Abstract

Akselos enables extremely fast simulations of large and complex parametrized systems. The two main ingredients to achieve this are:

1. The Reduced Basis (RB) Method, a powerful model order reduction technique for parameterized partial differential equations (PDEs), and

2. A component-based approach, which enables engineers to create large, reusable, and reconfigurable models in an efficient manner.

The combination of these two ingredients provides a new modeling framework that accelerates conventional Finite Element Analysis (FEA) without sacrificing accuracy, and we refer this framework as RB-FEA. In this whitepaper we give an overview of the mathematical principles behind Akselos’s RB-FEA simulation technology.

We begin in Section 1 by recalling the key ideas of conventional FEA, and we then explain how the RB method builds on conventional FEA in order to provide fast, parameterized models.

In Section 2, we discuss the FEA “substructuring” or “superelement” method. This is a classical approach that was first popularized in the 1970s, and which enables a component-based workflow for FEA. However, this method has a major limitation in practice: each time a change to a component is required (e.g. to modify geometry or material properties), then expensive FEA computations must be performed in order to update the component’s data accordingly. We then describe how the RB-FEA approach circumvents this issue by providing parameterized components which can be modified and re-solved extremely quickly.

In Section 3, we discuss nonlinear analysis in the context of RB-FEA. Akselos provides a hybrid solver that couples conventional FEA and RB-FEA in order to enable efficient nonlinear analysis of large systems.

We conclude with some example problems that demonstrate RB-FEA for steady-state, modal, and dynamic structural analysis in Section 4. Finally, we provide a summary in Section 5.
Contents

1. How does RB-FEA relate to FEA? 3
   1.1 Overview of FEA ................................................................. 3
   1.2 The Reduced Basis Method accelerates FEA while maintaining accuracy 3

2. Key to Success: Combining RB with a Component-Based Approach 6
   2.1 Basic principles of component-based approaches .......................... 6
   2.2 RB-FEA for Parameterized Component-based Modeling .................... 7

3. Hybrid Solver for Nonlinear Analysis 9

4. Numerical Examples 10
   4.1 Example: Steady-state analysis of a shiploader .......................... 10
   4.2 Example: Modal analysis of a shiploader shuttle .......................... 10
   4.3 Example: Transient analysis of a shiploader shuttle ....................... 11

5. Summary 12
1. How does RB-FEA relate to FEA?

1.1 Overview of FEA

FEA involves three major steps [8, 16]: (i) modeling of the system geometry, (ii) meshing of the geometry, and (iii) computational approximation of the Partial Differential Equation (PDE) solution. Steps (i) and (ii) are performed with CAD and meshing software tools, respectively, and then step (iii) requires an FEA solver.

Assuming the PDE is linear, then the steps listed above lead to a linear algebraic system of size $\mathcal{N}$:

$$
\begin{bmatrix}
[A] \\
[N] \\
[N] \\
[N]
\end{bmatrix} 
\begin{bmatrix}
[u] \\
[N] \\
[N] \\
[N]
\end{bmatrix} = 
\begin{bmatrix}
[f] \\
[N] \\
[N] \\
[N]
\end{bmatrix}
$$

with $\mathcal{N}$ denotes the number of degrees of freedom in the model, $A \in \mathbb{R}^{N \times N}$ is the stiffness matrix, $\mathbf{x} \in \mathbb{R}^{N}$ is the solution vector, and $f \in \mathbb{R}^{N}$ is the right hand side vector. (If the PDE is non-linear, then we obtain a nonlinear algebraic system, which is solved by first linearizing using Newton’s method. As a result, in either the linear or nonlinear case we must deal with large linear systems of equations as described above.)

FEA is very powerful because any geometry – even the most complex ones – can be represented using a mesh, and the methodology applies to most classes of PDEs. FEA is used to simulate many types of physics, e.g. elasticity, thermal, acoustics, electromagnetics, fluid dynamics, and can also deal with multi-physics in which multiple different types of physics are included in a single model.

