On Analyzing Quality of Data Influences on Performance of Finite Elements Driven Computational Simulations

Michael Reiter[1], Hong-Linh Truong[2], Schahram Dustdar[2],
Dimka Karastoyanova[1], Robert Krause[3], Frank Leymann[1], and Dieter Pahr[4]

[1] Institute of Architecture of Application Systems, Universität Stuttgart
{reiter,karastoyanova,leymann}@iaas.uni-stuttgart.de
[2] Distributed Systems Group, Vienna University of Technology
{truong,dustdar}@infosys.tuwien.ac
[3] Institute of Applied Mechanics (CE), Universität Stuttgart
krause@mechbau.uni-stuttgart.de
[4] Institute of Lightweight Design and Structural Biomechanics
Vienna University of Technology
pahr@ilsb.tuwien.ac.at

Abstract. For multi-scale simulations, the quality of the input data as well as the quality of algorithms and computing environments will strongly impact the intermediate results, the final outcome, and the performance of the simulation. To date, little attention has been paid on understanding the impact of quality of data (QoD) on such multi-scale simulations. In this paper, we present a critical analysis of how QoD influences the results and performance of basic simulation building blocks for multi-scale simulations. We analyze the impact of QoD for Finite Element Method (FEM) based simulation building blocks, and study the dependencies between the QoD of input data and results as well as the performance of the simulation. We devise and implement novel QoD metrics for data intensive, FEM-based simulations and show experiments with real-world applications by demonstrating how QoD metrics can be efficiently used to control and tune the execution of FEM-based simulation at runtime.

1 Introduction

For complex multi-scale simulations, e.g. to investigate structural changes within a human bone after a fracture of the arm, a common approach to perform scientific simulations is to transform the partial differential equations (PDEs) by means of the FEM to a system of linear or nonlinear matrix equations that must be solved. In such multi-scale simulations, FEM algorithms can be used at different scales, such as the skeleton, the bone structure and the bone cell simulations. Therefore, FEM algorithms play an important role in computational science. Because of their importance, understanding quality of these algorithms have attracted several research projects. However, most of them focus on subareas, for instance, optimizing matrix solver [1] or other performance aspects [2].

C. Kaklamanis et al. (Eds.): Euro-Par 2012, LNCS 7484, pp. 793–804, 2012.

FEM algorithms, in general, consume or produce input data (e.g. to describe the simulation object), intermediate results (e.g. results for intermediate time steps), the final result, other output data (e.g. status data), and internal used data (e.g. FEM grid or matrix). These types of data have large volume and some of them have complex data structures.

We argue that the quality of the input data, together with, the quality of algorithms and computing environments will strongly impact the intermediate results, the final simulation output, as well as the performance and storage provisioning of the simulation. This is particular applicable to multi-scale simulations in which the output data in one scale will be used as input data on the other scale. Not to mention that even in a single scale simulation, there are various steps, in which different types of data are processed and produced. Quality of data (QoD) can strongly affect the selection and operation of algorithms as well as computing and storage resources in these steps and in the data exchange among these steps and among scale-specific simulations. While several research papers have discussed possible computing environments and algorithms in detail for FEM, little attention has been paid on understanding the impact of QoD on the performance and resource/storage provisioning in phases of FEM-based simulations. Because FEM-based simulations in computational science are typical long running and produce large amount of data, and are expensive in terms of time and money [3,4]. Detecting poor data quality and able to understanding the impact of QoD in FEM-based simulations could potentially save time and money. In particular, being able to understand QoD influences on performance and resource provisioning for FEM-based simulations can also help to develop better strategies for pay-per-use resources in cloud computing environments.

In this paper, we focus on understanding major QoD metrics for FEM-based simulations and on analyzing the dependencies among QoD metrics to the quality of according intermediate results, performance, storage needs, and the QoD of the simulation as a whole. Fundamentally, we focus our work on general FEM-based simulation steps that can be considered as a basic building block for multi-scale simulations. We have developed several QoD metrics and analyzed trade offs between different QoD findings and simulation execution time. We present our experiments with a real world simulation. To the best of our knowledge, this is the first attempt to analyze the impact of FEM QoD metrics on simulations.

