The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5 – 9 June 2022, Oslo, Norway

A METHOD FOR COMMUNICATION BETWEEN USER MATERIALS DURING RUNTIME IN ABAQUS® ECCOMAS CONGRESS 2022

PHILIP F. ROSE[1,2], LUKAS MÜNCH[3], MARKUS LINKE[1] AND PETER MIDDENDORF[3]

[1] Hamburg University of Applied Sciences, Department of Automotive & Aeronautical Engineering,
Berliner Tor 9, 20099 Hamburg
email: Philip.Rose@HAW-Hamburg.de, https://www.haw-hamburg.de

[2] Universitat Politècnica de València
Camino de Vera, s/n. 46022 - Valencia, http://www.upv.es/es

[3] University of Stuttgart
Paffenwaldring 31, 70569 Stuttgart
email: lmuench@IFB.Uni-Stuttgart.de, https://www.ifb.uni-stuttgart.de

Key words: User-Subroutine, Communication during run time, Finite Element (FE), Composites, Abaqus®.

Abstract.

The application of the Finite Element Method (FEM) has developed considerably in recent decades. While in the early days of the FEM only linear-elastic material models were available in commercial Finite Element (FE) programs, today non-linear and also damage-considering material models are offered. But even these are often not capable of correctly representing the complex material behaviour of modern composite materials. Therefore, FE programs often provide the users with the option of integrating their own material models into the simulation. These programs, often called "user subroutines" or "user materials", represent an intensively researched area in the simulation of material behaviour, which is reflected in the large number of publications on such developments (exemplarily see [1-14]). These material models often require a large amount of data values if they are introduced within the simulation model by finite elements [2]. Especially in the field of composites there is a need to use a separate "user material" for each constituent material. Since there is a mutual interaction between the components in real composites, it is obvious that this interaction must also be represented

between the material models within a simulation in order to make accurate predictions about the material behaviour. To enable such an interaction between different material models in a simulation, a communication during runtime is required in which additionally needed data from the surrounding elements is exchanged between the user materials. In this paper, a method is presented to enable such information exchange during simulation runtime in the commercial FE software ABAQUS/CAE 2019 (Dassault Systèmes, Vélizy-Villacoublay, France) using external databases as well as structured global arrays and some built-in functions of the software. This allows the simultaneous application of several advanced user material models within one simulation and enable them to communicate during runtime, resulting in the possibility of making high accurate simulations of composite materials.

1 INTRODUCTION

The application of the finite element method (FEM) is becoming more and more widespread. Nevertheless, the material models in commercial Finite Element (FE) software programmes still have potential for optimisation. Therefore, many scientists are dealing with the development of their own material models (user material) in order to further expand the application spectrum of the FEM. One aspect of this is the cost-efficient analysis of high-cycle fatigue tests (HCF), for which the "cycle jump" is a suitable method [1-5, 10]. This correlates the simulation time with a virtual number of cycles and thus enables the fatigue behaviour to be simulated without having to model a real loading and unloading over millions of cycles. When designing carbon fibre reinforced plastics (CFRP), the complexity of the simulation requirements increases due to the very complex damage behaviour of these materials. Thus, a high number of developments can be found in the literature that either want to better predict the delamination behaviour with the help of user material [1, 4, 5] or address the damage behaviour of a single fibre layer [6-9]. A simultaneous application of several user materials is already possible as long as the user materials do not communicate [16]. Since methods such as the crack tip degradation approach [10] or the fibre-oriented continuum damage mechanics [11] rely on additional information such as the position of the crack front, the application of such material models requires communication between the elements within one material model. [15] shows in his work that delamination can be induced by matrix cracks. Therefore, if matrix crack-induced delamination are to be simulated, an exchange of information must also be possible between the material models involved in order to be able to numerically represent the emergence of a new possible delamination path. Therefore, a communication method developed for the application in the FE software Abaqus® (Dassault Systèmes, Vélizy-Villacoublay, France) is presented in the following, which enables such an interaction between several user materials. The information exchange is based on global arrays, which are supplied with the information about the neighbourhood relations of the individual elements through a pre-simulation analysis of the simulation model. Since simulations are nowadays typically calculated with several central process units (CPUs) and it is known that this does not result in a linear reduction of the calculation time [17], since the splitting and merging of the processes causes additional numerical effort, the influence of several CPUs on the calculation time when using the communication method is also investigated.

