International Journal of Fracture Fatigue and Wear, Volume 4

Proceedings of the 5th International Conference on
Fracture Fatigue and Wear, pp. 204-208, 2016

PARALLELIZATION APPROACHES FOR FRETTING FATIGUE SIMULATION

R. Trobec[1], M. Depolli[1], G. Kosec[1], S. Bordas[2], S. Tomar[2], K. Pereira[3] and M. Abdel Wahab[3]

[1]Department of Communication Systems, Jožef Stefan Institute, Slovenia
[2]Faculté des Sciences, de la Technologie et de la Communication, Université du Luxembourg, Luxembourg
[3]Department of Electrical energy, systems and automation, Ghent University, Belgium

Abstract: This paper is an initial investigation of parallelization approaches for the numerical simulation of fretting fatigue under realistic conditions, e.g. cyclic non-proportional loading. The major difficulty in parallelising such simulations is the severe non-linearity of involved mechanical processes, which spans over multiple spatial and temporal scales. Generally, the problem comprises modelling of the contact, friction, stick and slip conditions that result in non-linearities localised upon the interfaces between components, the crack initiation phase that can be simulated using damage accumulation, and crack propagation, where the regions surrounding the tips of the cracks typically require a much more precise treatment and therefore present much higher computational burden than the rest of the domain. In general, such highly non-linear problems are difficult to implement in parallel, primarily due to the load-balancing issues.

Keywords: parallel simulation, speed-up, scalability, computational complexity, fretting fatigue, crack initiation and propagation

1 INTRODUCTION

Failure in mechanical components due to fretting fatigue is popular research topic due to its immediate practical application, e.g. aerospace riveted joints [1] and many other industrial applications in mechanical and civil engineering. The crack initiation, propagation and lifetime of the fretting fatigue have been studied both experimentally [2] and numerically, within a finite element framework [3]. Most of the numerical studies simulate fretting fatigue by Continuum Damage Mechanics (CDM) for the analysis of crack initiation and Linear Elastic Fracture Mechanics (LEFM) for the analysis of crack propagation. Assuming proportional loading conditions and therefore constant ratio between mode I and mode II Stress Intensity Factors (SIFs) enables fast simulation on a single lumped cycle only [4]. The drawback of such simplified approach is that it might lead to incorrect crack paths and fatigue lives as reported in [5]. The cycle-by-cycle simulations, e.g. using Cohesive Zone Models (CZM) [6], admit the non-proportionality, however require significantly higher computational power. Additionally, a significant amount of time is needed for the calculation of reliable stresses and for the adaptive re-meshing. For this reason, FEM could be ineffective for simulating high-cycle fatigue, e.g. 10^7 fatigue cycles [7]. A possible alternative is an explicit Meshless Local Strong Form Method (MLSM) that has been proven as effective in several application fields [8-10], in particular, if implemented on parallel computers [11, 12]. The downside of the MLSM approach is that it does not guarantee that the equilibrium equations are strictly satisfied at each time step, and must satisfy stability conditions which might induce additional complications due to the existence of cracks in the domain.

The computational time of simulations grows rapidly with higher required accuracy, typically ensured with denser spatial discretization, i.e. more elements or computational nodes, and number of loading cycles. An important part of the numerical approach is thus the effective implementation of the solution procedure on modern computer architectures. The developments in the technology of the computer architectures are nowadays extremely vivid. The processing power can be increased either by increasing the processor's clock frequency or by increasing the number of processing units. The clock frequencies are approaching their physical limits; therefore the second option - increased number of processing units - is becoming more attractive. Parallel computers, available today in most desktop computers or computer servers, can compensate for the lack of performance of a single computer, but only in cases where an efficient parallelization of the computational method is known. Various application programming interfaces (APIs) for parallel programming are used to maximize the performance of parallel systems. Nowadays, the most widely used APIs for parallel programming are MPI for distributed-memory systems, and Ptreads and OpenMP for shared-memory systems [13]. Moreover, using graphical processing units (GPUs) for solving parallel problems is widely spreading. APIs that support parallel programming on GPUs are becoming more and more

popular, like CUDA and OpenCL [14, 15]. Solving crack problems in parallel is intrinsically difficult, and rarely attempted in practice due to the implementation complexity and the learning curve involved. Besides the parallelization techniques for FEM that are mature [16], considerable effort has been recently invested in the parallelization of adaptive eXtended FEM (XFEM) [17] for simulations of composites fracture.

