SAND2016-10948C

IMAC manuscript No.
(will be inserted by the editor)

Component Mode Synthesis with Large Contact Patches

Peter B Coffin · Adam R Brink · Nathan Crane · Mark Merewether

Received: date / Accepted: date

Abstract Component Mode Synthesis (CMS), specifically Craig-Bampton Reductions (CBRs), are popular methods for constructing reduced-order models of substructures for structural dynamics modeling. When the interfaces between these substructures are of limited complexity, such as a discrete set of bolts, these approaches work well, providing compact and portable models. With large interfaces relative to the volume, such as contact patches, these approaches become less convenient. Interfaces are generally represented as a series of nodes that do not retain interface-topology that may be critical to using more advanced contact algorithms or providing for contact between substructures. The interface may also lead to sufficient model complexity that each reduced model should be decomposed into multiple domains when included in

P Coffin *
MS 0557
Sandia National Labs
Albuquerque, NM Tel.: 1-505-845-8798
E-mail: pcoffin@sandia.gov

A Brink
MS 0840
Sandia National Labs
Albuquerque, NM Tel.: 1-505-284-9827
E-mail: arbrink@sandia.gov

N Crane
MS 0845
Sandia National Labs
Albuquerque, NM Tel.: 1-505-284-9808
E-mail: nkcrane@sandia.gov

M Merewether
MS 0845
Sandia National Labs
Albuquerque, NM Tel.: 1-505-284-7606
E-mail: mtmerew@sandia.gov

[0] Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract No DE-AC04-94AL85000.

system level models. In this work we outline the construction of a CBR where interface topology is retained so that models with CBR-CBR contact can be constructed. This is demonstrated with a set of example problems. We also investigate simple approaches to decomposing the CBR into multiple domains for parallel computing and their relative efficiency.

Keywords CMS · CBR · Craig-Bampton Reduction · Contact · Finite element analysis

1 Introduction

The Finite Element Method (FEM) is commonly used to model and analyze the vibration of mechanical structures [Inman and Singh(2001)]. To reduce the computational complexity and cost of performing mechanical vibration analyses, reduced order models of system components are often constructed using Component Mode Synethesis (CMS). CMS, introduced by [Hurty(1965)] and [Bampton and Craig(1968)], simplifies the component's complete FE model into one that is a function of a smaller set of Degrees Of Freedom (DOFs). Review of traditional CMS techniques are provided by [Hintz(1975)] and [Craig Jr(1985)].

CMS techniques, particularly the method described in [Bampton and Craig(1968)] and [Craig Jr(1995)] (Craig-Bampton Reduction), decompose the model into degrees of freedom (DOFs) associated with motion on the interface (constraint modes) and the degrees of freedom associated with motion internal to a component (fixed-interface modes). Using modal analysis, a small set of internal, fixed-interface modes are used to represent important component structural dynamics. If the number of interface connections is also small, as when components are connected by a few bolted joints, a compact model can be constructed with CMS.

A component interface that contains many FE faces or nodes yields a less compact model using the typical Craig-Bampton Reduction (CBR) since each of the DOFs are retained and are directly coupled to every other DOF on the interface. An approach to alleviate this problem is given by [Castanier et al(2001)Castanier, Tan, and Pierre], where a subset of characteristic modes on the interface replaces the typical constraint modes.

In this work we are interested in the use of CMS models where there is a large interface (relative to the volume) that may be in contact with other components. This use case poses two challenges: one, to accurately represent contact a detailed representation of the interface must be retained along with its topology for modeling by existing contact algorithms; two, for a typical CBR, this large number of interface DOFs (constraint modes) yields a large model that is completely coupled (dense) amongst those interface DOFs, which may become too large to represent on one processor.

We present numerical examples of CBR behavior when considering contact, enabled by retaining interface topology in Section 2.2. We also present preliminary efforts and results of decomposing a component with a large interface into multiple domains. The component is decomposed using typical FE domain decomposition tools and then a CBR is performed on each decomposed domain, resulting in a set of

coupled CBRs. We will discuss the benefits and pitfalls of this approach in Section 3.4. In Section 3 we study the behavior of these approaches with a selection of numeric examples.

This work was completed using a combination of Matlab scripts and preprocessing utilities along with the Sierra Mechanics ([Reese et al(2015)] and [Team(2011)]) FE codes and SEACAS domain decomposition tools from Sandia National Labs.

2 Method

2.1 Craig-Bampton Reduction Overview

In this work we utilize the CBR as our CMS technique. The CBR operates on the undamped FE model of a given component:

$$\mathbf{M}^{\alpha}\ddot{\mathbf{u}}^{\alpha} + \mathbf{K}^{\alpha}\mathbf{u}^{\alpha} = \mathbf{f}^{\alpha}, \tag{1}$$

where $\mathbf{u}^{\alpha}$ are the DOFs associated with the component, $\ddot{\mathbf{u}}^{\alpha}$ the second-order time derivatives of the displacement DOFs (acceleration), $\mathbf{f}^{\alpha}$ the external forces at nodes, $\mathbf{M}^{\alpha}$ the mass matrix and $\mathbf{K}^{\alpha}$ the stiffness matrix.