However, FEA software has some significant drawbacks:

- Steep learning curve: Its use requires expertise in computational engineering.
- Labor intensive: Building computational models is very time-consuming. For example, as pointed out above, the meshing step alone often takes several days.
- Computationally intensive: Even modest FEA models require heavy-duty computational resources, and simulation run-times often require hours or even days. The root cause of this is that assembling and solving large scale matrix/vector systems is highly computationally intensive.
- Simulations are “one-shot”: The CAD and meshing process must be repeated each time a design is changed.
- System size is limited: Solve times and memory requirements grow superlinearly with $\mathcal{N}$. This places a hard limit on model size that can be solved on any given workstation.

1.2 The Reduced Basis Method accelerates FEA while maintaining accuracy

To address these shortcomings, there has been a significant research effort on the Reduced Basis (RB) technology over the past 15 years, with leading research being done on this topic at MIT and EPFL, see for example [4, 7, 9, 13, 14, 15]. (Note that in this section we discuss the “single domain” RB method. We consider component-based RB-FEA in the next section.)

RB algorithms use FEA as the underlying technology, and add an extensive layer of mathematics and computational algorithms on top of FEA in order to:

- Accelerate the simulation time by orders of magnitude without compromising accuracy.
- Provide parameterized models which can be quickly and easily modified.
The key observation that leads to RB is that the discrete function space $X^N$ of dimension $N$ that is used in FEA to approximate the solution of a PDE is much larger than necessary.

Consider a parameterized PDE which is governed by the parameter vector $\mu \in \mathcal{D} \subset \mathbb{R}^p$. This means that we define a parameter domain $\mathcal{D}$ such that every choice of $\mu \in \mathcal{D}$ yields $p$ real numbers that are our parameters (these could be Young’s modulus, Poisson ratio, thermal conductivities, Reynolds number, geometric dimensions, etc.). In general each distinct choice of $\mu$ requires a new PDE solve. When we use FEA to perform these PDE solves, we effectively obtain a mapping from $\mu \in \mathcal{D}$ to $u^N(\mu) \in X^N$. The key point, then, is that in the context of parameterized PDEs we care only about the low dimensional “solution manifold” $\{u^N(\mu), \mu \in \mathcal{D}\}$, rather than the high dimensional FEA space $X^N$. For instance, if the PDE depends on one parameter, then the solution $u^N(\mu)$ lives on a one dimensional (nonlinear, but often very smooth) filament as represented in Figure 1. As a result we can develop a numerical method that is targeted at the low dimensional solution manifold, and this can be achieved using an approximation space of dimension $N \ll N$. This approximation space is built up using FEA solves of $N$ “snapshots” $u^N(\mu_i)$ at carefully chosen parameter values $\mu_i, 1 \leq i \leq N$, see the review article [9] for more details. In the RB framework, the $\mu_i$ are selected using the RB Greedy Algorithm, which has been developed and studied extensively in the RB literature and has been shown to provide near-optimal approximation properties in general. This methodology leads to a matrix/vector system of size $N$:

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} u \end{bmatrix} = \begin{bmatrix} f \end{bmatrix}.$$ 

In practice, we typically obtain $N \sim 10$ (compared to $N \geq 10^7$ for FEA). Hence the solve time is orders-of-magnitude faster with RB than with FEA, while typically retaining accuracy within 1% with respect to the FEA solution. It is important to note that there is no free lunch here: this huge computational speed up is obtained at the expense of prior “offline” computation of $N$ snapshots, of complexity $O(NN^{1+\alpha})$, where $\alpha > 0$ depends on the FEA matrix’s sparsity pattern and condition number. In practice, the RB data can be pre-computed and stored so that it is ready to be used for fast solves whenever needed.
Figure 1: The curve represents a one dimensional parametric manifold \( \{ u^N(\mu), \mu \in \mathcal{D} \} \), where in this case \( \mathcal{D} \subset \mathbb{R} \). We construct a RB space by sampling the solution manifold using the adaptive Greedy algorithm that has been developed extensively in the RB literature. Once an RB basis has been created then we use a Galerkin method to compute optimal RB solutions at any new parameter vector in \( \mathcal{D} \).
2. Key to Success: Combining RB with a Component-Based Approach