The rest of this paper is organized as follows: Section 2 discuss the background of our work. Section 3 presents and defines QoD metrics. Section 4 presents our prototype and experiments. Section 5 presents related work, followed by the conclusions and future work follows in Section 6.

2 Quality of Data Implications in FEM Based Simulations

2.1 Identifying Important Types of Data in FEM-Based Simulations

Different tools and frameworks exist to execute FEM based simulations. Some of them, e.g. Ansys[1], use pre-implemented functions and have a strong FEM focus

[1] http://www.ansys.com/

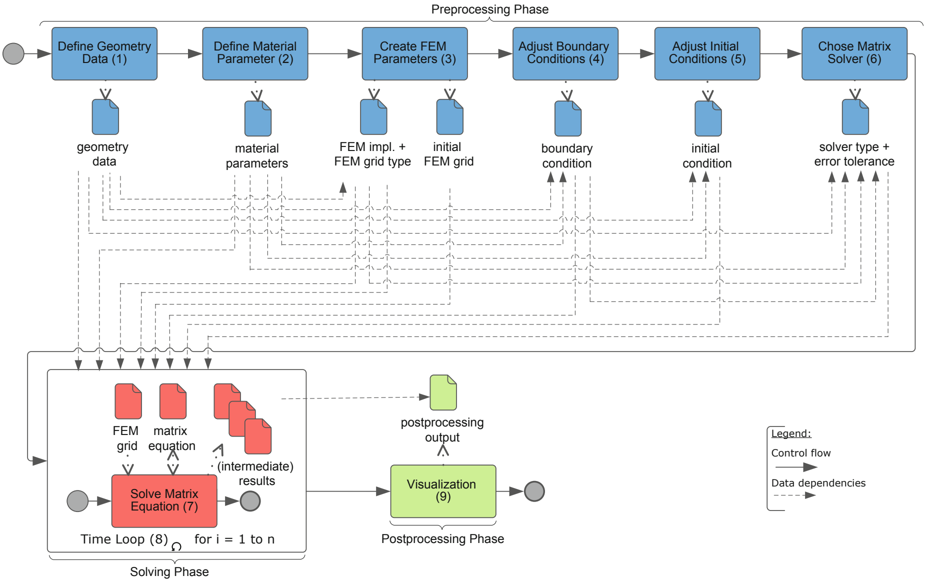

Fig. 1. Steps in a basic FEM-based simulation building block and important types of data and important data dependencies

while others could be part of a global technical computing environment, such as MatLab[2]. Physics specific FEM frameworks, like PANDAS[3], permit to create simulations in a very flexible way.

A FEM-based multi-scale simulation can have millions of basic simulation building blocks, e.g. a human skeleton has approximately 206 bones, each bone has thousands to millions of cells, where each cell can be simulated by a FEM-based simulation building block. Nevertheless a FEM-based simulation building block has a common procedure that can be divided into three phases: (i) *preprocessing*, (ii) *solving*, and (iii) *postprocessing*. The three phases can be divided again into different steps. Figure 1 illustrates the main steps for transient problems, in which the preprocessing phase includes steps 1 to 6, steps 7 and 8 belongs to the solving phase, and step 9 is in the postprocessing phase. In the preprocessing phase all relevant input data are collected, based on that an initial FEM grid and matrix equation can be generated. Matrix equations will be adapted and solved in the solving phase. The results are processed (e.g. visualized) in the postprocessing phase. In this paper, we focus on QoD metrics for the FEM-based simulation building block, in particular for the preprocessing and the solving phases.