The DOFs in the complete (α) component model are broken into two groups: DOFs that lie on the interface of the component, constraint DOFs (c) and DOFs that lie interior (I) to the component. The decomposed linear system is represented as:

$$\begin{bmatrix} \mathbf{M}_{II} & \mathbf{M}_{IC} \\ \mathbf{M}_{CI} & \mathbf{M}_{CC} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}_I \\ \ddot{\mathbf{u}}_C \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{II} & \mathbf{K}_{IC} \\ \mathbf{K}_{CI} & \mathbf{K}_{CC} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_I \\ \mathbf{u}_C \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_I \\ \mathbf{f}_C \end{Bmatrix}. \tag{2}$$

The CBR transforms the complete system of both constraint (c) and interior (I) DOFs into a system of the constraint DOFs (c) and a set corresponding to interior structural mode shapes (k):

$$\begin{bmatrix} \mathbf{M}_{kk} & \mathbf{M}_{kc} \\ \mathbf{M}_{ck} & \mathbf{M}_{cc} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}_k \\ \ddot{\mathbf{u}}_c \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{kk} & \mathbf{K}_{kc} \\ \mathbf{K}_{ck} & \mathbf{K}_{cc} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_k \\ \mathbf{u}_c \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_k \\ \mathbf{f}_c \end{Bmatrix}. \tag{3}$$

Note that the CBR results in system matrices that have the following form:

$$\begin{array}{rcl} \mathbf{M}_{kk} & = & I_{kk} \quad \text{(diagonal)} \\ \mathbf{M}_{kc} & = & \mathbf{M}_{ck}^{T} \quad \text{(dense)} \\ \mathbf{M}_{CC} & = & - \quad \text{(dense)} \\ \mathbf{K}_{kk} & = & \Lambda_{kk} \quad \text{(diagonal)} \\ \mathbf{K}_{kc} & = & \mathbf{K}_{ck}^{T} \quad = \mathbf{0} \\ \mathbf{K}_{CC} & = & - \quad \text{(dense)}. \end{array} \tag{4}$$

Due to the form of these matrices, the computational complexity of the reduced model quickly becomes driven by the interface when more than a few element faces or points lie on the interface. That is:

$$N_k \gg N_C, \tag{5}$$

the number of interface DOFs is greater than the number of fixed interface DOFs. For large interfaces this means either controlling this growth in complexity via a

method such as characteristic constraint or being able to decompose the component or submodel.

In this work, the Sierra/Structural Dynamics (SD) code is used to compute the CBR for a given component. Analysis of the system model, often involving contact between components, is performed in Sierra/Solid Mechanics (SM) using an explicit time solver. In the Sierra mechanics codes, CBRs are represented in the construct of a superelement. A superelement is simply a generic linear element where the nodes, DOFs, and element matrices are defined in input file. This information is read into the code, nodes and DOFs are correlated between the superelement and the rest of the existing mesh and the superelement matrices are used to compute the element response.

2.2 General Contact

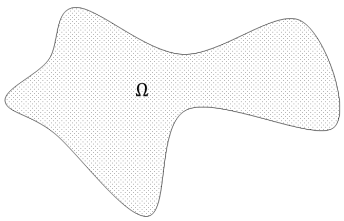
In the Sierra/SM code, contact is either defined between a set of nodes and a set of element faces or between two sets of element faces. Considering a superelement, contact can only be defined between the nodes on the superelement and faces in the rest of the FE mesh. One key limitation of this is that superelements can not be in contact with each other, as superelements do not retain topology information about their interfaces.

The most convenient solution to this problem would be to augment the file defining the superelement with information about the interface topology. This information could be passed along to the typical contact algorithms within the finite element code and little further effort would be necessary. Due to the complexity of code changes necessary for this approach, an alternative approach is used to demonstrate superelement-superelement contact here.

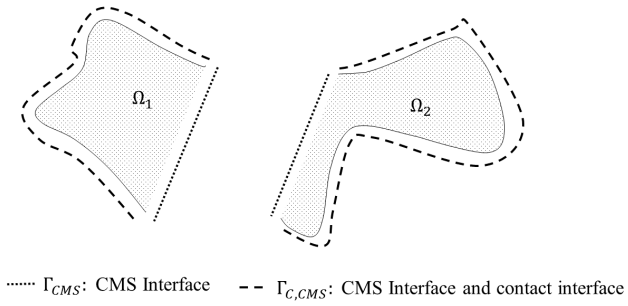
In this work we demonstrate contact with superelements by introducing fictitious shell elements at the superelement interface. This is a relatively simple intermediate step completed by a Matlab script. The original FE mesh of the component that is subject to the CBR contains both a nodeset and sideset that define the interface (constraint) nodes for the CBR. The sideset provides topology information about the nodes that make up the element faces that is then directly used to add blocks of shell elements to the FE mesh that the CBR is later be included in.