2 GENERAL SPECIFICATIONS

Many scientists develop their own material models with the aim of mathematically describing the complex material behaviour of modern composite materials [1-11], and efficient approaches to fatigue simulation are of particular importance. The so-called energy-based cycle jump is a method that enables the simulation of very high fatigue cycles in the area of high cycle fatigue (HCF) in an economically justifiable time [1, 2, 10]. However, in order to provide correct calculation results, this approach requires the availability of historical information for an element to be calculated. In addition, the crack tip degradation approach postulates that the fatigue-induced damage and associated degradation of the material properties only takes place in the area of the crack tip [2, 10], therefore such material models require knowledge about which elements are part of a crack front, so that only these elements experience degradation through the fatigue algorithm. In the simulation of CFRP, this approach is mainly used in the calculation of delamination growth [1, 2, 10]. Similarly, material models for the fibre layers are being developed to represent the different forms of damage in CFRP [6 - 9]. The numerical calculation of matrix cracks often shows a strong dependence on the discretisation [12-14], meaning the structure and orientation of the mesh. S. Hallett and S. Mukhopadhyay [11] have developed a method that overcomes this dependency by knowing which elements are the neighbours of a considered element, numerically controlling the propagation direction of a crack and allowing it to propagate only along the fibre direction as observed in the real world. However, these methods require a preparation of the simulation model in order to be able to provide the required information during the simulation runtime. In addition, real CFRP components exhibit the phenomenon often referred to as "crack jump" [118], meaning the matrix crack-induced plane change of a delamination, highlighted in Figure 1.

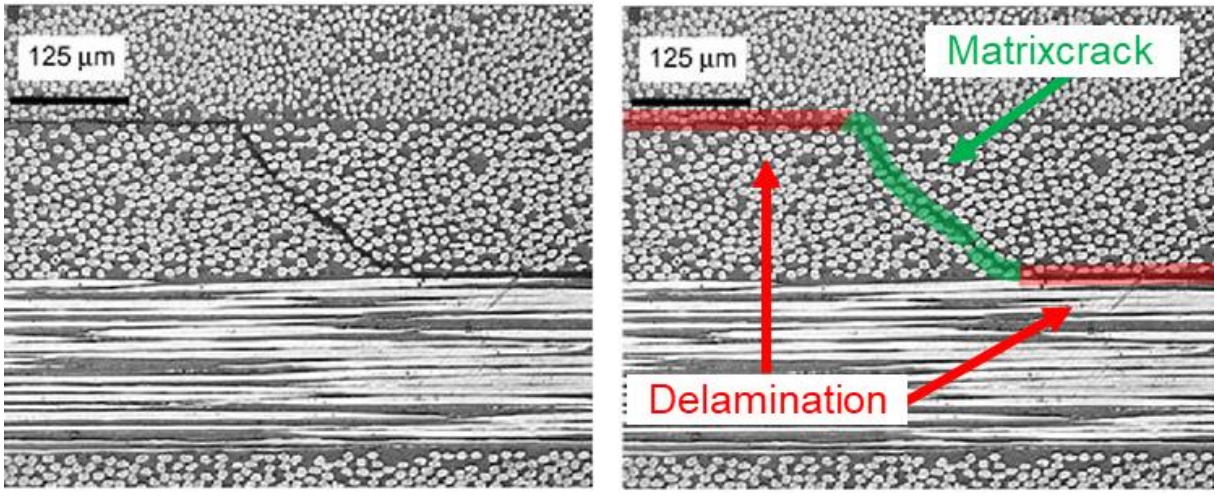

Figure 1: Crack jump [18]

Furthermore, [15] shows that matrix cracks can be considered as an initiator for delamination, especially under fatigue loads. Therefore, a simultaneous application of the previously described user materials requires communication between the material models, so that a matrix

crack calculated by the material model of the fibre layer informs the user material for the delamination that the corresponding element is now part of a new crack front and thus the crack tip degradation approach is now to be activated for this element.