This paper describes an initial investigation of the parallelization approaches and opportunities for the numerical simulation of fretting fatigue cracks under real conditions, e.g. high resolution cyclic non-proportional loading, with high computational requirements that exceed the computational capabilities of a single computer. In Section 2, the crack simulation method is functionally decomposed. In Section 3, opportunities for the parallel speed-up of the crack simulations are analysed together with an approximate estimation of most promising numerical techniques.

2 FRETTING FATIGUE SIMULATION

The fretting fatigue solution methodology can be decomposed into independent functional blocks that can be executed on different computers. Note that the sub-functions cannot be always executed in parallel because of the data dependencies, i.e. results of a block are inputs of another block. Some functional blocks can be further decomposed by domain decomposition and implemented again in parallel on several computing cores.

Let us consider a generic simulation of a high number of cycles in fretting fatigue crack life. A birds view on such a functional decomposition results in (i) modelling of domain geometry, crack initiation, crack propagation and loading cycle, which mostly needs human resources, (ii) discretization, numerical solution, and parameter sweep, which require computational resources. Each block can be further decomposed. For example, the crack initiation phase is usually composed of (i) determination of geometry, (ii) modelling and simulation of contact's dynamic, (iii) testing of initiation criteria. The crack propagation phase is usually composed of the following steps: (i) computation of the stress intensity factors, (ii) verification of the crack stability, and (iii) computation of the direction of crack propagation. A principal functional decomposition of a fretting fatigue crack simulation is shown in Fig. 1. The blocks that can run in parallel are shown on a symbolic time axis.

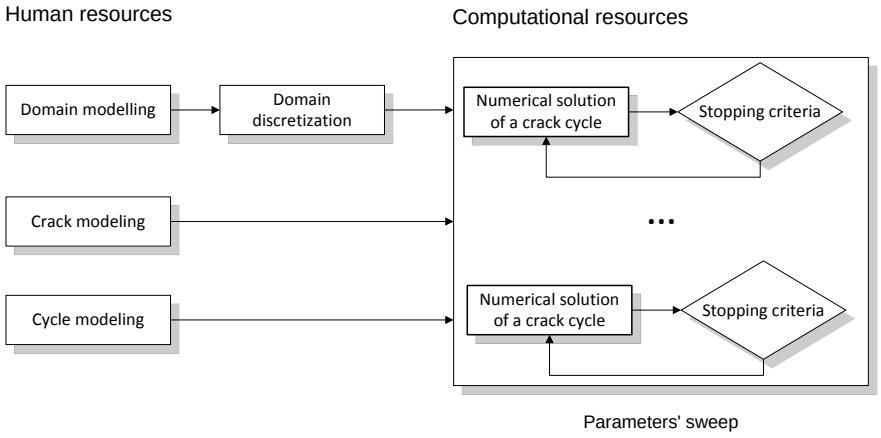

Fig. 1 Main functional blocks of the simulation of crack propagation under high-cycle fretting fatigue conditions

3 PARALLELIZATION AND COMPLEXITY ESTIMATION

In order to simulate millions of nodes and millions of fretting cycles, the parallelization of the fretting fatigue cracks simulation programs is required. Thousands of computing cores are available today in distributed-memory computing clusters with interconnected processing nodes that incorporate a multiple shared-memory processors and optionally also many-core accelerators, such as GPUs, for number crunching. Such hybrid architectures requires also hybrid parallelization strategy that incorporates thread-level parallelization, e.g. OpenMP in each computing node, combined with a node-level parallelization, e.g. problem decomposition and message passing communication (MPI). There exist some successful attempts to address the non-linear mechanics problems with the hybrid parallelization approach [18], with the aim to preserve acceptable computational scalability with high number of computational nodes in a wide area of applications. It seems that adaptivity [19] and XFEM [20] are appropriate solution methodologies for the parallel simulation of the fretting fatigue crack propagation. However, alternative algorithms can be used that are less optimal on a

single computer but very scalable and therefore appropriate for the parallel computers. For example, the limitations posed by re-meshing after each propagation of the crack can be solved by the local meshless method [21], which enables relatively simple and efficient adaptive implementation either in weak or strong formulation [22].

Another important aspect of the parallelization is the load balancing and minimal requirement for global communication [23]. Unbalanced system will suffer from increased computational time, since overloaded processors will be late with their output, on the other hand, global communication could limit the scalability because of congested communication network. Both aspects must be considered during the parallelization process. For example, in numerical solution of a crack simulation, the solution of the system of Partial Differential Equations (PDE) represents one of the major parts of the solution procedure. There are several possible approaches of addressing this problem; one possibility is to use a MLSM with the shared memory parallelization [11]. A complete localization of numerical treatment in MLSM enables directly applied parallelization through the OpenMP API on multi core CPUs and many-core accelerators. The OpenMP enabled program forks on request into several threads and processes the assigned task concurrently on multiple processor cores. The run time environment allocates the threads to the available processor cores. After the execution of the parallel sections of the code, OpenMP joins threads back into a single thread and the program continue with sequential execution. The schematic block diagram of the parallel solution based on MSLM is shown in the left part of Fig 2.