2.3 Decomposition

A simple approach to reducing CMS complexity for components with large interfaces is to use traditional FEA domain decomposition tools. We propose decomposing a component into n-domains and constructing a CMS reduction of that domain. In each decomposed domain, the CMS interface is a union of the original component interface on that domain and the interface between the decomposed domain and its neighbors. Figure 1 depicts this process, in which the outer surface of the component represents both CBR interface nodes and contact surface. For the purpose of contact, fictitious shell elements will be located here. The shell elements and CBR interface share



(a) Original Component Domain



(b) Decomposed Domains CBR Interfaces

Fig. 1 Decomposition approach creating multiple CBRs within original component. Here the outer surface of the component will be in contact with other components, the decomposed component domains are linked via their contiguous nodes and DOFs.

nodes and DOFs so constraints tieing them are unnecessary. The internal component interfaces resulting from decomposition are additional CBR interface nodes. These interface nodes are shared between decomposed domain CBRs and naturally provide the connection between them.

To study the expected complexity of these decompositions we compute the number of DOFs associated with decomposing a cube. We assume the cube has a structured mesh, with consistent sizing in all directions and an even number of elements along each edge. We compute resulting decompositions based on cutting the cube N_{cuts} times in each direction. The number of total domains then is:

$$N_{Domain} = (N_{cuts} + 1)^3, \tag{6}$$

where the total number of DOFs $N_{Dofs,Total}$ is:

$$N_{Dofs,Total} = (N_{EPL} + 1)^3, \tag{7}$$

where N_{EPL} is the number of elements per line (or edge). The number of nodes per edge N_{PL} in the decomposed domain is:

$$N_{PL} = \left(\frac{N_{EPL}}{N_{cuts}+1} + 1 \right). \tag{8}$$

The number of interface DOFs per domain (assuming 3 DOFs per node) is:

$$N_{IfaceDofsPerDom} = 3\left(6N_{PL}^2 - 12N_{PL} + 8\right). \tag{9}$$

This means that the approximate number of interface DOFs per domain ($N_{IFDofsPerDom}$) is:

$$N_{IFDofsPerDom} \approx 18 \frac{N_{EPS}^2}{N_{DOM}^{2/3}}. \tag{10}$$

In terms of total component DOFs this is:

$$N_{IFDofsPerDom} \approx 18 \left(\frac{N_{Dofs,Total}}{N_{DOM}} \right)^{2/3}. \tag{11}$$

The total number of interface DOFs in all domains ($N_{IFDofsTot}$) is therefore:

$$N_{IFDofsTot} \approx 18 N_{Dofs,Total}^{2/3} \, N_{DOM}^{1/3}. \tag{12}$$

Assuming that for large interfaces, the total number of nonzero matrix entries (in either the mass or stiffness matrix) is dominated by the interface-interface terms (N_{IFDOF}^2), the number of nonzero entries per domain ($N_{NNZPerDom}$) is:

$$N_{NNZPerDom} \approx 36 \left(\frac{N_{Dofs,Total}}{N_{DOM}} \right)^{4/3}. \tag{13}$$

The total across all domains is then:

$$N_{NNZTot} \approx 36 \left(\frac{N_{Dofs,Total}^4}{N_{DOM}} \right)^{1/3}. \tag{14}$$

This can be compared with a regular FE mesh compresed of 8-node hexahedral elements where the number of nonzero entries would be bound as:

$$N_{FE,NNZCube} < N_{EPL}^3 \, N_{DOFPerElem}^2, \tag{15}$$

where N_{EPL} is the number of element edges per line and $N_{DOFPerElem}$ is the number of DOFs per element (8 Nodes x 3 Dimensions = 24). We can again consider a decomposition with N_{cuts} slices in each dimensions so that the number of nonzero entries on a given domain is:

$$N_{FE,NNZPerDomain} < \frac{N_{Dofs,Total}}{N_{Domain}} \, N_{DOFPerElem}^2. \tag{16}$$

The total nonzero entries across all domains is:

$$N_{FE,NNZTotal} < N_{Dofs,Total} \, N_{DOFPerElem}^2. \tag{17}$$

Table 1 Problem details for block-block impact (Figure 2).

Parameter	Value
Left Block Dimensions (m)	$[0.0508, 0.0254, 0.0381]$
Right Block Dimensions (m)	$[0.0508, 0.0254, 0.0381]$
Left Block Initial Position (center of body, m)	$[-0.0279, 0, 0]$
Right Block Initial Position (center of body, m)	$[0.0381, 0, 0]$
Number of Elements Per Block	3000
Total Number of DOFs in each CBR (constraint + internal)	4406
Elastic Modulus (Pa)	5.17×10^{12}
Poisson Ratio	0.3
Density (kg/cu. meter)	7836
Left Block Initial Velocity (m/sec)	$[3810, 0, 0]$
Right Block Initial Velocity (m/sec)	$[0, 0, 0]$

These estimates show that the number of nonzero matrix entries may scale better in the original FE mesh than in the CMS model. This reinforces that a traditional CBR is a poor choice for domains with large interfaces. Decomposition is not effective at reducing the CMS models' complexity below that of the original FE model. In this work we constrain ourselves to a traditional CBR as it retains the DOFs of the original element faces that will be used by contact algorithms. The results of this section indicate that more work is necessary to enable the use of more advanced interface models (such as characteristic constraint modes) while modeling contact.