3 COMMUNICATION METHOD

With the help of a "user material", the achievable accuracy in the prediction of fatigue behaviour can be increased, but above all, the HCF regime can be analysed with acceptable calculation times using approaches such as the "cycle jump". Since composites have many different damage forms due to their layered structure and the combination of different materials, which also interact with each other, the prediction requires several "user materials" so that a suitable mathematical material model is available for the specific damage form. In addition, a method is needed that can numerically represent the interaction of the individual forms of damage, which is made possible in the context of this publication with the use of communication between the individual "user materials" during the simulation runtime. In the following, an overview of the communication method is given. This is followed by a more detailed description of the necessary model preparation. Subsequently, the implementation of the communication method in the FE software Abaqus® is explained.

3.1 Overview

The simulation in the FE software Abaqus® typically starts with the creation of a so-called input file (.INP) which contains all information of the simulation model for the solver. For example, the geometry of the structure to be simulated as well as the number, type and position of the elements. This file is analysed before the simulation is started. This process is explained in more detail in the following chapter 3.2. After the analysis and the resulting database, the information necessary for the "user material" described in chapter 2 can be provided during the simulation when it is started. By knowing the neighbourhood relationships of the elements even across the boundaries of a "part", it is possible to exchange information between two different "user materials" and thereby trigger specific functions of the related "user material". As shown in Figure 2, the communication method requires a preparation of the simulation model, which results in an external database. This database is also required without communication for the material models crack tip degradation approach [2, 10] and fibre-oriented continuum damage mechanics [11], so that this additional effort cannot be attributed to the communication method alone. In addition, the user subroutine is shown, which on the one hand contains the Abaqus® subroutine "VEXTERNALDB", which makes it possible to execute commands at specific points in time, such as the start of an increment [16], and on the other hand also the subroutine "VUMAT", with which Abaqus® makes it possible for users to integrate their own material models into an explicit simulation. The VUMAT (Main) is only responsible for calling the correct VUMAT based on the material assigned to the element. The other two VUMAT subroutines in Figure 2 then represent the individual material models for the fibre layer and delamination.

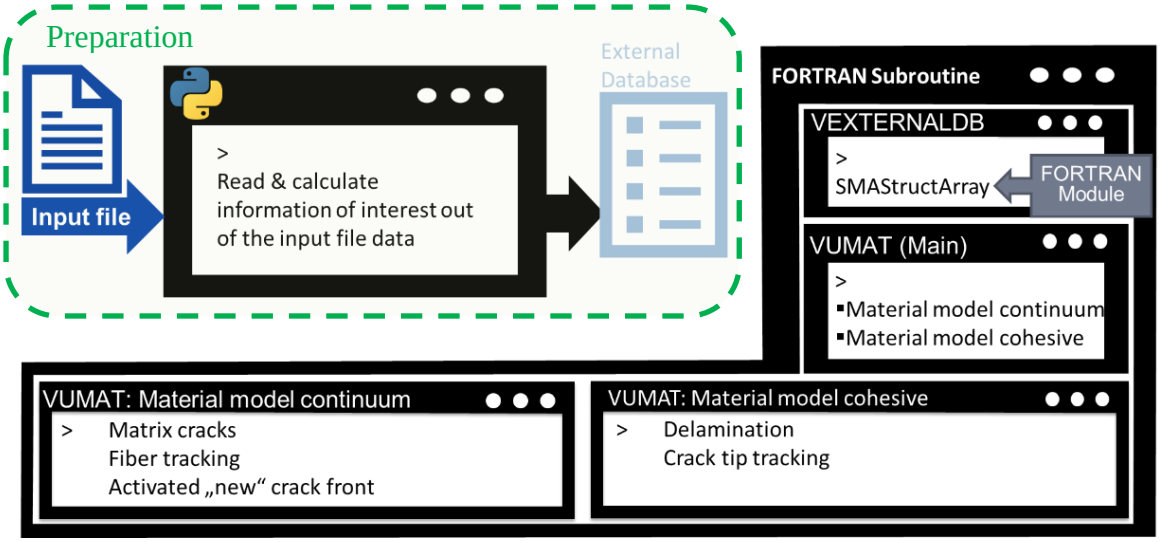

Figure 2: Overview of the communication methode

3.2 Preparation

For specific functions of the "user materials", information of the surrounding elements is required in addition to integration point (IP) information such as the historically highest load of an integration point (see chapter:2). This information must be known before the start of the simulation calculation and must be available for the "user materials" during the simulation. For this purpose, the .INP file is first analysed using a Python script and a row is created in an external database for each IP. In the row, information such as the associated element, the coordinates of the integration point and the neighbouring elements of the IP are stored. This shows a very strong dependency on the time required by the number of elements or integration points, which is discussed in more detail in chapter 5.2.