From our study, we determine five relevant *classes of data* in the preprocessing and solving phases: (i) *PDE driven input data* (input data), (ii) *FEM driven*

[2] http://www.mathworks.com/products/matlab/

[3] http://www.mechbau.uni-stuttgart.de/pandas/index.php

input data (input data), (iii) *internal/temporary FEM data* (output/input data), (iv) *intermediate results* (output data), and (v) *final result* (output data). PDE and FEM driven input data are generated during the preprocessing phase. Most of the internal and temporary FEM data as well as intermediate and final results are computed at the solving phase. Generally, the QoD in these classes is critical for the quality of the simulation result, the performance, and the amount of data. Several types of data that can influence the QoD of FEM based simulations, such as the *geometry, material parameter, FEM interpolation, FEM grid type, boundary condition, initial condition, matrix solver type, matrix error tolerance, matrix equation, FEM grid,* and *time step*[4].

2.2 Defining and Evaluating Quality of Data Metrics

In general, there can be several QoD metrics. However, [6] pointed out that the term QoD is used in different domains with several meanings and must be regarded specific to the application domain. QoD can be measured with the aid of metrics [6]. A metric must be provided by one or more well defined measurement method. Furthermore, several meta information are necessary: (i) where the measurement is taken, (ii) what data are included, (iii) the measurement device, and (iv) the scale on which results are reported. To evaluate QoD, we distinguish between QoD verification and validation with respect to the FEM phases. Related to the IEEE-STD-610 definition of validation and verification in software products [7] QoD verification and validation is defined as follow:

Definition 1 (Quality Of Data Verification). *The process of evaluating quality of data during or at the end of the preprocessing phase to determine whether it satisfies specified quality of data demands.*

Definition 2 (Quality Of Data Validation). *The process of evaluating quality of data during or at the end of the solving phase to determine whether it satisfies specified quality of data demands.*

The QoD Verification definitions means that verification makes sure that the simulation (i) fulfills the specification of the simulation object, (ii) is derived from the specification of the simulation object that the simulation fulfills the specification of the PDE, and (iii) is derived from the specification of the PDE that the simulation fulfills the specification of the FEM; or for short: computes the problem right. The QoD Validation definitions means that validation make sure that the simulation actually fulfills the simulation intention; or for short: computes the right problem.

2.3 Identifying Factors Influencing Quality of Data

Each step has different influences to simulation characteristics like QoD, data quantity, or performance. In Figure 1, in step 1, the dimension (e.g. 3 D) or

[4] Due to the lack of space, we provide a supplement report to document important types of data and their possible influences as well as possible FEM-based specific QoD metrics in [5].

complexity (e.g. smoothness) of the simulation object – described by *geometry data* – has strong influences to the size of the result data and to the performance. The *material parameters* in step 2 approximate the material behavior within the PDE based on a given physics. Changes to material parameter influences the behavior of the PDE. In step 3, data that describe the *FEM grid type*, the *FEM implementation*, and an *initial FEM grid* are involved. Hence, interpolation functions (e.g. power series) affect the performance and the result quality. The size, complexity, and type of the FEM grid strongly influence the performance, the data quantity, and the QoD. *Boundary condition* and *initial condition* in step 4 and 5 could be defined with several accuracies and numerical characteristics which affect the QoD and the performance. A matrix *solver type* and, in some cases, an *error tolerance* must be chosen in step 6. Each solver has different characteristics in relation to error behavior, QoD, and performance.

Based on a *FEM grid* a *matrix equation* (step 7) could be solved with different precisions which influence the performance and the result quality. Additional internal or temporary data, such as the Jacobian determinant, influence the quality of the *(intermediate) result.* In step 8 the increment of *time step* δt_i influences the accuracy, the performance, and the data quantity.

Furthermore, there are relevant constrains between the steps regarding to QoD. Figure 1 depict essential data dependencies that need to be observed. Considering the important role of quality of data, we present novel QoD metrics for FEM based simulations and discuss the influences.

3 QoD Metrics for FEM-Based Simulations

QoD metrics are determined based on our evaluation important types of data that influence FEM based simulations detailed in [5]. We distinguish between *basic* and *constrained* input data as characteristics of types of data. As shown in Figure 1, basic input data have no strong dependencies to other input data and are typically data to describe the PDE. Constrained input data are essentially dependent on basic input data and can have relations to other constrained input (or combined output/input) data.