3 Numerical Examples

In this section we study the behavior of CBRs in contact using three examples and the effectiveness of decomposition with another example. First, the mechanical impact of two identical blocks where the influence of CBR interface is demonstrated. The second example is the impact of two blocks with dissimilar stiffness and density in which the shape of one block is varied to show the impact of curvature. In the third example a single CBR has a sharp traction load applied to one of its faces, further clarifying the influences of CBR interface definition observed in the first example. The gains in computational cost via decomposition are demonstrated in the last numerical example.

3.1 CBR-CBR Contact: Identical Blocks

We first demonstrate contact between superelements, particularly CBRs, with the problem depicted in Figure 2.

Throughout the numerical examples in this work we depict the motion of the two blocks via plots of position over time, demonstrated in Figure 3. In these examples, the x-axis is the axial direction, the primary direction of motion. In examples with only bricks (first and third examples), we plot the average x-position of the brick faces that come into contact. In the second example, the average position of each entire block is plotted.

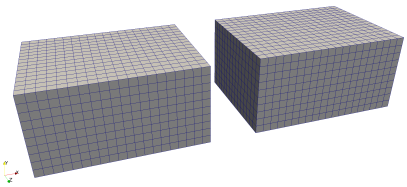
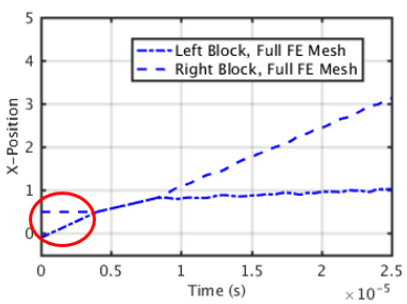
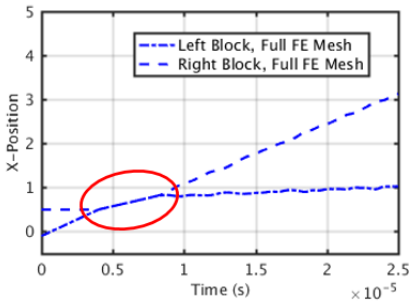


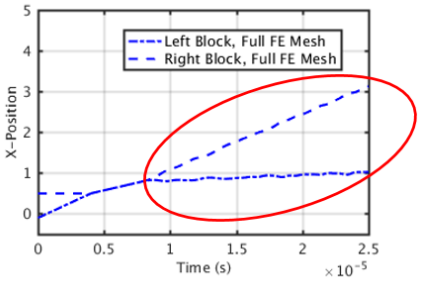
Fig. 2 Test problem for CBR-CBR contact, two blocks. Right-most block initially at rest, left-most block initially in motion with a velocity of 3810 m/sec to the right.



(a) Time before block contact



(b) Block contact

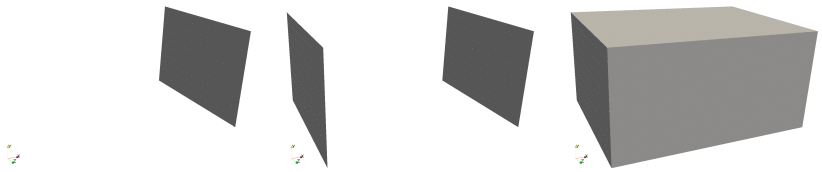


(c) Block separation

Fig. 3 Plotted left and right face positions over time during test problem of Figure 2. These are the right-most face of the left block and the left-most face of the right block, respectively.

In Figure 3 there are three clear segments of time during the simulation: 3a is the time before the blocks impact, 3b is the time during which the two blocks are in close contact and finally 3c shows the time after the blocks have separated and are traveling away from each other.

An important consideration for any CBR is the proper choice of the set of nodes and element faces to define as the CBR interface. In a typical reduction the connections are known and are the obvious choice for the interface definition. When considering contact, a large area is generally necessary. In our example there are a few logical



(a) Only surface expected to be in contact.

(b) Both front and back surfaces of block.

(c) Entire outer surface of block.

Fig. 4 Choices of surface for CBR interface definition when constructing CBR for test contact problem.

options for defining the CBR interface, these are shown in Figure 4. Figure 4a represents the most compact interface, the face of the brick that contacts the other brick (the right-most face on the left brick and the left-most face on the right brick). A more general interface is to include both the left and right faces as in Figure 4b, such that both blocks are identical. The most conservative choice, allowing for completely general contact between the brick and other components, is shown in Figure 4c, the entire outer face of the brick.

As shown in Figure 5, the choice of interface for the CBR has a significant influence on the dynamics of the CBR. Here the total number of DOFs (interface + internal) for the CBR is held constant with varying interface. The total number of DOFs is chosen to be a large percentage of the original FE mesh's, retaining much of its dynamics. In Figure 5 the dark blue curves are the results from a regular FE mesh, that is no CBR on either brick. It is clear from the green curves that if only the contacting interfaces Fig. 4a) are defined, the resulting dynamics are drastically different than with the original FE mesh. They behave as if there is no elasticity in the contact, the two faces coming into contact and never separating, continuing along together. As more interface is defined (Figs 4b and 4c), the dynamics more closely match the traditional FE mesh result.