3.3 Communication process

The communication process can be divided into 4 phases, which are described in the following.

1. start of the simulation calculation
2. first CALL of a "user material"
3. event that triggers the communication
4. the recognition of a communicated information

1. Start of the simulation
When the simulation is started, a global array is created using the SMAStructArray function. This allows access to the contained data from each participating subroutine. The global array has a separate row for each IP and by using a Fortran module, the structure of the global array is defined so that each column can store different types of data. The global array is the core of

the communication because the actual exchange of information takes place via it. As can be seen in Table 1, in the simplest form the following information is stored in the global array. The IP-ID is, as the name suggests, an identification number of an integration point. In the next column, the corresponding coordinates in the global coordinate system X, Y, Z as well as the elements that are affected by a possible communication are stored.

Table 1: Structure of the global array

Global Array	Integration Point ID	Coordinates of IP (x,y,z)			NTA ID	NTB ID	Information
IP 1							
⋮							
IP n-1							
IP n							

In the example shown here, these elements are the adjacent elements in the thickness direction of the simulated laminate. NTA stands for the neighbouring element in positive thickness direction (neighbour in thickness above) and NTB (neighbour in thickness below) for the one in negative thickness direction (Z). The last column is used for the actual information to be exchanged between the "user materials".

2. First CALL of a User-Material (VUMAT)
When the respective VUMAT for matrix crack or delamination is called for the very first time, a coordinate comparison is first carried out, whereby the material point coordinates (Coords MP) of the current IP provided by Abaqus® are searched in the global array. If a match is found, all information of this row from the global array is stored in local Solution Depended Variables (SDV). This eliminates the need to constantly open the external database in the further course of the simulation calculation, which significantly reduces the time required for the simulation. In addition, the row numbers of the NTA and NTB elements are also stored locally as SDVs, which avoids searching for the neighbour elements row in the global array in the event of an information exchange which also saves time.

3. Event that trigger the communication
In the case that a current calculation result of an integration point requires a communication to the neighbouring element, the respective row in the global array is first called up by the row number of the NTA and NTB elements and, according to the calculation result of the current IP, information is entered in the corresponding column "Information" in the rows of the NTA and NTB elements. In this way, it is achieved that an IP with its associated "user material" sends information to another IP with its associated "user material". How this is then received is explained in the fourth and final step.

4. The recognition of a communicated information
After the previously calculated integration point has stored the information to be communicated in the global array in the row of the information receiver, as soon as the information receiver has become the currently calculated IP, it is checked whether a change has taken place in the global array in the row of the now current integration point, if this is the case as in point 3 "event that trigger the communication", this change is recognised and stored in the associated SDV, which finalises the information exchange.

4 DEMONSTRATION EXAMPLE

In order to illustrate how the communication method mentioned above works, a demonstration example that illustrates the exchange of information in the simplest way possible is presented. For this purpose, the phenomenon of crack jumps as shown in Figure 1 is used. The model has three parts (see Figure 3), the upper and lower part is modelled with cohesive zone elements (COH3D8) and assigned to the material model of the delamination. The part in between is modelled with solid elements (C3D8R) and represents a fibre layer, in which the middle part is modelled with a "predetermined breaking point" in order to simulate a targeted formation of a matrix crack in this area of the fibre layer. Therefore, a displacement is applied to the right-hand side of the model via a reference point connected to the right-hand end face, and the fixed clamping of the elements on the left-hand side results in a tensile load acting in the X direction.