We consider QoD as a tuple of characteristics and goodness [8]. A characteristics of data will be analyzed without any simulation context, while the goodness of the characteristics of data will be evaluated with respect to the specific simulation context. To simplify the wording, we use the term QoD metric result in the following as a synonym for the goodness value of the QoD tuple. Regarding to [6], a QoD metric result will have a value calculated by a well-defined quality objective. In this paper, we use for the quality objective an interval from 0 to 1. Hence, we defined the QoD metric result for FEM-based simulations as follows:

Definition 3 (Quality of Data / Goodness). *The Quality of Simulation Data / Goodness (or QoD metric result) is an objective represented by a value* $Q \in [0,1]$ *that determines the quality of input and output data of a FEM based simulation with both limits* $0 =$ *bad and* $1 =$ *good.* $Q \in (0,1)$ *determines a quality between bad and good.*

Table 1. Selected Quality of Data metrics for FEM-based simulations

QoD Metrics	**Description**
Geometry Accuracy	Metric related to Figure 1 step 1 to verify geometry data of the PDE
Material Parameter Accuracy	Metric related to Figure 1 step 2 to verify material parameters of the PDE
Interpolation Accuracy	Metric related to Figure 1 step 3 to verify interpolation functions of FEM implementation data
FEM Grid Type Adequacy	Metric related to Figure 1 step 3 to verify the FEM grid type
FEM Grid Accuracy	Metric related to Figure 1 step 3 to verify the fineness of the (initial) FEM grid
Boundary Condition Accuracy	Metric related to Figure 1 step 4 to verify the accuracy of the boundary condition of the PDE
Initial Condition Accuracy	Metric related to Figure 1 step 5 to verify the accuracy of the initial condition of the PDE
Matrix Solver Accuracy	Metric related to Figure 1 step 6 to verify the accurate selection of a matrix solver type for the FEM
Matrix Solver Error	Metric related to Figure 1 step 6 to verify the maximal error tolerance for a matrix solver type for the FEM
FEM Element Condition	Metric related to Figure 1 step 7 to validate the condition of an element within a FEM grid
Matrix Equation Condition	Metric related to Figure 1 step 7 to validate the numerical condition of a matrix within a matrix equation
Vector Condition	Metric related to Figure 1 step 7 to validate the numerical condition of a vector within a matrix equation
Time Discretization Accuracy	Metric related to Figure 1 step 8 to validate the accuracy of a time step

Based on this definition a value $Q_{QoDmetric}$ determines the QoD metric result with respect to the specific *QoDmetric*. Corresponding to the influencing factors, Table 1 presents a list of QoD metrics we have developed and their associated data as well as data types described in Section 2 (see our supplement report [5] for detailed explanation).

We introduce in detail three implementations of QoD metrics: Material Parameter Accuracy (Q_{MPA}), Matrix Solver Accuracy (Q_{MSA}), and Vector Condition (Q_{VC}). Q_{MPA} (Figure 1 step 2) and Q_{MSA} (step 6) verify the quality in the preprocessing phase. Q_{MPA} is based on the quality of PDE driven basic input data and will be used if the implemented parameter approximates a known parameter. Q_{MSA} is premised on the quality of FEM driven constrained input data and makes a statement about the expected accuracy of the numerical solution before the solving phase starts. With this knowledge a time and money consuming simulation can be adopted or aborted if a poor QoD is expected. Q_{VC} (step 7) of both vectors $\mathbf{b}$ and $\mathbf{x}$ concerns the solving phase and determines the quality of constrained internal/temporary FEM data within a matrix equation $\mathbf{Ax} = \mathbf{b}$. In contrast to Q_{MSA}, Q_{VC} validates and makes a statement about the real condition of the numerical solution at the solving phase. Nevertheless, even as the solving phase a simulation can be adopted or aborted to save time and money.