The influence of the number of fixed interface (internal) modes retained is shown in Figure 6. We see here that the number of modes retained has little influence on the solution, all of the curves lieing on top of each other. This indicates that the behavior in this example is driven by constraint (interface) DOFs and modes.

The differences in these contact results are likely due to the dissimilar nature of the basis functions and construction that are used for the constraint and component modes. The internal, component modes are the dynamic modes with the entire interface fixed. Each interface mode is the result of a static analysis with unit displacement applied to that particular interface DOF, while all other interface DOFs are fixed to zero. That is, the behavior due to interface DOFs are the result of a static condensation, while the component DOFs represent a dynamic construction.

A simple example of inaccuracies is to consider the steady state solution to a 1D bar element under constant acceleration. In this example, one of the two nodes is considered interface (u_1). This means that the constraint mode would be: $u_1 = u_2 = 1$, and the component mode would be: $u_1 = 0$, $u_2 = -1$. Assuming a constant

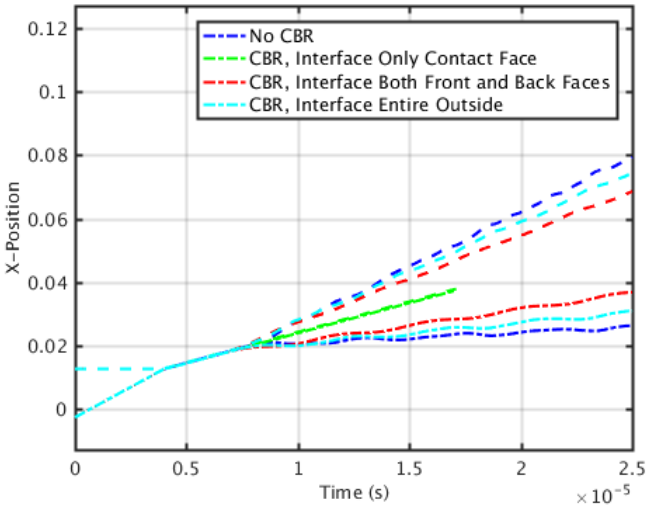


Fig. 5 Comparison of motion resulting from the choice of interface region when defining CBR. Here the total number of DOFs in the CBR remains constant, one fewer than the total number of DOFs in the original FE Mesh.

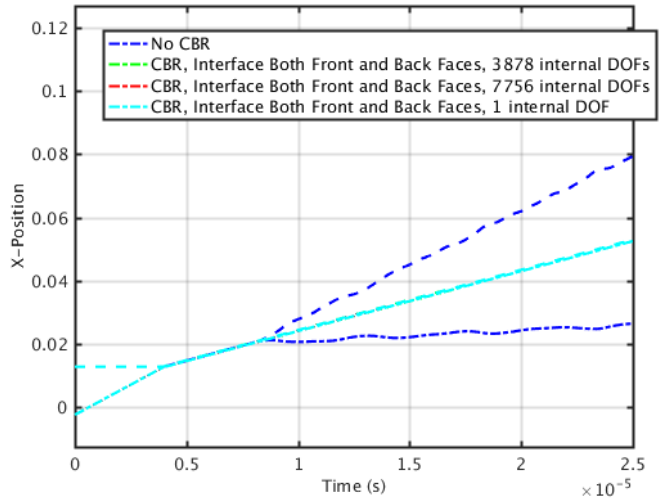
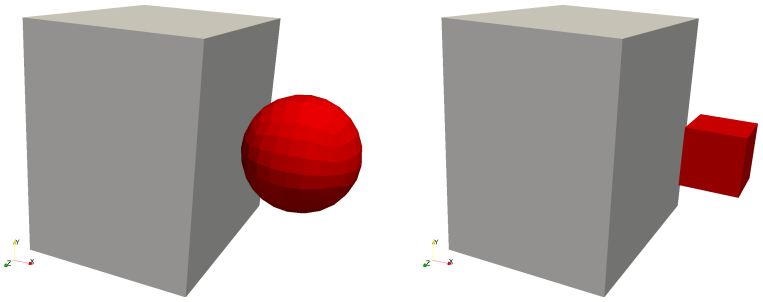


Fig. 6 Comparison of motion resulting from varying the number of fixed interface modes retained in the CBR. The differences are small, all curves for the CBR result lieing on top of each other visually.



(a) Sphere impacting stiff, dense brick. (b) Brick impacting stiff, dense brick.

Fig. 7 Depiction of block geometries for comparison of CBR contact response with flat vs curved surfaces.

Table 2 Problem details for dissimilar block-block impact (Figure 7).