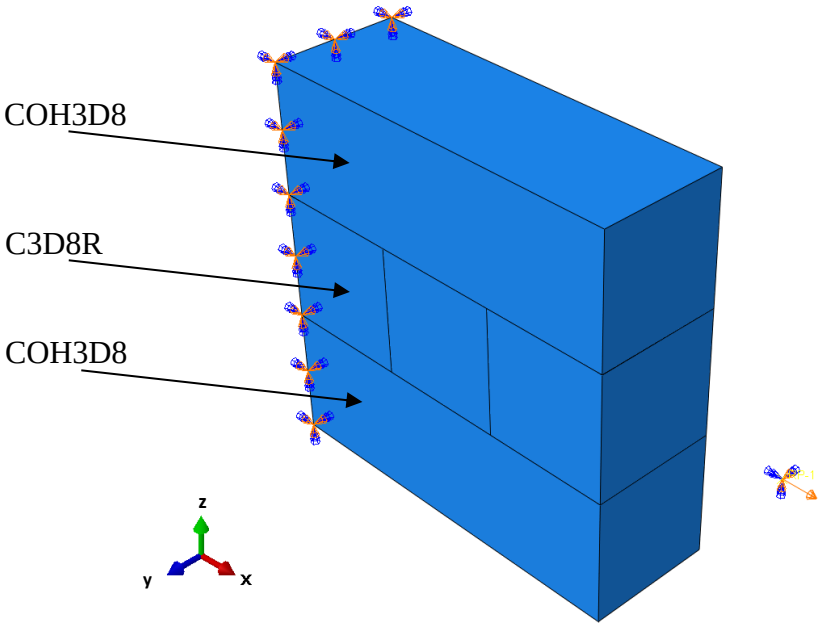

Figure 3: Demonstration model

5 RESULTS AND DISCUSSION

In order to illustrate the communication between two user materials during the simulation runtime, chapter 5.1 shows the example model from chapter 4 before at the formation of a matrix crack. In addition, chapter 5.2 shows the time consumption for the analysis of the simulation model by the Python script described in chapter 3.2. Furthermore, the influence of the use of several CPUs on the total time consumption of a simulation when using the communication method is presented.

5.1 Communication

The example model shown in chapter 4 is illustrated in the following figure 4 at two different points in time. On the left is time 1 at which no matrix crack has yet occurred and consequently no information has yet been exchanged between the two individual material models, i.e. the fibre layer damage and the delamination. On the right is time 2 at which a matrix crack has occurred in the middle element, which is represented by SDV 1. Below this, SDV 10 is shown and it can be seen that the two elements that are adjacent to the element with the matrix crack have received information by SDV 10. In the VUMAT code for delamination, the crack tip degradation approach can now be activated by the change of SDV 10 and so the crack jump phenomenon can be simulated in a numerically controlled way.

