancements in terms of validation loss reduction and uncertainty propagation performance. The logarithmic mapping of eigenvalues appears to be a key factor in this enhanced performance.

While our study focused on a simple, single tensor input problem, future research should explore these layers in more complex, high-dimensional scenarios which could further highlight advantages or disadvantages for each layer. Also, the current implementation, which prioritizes simplicity, could benefit from further hyperparameter tuning and advanced techniques such as batch normalization to address the observed performance variability.

REFERENCES

[1] D. L. Corwin and E. Scudiero, "Field-scale apparent soil electrical conductivity," *Soil Science Society of America Journal*, vol. 84, no. 5, pp. 1405–1441, 2020. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/saj2.20153.

[2] H. Ahmadi Moghaddam and P. Mertiny, "Stochastic Finite Element Analysis Framework for Modelling Mechanical Properties of Particulate Modified Polymer Composites," *Materials*, vol. 12, p. 2777, Jan. 2019. Number: 17 Publisher: Multidisciplinary Digital Publishing Institute.

[3] G. B. Saturnino, A. Thielscher, K. H. Madsen, T. R. Knösche, and K. Weise, "A principled approach to conductivity uncertainty analysis in electric field calculations," *NeuroImage*, vol. 188, pp. 821–834, Mar. 2019.

[4] S. K. Shivanand, B. Rosić, and H. G. Matthies, "Stochastic modelling of symmetric positive definite material tensors," *Journal of Computational Physics*, vol. 505, p. 112883, May 2024.

[5] B. Sudret, S. Marelli, and J. Wiart, "Surrogate models for uncertainty quantification: An overview," in *2017 11th European Conference on Antennas and Propagation (EUCAP)*, pp. 793–797, Mar. 2017.

[6] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, Dec. 2021.

[7] R. K. Tripathy and I. Bilionis, "Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification," *Journal of Computational Physics*, vol. 375, pp. 565–588, Dec. 2018.

[8] M. Lutter, C. Ritter, and J. Peters, "Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning," July 2019. arXiv:1907.04490 [cs, eess, stat].

[9] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019.

[10] K. Xu, D. Z. Huang, and E. Darve, “Learning constitutive relations using symmetric positive definite neural networks,” *Journal of Computational Physics*, vol. 428, p. 110072, Mar. 2021.

[11] K. Linka, M. Hillgärtner, K. P. Abdolazizi, R. C. Aydin, M. Itskov, and C. J. Cyron, “Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning,” *Journal of Computational Physics*, vol. 429, p. 110010, Mar. 2021.

[12] K. Garanger, J. Kraus, and J. J. Rimoli, “Symmetry-enforcing neural networks with applications to constitutive modeling,” Dec. 2023. arXiv:2312.13511 [cond-mat].

[13] Z. Huang and L. Van Gool, “A Riemannian Network for SPD Matrix Learning,” Dec. 2016. arXiv:1608.04233 [cs].

[14] D. Brooks, O. Schwander, F. Barbaresco, J.-Y. Schneider, and M. Cord, “Riemannian batch normalization for SPD neural networks,” Sept. 2019. arXiv:1909.02414 [cs, stat].

[15] W. J. Schuttert, M. I. Abdul Rasheed, and B. Rosić, “Modeling variability of the electrical conductivity tensor for the induction welding of composites.,” *Proceedings in Applied Mathematics & Mechanics*, 2024. [In press}.

[16] K. V. Mardia and P. E. Jupp, *Directional Statistics.* Chichester: John Wiley & Sons, Ltd, 2000.

[17] P. Gouin O’Shaughnessey, M. Dubé, and I. Fernandez Villegas, “Modeling and experimental investigation of induction welding of thermoplastic composites and comparison with other welding processes,” *Journal of Composite Materials*, vol. 50, pp. 2895–2910, Sept. 2016.

[18] *COMSOL Multiphsyics® v. 6.2.*