Another approach is to solve PDEs on distributed system such as a computer cluster with appropriate domain decomposition as shown in the right part of Fig 2. An important part of the implementation is the communication between computing nodes. In efficient distributed algorithms, the communication load must be minimized while the computation load must be well balanced between processors; otherwise, some computers would mostly wait for jobs. Although the communication plays an important role also in shared memory, in terms of data access latencies on different memory levels, it is much more pronounced in the distributed execution, where the interconnection topology plays a crucial role [23]. Note that significant speed-ups which scale on thousands of computing nodes are only possible on distributed computers.

The computational and memory cost of the solution procedure are mostly governed by the complexity of the numerical method used. In general both, memory and computational complexities, are functions of the number of discretization nodes N [11], however, there could be a significant difference in the constant. For example, estimated computational complexity of FEM system matrix construction is $O(N[logN+n_e])$, where n_e is the number of discretization nodes per element, while the complexity of the meshless local Petrov-Galerkin (MLPG) method, which is basically a meshless variant of FEM, is $O(Nn_q[logN+n_s m^2])$, where n_q is the number of quadrature points, n_s is the number of support nodes and m is the order of shape function basis; alternatively, the complexity of MLSM, a meshless variant of FDM [21], is $O(N[logN+n_s m^2])$. All methods ultimately result in a sparse and possibly banded non-linear system that can be solved by an iterative method with $O(Nd)$ operations, where d is the width of system matrix band.

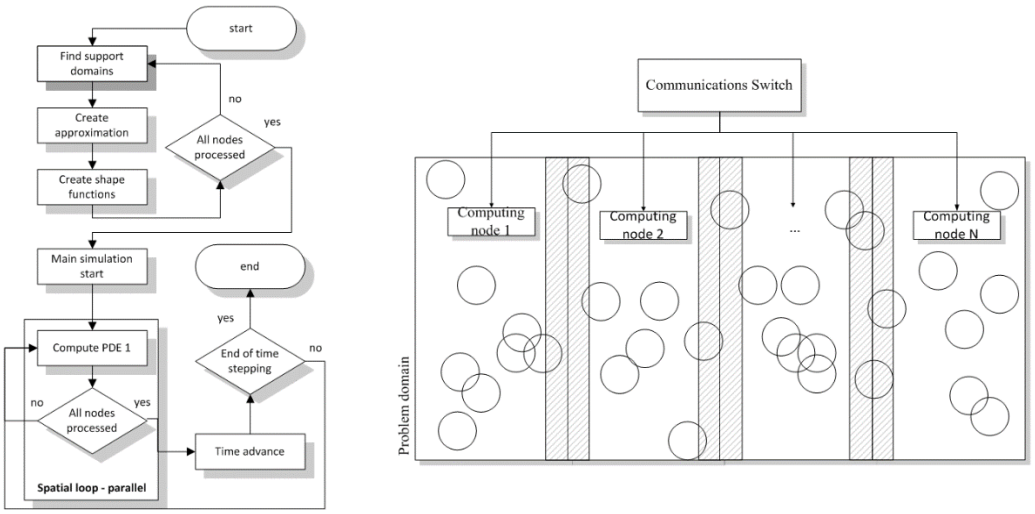

Fig. 2 A block diagram of OpenMP parallel solution based on MLMS on multicore computers (left) and a problem domain decomposition for parallel solution on interconnected computing nodes (right).

4 CONCLUSIONS

This paper is a short review of possibilities for speeding up the fretting fatigue simulations in order to approach the real situations with millions of fatigue cycles. The functional decomposition followed by the parallelization of numerical solution of each cycle, on a heterogeneous computer with hundreds of interconnected nodes, each with several cores, seems to be a viable approach. Near ideal speed-ups, as reported for other solid mechanics simulations, can be expected for the simulation of a single fatigue cycle, if using methodologies with local approach that reduce the frequency or complexity of re-meshing. The problem of inherently sequential nature of the fatigue simulation seems to remain a serious bottleneck, unless the crack history could be modelled differently, for example by a simulation of only few consecutive cycles and extrapolation of the results. On the other hand, multiple simulations with different parameters, which are often used in the real problem, are trivial for parallelization.