3.1 Material Parameter Accuracy

Most simulations run with estimated material parameters or average values. The QoD metric Q_{MPA} helps to verify implications of inaccurate material parameters. The correctness of the material parameter to describe the phenomenological behavior of the material based on a given physics depends on the accurate description of all relevant parameter. We define the error rate of all implemented material parameter: Given a region bm that describes the model of the simulation object and an interval $[t_0, t_n]$ that describes the simulation time period. Given a function $rMP(x,t)$ that describe the phenomenological behavior of the real material and a function $mMP(x,t)$ that describe the phenomenological behavior of the material in the implemented FEM model. The characteristic of material parameter accuracy Q_{MPA} without respect to the specific simulation context is defined by $min(\left|\frac{mMP(x,t)}{rMP(x,t)}\right|, \left|\frac{rMP(x,t)}{mMP(x,t)}\right|)$ for all $x \in bm$ and $t \in [t_0, t_n]$. Based on this definition the error rate of a specific material parameter j can be defined in the same manner.

3.2 Matrix Solver Accuracy

A matrix solver implements numerical methods to solve matrix equations and approximates (in most cases) the exact solution of the matrix equation. Q_{MSA} helps to verify implications of numerical problems. x_e is the exact solution of a test matrix equation $\mathbf{Ax} = \mathbf{b}$ and x_m is the numerical solution by using matrix solver m. We define the characteristics of the matrix solver accuracy Q_{MSA} by $\|x_e - x_m\|$ with $\|\|$ is a appropriate norm. If it is not possible to determine the exact solution x_e objectively a domain expert can determine the characteristics of Q_{MSA} subjectively.

3.3 Vector Condition

To solve a matrix equation $\mathbf{Ax} = \mathbf{b}$ with numerical methods, the condition of the vector $\mathbf{b}$ (and of $\mathbf{A}$) influences the solving performance and the quality of the solution $\mathbf{x}$. Q_{VC} helps to validate implications of numerical problems. If the difference regarding the least absolute value and the maximum absolute value of $\mathbf{b}$ is "too big" numerical errors can be estimated. b_l is the least absolute value with $b_l \neq 0$ and b_m the maximum absolute value of $\mathbf{b}$. The characteristics of the vector condition without respect to the specific simulation context is defined by $Q_{VC} = \frac{b_l}{b_m}$. We define the vector condition for the vector $\mathbf{x}$ in the same way. For this, we replace vector $\mathbf{b}$ and the values b_l and b_m by vector $\mathbf{x}$ and the values x_l and x_m.

4 Evaluating the Influence of Quality of Data Metrics

4.1 QoD Evaluation Framework

In order to measure, monitor, and evaluate QoD metrics for FEM-based simulations, we utilize our extensible QoD Evaluation Framework for workflows developed in [8]. Generally, with this framework, we can determine QoD in a very flexible way: (i) platform and language independent metrics and interpretations can be invoked, (ii) separate metrics as well as metrics that are included in comprehensive algorithms (e.g. a solver that include algorithms to determine the Jacobian determinant) can be used, and (iii) objective (automatically determined by a computer) and subjective (manual determined by a human) QoD determination is supported. During runtime, relevant data is needed for the determination of QoD is passed to the QoD Evaluation Framework by values or by references. This approach enables us to shift data to QoD metrics and interpretations or shift QoD metrics and interpretations to data to improve the performance.

Conceptually, Figure 2 describes how we utilize the QoD Evaluation Framework for understanding the dependencies among QoD for inputs, intermediate results, and final outcomes, as well as the influences of QoD on the performance and resource provisioning of the simulation. All necessary information about available QoD metrics and interpretations are stored into `Metric Definition`. Implementations of QoD metrics and interpretations can be found in `Software-based Evaluator` for automatic QoD determination and in `Human-based Evaluator` for manual QoD determination. Both kinds of evaluators and the data provisioning are managed by `Manager`. Furthermore, the `Manager` has all meta information described in Section 2.2. For QoD determination the `Manager` searches `Metric Definition` for an appropriated QoD metric and a corresponding QoD `Evaluator` and passes information about data (in XML messages) to the `Evaluator`. The `Evaluator` analyzes the specified data about their characteristics and returns XML-based `Metric Result` – including values of QoD metrics – to the `Manager`. The metric results will be analyzed and displayed together with, e.g., performance, failures, or storage informations.