2.1 Basic principles of component-based approaches

Several FEA vendors (e.g. MSC, ANSYS, Simulia) provide component-based solvers. In the context of static problems, these approaches are typically referred to as “substructuring” or “superelements,” and for dynamic problems they are referred to as the Craig-Bampton method or Component Mode Synthesis (CMS). These approaches involve subdivision of models into simpler components in order to facilitate the design, meshing, and solution steps. This type of subdivision is illustrated in Figure 2.

Component-based approaches provide the following benefits:

- Division of labor: different substructures are taken care of by different design teams, which can work independently and in parallel as long as the interfaces between substructures remain compatible.
- Taking advantage of repetition: many structures are built of many similar components, in which case it is only necessary to deal with each archetypal component once.
- Modularity makes it easier to modify systems: global systems can be changed by modifying or replacing individual components.
- Solve time acceleration by pre-computing data for component interiors.

For example, consider the simple case of a system which is broken down into two components corresponding to subdomains $\Omega_1$ and $\Omega_2$ with interface $P$. 
We can group the degrees of freedom into degrees of freedom internal to each subdomain \((u_{\Omega_1}, u_{\Omega_2})\) and degrees of freedom \(U\) belonging to the interface. The number of degrees of freedom on the interface can also be truncated, and this is commonly done in CMS, for example. The FEA linear system then be written as

\[
\begin{bmatrix}
A_P & A_P^{T,\Omega_1} & A_P^{T,\Omega_2} \\
A_P,\Omega_1 & 0 & 0 \\
A_P,\Omega_2 & 0 & A_P^{T,\Omega_2}
\end{bmatrix}
\begin{bmatrix}
U \\
\Omega_1 \\
\Omega_2
\end{bmatrix}
= \begin{bmatrix}
f_P \\
f_{\Omega_1} \\
f_{\Omega_2}
\end{bmatrix}
\]

We then follow a static condensation procedure that allows elimination of the degrees of freedom internal to each subdomain: we compute the Schur complement matrix and right hand side

\[
A = A_P - A_P^{T,\Omega_1} A_{\Omega_1}^{-1} A_P,\Omega_1 - A_P^{T,\Omega_2} A_{\Omega_2}^{-1} A_P,\Omega_2
\]

\[
F = f_P - A_P^{T,\Omega_2} A_{\Omega_2}^{-1} f_{\Omega_2} - A_P^{T,\Omega_2} A_{\Omega_2}^{-1} f_{\Omega_2}
\]

which yields the static condensation system

\[
\begin{bmatrix}
A \\
U
\end{bmatrix}
\begin{bmatrix}
N_P \\
N_P \times 1
\end{bmatrix}
= \begin{bmatrix}
F \\
N_P \times 1
\end{bmatrix}
\]

This system is now of size \(N_p\), the number of degrees of freedom on the interface \(P\), which is much smaller than \(N\). Hence this system is significantly faster to solve than the original FEA system. However, assembling the static condensation system is very costly. For example, computation of \(A_{\Omega_2}^{-1} A_P,\Omega_2\) corresponds to \(N_p\) FEA solutions on the subdomain \(\Omega_1\).

In practice the data for each component is typically pre-computed and stored so that it can be used for subsequent system-level solves. But of course if any change is made to a component, then the pre-computation phase must be repeated. This leads to a cumbersome workflow where updating models is a laborious and computationally intensive process. RB-FEA addresses this issue, as discussed in the next section.