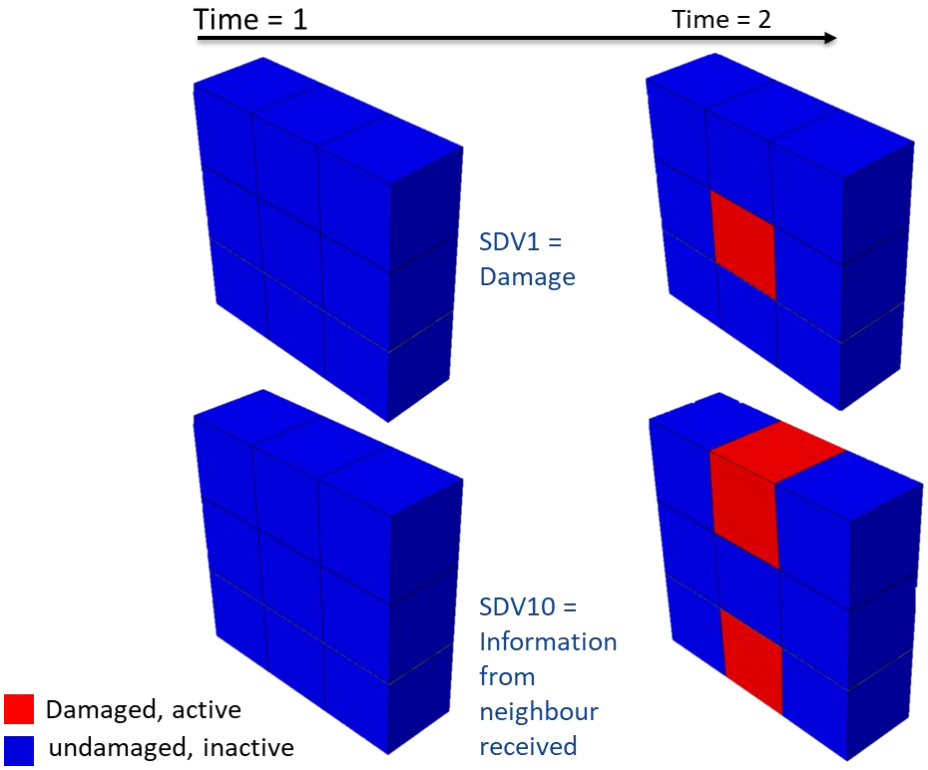

Figure 4: Communication in Abaqus®

5.2 Time consumption

As already mentioned, the .INP file of the simulation model must be analysed before the simulation can be performed (see chapter 3.2). This process is carried out by a script written in Python and determines, among other things, the neighbourhood relations of the integration points or elements, so it is obvious that this process has a dependency on the integration points in the model. Figure 5 clearly shows the strongly exponential growth of the time consumption for preparation by the logarithmic scaling. While the time consumption for 1,000 elements is still far below one minute, it is already almost 7 minutes for 10,000 elements with a single central processing unit (CPU) with a frequency of 2.3 GHz. For this reason, the preparation script was extended for the use of the Massage Passing Interface (MPI) method so that any number of CPUs can carry out the analysis of the input file in parallel.

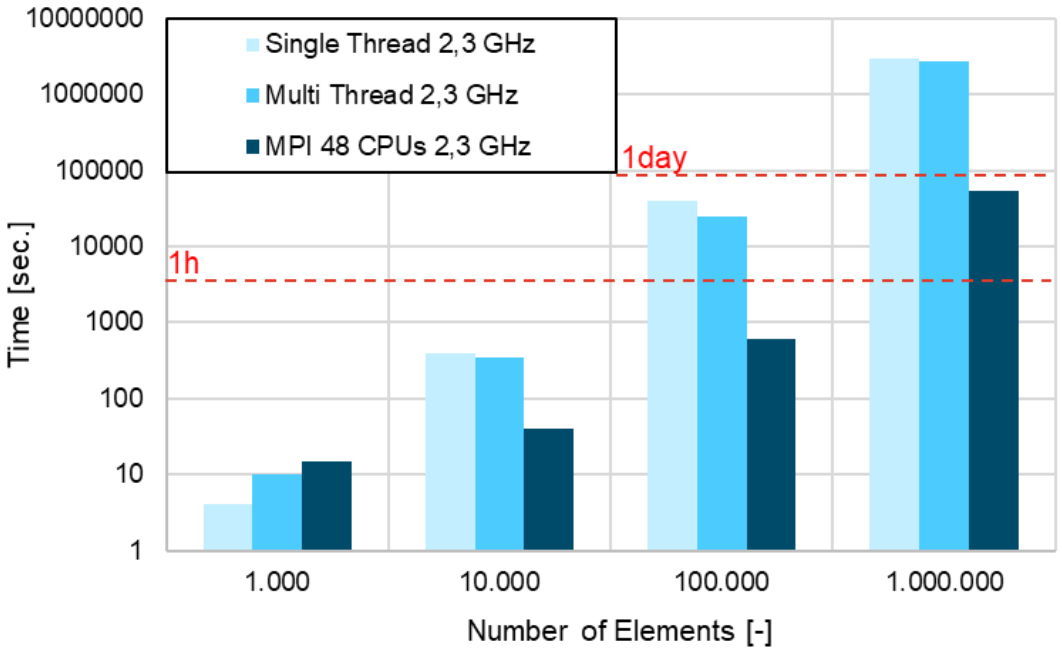

Figure 5: time consumption of the preparation

In this way, for a model with 100,000 elements and 220,000 integration points, the analysis time can be reduced from 10.9 hours with one CPU to about 1.3 hours with 48 CPUs. Since the actual communication takes place during the simulation runtime, this also has an impact on the time consumption for the simulation itself. Since it is common in the application of FEM simulations to use several CPUs for the calculation, the influence of the number of CPUs used on the total calculation time is shown in the following.