5 ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial support of the Research Foundation - Flanders - (FWO), The Luxembourg National Research Fund (FNR) and Slovenian Research Agency (ARRS) in the framework of the FWO Lead Agency project: G018916N ‘Multi-analysis of fretting fatigue using physical and virtual experiments’.

6 REFERENCES

[1] Hojjati-Talemi R, M. AW. Finite element analysis of localized plasticity in Al 2024-T3 subjected to fretting fatigue. Tribol Int. 2012;55:805-14.

[2] Hojjati-Talemi R, Abdel Wahab M, De Pauw J, De Baets P. Prediction of fretting fatigue crack initiation and propagation lifetime in cylindrical contact configuration. Tribology Int. 2014;76:73-91.

[3] Proudhon H, Basseville S. Finite element analysis of fretting crack propagation. Eng Fract Mech. 2011;78:685-94.

[4] Abdel Wahab M. Simulating mode I fatigue crack propagation in adhesively-bonded composite joints. Fatigue and fracture of adhesively-bonded composite joints: Elsevier; 2015. p. 323 - 44.

[5] Giner E, Sukumar N, Denia FD, Fuenmayor FJ. Direction of crack propagation in a complete contact fretting fatigue problem. Int J Fatigue. 2014;58:172-80

[6] Hillerborg A, Modeer M, Petersson PE. nalysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements. Cem Concr Res. 1976;6:773-81.

[7] Kim K, Yoon MJ. Fretting fatigue simulation for aluminum alloy using cohesive zone law approach. Int J Mech Sci. 2014;85:30-7.

[8] Kosec G, Trobec R. Simulation of semiconductor devices with a local numerical approach. Eng Anal Bound Elem. 2015;50:69-75.

[9] Kosec G, Šarler B. Simulation of macrosegregation with mesosegregates in binary metallic casts by a meshless method. Eng Anal Bound Elem. 2014;45:36-44.

[10] Kosec G. A local numerical solution of a fluid-flow problem on an irregular domain. Adv Eng Softw. 2016 10.1016/j.advengsoft.2016.05.010.

[11] Kosec G, Depolli M, Rashkovska A, Trobec R. Super linear speedup in a local parallel meshless solution of thermo-fluid problems. Comput Struct. 2014;133:30-8. 10.1016/j.compstruc.2013.11.016.

[12] Kosec G, Zinterhof P. Local strong form meshless method on multiple Graphics Processing Units. CMES-Comp Model Eng. 2013;91:377-96.

[13] Pacheco PS. An Introduction to Parallel Programming. Burlington: Morgan Kaufmann Publishers; 2011.

[14] Kirk DB, Hwu WW. Programming Massively Parallel Processors. Burlington: Morgan Kaufmann Publishers; 2010.

[15] Domínguez JM, Crespo AJC, Valdez-Balderas D, Rogers BD, Gómez-Gesteira M. New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters. Comput Phys Commun. 2013

[16] Babuska I. Some aspects of parallel implementation of the finite-element method on message. Parallel Algorithms for Numerical Linear Algebra. 2014;1:157.

[17] Vigueras G, Sket F, Samaniego C, Wu L, Noels L, Tjahjanto D, et al. An XFEM/CZM implementation for massively parallel simulations of composites fracture. Compos Struct. 2015;125:542-57. 10.1016/j.compstruct.2015.01.053.

[18] Casoni E, Jérusalem A, Samaniego C, Eguzkitza B, Lafortune P, Tjahjanto DD, et al. Alya: Computational Solid Mechanics for Supercomputers. Arch Comput Method E. 2015;22:557-76. 10.1007/s11831-014-9126-8.

[19] Waisman H, Berger-Vergiat L. An adaptive domain decomposition preconditioner for crack propagation problems modeled by XFEM. 2013;11:633-54. 10.1615/IntJMultCompEng.2013006012.

[20] Vigueras G, Sket F, Samaniego C, Wu L, Noels L, Tjahjanto D, et al. An XFEM/CZM implementation for massively parallel simulations of composites fracture. Compos Struct. 2015;125:542-57. http://dx.doi.org/10.1016/j.compstruct.2015.01.053.

[21] Trobec R, Kosec G. Parallel scientific computing : theory, algorithms, and applications of mesh based and meshless methods: Springer; 2015.

[22] Kosec G, Šarler B. H-adaptive local radial basis function collocation meshless method. CMC-Comput Mater Con. 2011;26:227-53.

[23] Trobec R, Šterk M, Robič B. Computational complexity and parallelization of the meshless local Petrov-Galerkin method. Comput Struct. 2009;87:81-90.