Parameter	Value
Left Block Dimensions (m)	$[0.0762, 0.1016, 0.1016]$
Right Block Dimensions (m)	$[0.0254, 0.0254, 0.0254]$
Left Block Initial Position (center of body, m)	$[-0.0279, 0, 0]$
Right Block Initial Position (center of body, m)	$[0.0381, 0, 0]$
Number of Elements Per Block, Left	6000
Number of Elements Per Block, Right (sphere, brick)	864 , 1000
Number of internal modes	200
Number of constraint DOFs (sphere, brick)	654 , 1806
Elastic Modulus, Left Block (Pa)	5.17×10^{14}
Elastic Modulus, Right Block (Pa)	5.17×10^{12}
Poisson Ratio	0.3
Density, Left Block (kg/cu. m)	7.836×10^{4}
Density, Right Block (kg/cu. m)	7.836×10^{3}
Left Block Initial Velocity (m/sec)	$[0, 0, 0]$
Right Block Initial Velocity (m/sec)	$[3810, 0, 0]$

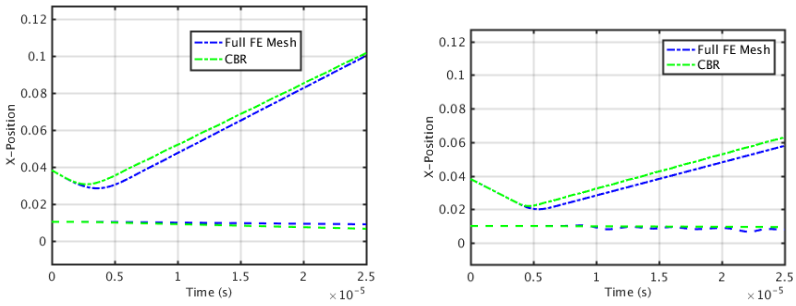
acceleration and steady-state solution:

$$\ddot{\mathbf{u}}_k = \mathbf{K}_{kk}^{-1}\, \mathbf{K}_{kc}\, \mathbf{u}_c, \tag{18}$$

and considering Eqn. 4, it is clear that for a unit acceleration at the interface, there is no response in the rest of the component. This does not reflect the response that would be expected for an acceleration only applied at the interface.

3.2 CBR-CBR Contact: Dissimilar blocks

Here we study the relative response between the traditional FE mesh and CBR models of two different impactor geometries. Figure 7a shows the first geometry, a sphere. Figure 7b shows the second geometry, a brick, similar to that of the first example. Both geometries are impacting a larger, more stiff and more dense brick, simulating a relatively solid body. The complete problem details are outlined in Table 2.



(a) Sphere impacting stiff, dense brick. (b) Brick impacting stiff, dense brick.

Fig. 8 Comparison of motion between traditional FE Mesh and CBR model with a soft object impacting a stiff, dense brick.

Figure 8 depicts the differences in CBR behavior resulting from varying impactor geometry. Qualitatively in this example, the response of the CBR of the sphere and of the brick match the full FE mesh behavior similarly. The number of internal modes retained for the CBR was held constant. Due to the geometry to the two bodies, this means that the brick (Fig 8b) had nearly three times the total number of DOFs as the sphere (Fig. 8a).

3.3 CBR with Force Load

To more carefully study the influence of the interface definition on CBR behavior we load a single CBR with a sharp load. This load is designed to be short in duration to excite higher frequency behavior as one may expect contact to also do. The problem details are found in Table 3. Here the load is applied to the nodes of the center-most two element faces on one of the brick faces. To allow the CBR to retain the most information, the total number of DOFs associated with each CBR remains constant. This number is nearly the same as the total number of DOFs associated with the full FE mesh. Due to solver limitations we retain 1 fewer DOF in the CBR representation than in the full FE mesh. We study this problem both using the Sierra/SM and Sierra/SD codes.

In Figure 9 a comparison of the average position of the loaded face is shown, highlighting differences between the results produced by Sierra/SM using an explicit time solver and the original FE model in Sierra/SD (where the CBR was created). In this figure all of the Sierra/SM explicit runs lie on top of each other (visually), regardless of the incorporation of the fictitious shell elements or their parameters, indicating that the fictitious shell elements have little influence. This also indicates that for a simple load, the interface definition is unimportant. Future study of the contact algorithms used by Sierra/SM is necessary to understand how the CBR construction may be leading to the differences observed in Section 3.1.

To verify the CBR for this problem, we show Figure 10, which shows that in Sierra/SD the full FE mesh behaves identically to the CBRs within visual tolerances.

Table 3 Problem details for brick with force load.

Parameter	Value
Brick Dimensions (m)	$[0.0508, 0.0254, 0.0381]$
Number of Elements Per Block	3000
Total Number of DOFs in each CBR (constraint + internal)	1781
Elastic Modulus (Pa)	5.17×10^{12}
Poisson Ratio	0.3
Density (kg/cu. m)	7836
Load location	Center-most two element faces on x-max face
Load Direction	$[-1, 0, 0]$
Load Function	Linear interpolation
Interpolation Time Values	$[0, 3 \times 10^{-7}, 6 \times 10^{-7}]$
Interpolation Load Values	$[0, 6.90 \times 10^{8}, 0]$
Fictitious Shell Elastic Modulus (Pa) [Large, Small]	$[6.9 \times 10^{3}, 6.9 \times 10^{-5}]$
Fictitious Shell Density (kg/cu. m) [Large, Small]	$[0.11, 1.1 \times 10^{-5}]$
Fictitious Shell Thickness (m) [Thin, Thick]	$[2.5 \times 10^{-10}, 2.5 \times 10^{-4}]$