It is known that by parallelising computations, the time consumption does not decrease linearly with the number of CPUs [17], since splitting, allocating and then merging the computation results causes an additional numerical effort. Since the communication method presented here uses global arrays and these are created separately for each CPU or domain when using multiple CPUs and there is no automatic synchronisation of the information between the individual

global arrays, this synchronisation of the global arrays must be carried out by the method itself. For this purpose, at the start of each increment, each global array is sent to the root processor, which examines the arrays for changes and, if a change is found, sends back an updated global array to all CPUs. Figure 3 clearly shows that the synchronisation of the CPU-specific global arrays is so time-consuming that there is no further time saving with more than 8 CPUs. It must be mentioned that this was only tested on a model with 10,000 elements and a total of 22,000 integration points; it cannot be ruled out that with larger models, even with more than 8 CPUs, a further reduction in the total calculation time can be achieved.

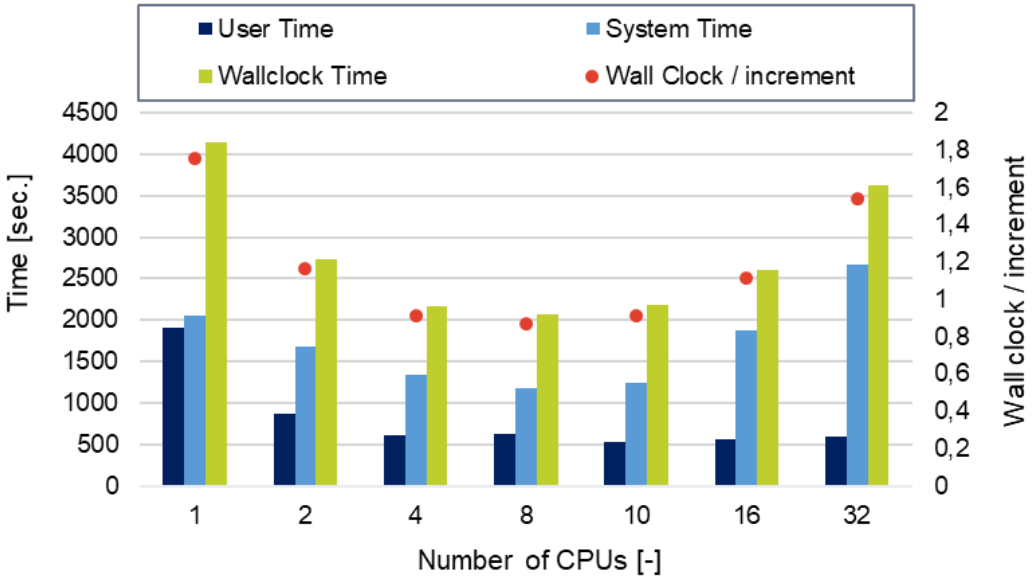

Figure 6: Influence of using several CPUs onto the time consumption of the simulation

6 SUMMARY

The material models available in commercial FE software are often not sufficient to simulate the complex damage behaviour of modern composite materials. Therefore, many scientists develop their own material models, so-called "user materials", with the aim of improving the prediction accuracy of individual damage phenomena [1-14]. Especially in modern approaches for the simulation of high cycle fatigue tests like the cycle-jump [1-5] in combination with the crack tip degradation approach [2, 10], the material model needs information that the FE software itself does not provide during the simulation calculation. This information is typically obtained by analysing the input file and stored in external databases [2, 11]. If matrix crack-induced delamination is to be simulated, an additional exchange of information between the individual material models is required. In order to make this possible, a communication method was developed which enables an exchange of information between several "user materials" during the simulation runtime. The presented communication method shows the intended function and enables an information exchange between elements even if they are assigned to different material models. This allows an interaction of different user materials during the

simulation runtime, which can be applied far beyond the limits of the mentioned example of fibre-reinforced plastics. Like many modern material models [2, 10, 11], the communication method also requires additional information that must be known before the actual simulation calculation. This is done with the help of a Python script which analyses the input file of the simulation and derives the necessary information from it and stores it in an external database. A row is created in the database for each integration point and contains, in addition to an identification number of the integration point, further information such as its global coordinates. To avoid a constant opening and closing of the external database, a global array is created at the beginning of the simulation calculation in which all data of the external database are stored. The element-specific data is then stored in local variables (SDV). This also includes the information about which elements are the neighbours of a considered element, so that in the case of information to be exchanged with the neighbour, its row can be directly addressed in the global array and information can be stored in it, which is registered during the calculation of this element and the material model is further calculated according to the received information.
Since simulations are typically calculated with several CPUs in parallel, the influence of several CPUs on the total calculation time was studied. The use of up to 8 CPUs showed a reduction in the time consumption. When more than 8 CPUs were used, the simulation time increased again. This increase is due to the fact that a separate copy of the global array is created for each CPU and it is necessary to update these arrays with each other at the start of each increment.