### 2.2 RB-FEA for Parameterized Component-based Modeling

Akselos’s RB-FEA method uses the same static condensation based approach for connecting components together as described in the previous section. However, the key difference is that RB-FEA uses the RB method on component interiors. This is a major improvement over conventional superelement approaches because now each component is parameterized, and users can easily modify models by changing parameters and re-solving, without needing to perform any expensive FEA computations.

---

1Here we consider the static case, but the dynamic case is similar.
This approach is described in the journal article “A Static Condensation Reduced Basis Element Method: Approximation and A Posteriori Error Estimation” by Dr. Phuong Huynh, Dr. David Knezevic and Prof. A.T. Patera [6]. We also note that, similar to CMS, Akselos's RB-FEA method uses a truncated set of degrees of freedom on component interfaces. Several interface degree of freedom truncation schemes that are accurate and efficient for parameterized models have been studied in the literature, including empirical modes, pairwise training, and so-called optimal modes [1, 2, 3, 5, 10, 11]. Akselos leverages each of these approaches in order to ensure fast and accurate analysis of component-based systems.

Figure 3 illustrates the RB-FEA workflow. The key steps are: (i) development of parameterized components, (ii) assembly of component-based models, (iii) configuration of the models by setting geometric and material parameters, and (iv) performing fast RB-FEA solves for the configured models. Steps (ii), (iii), and (iv) can be performed very quickly, whereas step (i) requires computationally intensive FEA calculations at the component level to obtain the pre-computed RB-FEA data.
3. **Hybrid Solver for Nonlinear Analysis**

A major limitation of the component-based formulations introduced in Section 2 is that they are inherently restricted to linear analysis since the elimination of component interior degrees of freedom relies on linearity of the PDE. To resolve this issue, Akselos has developed a unique hybrid solver that provides a tight coupling between conventional FEA and RB-FEA. This means that we can use RB-FEA in "linear regions," and conventional FEA in "nonlinear regions." In cases where a model contains localized nonlinearities (which is common in practice, e.g. for localized failure analysis, or localized contact), this provides significant speedup compared to solving the entire model with FEA.

Using this approach Akselos provides accelerated analysis for the full range of nonlinearities that can be modeled with FEA, including contact, friction, plasticity, and finite strain elasticity. A journal article that provides a full discussion of this hybrid solver methodology is currently in preparation.

The list of solve types that are provided by Akselos's RB-FEA, FEA, and Hybrid solvers is shown in Figure 4.
4. Numerical Examples

In this section we demonstrate the capabilities of Akselos RB-FEA solver for linear problems. We consider three structural analysis model problems: (i) steady-state, (ii) modal analysis, and (iii) dynamic analysis, and in each case we demonstrate a speed up of between 700 and 900 compared to conventional FEA. Also, we note that the problem sizes considered here are modest. For larger models the speed up factor is routinely well over 1000.

4.1 Example: Steady-state analysis of a shiploader

A shiploader is a large steel structure that supports a belt and is used to carry minerals from trucks on the ground to ships on the water. The complete 3D model of a shiploader is shown in Figure 5 (a). Using FEA, the mesh for the complete model has 1.8 million nodes, 6.3 million elements (tetrahedral elements), and the final finite element system has 5.5 million degrees of freedom. In order to use Akselos simulation technology, this model is decomposed in 323 components, with 65 different component types. Some of these components are shown in figure (c). Some component types, such as trusses, are reused several times in the model. In this case, the total data footprint for Akselos precomputed components is 5.1 GB.

With Akselos, the structural analysis results (displacements) are obtained within 2 seconds, whereas it takes 30 minutes with a standard FEA. Akselos technology thus enables a massive speed up of 900 while preserving FEA accuracy (maximum error 1%). After a postprocessing of the displacement solution field, we can display the stresses in the structure as shown in figure (b). Some of the components in the Akselos model are parametrized, such as a truss with a crack that can be moved laterally. The user can change the location of the crack without remeshing, recompute the global displacement solution in 2 seconds, and then visualize the stresses as shown in figure (d).