4.2 Experiments

To analyze the influence of QoD metrics and goodness we used simulations developed atop the PANDAS framework. PANDAS was designed for simulations of multiphasic materials [9]. We used it for two reasons: It represents a typical FEM based simulation framework that can be used within a workflow environment (e.g. to perform multi scale simulations) and FEM-based basic simulation building blocks can be structured into the steps shown in Figure 1 [10].

We implement two simple but well proved simulations [11] simulating the same problem but having fundamental different characteristics in terms of the elasticity of the boundary: "fluid-saturated elastic column in an impermeable rigid tub" (EC) and "rigid slab on a fluid-saturated elastic half space" (RS). Furthermore, two solvers (GMRES and BiCG) with different numerical behaviors are utilized.

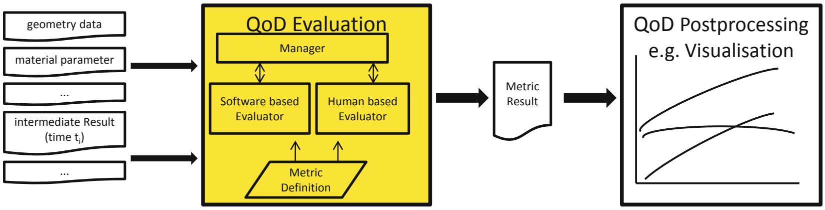

Fig. 2. Components and steps in evaluating QoD using the QoD Evaluation framework

To analyze the impact of QoD the presented QoD metrics Material Parameter Accuracy (Q_{MPA}), Matrix Solver Accuracy (Q_{MSA}), and Vector Condition (Q_{VC}) as well as performance measurement metrics (time in hour) are used. The goodness of QoD metrics with respect to the simulation context is defined based on domain expert's knowledge (due to the lack of space, the exact setting of the experiments can be found in [5]). We execute the experiments on a four CPU machine because it is sufficient to calculate our QoD metrics for FEM-based basic simulation building blocks, in particular for the preprocessing and the solving phase shown in Figure 1, of multi-scale simulations. Furthermore, it clarifies the analysis of the influence of QoD metrics and performance. Nevertheless, in practice we could use our approach to control multi-scale simulations with the aid of QoD findings within a multiple of computational resources when a large of basic simulation building blocks are employed.

To demonstrate the applicability and strength of our novel approach we have performed different correlations. We present three examples in detail: (i) the correlation between the goodness of Q_{MPA} and Q_{VC}, (ii) the correlation between the goodness of Q_{MPA} and the fault behavior, and (iii) the correlation between the goodness of Q_{MSA} and the performance. Several other correlations can be implemented. Examples are the correlation between the goodness of a QoD metric and storage provisioning or the correlation between the goodness of QoD metrics determined in different scales with a multi-scale simulation. Due the lack of space we present only selected findings in the following.

QoD-QoD Correlation: The QoD-QoD correlation (Figure 3 left) describes the relation between the goodness of Q_{MPA} (material parameter accuracy) and Q_{VC} (vector condition). Although the EC and the RS simulations are similar only with the exception of the flexibility of the boundary, the results of the Q_{VC} differ considerably. The RS simulation is less sensitive to material parameter changes than the EC simulation. As shown in Figure 1 the Q_{VC} of matrix data outputted in step 7 (vector $\mathbf{x}$) depends on Q_{MPA} in step 2. In the RS simulation a poor QoD within material parameter data do not influence the limited or good QoD of the vector condition. In contrast, at the EC simulation the Q_{VC} is overall poor and tends to have critical values when $Q_{MPA} \leq 0.7$. Based on this QoD-QoD correlation findings we concluded that: (i) the RS simulation has an uncritical numerical behavior at the vector $\mathbf{x}$ with respect to a not well accurate material parameters, (ii) the EC simulation has overall a poorer numerical behavior in