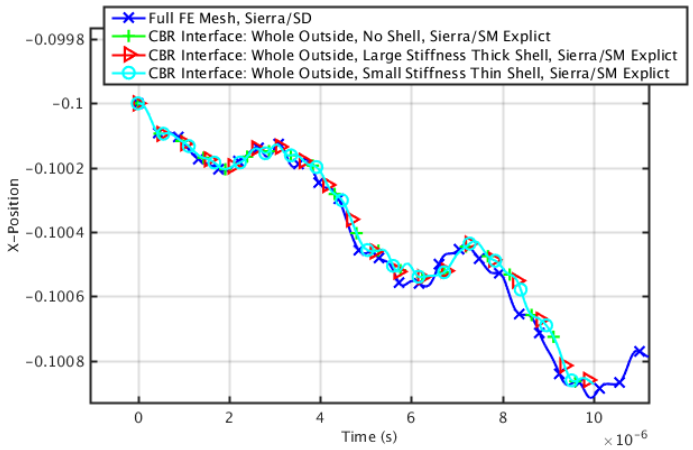


Fig. 9 Comparison of motion between traditional FE Mesh and CBR models with pressure load applied to two element faces. Here showing differences between Sierra/SM Explicit solution and original Sierra/SD solution.

In this comparison we show the Sierra/SD solution with no fictitious shell elements since they were shown to be not influential in Figure 9. This was also confirmed for Sierra/SD results not shown here. The time-step is varied to ensure that does not impact the solution, also shown in Figure 10.

As a final step we show Figure 11 to observe the influence of time step on the Sierra/SM solution. Time step does not appear to significantly influence the observed behavior. Here the time step is a multiple of the smallest element critical time step; this multiple is identified in the legend of Figure 11. Additionally, in this Figure the full FE mesh is run in Sierra/SM explicit. In this case the default 8-node hexahedral element formulation in Sierra/SD and Sierra/SM is different, Sierra/SM using a fully integrated bubble function element while Sierra/SM uses an underintegrated element

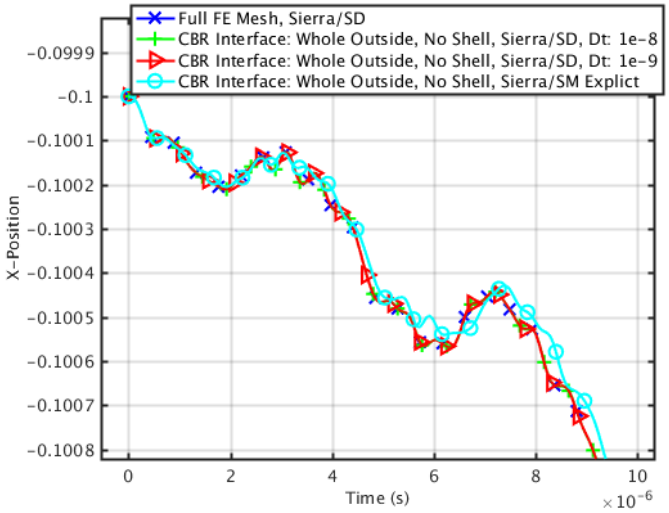


Fig. 10 Comparison of motion between traditional FE Mesh and CBR models with pressure load applied to two element faces. Here showing the similarities between Sierra/SD solution with CBRs and with the original FE mesh.

Table 4 Problem details for timing test.

Parameter	Value
Block Dimensions (m)	$[1.91 \times 10^{-2}, 1.91 \times 10^{-2}, 1.91 \times 10^{-2}]$
Total Number of Internal Modes Retained in each CBR	50
Elastic Modulus (Pa)	5.17×10^{12}
Poisson Ratio	0.3
Density (kg/cu. m)	7.836×10^{3}

with hourglass control. The author was unable to run the CBR in Sierra/SM using an implicit time solver due to convergence issues. The observed strain in the full FE mesh observed in Sierra/SM was $< 0.5\%$.

3.4 Decomposition Performance

The decomposition approach is implemented with a series of scripts combined with both the Sierra/SD and SM FE codes. We test the gains in computational time (wall clock) for this decomposition approach by varying the mesh refinement of a brick and the number of decomposition domains. The test problem is a cube with all displacements fixed on one face and a constant force applied to the opposite face in a direction normal to the face. More detailed problem parameters are described in Table 4. Spot checking showed that displacements for varying decompositions were identical within the solver precision.