<table>
<thead>
<tr>
<th>Speed up vs. FEA</th>
<th>900</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum error vs. FEA</td>
<td>1%</td>
</tr>
</tbody>
</table>

4.2 Example: Modal analysis of a shiploader shuttle

Here we consider a subpart of the shiploader structure presented on the previous page: the shuttle is the part of the shiploader that can slide in and out in order to adjust the length of the structure. The full shuttle model with its components is shown on page 6. This model is decomposed in 69 components, with 31 different component types. Regarding FEA, the mesh for the complete model has 440 thousands nodes, 1.6 million tetrahedral elements, and the final finite element system has 1.3 million degrees of freedom.

With Akselos simulation technology, the modal analysis results (first five displacement eigenmodes) are obtained within 0.5 seconds, whereas it takes 6 minutes with a standard FEA. Akselos technology allows thus enables a speed up of 700. Figure 6 (a) shows the shuttle model with the locations (indicated by lock symbols) where clamping boundary conditions have been imposed. Figures 6 (b) and (c) show the first and fifth eigenmodes respectively. Please see [12] for details.

<table>
<thead>
<tr>
<th>Speed up vs. FEA</th>
<th>700</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum error vs. FEA</td>
<td>1%</td>
</tr>
</tbody>
</table>
4.3 Example: Transient analysis of a shiploader shuttle

Here we consider the same shuttle structure as above, but we introduce a small modification: we removed the diagonal trusses in order to decrease the lateral stiffness of the structure. This modification of the geometry doesn’t require remeshing since we only need to remove a few components in the model. This illustrates one of the advantages of Akselos component-based models, namely the ability to modify the topology of simulation models without the need to remesh.

We then perform a transient analysis of the structure after a lateral shock on the end of the structure (normal load applied at t=0s), with the same clamping boundary conditions as in Figure 6 (a) in the previous page. Figures 7 (a) to (d) show the evolution of the displacement with respect to time, where we observe the structure oscillating around its resting position. The transient analysis is computed using modal superposition, hence the speed up obtained with respect to FEA is the same as the one obtained for modal analysis.

<table>
<thead>
<tr>
<th>Speed up vs. FEA</th>
<th>700</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum error vs. FEA</td>
<td>1%</td>
</tr>
</tbody>
</table>
5. Summary

In this report we outlined the mathematical background of Akselos’s RB-FEA simulation technology. The main ingredients are the Reduced Basis Method in combination with a component-based approach. This allows engineers to easily construct large, complex, parametrized models, and perform extremely fast simulations. We also discussed Akselos’s hybrid solver which enables nonlinear analysis by tightly coupling RB-FEA and conventional FEA within a single global model. The capabilities provided by Akselos’s solvers reduces or eliminates the key computational bottlenecks in the analysis of large-scale engineering models, and therefore opens up many new possibilities, including:

- **Design optimization.** An optimization algorithm can be used with Akselos’s solvers to efficiently identify an optimal configuration of a parameterized model.

- **Parameter fitting.** Identify model parameters such that an RB-FEA model matches sensor data from a physical asset. This requires many iterations to find a good fit, and hence is fast with RB-FEA and often unfeasible with conventional FEA.

- **Statistical analysis.** Monte Carlo analysis of a system involves performing many solves for many different parameter values. This enables uncertainty quantification or sensitivity analysis of complex systems, which in turn provide critical insights into maintenance and operation of real-world engineering systems.


About Akselos

Akselos is a digital technology company headquartered in Switzerland, with operations in Europe, the USA and South East Asia. The company has created the world’s most advanced engineering modelling, and fastest simulation technology, to protect the world’s critical infrastructure today and tomorrow. The technology has the power to revolutionise how we build and manage our critical infrastructure, and pushes the boundaries of what modern engineering and data analytics can achieve. Developed by some of the world’s best minds, the MIT-licensed technology builds something far beyond the capability of a conventional digital twin – a digital guardian that allows operators to not only monitor an asset’s condition in real time, but helps them to see the future.