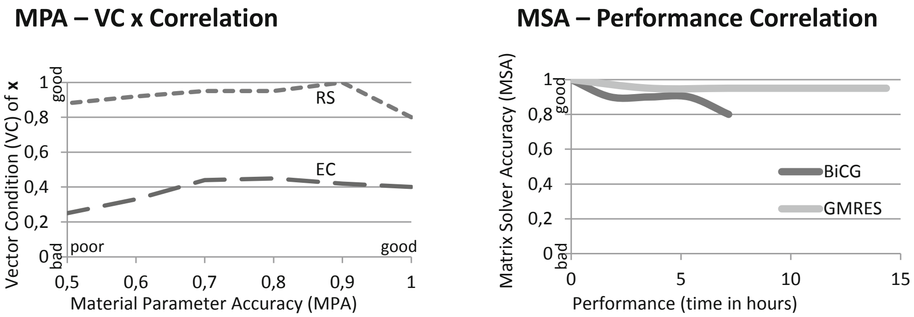

Fig. 3. Results of EC and RS simulations relating to parameter accuracy (MPA) and vector condition (VC) (left) and results of EC simulations relating to matrix solver accuracy (MSA) and performance (right)

solving the vector **x**, (iii) the EC simulation shows an critical numerical behavior at the vector **x** with respect to a not well accurate material parameters. As a consequence of the findings, our tool indicates that EC simulation should use correct material parameters as well as suitable matrix solvers that can handle a poor vector condition.

QoD-Fault Correlation: The QoD-Fault correlation describes the relation between the goodness of Q_{MPA} and the fault behavior (termination of the simulation framework) by executing the EC simulation. When $Q_{MPA} \leq 0.3$ the simulation framework PANDAS frequently terminates the calculation during the solving phase with error messages (no convergence). Overall we observed that in one of five simulations PANDAS terminated at the initial time step δt_0, in three of five simulations PANDAS terminated before the time step δt_{5000}, and in one of five simulations ended without error. Based on this knowledge a simulation should be adapted at the preprocessing phase with material parameters that have a sufficient QoD value in order to avoid simulation failures.

QoD-Performance Correlation: The QoD-Performance correlation describes the relation between Q_{MSA} and runtime of the EC simulation in hours. Figure 3 (right) shows that the used solvers have a different behavior: for 5000 time steps, the BiCG matrix solver took a half of the execution time (7 hours and 10 minutes) that the GMRES solver (14 hours and 21 minutes) required. However, for longer running simulations the GMRES solver provides results with a good QoD (0.95) but the BiCG solver calculates results that differs slightly from a proven result and the achieved QoD is limited to 0.8. Hence, our QoD framework indicates that (i) the GMRES solver provides reliable results, (ii) for simulations that run with a relative small number of time steps, the BiCG solver produces good quality of results in a fast way. By using our framework, scientists can find out the maximum number of time steps on which the Q_{MSA} is useful for their specific simulations.

5 Related Work

This section presents related work in the fields of QoD influences for computational simulations. Batini et al. present general concepts, methodologies, and techniques as well as a process to determine QoD [6]. We adopt it for the field of FEM based computational simulations. Hey et al. pointed out the principle need to observe QoD in scientific applications [3]. Heber and Gray concertize this need for FEM driven computational simulations [4] and have implemented a runtime environment for FEM based simulation [4,12]. But those implementations do not support any QoD evaluation.

The principles of the FEM are characterized by [13]. Those principles includes simulation dependent QoD aspects, e.g., basic approaches for error minimizing in the overall FEM and for several FEM steps. But the authors do not focus on simulation independent QoD metrics. Several approaches were already created for error minimizing in FEM. Lots of papers discuss special aspects in one step in detail. An example is description of the implementation of the concept of geometric multigrid algorithms and hierarchical local grid refinement [1]. Those kinds of paper depict in each case not the whole simulation (preprocessing, solving, and postprocessing phase). Nevertheless, we can use those approaches within specific QoD metrics.