REFERENCES

[1] Bak, B.L.V. and Turon, A. and Lindgaard, E. and Lund, A.: A benchmark study of simulation methods for high-cycle fatigue-driven delamination based on cohesive zone models, *Composite Structures* Vol. 164, *pp. 198–206,* Elsevier, (2017)

[2] Sachse, R. and Pickett, A. K. and Essig, W. and Middendorf, P.: Experimental and numerical investigation of the influence of rivetless nut plate joints on fatigue crack growth in adhesively bonded composite joints. *International Journal of Fatigue* Vol. 105, pp. 262–275, (2017)

[3] Van Paepegem, W. and Degrieck, J.: Fatigue degradation modelling of plain wovenglass/epoxy composites, *Composites Part A*, Vol 32, pp. 1433–1441, (2001)

[4] Turon, A. and Costa, J. and Camanho, P. and Dávila, C.: Simulation of delamination in composites under high-cycle fatigue. *Composites Part A: Applied Science and Manufacturing*. Vol. 38, pp. 2270-2282. (2007)

[5] Tao, C. and Yao, W. and Ji, H.: A Novel Method for Fatigue Delamination Simulation in Composite Laminates. *Composites Science and Technology*. Vol. 128, (2016)

[6] Deuschle. H.: 3D failure analysis of UD fibre reinforced composites : Puck's Theory within FEA. Dissertation, Institute for Statics and Dynamics of Aerospace Structures, University of Stuttgart (2010)

[7] Lapczyk, I. and Hurtado, J. A.: Progressive damage modeling in fiber-reinforced materials, Composites Part A: Applied Science and Manufacturing, vol. 38, no. 11, pp. 2333-2341, (2007)
[8] Pham, D. C. and Cui, X. and lua, J.: A discrete crack informed 3D continuum damage model and its application for delamination migration in composite laminates, Composites Part B Engineering May (2019)
[9] Wang, J. and Pineda, E. and Ranatunga, V. and Smeltzer, S.: 3D Progressive damage modeling for laminated composite based on crack band theory and continuum damage mechanics. Conference: American Society for Composites 30th Technical Conference (2015)
[10] Kawashita, L. F. and Hallett, S. R.: A crack tip tracking algorithm for cohesive interface element analysis of fatigue delamination propagation in composite materials, International Journal of Solids and Structures, Jg. 49, Nr. 21, S. 2898–2913, issn: 00207683. doi: 10.1016/j.ijsolstr.2012.03.034. (2012)
[11] Mukhopadhyay, S. and Hallett, S.: A directed continuum damage mechanics method for modelling composite matrix cracks. Composites Science and Technology. Vol. 176. pp. 1-8. (2019)
[12] Bazant, Z. P. and Belytschko, T.: Wave propagation in a strain-softening bar: exact solution, J. Eng. Mech. Vol. 111 pp. 381–389, (1985)
[13] Bazant, Z.P.: Crack band theory for fracture of concrete, Mater. Struct. Vol. 16, pp. 155–177. (1983)
[14] Bazant, Z.P. and Oh, B.H.: Microplane model for progressive fracture of concrete and rock, J. Eng. Mech. Vol. 111. pp. 559–582. (1985)
[15] Müller, A.: Damage characterisation on fibre-plastic composites in the vibration test by means of X-ray fraction topography taking into account the matrix properties, BAM dissertation series, Vol. 162, (2018)
[16] Dassault Systems, SIMULIA User Assistance 2019, Abaqus-User Subroutines-Abaqus/Explicit User Subroutines
[17] Bergan, A. C.: Computational Performance of Progressive Damage Analysis of Composite Laminates using Abaqus/Explicit with 16 to 512 CPU Cores, *NASA Langley Research Center Hampton, VA, United States, (*2019)
[18] Birur A.: Time-dependent damage evolution in multidirectional polymer composite laminates. Master Thesis, Department of Mechanical and Manufacturing Engineering, University of Manitoba, Canada; (2008)