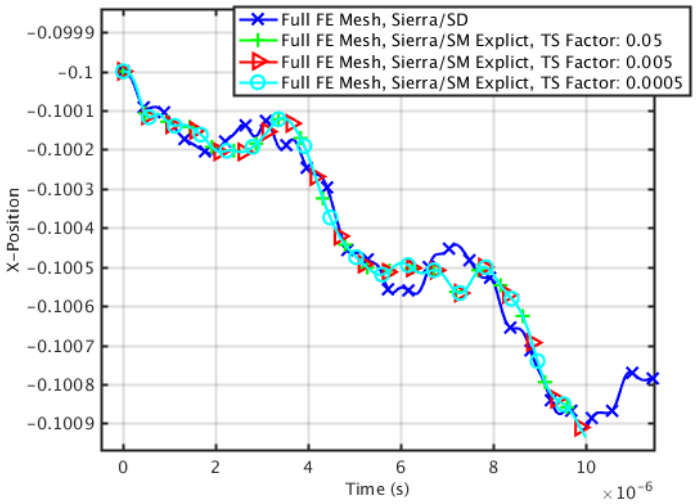


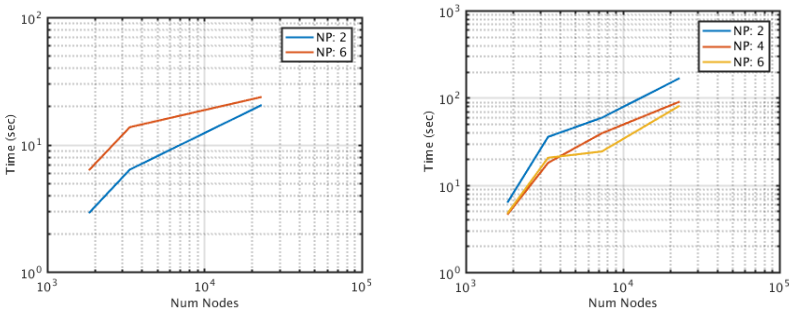
Fig. 11 Comparison of motion between traditional FE Mesh and CBR models with pressure load applied to two element faces. Here showing the differences between the Sierra/SD and Sierra/SM solutions using the full mesh. The impact of time step is also shown here.

The impact on computation time while using an explict solver is shown in Figure 12, more domains requiring more computational time. In order to handle these mass matrices in an explicit code, a solve of the assembled mass matrices for all coupled CBRs is required. In the current implementation, this requires a serial solve of the assembled CBR mass matrices of the entire decomposed component. This negates any other gains in computational efficiency from this decomposition approach. A parallel linear solve for this step would be more ideal, but still relatively expensive.

This decomposition approach for CMS holds more promise in the context of an implicit time solve as the non-diagonal mass matrices on each domain yield no such additional computational costs. Figure 12b shows the timing results using an implicit time solver. Here the timing results are more promising, showing that with more decomposition domains, the total computational cost decreases.

4 Conclusions

We have demonstrated that contact can be incorporated into CMS methods using fictitious elements utilizing standard finite element contact algorithms and that components with large interfaces can be decomposed into multiple CBRs with with a savings in number of DOFs per CBR. Demonstration of CBR-CBR contact also showed that the behavior is strongly dependent on the interface nodes of the CBR. Further testing with no contact showed that these interface dependent differences were only observed with contact. Small differences between the original FE mesh or CBR solution in Sierra/SD and those in Sierra/SM were still observed.



(a) Explicit time solver (b) Implicit time solver

Fig. 12 Comparison of computational cost with decomposition on different numbers of processors (NP) and for different time solvers.

Decomposing a component with a large interface into multiple CBRs was shown to hold promise for situations where memory is limited but not be a good solution to speed up explicit solutions. The requirement of assembling and solving the coupled mass matrices in an explicit method negates the possible impact of this approach. FE analyses using an implicit time solver may find this decomposition beneficial as was demonstrated in testing. A simple estimation of nonzero matrix entries associated with the decomposed CBR indicate that this approach does not offer any significant advantage over retaining the original FE mesh, and may be worse. At a fundamental level, CBRs with complex interfaces present a significant challenge to incorporate in a computationally efficient manner in an explicit FE method. Approaches combining more advanced interface models (such as characteristic constraint modes) with the modeling of contact appear to be necessary to achieve the reduced complexity desired of CMS methods.

References

[Bampton and Craig(1968)] Bampton MC, Craig RR JR (1968) Coupling of substructures for dynamic analyses. Aiaa Journal 6(7):1313–1319

[Castanier et al(2001)Castanier, Tan, and Pierre] Castanier MP, Tan YC, Pierre C (2001) Characteristic constraint modes for component mode synthesis. AIAA journal 39(6):1182–1187

[Craig Jr(1985)] Craig Jr RR (1985) A review of time-domain and frequency-domain component mode synthesis method

[Craig Jr(1995)] Craig Jr RR (1995) Structural dynamics: an introduction to computer methods. Society for Experimental Mechanics, Inc, 7 School St, Bethel, CT 06801, USA, 1995 527

[Hintz(1975)] Hintz RM (1975) Analytical methods in component modal synthesis. AIAA Journal 13(8):1007–1016

[Hurty(1965)] Hurty WC (1965) Dynamic analysis of structural systems using component modes. AIAA journal 3(4):678–685

[Inman and Singh(2001)] Inman DJ, Singh RC (2001) Engineering vibration, vol 3. Prentice Hall Upper Saddle River

[Reese et al(2015)] Reese GM, et al (2015) Sierra structural dynamics theory manual. Tech. rep., Sandia National Laboratories (SNL-NM), Albuquerque, NM (United States)

[Team(2011)] Team SSM (2011) Sierra/solid mechanics 4.22 users guide,. SAND2011-7597, Sandia National Laboratories