Beyond that, performance analysis of FEM based simulation is well studied as well as optimization, e.g., in [2,14]. Our work is different as we focus primly on understanding QoD and only effected by this on research questions like performance or data quantity. In existing FEM tools or frameworks, such as Ansys, MatLab, or PANDAS QoD checking has not been provided. To our best knowledge, there is no work to define QoD metrics for FEM driven computational simulations in general and to study how QoD metrics impact on simulations.

6 Conclusions and Future Work

This paper analyzes common steps in FEM based computational simulations and proposes novel QoD metrics for the evaluation of the QoD and performance of FEM-based simulations. We have investigated how such metrics could provide benefit for understanding the inter-dependencies among QoD of inputs, intermediate data, and outputs as well as the performance needs of entire simulations. By monitoring and analyzing QoD influences with two real world simulation examples, we have shows that different types of QoD analyses could be very useful for understanding and steering FEM-based simulations.

We concentrate on basic simulation building blocks that can be used for general FEM-based multi-scale simulations. Therefore, new metrics and analysis methods must be investigated for data exchanged among different building blocks within and across specific simulation scales as well as for specific FEM approaches. In our future work, we focus on steering multi scale simulations and adapting resource provisioning based on QoD influences.

Acknowledgments. The work presented in this paper has partly been funded by the DFG Cluster of Excellence Simulation Technology (http://www.simtech.uni-stuttgart.de) (EXC310).

References

1. Bastian, P., Blatt, M., Dedner, A., Engwer, C., Kloefkorn, R., Ohlberger, M., Sander, O.: A generic grid interface for parallel and adaptive scientific computing. part i: abstract framework. Computing (2) (2008)
2. Chen, P., Zheng, D., Sun, S., Yuan, M.: High performance sparse static solver in finite element analyses with loop-unrolling. Adv. Eng. Softw. (4) (2003)
3. Hey, A., Tansley, S., Tolle, K.M.: The fourth paradigm: data-intensive scientific discovery. Redmond, Wash.: Microsoft Research (2009)
4. Heber, G., Gray, J.: Supporting finite element analysis with a relational database backend, part i: There is life beyond files. CoRR (2007)
5. Reiter, M., Truong, H.L.: Supplement report for quality of data implications. Technical report (2012), `http://www.iaas.uni-stuttgart.de/institut/mitarbeiter/reiter/Report/FemQoD.pdf`
6. Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques. Data-Centric Systems and Applications. Springer (2006)
7. IEEE: IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries, IEEE std 610-1990 (1990)
8. Reiter, M., Breitenbuecher, U., Dustdar, S., Karastoyanova, D., Leymann, F., Truong, H.L.: A novel framework for monitoring and analyzing quality of data in simulation workflows. In: 7th IEEE e-Science International Conference (2011)
9. Ehlers, W.: Foundation of multiphasic and porous material. In: Ehlers, W., Bluhm, J. (eds.) Foundation of Multiphasic and Porous Material. Springer (2002)
10. Reimann, P., Reiter, M., Schwarz, H., Karastoyanova, D., Leymann, F.: Simpl – a framework for accessing external data in simulation workflows. In: 14. GI-Fachtagung Datenbanksysteme für Business, Technologie und Web (2011)
11. Ehlers, W., Eipper, G.: Finite elastic deformations in liquid-saturated and empty porous solids. Transport in Porous Media (1999)
12. Heber, G., Gray, J.: Supporting finite element analysis with a relational database backend, part ii: Database design and access. CoRR (2007)
13. Zienkiewicz, O., Taylor, R., Zhu, J.: The Finite Element Method - Its Basis & Fundamentals, 6th edn. Elsevier Ltd., Butterworth-Heinemann (2005)
14. Shadid, J., Hutchinson, S., Hennigan, G., Moffat, H., Devine, K., Salinger, A.G.: Efficient parallel computation of unstructured finite element reacting flow solutions. Parallel Computing (9) (1997), Parallel computing methods in applied fluid mechanics